



Scolang

LISTENER -> ACTION

vv2282 - Varun Varahabhotla

jc4697 - Jackson Chen

sa3433 - Sambhav Anand

sr3355 - Sushanth Raman

kv2295 - Kanishk Vashisht

Introduction	6
Purpose	6
Use Cases	7
Tutorial	7
Environment Setup	7
Install Homebrew	7
Install Docker	7
Dockerfile	7
Enter the Environment	8
Within the Docker Container	9
Compilation	9
Creating Programs	9
Listeners and Actions	9
Running the Testing Script	10
Language Reference Manual	10
Basic Data Types	10
Advanced Data Types	10
Lexical Convention	11
Identifiers	11
Comments	11
Keywords	11
Operators	13
Syntax Notation and Program Structure	15
Operator Precedence	15
Declaration	15
Assignment	15
Binding	16
Chaining	17
Control Flow	17
Custom-Defined Functions	18
Built in Functions	18
Program Structure	19
Code Example	19

GCD:	19
Hello World Through Contracts:	20
Architecture	21
Diagram	21
Compiler	21
Scolang.ml	21
Scanner.mll	21
Parser.mly	21
Ast.ml	22
Semant.ml	22
Sast.ml	22
Codegen.ml	23
Project Plan	23
Planning Process	23
Specification Process	24
Development Process	24
Testing Process	24
Team Roles	24
Timeline	25
Communication	26
Trello	26
Facebook Messenger	26
Version Control	26
Master Contribution	27
Individual Contribution	27
Programming Style Guide	28
Test Plan & Automation	29
Example Tests	30
Test-func.sav	30
Test-Func.out	31
Test-prim-assignment.sav	31
Test-prim-assignment.out	32
Test-Fail-expr7.sav	32
Test-Fail-expr7.out	32
Test Automation	33
Testing Environment	34
Lessons Learnt	35

Kanishk Vashisht	35
Sambhav Anand	35
Jackson Chen	35
Sushant Raman	36
Varun Varahabhotla	36
Appendix	36
Makefile	37
Testing Script	38
Testing.sh	38
Testall.py	39
Stdlib	39
Scolang_stdlib.h	41
Scolang_stdlib.c	41
Scolang.ml	47
Parser.mly	48
Ast.ml	48
Semant.ml	56
Sast.ml	64
Scanner.mll	67
Codegen.ml	70
Compile.sh	80
Dockerfile	80
.travis.yml	81
Test Suite	82
Fail-action1.sav	82
Fail-action2.sav	82
Fail-assign1.sav	82
Fail-assign2.sav	82
Fail-assign3.sav	83
Fail-assign4.sav	83
Fail-assign5.sav	83
Fail-assign6.sav	83
Fail-assign7.sav	83
Fail-assign8.sav	84
Fail-assign9.sav	84
Fail-assign10.sav	84
Fail-assign11.sav	84
Fail-assign12.sav	85

Fail-Contract1.sav	85
Fail-Contract2.sav	85
Fail-Contract3.sav	85
Fail-dead1.sav	86
Fail-dead2.sav	86
Fail-dead3.sav	87
Fail-dead4.sav	87
Fail-dead5.sav	88
Fail-dead6.sav	88
Fail-expr1.sav	88
Fail-expr2.sav	89
Fail-expr3.sav	89
Fail-expr4.sav	89
Fail-expr5.sav	89
Fail-expr6.sav	89
Fail-expr7.sav	90
Fail-expr8.sav	90
Fail-expr9.sav	90
Fail-for1.sav	90
Fail-for2.sav	91
Fail-for3.sav	91
Fail-for4.sav	91
Fail-for5.sav	91
Fail-for6.sav	91
Fail-func1.sav	91
Fail-func2.sav	92
Fail-func3.sav	92
Fail-func4.sav	92
Fail-func5.sav	92
Fail-func6.sav	93
Fail-func7.sav	93
Test-Arithmetic.sav	93
Test-for.sav	94
Test-func.sav	94
Test-if.sav	95
Test-prim-assignment.sav	96
Test-print.sav	97
Test-return.sav	97
Test-while1.sav	98

Test-while2.sav
Git Commits

98
98

SCoLang : Small Contract Language

Introduction

A scripting language made with the premise that everything is a contract waiting to execute. Each contract has defined listeners, including listeners which listen for other contracts recursively. When all listeners are satisfied, a set of contract conditions get executed.

Purpose

We live in a world full of routine tasks. Every day, millions of people take repeated actions based on guaranteed stimuli - all the way from turning on the lights when they wake up to making git commits after a function is declared. The scope of these actions is broad, and we envision a general purpose language that can make a dent in this scope. Which is why we're writing a language that works by listening and reacting inspired by blockchain smart contracts. This is the language that will be usable for anything from simple load balancing to complex IOT applications.

Use Cases

- 1) Notifying users about certain events (e.g. weather drops below 40 degrees)
- 2) Distributed system management
- 3) Recurring tasks (e.g every time it hits 7am turn off Nightshift).
- 4) Cycle numeric operations (eg. GCD)
- 5) Triggering events based on physical sensors (e.g. if multiple burglar alarms go off, we can trigger multiple events accordingly)

Tutorial

Environment Setup

SCoLang was developed utilizing Ocaml, meaning that OCaml should be installed before beginning to program on Ocaml. The quickest way to create the environment to develop SCoLang code is to run a docker instance with the environment variables already setup. The easiest way to do is through running a docker instance with all the compiler software defined for you.

Install Homebrew

Run this command within terminal:

```
/usr/bin/ruby -e "$(curl -fsSL  
https://raw.githubusercontent.com/Homebrew/install/master/install)"
```

Install Docker

Run the following command into terminal

```
brew install docker
```

Dockerfile

Save the following content into a file named dockerfile with no extension.

```
# Based on 18.04 LTS  
FROM ubuntu:bionic  
  
RUN apt-get -yq update && \  
    apt-get -y upgrade && \  
    apt-get -yq --no-install-suggests --no-install-recommends install \  
    ocaml \  
    menhir \  
    llvm-6.0 \  
    llvm-6.0-dev \  
    m4 \  
    git \  
    aspcud \  
    ca-certificates \  
    python2.7 \  
    pkg-config \  
    vim \  
    nano \  
    cmake \  
    opam  
  
RUN ln -s /usr/bin/lli-6.0 /usr/bin/lli  
RUN ln -s /usr/bin/lld-6.0 /usr/bin/lld  
  
RUN opam init
```



```
RUN opam install \
  llvm.6.0.0 \
  ocamlfind

WORKDIR /root

ENTRYPOINT ["opam", "config", "exec", "--"]

CMD ["bash"]
```

Enter the Environment

To Enter the Docker Environment Run the following commands in the directory with the dockerfile

```
docker build . -t [container-name] -f Dockerfile
docker run -t -d [container-name]
r
```

Within the Docker Container

Once within the docker container, clone the github repository

```
git clone https://github.com/KanishkVashisht/scolang.git
```

Compilation

Once the environment is set up and the source code is, to compile your project run enter the main 'scolang' directory and run the compile.sh script with your file that you intend to run

The content of compile.sh are:

```
#!/bin/bash

make scolang.native
./scolang.native $1.sav > $1.ll
llc $1.ll > $1.s
gcc -o $1 $1.s stdlib/scolang_stdlib.o -no-pie
rm -rf $1.s $1.ll
```

The compile script is run with the following command

```
./compile.sh [file-name-without-extension]
```

Creating Programs

Scolang programs have the extension '.sav'. The following is an example of Hello world in scolang

```
print("Hello World");
```

Since Scolang is a scripting language, there is no need to declare a main, and one can begin coding right away.

Listeners and Actions

As stated above, the premise of our language is that everything is a contract waiting to execute. This means that there are two objects 'listeners' and 'actions' which allow for users to be able to implement IFTTT interactions. Listeners wait for certain conditions to be met, such as a http request, or a set of conditions, while 'actions' are a set of code which is executed once a set of conditions satisfy a listener. Actions are bound to listeners through the '->' syntax. The following is an example which shows both in action:

```
Listener a = {  
  webhook(6969);  
  resolve 1;  
};  
  
Action b = {  
  
  print("Executed");  
  
};  
  
a -> b;
```

Running the Testing Script

The testing script can be run with the following shell command:

```
./testing.sh
```

This will run through a series of automated tests located within the /tests directory to ensure that the language is running properly and there is a coverage of edge cases within the language.

Language Reference Manual

Basic Data Types

Types	Description
int, float	Regular primitive types
bool	Ternary data type: true, false, null
String	A text value. Can be a single char or an arbitrary number of chars concatenated together.

Advanced Data Types

Data Type	Description
Contract	A structure that stores associated listeners with actions. At a high level, this binds listeners to actions and then triggers actions if conditions for listeners are satisfied.
Listener	A data structure that stores a boolean condition. It also handles checking to see if the condition is met. (e.g. memory usage exceeds a certain amount)
Action	A data structure that stores an event. This event is meant to be triggered by a contract if the conditions for the associated listener(s) are meant. (e.g. turn lights on)

Lexical Convention

Identifiers

Identifiers consist of one or more characters with a leading alphabetic character followed by any number of alphanumeric characters or underscores.

Comments

```
/* Comment */  
/*  
    Multi-line Comment  
*/
```

Keywords

Keywords	Description
Contract	A structure that stores associated listeners with actions. At a high level, this binds listeners to actions and then triggers actions if conditions for listeners are satisfied. Contracts are immutable.
Listener	A data structure that stores a boolean condition. It also handles checking to see if the condition is met. (e.g. memory usage exceeds a certain amount). Listeners are immutable.
Action	A data structure that stores an event. This event is meant to be triggered by a contract if the conditions for the associated listener(s) are meant. (e.g. turn lights on). Actions are immutable.

for	standard for loop (for control flow) that follows the syntax: <i>for(initialization; terminal condition; increment) block</i>
while	standard while (for control flow) that follows the syntax: <i>while(condition) block</i>
if	standard if (for control flow) that follows the syntax: <i>if(condition) block</i>
else	standard else (for control flow) that follows the syntax: <i>else block</i> Note <i>else</i> must follow an <i>if</i> statement (and a variable number of <i>elif</i>).
return	Return a specified variable or data type. Mustn't necessarily return anything. Return is only allowed inside of actions.
resolve	Complete a listener. Resolve means the listener successfully executed. Resolve will set the listener value to be 1 (or true).
reject	Complete a listener. Reject means the listener failed in its execution. Reject will maintain the listener value at 0, and will no longer be able to change to 1.
break	Breaks the current control flow
continue	Skips the current iteration within an iterator
void	Null value

Operators

Unary Operators:

Operators	Description
-----------	-------------

!	Not
---	-----

Binary Operators:

Operators	Description
==	Equal to comparative
!=	Is not equal to
<	Less than
>	Greater than
<=	Less than or equal to
>=	Greater than or equal to
&&	And operator
	Or operator

Special Operators:

Operators	Description
=	Assignment
->	Binds listeners to actions to create a contract
()	Function call -- should match the signature of the function being called exactly

Mathematical Operators

Operators	Description
+	Addition
-	Subtraction
*	Multiplication

/	Division
%	Modulo
<<	Bitshift left
>>	Bitshift right
^	Binary XOR
&	Binary AND
	Binary OR

Punctuators:

Notation	Description
[]	Array assignment or array index referencing
()	Statement Precedence (e.g in a mathematical expression)
{ }	Code Blocks
;	Statement boundary. Must be at the end of each statement to signal such.

Syntax Notation and Program Structure

Operator Precedence

The order of operators is as follows:

Operator	Precedence
-> []	right
!	right
* / %	left

+ -	left
>> <<	left
> < >= <=	left
== !=	left
& ^	left
&&	left
=	right

Declaration

```
<type> <name>
int a; /* Initiates variable a of type int */
```

Assignment

Basic:

```
<type> <name>
int a; /* Initiates variable a of type int */
```

Advanced:

```
<type> <varname> = <codeblock>
    Action display = {
        print(a);
    }
Listener listen = {
    while(a != b)
    {
        if(a > b)
        {
            a -= b;
        }
    }
}
```



```
    }
    elif( b > a)
    {
        b -= a;
    }
}
```

Binding

```
Contract c = <expression of listeners> -> <action(s)>
```

Example:

```
Listener l1 = {...}
Listener l2 = {...}
Action a = {...}
Contract c = l1 || l2 -> a
```

The expression of listeners must resolve to a value of 1 in order for the actions to be executed. Note that Listeners that have not resolved are at the state of 0. All of the actions will be executed in random order once the expression is satisfied. Each variable in the listener expression must be a single listener (i.e it cannot be an array of listeners).

If the user would like to terminate a contract from happening, the individual listeners will need to be individually rejected.

Chaining

Example:

```
Contract c = ...
```

```
Contract d = c -> a2
```

In this case, we chain contract c to action a2. Here, contract c functions as a listener to action a2. Once the contract c is completed, then the actions associated with contract d are executed.

Chaining is also a way to enforce order in executing actions, since the bind function executes the array of actions in random order.

Control Flow

```
if/else structure
    if(condition){
        statement
    }
}
else {
    statement
}

for structure
    for(initialization; terminal condition; increment) {
        statement
    }

while structure
    while(condition) {
    }
```

Custom-Defined Functions

Scolang supports user-defined functions. Functions are defined in the following way:

```
<return type> <function name>(<formal vars>) { <statements> }
```

The function declaration is similar to the way functions are declared in C. Every function has a return-type attached to it, and that is specified as the first parameter. The formal variables is a comma-separated list of variable-types and variables (in that order). The function itself is composed of various statements, that ultimately end with a *return* statement. The return statement needs to return the matching return type of the function.

Built in Functions

Syntax	Functions
webhook(int port)	Initiates a connection to a specified port. Returns a boolean (default 0, set to 1 for a tick if the endpoint is requested).
printf(float)	Print a float that is passed into the function into standard output.
printi(int)	Print an int that is passed into the function into standard output.
print(String str)	Print a string that is passed into the function into standard output.
query(String Str)	sends an http call to a given endpoint
system_call(string str)	emulates the C system fx

Program Structure

A Scolang program is expected to have function declarations in the beginning of the program, and then user-defined execution of the functions in the latter part of the program. Note that the user can forgo function-based program entirely, and simply execute chains of contracts instead -- though this is not recommended.

Within a function, the structure should be in the following order:

1. Variable Declarations
2. Listener Declarations
3. Action Declarations
4. Contract Declarations (through bindings)
5. Return statement

Code Example

GCD:

```
long a = 10; /*or input*/
long b = 4;
long t;

Listener keep_going = { if(b != 0){ resolve(); } };
Listener found = {if(a==b){ resolve(); }};

Action subtract = {
    if(a > b) {
        t = b;
        b = a % b;
        a = b;
    }
};

Action display = {
    print(a);
};

keep_going -> subtract;
found->display;
```

Hello World Through Contracts:

```
bool helloHook = webhook('/hello');

Listener hello = {
  if (helloHook) {
    resolve();
  }
}

Action print_hello = {
  print("Hello World!");
}

hello -> print_hello;
```

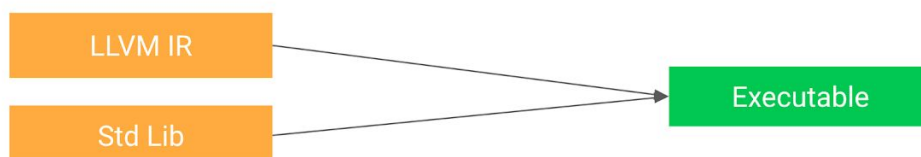
Architecture

Diagram

Ocaml Compiler



IR to Executable



Compiler

Scolang.ml

This is the top-level of the Scolang compiler. It invokes the scanner, parser, semant and codegen modules which inturn generates the LLVM IR. It contains various flags that allows the user to stop the chain at various stages, outputting an abstract syntax tree, a semantically checked abstract syntax tree, and of course the LLVM IR.

Scanner.mll

The scanner reads Scolang source code and does the lexical analysis. If there are no illegal character issues, such as a malformed variable name construction, the scanner then passes on the tokens to the parser.

Parser.mly

Reads the tokens passes through from the scanner module and checks if they are syntactically correct. This is where associativity rules are also defined to avoid shift-reduce conflicts. If there are no syntax issues, an abstract syntax tree will be formed.

The hierarchy of types within a Scolang program is as follows: program, function, statement, expression. The parser deconstructs the program into a list of function declarations and a list of statements. The reason for this design decision was that we wanted to make Scolang an imperative language where the user would be able to quickly write scripts in setting up Contracts (ease of use is a primary selling point of this language), but also allow the user to simultaneously add more sophisticated logic that can be encapsulated within various functions.

Ast.ml

This is the abstract syntax tree generated from the parser of the Scolang code. The AST file also contains string conversion functions for the various types that are present within the language (as specified in the parser.mly section). These functions are used for debugging purposes or for printing errors out.

One language design decision that was made in the Ast was to wrap primitives (int, float, Contracts, etc) within a higher order datatype called typ. Typ would also encapsulate aggregate data types of primitives as well in order to avoid having two separate data types for singular primitives and for aggregate primitives. However, aggregate data types were cut from scope.

Another design decision was to upgrade variable initialization to be at the statement level. This adds the functionality of allowing the user to declare and initialize variables anywhere within

statement blocks of code. This, again, goes in line with the theme of having an easy to use programming language.

Semant.ml

Checks the AST to ensure that it is semantically correct. If all the AST checks are correct, passes it into the Codegen

Semant generates a semantically checked abstract syntax tree by mapping each of the AST types into its equivalent SAST type. This decision was made because this be a fail-safe way to force Codegen to use semantically checked input types and avoid any potential issues of it using unchecked input types.

The overall structure of this semant file checks for legality of variable declarations, then adds said variables to a StringMap that continuously refers to later. The checker then checks functions, statements and expressions within a nested order based on the input program.

Sast.ml

This files defines the semantically checked abstract syntax tree. This tree should be a near one to one mapping with the abstract syntax tree, since every type is just the semantically checked variant of its related type within the AST. One key difference is that every semantically checked expression also has a primitive type attached to it, since all expressions in Scolang should operate on a specific type of primitive.

This file also contains many string conversion functions that are useful for debugging and error printing purposes within the Codegen.

Codegen.ml

Codegen takes in the semantically checked AST and converts the source code into LLVM byte code. The Codegen links in standard library code written in C that are used for custom defined functions like webhook, and bind.

Earlier it was mentioned that this language was designed to be imperative (the program can just be a list of statements without functions). However, the Codegen only executes code within functions, and it simply incorporates the imperative statements within a "main" function that behaves like it does in C.

The codegen keeps tracks of different variable maps to distinguish global variables from local variables, and to ensure standard scoping behavior is properly implemented.

Actions and Listeners are passed into the standard library as function pointers, since these variables are unique in that they take in statement code blocks rather than an expression that computes to some primitive value. Thus within Codegen, actions and listeners are implemented as special functions that are stored in separate function declaration StringMaps that are then invoked when the bind function is called.

Project Plan

Planning Process

Our team met twice a week during the development of our language. The first meeting would occur after class on Monday with Jennifer, who was the TA assigned to our group. The second Meeting would typically occur Tuesday nights, where we would in the earlier stages make decisions for designing and implementing our language and later ended up being focused more on developing our language. These meetings would act as an opportunity to touch base on code that we were individually contributing to.

All shared documentation resided on our Columbia Google Drive Directory, which included meeting notes with TA, group meeting notes, any miscellaneous documentation and all Scolang related documented files. Other tools we utilized will be described below.

Specification Process

We initially had two different language ideas, and after discussing the merits and challenges of both with the TA's, ended up having a vote between the two languages to decide which one we would decide to create. During this process, we had to compare the complexity of both (with the other language being a vector-based images processing language) along with where we believed our interest lay. This ended up being the smart contracts language which was based upon the smart contracts associated with the cryptocurrency Ethereum.

Development Process

During the beginning development of our language, we would meet in person and contribute together utilising codeshare. This was primarily due to our inexperience in OCaml, but also to concurrently verify the code being written. Later in the semester as our focus shifted more towards Semant and Codegen, We split into two groups to contribute in parallel. Throughout the entire process, we maintained a pair programming structure. If a member of the group wasn't there whose work would be touched, we would make sure to leave detailed notes. The tools we

utilized to ensure that everyone was making meaningful contributions and working on relevant issues will be detailed below.

Testing Process

We developed a large automated testing suite which was one of the first things coded on how we intended to use our language and edge cases where we thought issues could arise along with the expected output. We would run the test suite automatically and would work on developing towards getting our tests to pass. This was an invaluable tactic as it allowed us to understand and see a physical representation of changes we were making in semant and codegen. Our test cases were written in Scolang Files (which have a .sav extension) along with a .out file with the respective name to use as a comparison for our testing script. The testing suite will further be detailed below.

Team Roles

While the table below shows the individual roles we operated under, the contributions deviated significantly from the signed upon responsibilities. As we got later in the process, we began pair programming as we found that to be the most effective way of treading unfamiliar territory with both OCaml and the LLVM module. While the roles helped to guide what we would focus on initially, as the project progressed roles became more dynamic.

People	Roles
Jackson Chen (Language Guru)	Parser, AST, Semant SAST, Codegen
Kanishk Vashisht (Tester)	Scanner, Codegen, Stdlib, Testing Script, Tests, Environment
Sushant Raman (Systems Architect)	Parser, Semant, SAST, Stdlib, Scolang.ml
Sambhav Anand (Systems Architect)	Scanner, Semant, Demo, StdLib, Scolang.ml
Varun Varahabhotla (Project Manager)	AST, Semant, Tests, Report, Environment Setup

Timeline

Date	Objectives and Achievements
------	-----------------------------

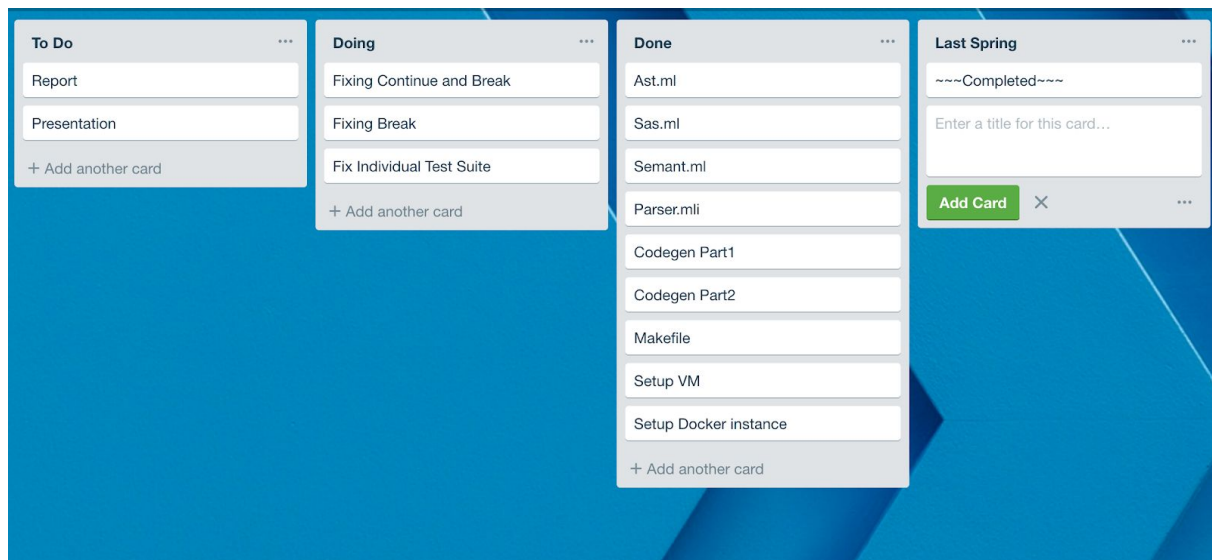
Sep 15	First group meeting, where different language ideas were proposed.
Sept 18	Voted on language of choice and complete the language proposal
Oct 2	Language design, initial version of the LRM
Oct 16	Scope cutting on language choices, begin AST & Parser, write test suites
Oct 30	Complete AST and Parser, complete test suites
Nov 13	Rough version of codegen, Hello World!, Start devops (CI, Docker, and Ubuntu Server)
Nov 27	Start semant and rework codegen, complete SAST, more scope cutting
Dec 4	Completing semant and codegen, complete standard library, integration
Dec 11	Debugging, work on demo
Dec 18	Finalization of demo, create presentation and final report

Note that towards the end, we moved towards a one week sprint to achieve more progress on the various projects that were going on simultaneously (semant, codegen, standard library, demo constructing, devops).

Communication

Trello

We followed Agile Principles and ran Bi-Weekly sprints of what we were trying to achieve. We utilized Trello for the Kaba Boards. The following is an example of our Kaba Board:



Facebook Messenger

Our primary form of communication including text, and voice was through Facebook Messenger, where we had a dedicated group for our language. Through this group chat, we ran video chats, VOIP calls, and shared important documents.

Version Control

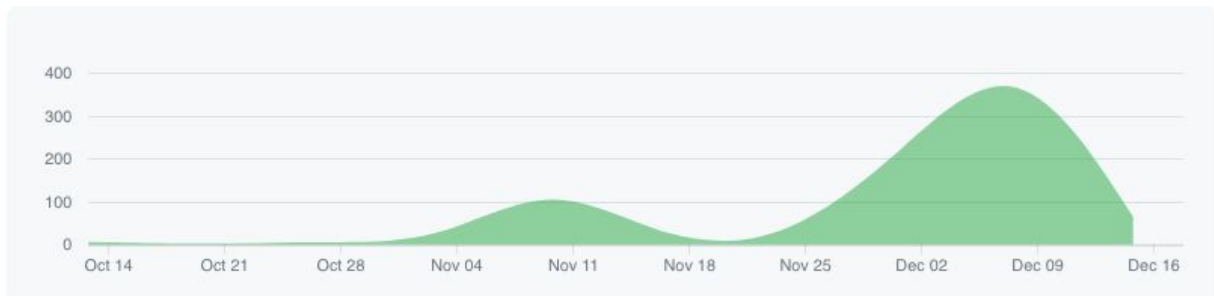
Throughout our project we utilized a private git repository to implement version control on our project. This allowed for cohesion during the development phase of our language through ensuring that everyone was working on the latest version of the code. Furthermore version control provided the benefit of ensuring that if there were conflicts, they could properly be sorted and the intended version used. The following is the insights of our version control:

Master Contribution

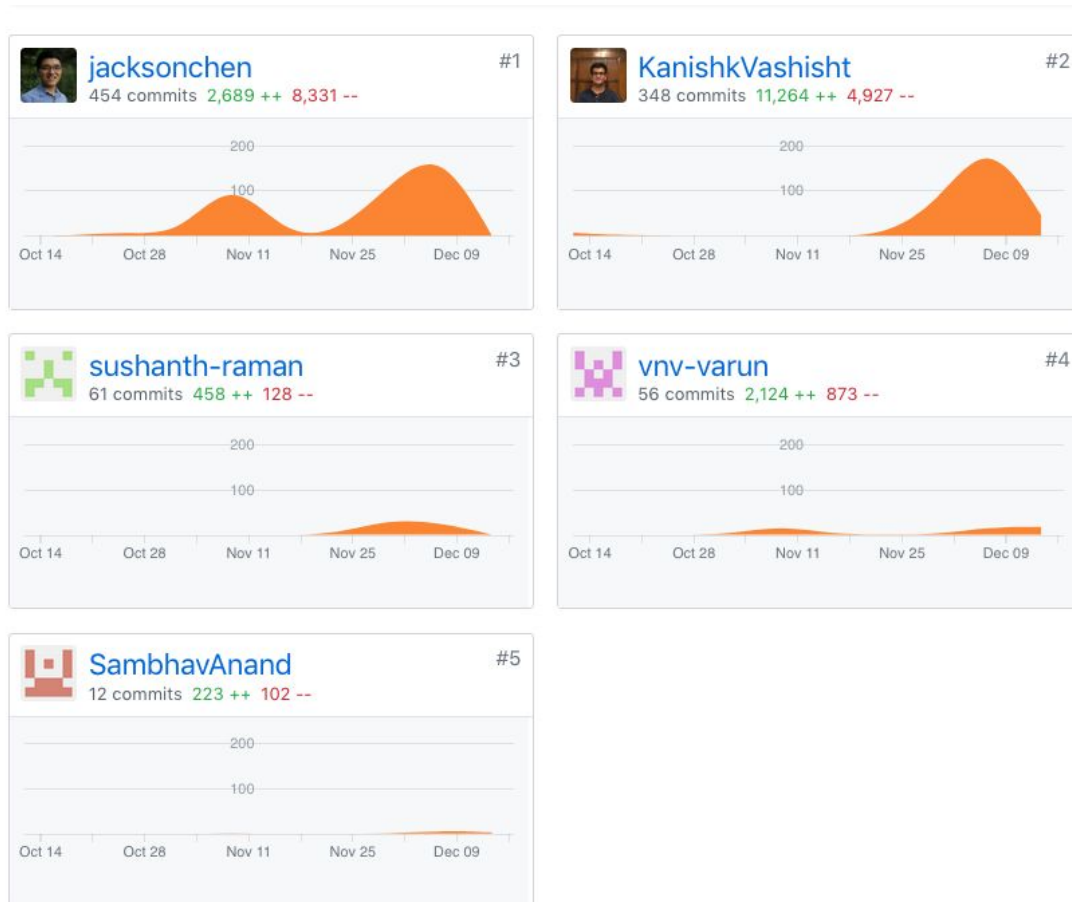
Oct 14, 2018 – Dec 19, 2018

Contributions: Commits ▾

Contributions to master, excluding merge commits



Individual Contribution



Note that there was a lot of effort put into other aspects (devops, project management, demo setup) that are not reflected in the GitHub history.

Programming Style Guide

We all utilized Atom as our primary text editor along with the **language-ocaml** package.

Tab Size: 2

Commenting Convention:

All comments should begin immediately following the opening comment symbols and end immediately following the closing comment symbol.

```
/* Comment */  
/*  
    Multi-line Comment
```

```
*/
```

Parser Rule Convention:

All patterns associated with a rule should be aligned with two spaces before it followed by a pipe (|) followed by another space. The exception is that the first pattern will not have a pipe before it and will be aligned with four spaces before it. Actions should also align vertically with each other, in other words, all action blocks should be aligned by one space after the longest pattern within the rule. Action blocks should also have a single space separating either curly brace from the text of the action.

```
rule:
    first_pattern { first_action }
  | second_pattern { second_action }
```

When a rule accepts an empty string, or nothing, the matching pattern should contain a comment saying “nothing”. The alignment conventions from above also apply to this comment, treating it as if it was another pattern. An example is as follows:

```
rule:
    /* nothing */ { first_action }
  | second_pattern { second_action }
```

AST Type Convention:

For types with a single pattern, the definition should be on one line

```
type a_type = Name of b_type
```

Note that names must start with a capital letter and continue to be CamelCase throughout. For types with multiple patterns, the same alignment rules from the parser apply, without the action statements. The equals sign needs to be padded with a space to its left, and should be the last character within the type definition line.

```
type a_type =
    first_name of first_type
  | second_name of second_type
```

OCaml Coding Convention:

OCaml function assignments should be multi-line. The `let` statement, name of the variable (in snake case) and the equals sign should be on one line. The body of the function should follow

on new lines, with each line of the body being indented by two spaces. Then an `in` statement ends the declaration, sitting at the same indentation as the `let` statement.

```
let some_name =  
  (* some code *)  
in
```

However, in situations of pattern matching, the pipes are allowed to be at the same level of indentation as the `let` statement. Pipes may also be indented with two spaces as well -- these two conventions are both accepted. The following is an example:

```
let some_name = function  
  something -> something_else  
| another -> another_thing  
in
```

OCaml code should also be properly commented with explanations, inputs, outputs, and section labeling to make it much easier to read by a human eye. Furthermore, error messages should also be as verbose and descriptive as possible for easier debugging purposes.

Test Plan & Automation

The full scripts for our running our tests and the tests themselves can be found in the appendix below. Our primary goal of the tests was to not only test successful code, but to also test all edge cases of assignment operators, control flow, built in functions, comments, function calls, scoping and to ensure that code was failing where we intended it to for the same listed types. To achieve this we had two sets of tests, fail-tests which were intended to fail along with a set of passing tests (labeled tests-[name]) which were intended to pass. Here we would also output the expected output and then the testing script would compare to ensure that the output was what was expected.

Example Tests

Test-func.sav

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

```

}

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

int foo(int a)
{
    return a;
}

int bar(int a, bool b, int c) { return a + c; }

int a;
a = add(39, 3);
printi(a);

int i;
i = 1;

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

printi(i);

printi(bar(17, false, 25));

```

Test-Func.out

```

42
2
42

```

The above program tests to see if functions are properly being called, and if they are returning the type specified in the function declaration.

Test-prim-assignment.sav

```

int a = 10;

```

```
int b = 5;
printi(a);
printi(b);
b = a;
printi(a);
printi(b);
```

```
float f1 = 5.5;
float f2 = 3.30;
printf(f1);
printf(f2);
f2 = f1;
printf(f1);
printf(f2);
```

```
String s1 = "Hello";
String s2 = "World";
```

```
print(s1);
print(s2);
```

```
s2 = s1;
```

```
print(s1);
print(s2);
```

Test-prim-assignment.out

```
10
5
10
10
5.5
3.3
5.5
5.5
Hello
World
Hello
```



```
Hello
```

This is yet another successful test which is highlighting and testing various forms of assignment, to ensure that variables and values are being passed as intended.

Test-Fail-expr7.sav

```
//contract op non-contract

Listener a = {resolve();}
Action b = {print("test");}
int d = 100;

Contract c = a->b;
c = c*d;
```

Test-Fail-expr7.out

```
Fatal error: exception invalid type operation
```

This is a failed test example, here an invalid operation is happening with an object of type contract being multiplied by an object of integer. This is not a valid arithmetic operation as and thus provides an error.

Test Automation

We utilized a script called testing.sh which automatically runs all of our tests and creates a log file. The testing.sh script calls a python script, named testing.py which runs through the commands. The following is the output of the testing.sh script:

```
root@4f992c1b9b22:~/scolang# ./testing.sh
starting test...
[Passed] test-for
[Passed] fail-assign8
[Passed] fail-dead4
[Passed] fail-assign12
[Passed] fail-expr8
[Passed] fail-assign11
[Passed] fail-for6
[Passed] fail-func5
[Passed] fail-assign10
[Passed] fail-expr9
[Passed] fail-func7
[Passed] fail-dead6
[Passed] test-while1
[Passed] fail-action1
[Passed] fail-func1
[Passed] fail-contract2
[Passed] fail-expr6
[Passed] test-return
[Passed] fail-dead3
[Passed] test-prim-assignment
[Passed] fail-while
[Passed] fail-for5
[Passed] fail-assign3
[Passed] test-world
[Passed] fail-for4
[Passed] fail-func3
[Passed] fail-func6
[Passed] fail-func2
[Passed] fail-dead2
[Passed] test-func
[Passed] fail-assign7
[Passed] fail-expr3
[Passed] fail-action2
[Passed] fail-contract3
[Passed] fail-assign2
[Passed] fail-expr5
[Passed] fail-dead5
[Passed] fail-expr7
[Passed] fail-assign9
[Passed] fail-contract1
[Passed] fail-expr1
[Passed] fail-assign5
[Passed] fail-assign6
[Passed] test-while2
[Passed] test-if
[Passed] test-print
[Passed] fail-assign4
[Passed] fail-dead1
[Passed] fail-expr2
[Passed] fail-for2
[Passed] fail-assign1
[Passed] fail-expr4
[Passed] test-arithmetic
[Passed] fail-for1
[Passed] fail-func4
[Passed] fail-for3
Passed 56 out of 56 tests.
root@4f992c1b9b22:~/scolang#
```

Additionally the following is the output of a sample log.txt file generated through running this script. This example has failed test cases to highlight the difference:

```
====
```

```
STARTING TESTING
```

```
2018-12-19 00:46:10.857834
```

```
FAILED TESTS:
```

```
test-for, test-prim-assignment, test-if, test-return, test-print,  
test-func, test-while1, test-while2, test-arithmetic, test-world
```

```
PASSED TESTS:
```

```
fail-dead6, fail-expr8, fail-expr9, fail-action1, fail-dead5, fail-dead4,  
fail-action2, fail-dead1, fail-dead3, fail-dead2, fail-while, fail-assign9,  
fail-assign8, fail-for2, fail-assign6, fail-assign7, fail-for3, fail-for1,  
fail-assign5, fail-assign4, fail-for4, fail-assign1, fail-for5,  
fail-assign3, fail-assign2, fail-for6, fail-assign12, fail-func4,  
fail-expr1, fail-func5, fail-assign11, fail-func7, fail-expr2, fail-expr3,  
fail-func6, fail-assign10, fail-func2, fail-expr7, fail-expr6,  
fail-contract1, fail-func3, fail-func1, fail-contract3, fail-expr4,  
fail-expr5, fail-contract2
```

```
Successes : 46, Failures: 10
```

```
ENDED TESTING
```

```
====
```

Testing Environment

We had a multifaceted approach towards our testing environment. To standardize all development environments we utilized docker containers which allowed for consistency during development which mitigates issues that arise from different OS's and settings. Furthermore we set up a VM on Digital Ocean which we linked to Travis CI for continuous integration testing. This allowed it so that we all had access to a working instance of our code, but also to automatically deploy and test our code through Travis CI.

Lessons Learnt

Kanishk Vashisht

As a part of this PLT Project, I learned the vital importance of continuous testing using CI and why it's a beneficial tool. I also learned how tools like Menhir are much better simulators than just scaffolding code or hit and trial. I was also familiarized with functional programming and ended on a high note finding it to be elegant and more straightforward than storing and manipulating values in objects. More personally, I learned the importance of working together as

a team and playing to everyone's strengths to optimize completion. Lastly, as Edwards says, I also learned that solving any CS Problem is a matter of adding another layer of indirection.

Sambhav Anand

I would highly recommend this class to anyone who's interested in improving their software engineering skills. During the course of the semester, I learned a lot while working in a team environment to build our compiler, but one learning that stood out the most was being understanding towards other team members. The whole team carries everyone, and everyone has to do their job assigned to them. I always thought of myself as someone who wrote code on his own and wouldn't, but after this experience, I realized that having a smart and talented team of engineers working towards the same problem with a common goal is one of the best ways to improve one's skills.

Jackson Chen

The process of creating this language and taking this class provided many invaluable lessons that can't be found in other academic or even industry settings. I learned the following lessons:

- How to work on in a group environment within an academic setting. This is particularly challenging because of the lack of incentive structure to keep a steady pace throughout the semester when there are no deadlines
- Learned lots of OCaml, Llm, how a compiler works. This was my first introduction to functional programming, and I was blown away by how condensed and logical the programming paradigms were.
- Programming languages and compilers used to be black boxes that I never understood fully. Understanding the implementation of programming languages has made me a more aware and better (applications) programmer in the process
- Gained a ton of appreciation for programming languages, I've been particularly humbled by how we initially scoped our work through the LRM to increase feasibility, but even that wasn't enough
- Cutting scope was a big technical and feasibility task. As projects progress and the realization that not 100% of the expected features will be complete by a timeline, I learned to figure out where to prioritize work, how to effectively cut scope (without endangering the purpose of the language), and how to negotiate with team members around this
- Gained 4 amazing new friends

Sushant Raman

One crucial area I learned from this project was how to work collaboratively in a group and set clear accountability guidelines. Initially, we found ourselves falling behind as we hadn't divided tasks and set deadlines. Over time, we started to see that setting clear deadlines and used Trello to divide responsibilities. This approach helped us significantly improve productivity and outcomes. What I primarily learned is that excellent teamwork and project management skills are critical to driving success in large projects. Usually, the technical elements of a plan aren't as difficult as organizing the people.

Varun Varahabhotla

One of the most important things I got out of this class was to demystify a lot of the 'magic' surrounding software engineering. I would always wonder how a programming language was written and operates and my understanding was primarily that it converts code into binary which the computer could understand. The remaining middle part was always a mystery to me. Not only did I get an understanding of what this mysterious middle layer did, but thanks to that experience combined with utilising OCaml, I feel like I can conquer most commercial programming languages. Another topic this course exposed me to was the many continuous integration and testing tools available in software engineering settings. This was the first time I had to go beyond Git and utilize Docker, Cloud Hosted Virtual Machines, and Travis CI which I had never had exposure to. Also the friendships that are made through achieving a large goal is something I hope to continue doing.

Appendix

Makefile

```
.PHONY: all
all : clean scolang.native scolang_stdlib.o

.PHONY: scolang.native
scolang.native :
    eval `opam config env`; ocamlbuild -use-ocamlfind -pkg
llvm,llvm.analysis -cflags -w,+a-4 \
    scolang.native

# "make clean" removes all generated files
```

```
.PHONY : clean
clean :
    ocamlbuild -clean
    rm -rf testall.log *.diff scolang scanner.ml parser.ml parser.mli
    rm -rf webhook
    rm -rf *.cmx *.cmi *.cmo *.cmx *.o *.s *.ll *.out *.exe

# More detailed: build using ocamlc/ocamlopt + ocamlfind to locate LLVM
OBJS = ast.cmx codegen.cmx parser.cmx scanner.cmx semant.cmx sast.cmx
scolang.cmx

scolang : $(OBJS)
    ocamlfind ocamlopt -linkpkg -package llvm -package llvm.analysis
$(OBJS) -o scolang

scolang_stdlib.o : stdlib/scolang_stdlib.c
    gcc -c -g -o stdlib/scolang_stdlib.o stdlib/scolang_stdlib.c

scanner.ml : scanner.mll
    ocamllex scanner.mll

parser.ml parser.mli : parser.mly
    ocamlyacc parser.mly

%.cmo : %.ml
    ocamlc -c -g $<

%.cmi : %.mli
    ocamlc -c -g $<

%.cmx : %.ml
    ocamlfind ocamlopt -c -package llvm $<

ast.cmo :
ast.cmx :
codegen.cmo : ast.cmo sast.cmo
codegen.cmx : ast.cmx sast.cmx
parser.cmo : ast.cmo parser.cmi
parser.cmx : ast.cmx parser.cmi
parser.cmi : ast.cmo
sast.cmo :
sast.cmx :
```

```
scanner.cmo : parser.cmi
scanner.cmx : parser.cmx
scolang.cmo : semant.cmo scanner.cmo parser.cmi codegen.cmo ast.cmo
sast.cmo
scolang.cmx : semant.cmx scanner.cmx parser.cmx codegen.cmx ast.cmx
sast.cmx
semant.cmo : sast.cmo ast.cmo
semant.cmx : sast.cmx ast.cmx
test.cmo : scanner.cmo parser.cmi ast.cmo
test.cmx : scanner.cmx parser.cmx ast.cmx
```

Testing Script

Testing.sh

```
#!/bin/bash
python3 testall.py
```

Testall.py

```
import os
import filecmp
import datetime

"""CONFIG ZONE"""
TESTS = "./newtests"
LOGFILE = "./logfile.txt"
"""END CONFIG ZONE """

class bcolors:
    HEADER = '\033[95m'
    OKBLUE = '\033[94m'
    OKGREEN = '\033[92m'
    WARNING = '\033[93m'
    FAIL = '\033[91m'
    ENDC = '\033[0m'
```

```

BOLD = '\033[1m'
UNDERLINE = '\033[4m'

def compile(filename):
    os.system("touch {0}.trial.out".format(filename))
    os.system("./scolang.native {1}/{0}.sav >> {0}.ll
2>{0}.trial.out".format(filename,TESTS+"/"))
    if("fail" in filename): return
    os.system("llc {0}.ll > {0}.s".format(filename))
    os.system("gcc -o {0} {0}.s stdlib/scolang_stdlib.o -lcurl
-no-pie".format(filename))
    os.system("./{0} > {0}.trial.out".format(filename))
    return

def clean(filename):
    os.system("rm -rf {0}.ll {0}.s ./{0} {0}.trial.out".format(filename))
    return

os.system('echo "====\nSTARTING TESTING" >>'+LOGFILE)
os.system("echo \''+str(datetime.datetime.now())+'\n\'>> "+LOGFILE)
#os.system('echo "Making scolang" >> {}'.format(LOGFILE))
#os.system("make")
print("starting test...")

filenames = os.listdir(TESTS)
files = []
successes, failures = [],[]

for file in filenames:
    if(".sav" not in file):
        continue
    filename = file[:file.index(".sav")]
    if("fail" in file):
        files.append(filename)
    elif("test" in file):
        files.append(filename)
    else:
        print("ERRONEOUS FILE :"+filename)

for filename in files:
    compile(filename)

```



```

    try:
        success = filecmp.cmp(TESTS+"/"+filename+".out",
filename+".trial.out")
    except:
        success = False
        print(filename + " or output not found")

    if(success):
        successes.append(filename)
        print(bcolors.OKGREEN+"[Passed] "+bcolors.ENDC + filename)
    else:
        failures.append(filename)
        print(bcolors.FAIL+"[Failed]
"+bcolors.ENDC+bcolors.UNDERLINE+filename + bcolors.ENDC)

    clean(filename)

os.system('echo "\nFAILED TESTS: \n '+(", ".join(failures))+'" >>
'+LOGFILE)
os.system('echo "PASSED TESTS: \n '+(", ".join(successes))+'">> '+LOGFILE)
os.system('echo "Successes : {}, Failures: {}\n" >>
{}'.format(len(successes),len(failures),LOGFILE))
print(bcolors.HEADER+bcolors.BOLD+"Passed "+str(len(successes))+ " out of
"+str(len(failures) + len(successes))+ " tests."+bcolors.ENDC)
os.system('echo "ENDED TESTING \n=== \n" >> '+LOGFILE)

```

Stdlib

Scolang_stdlib.h

```

#ifndef _MYLIB_H_
#define _MYLIB_H_
    extern int query(char* endpoint);
    extern int webhook(int port_user);
    extern int bind_val(int(*listener)(),void(*action)());
    extern int bind_port(int(*listener)(int),void(*action)(), int port);
    extern int bind_p(int(*listener)(int),void(*action)());
    extern int system_call(char* query);

#endif

```

Scolang_stdlib.c

```
#include <stdlib.h>
#include <unistd.h>
#include <stdio.h>
#include <errno.h>
#include <sys/socket.h>
#include <sys/types.h>
#include <netinet/in.h>
#include <string.h>
#include <time.h>
#include <arpa/inet.h>
#include <netdb.h>
#include "scolang_stdlib.h"
#include <curl/curl.h>
#include <curl/easy.h>

int print(char *s){
    printf("%s", s);
    printf("\n");
    return 1;
}

static void die(const char* msg) {
    fprintf(stderr, "%s", msg);
    exit(1);
}

int bind_val(int(*listener)(),void(*action)())
{
    if(fork() == 0){
        return 1;
    }
    else{
        int resolve = 0;
        if((*listener)())
        {
            resolve = 1;
        }
    }
}
```

```

        (*action)();
    }
    exit(0);
}
}

int bind_port(int(*listener)(int),void(*action)(), int port)
{
    int resolve = 0;
    if((*listener)(port))
    {
        resolve = 1;
        (*action)();
    }
    return resolve;
}

int bind_p(int(*listener)(int),void(*action)())
{
    int lower = 1000;
    int upper = 9999;
    int port = 0;
    srand(time(NULL));
    port = rand() % upper;

    if(port < lower)
    {
        port += lower;
    }
    fprintf(stderr, "listening on port numbr: %d \n",port);

    int resolve = 0;
    if((*listener)(port))
    {
        resolve = 1;
        (*action)();
    }
    return resolve;
}

int webhook(int port_user)

```

```

{
    int sockfd;
    int resolve;
    struct sockaddr_in self;
    char buffer[1024];

    int lower = 1000;
    int upper = 9999;
    int port = 0;

    srand(time(NULL));
    port = port_user;
    if(port < lower)
    {
        port += lower;
    }
    fprintf(stderr, "\nlistening on port number: %d \n", port);

    resolve = 0;
    sockfd = socket(AF_INET, SOCK_STREAM, 0);
    bzero(&self, sizeof(self));
    self.sin_family = AF_INET;
    self.sin_port = htons(port);
    self.sin_addr.s_addr = INADDR_ANY;

    if(bind(sockfd, (struct sockaddr*)&self, sizeof(self)) != 0)
    {
        perror("socket--bind");
        exit(errno);
    }

    if(listen(sockfd, 20) != 0)
    {
        perror("socket--listen");
        exit(errno);
    }

    int response;
    struct sockaddr_in client_addr;
    int addrlen=sizeof(client_addr);
    response = accept(sockfd, (struct sockaddr*)&client_addr, &addrlen);

```

```

if(response > 0)
{
    resolve = 1;

}
close(response);
close(sockfd);
return resolve;
}

/*int query(char* query_string)
{
    int sock;
    struct sockaddr_in server_addr;
    struct hostent *he;
    char buf[4096];
    char server_name[200];
    char *server_ip;
    char *serverPort;
    char filePath[200];
    char *fname;
    int i;
    for(i=0;i<strlen(query_string);i++){
        if(query_string[i]=='/')
            break;
    }
    memset(server_name, '\0', sizeof(server_name));
    strncpy(server_name, query_string, i);
    server_name[i] = '\0';
    memset(filePath, '\0', sizeof(filePath));
    strcpy(filePath, strchr(query_string, '/'));
    if((he = gethostbyname(server_name))==NULL){
        die("gethostbyname failed");
    }

    server_ip = inet_ntoa(*(struct in_addr *)he->h_addr);
    if ((sock= socket(PF_INET, SOCK_STREAM, IPPROTO_TCP))<0) {
        die("socket failed");
    }

    memset(&server_addr, 0, sizeof(server_addr));
    server_addr.sin_family = AF_INET;

```

```

server_addr.sin_addr.s_addr = inet_addr(server_ip);
server_addr.sin_port = htons(80);

if(connect(sock, (struct sockaddr *)&server_addr,
sizeof(server_addr)<0){
die("connect failed");
}

snprintf(buf, sizeof(buf), "GET %s HTTP/1.1\r\n"
"Host: %s:%s\r\n"
"\r\n",
filePath, server_name, "80");
if(send(sock, buf, strlen(buf), 0) != strlen(buf)){
die("send failed");
}

FILE *f;
if((f=fdopen(sock, "r"))==NULL) {
die("fopen failed");
}

if(fgets(buf, sizeof(buf), f)==NULL){
if(ferror(f)){
die("IO error");
}
else {
fprintf(stderr, "server terminated connection without
response");
fclose(f);
exit(1);
}
}
if (strncmp("HTTP/1.0 ", buf, 9) != 0 && strncmp("HTTP/1.1 ", buf, 9)
!= 0) {
fprintf(stderr, "unknown protocol response: %s\n", buf);
fclose(f);
exit(1);
}
char response_code[4];
memset(response_code, '\0', sizeof(response_code));
strncpy(response_code, (buf+9), 3);
response_code[3] = '\0';

```

```

    fclose(f);
    return atoi(response_code);
}*/
int query(char* make_query) {
    char *query = malloc(1000);
    strcpy(query, make_query);
    char *endpoint = strtok(query, " ");
    char *method = strtok(NULL, " ");
    char *raw_data = strtok(NULL, " ");
    curl_global_init(CURL_GLOBAL_ALL);
    CURL *hnd = curl_easy_init();

    curl_easy_setopt(hnd, CURLOPT_CUSTOMREQUEST, method);
    curl_easy_setopt(hnd, CURLOPT_URL, endpoint);

    struct curl_slist *headers = NULL;
    headers = curl_slist_append(headers, "Postman-Token:
80f28757-6ef8-4444-abac-7dcb1dd7b204");
    headers = curl_slist_append(headers, "cache-control: no-cache");
    curl_easy_setopt(hnd, CURLOPT_HTTPHEADER, headers);

    curl_easy_setopt(hnd, CURLOPT_POSTFIELDS, raw_data);

    CURLcode ret = curl_easy_perform(hnd);
    free(query);
    return 1;
}

int system_call(char* query_string)
{
    return system(query_string);
}

```

Scolang.ml

```

(* Top-level of the MicroC compiler: scan & parse the input,
   check the resulting AST, generate LLVM IR, and dump the module *)

type action = Ast | Sast | LLVM_IR | Compile

```

```

let _ =
  let action = ref Compile in
  let input = ref "" 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: ./scolang.native [-a|-l] [file]" in
  Arg.parse speclist (fun s -> input := s) usage_msg;
  let channel = if !input = "" then
    stdin
  else
    open_in !input
  in
  let lexbuf = Lexing.from_channel channel in (*reading in the input*)
  let ast = Parser.start_state (*parser takes the tokens, pattern matches
  builds an abstract syntax tree*) Scanner.token lexbuf (*scanner takes
  lexbuf and adds tokens to it *)in
  let sast = Semant.check ast in (*Semantically checks it, adds types where
  missing and creates SAST*)
  match !action with
  | Ast -> print_string (Ast.string_of_program [] 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 (*codegen takes sast, matches
  llvm api's to our code and generates some form of IR *) in
    Llvm_analysis.assert_valid_module m; (*LLVM does syntax checking on
  IR*)
    print_string (Llvm.string_of_llmodule m) (*Prints out IR*)

```

Parser.mly

```

%{
open Ast
%}

%token INT FLOAT BOOL STRING VOID CON ACT LIS EQ NOTEQ RESOLVE REJECT

```



```

%token PLUS MINUS TIMES DIVIDE MODULO EOF OPENSQ CLOSESQ OPENPAR CLOSEPAR
OPENBLOCK CLOSEBLOCK SEMICOLON COMMA
%token ARRDEC ASSIGN NULL AND OR
%token BAND BOR BRIGHT BLEFT BXOR
%token LT LTE GT GTE
%token NOT
%token BIND
%token FOR WHILE IF ELSE BREAK CONTINUE RETURN
%token<string> VARIABLE
%token<bool> BOOL_LIT
%token<int> INT_LIT
%token<string> STRING_LIT
%token<float> FLOAT_LIT

%nonassoc NOELSE
%nonassoc ELSE
%right ASSIGN
%left AND OR
%left BAND BOR BXOR
%left EQ NOTEQ
%left LT LTE GT GTE
%left BLEFT BRIGHT
%left PLUS MINUS
%left TIMES DIVIDE MODULO
%right NOT
%right BIND ARRDEC

%start start_state
%type<Ast.program> start_state
%%

start_state:
  decls EOF { $1 }

decls:
  /* nothing */ { [], [] }
  | decls statement { fst $1, (snd $1 @ [$2]) }
  | decls fdecl { (fst $1 @ [$2]), snd $1 }

fdecl:
  vartype VARIABLE OPENPAR formals_opt CLOSEPAR OPENBLOCK stmt_list

```

```

CLOSEBLOCK
    { { typ = $1;
      fname = $2;
      formals = $4;
      body = List.rev $7 } }

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

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

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

statement:
  | expr SEMICOLON { Expr($1) }

  /* control flow */
  | FOR OPENPAR expr SEMICOLON expr SEMICOLON expr CLOSEPAR statement {
For($3, $5, $7, $9) }
  | OPENBLOCK stmt_list CLOSEBLOCK { Block(List.rev $2) }
  | WHILE OPENPAR expr CLOSEPAR statement { While($3, $5) }
  | IF OPENPAR expr CLOSEPAR statement %prec NOELSE { If($3, $5, Block([]))
}
  | IF OPENPAR expr CLOSEPAR statement ELSE statement { If($3, $5, $7) }
  | BREAK { Break }
  | CONTINUE { Continue }

  /* Return */
  | RETURN expr SEMICOLON { Return($2) }
  | RETURN SEMICOLON { Return(NoExpr) }

  /* Variable assignment */
  | vartype VARIABLE SEMICOLON { VarInit($1, $2, NoExpr) }
  | vartype VARIABLE ASSIGN expr SEMICOLON { VarInit($1, $2, $4) }
  | vartype VARIABLE ASSIGN OPENBLOCK stmt_list CLOSEBLOCK SEMICOLON {
LisActInit($1, $2, (List.rev $5)) }

```

```

/* Binding */
| VARIABLE BIND VARIABLE SEMICOLON { Bind($1, $3) }

/* Completion */
| RESOLVE expr SEMICOLON { Resolve($2) }
| REJECT SEMICOLON { Reject }

expr:
| BOOL_LIT      { BoolLit($1) }
| INT_LIT       { IntLit($1) }
| FLOAT_LIT     { FloatLit($1) }
| STRING_LIT    { StringLit($1) }

/*Vars*/
| VARIABLE { Var($1) }

/*Logical*/
| expr LT expr { Binop($1, Lt , $3) }
| expr LTE expr { Binop($1, Lte , $3) }
| expr GT expr { Binop($1, Gt , $3) }
| expr GTE expr { Binop($1, Gte , $3) }
| expr EQ expr { Binop($1, Eq , $3) }
| expr NOTEQ expr { Binop($1, NotEq , $3) }
| expr AND expr { Binop($1, And, $3) }
| expr OR expr { Binop($1, Or, $3) }

/*Unary*/
| NOT expr { Unop(Not, $2) }

/*Arithmetic*/
| expr PLUS expr { Binop($1, Plus, $3) }
| expr MINUS expr { Binop($1, Minus, $3) }
| expr TIMES expr { Binop($1, Times, $3) }
| expr DIVIDE expr { Binop($1, Divide, $3) }
| expr MODULO expr { Binop($1, Modulo, $3) }

/*BITWISE*/
| expr BAND expr { Binop($1, Band, $3) }
| expr BOR expr { Binop($1, Bor, $3) }
| expr BRIGHT expr { Binop($1, Bright, $3) }
| expr BLEFT expr { Binop($1, Bleft, $3) }
| expr BXOR expr { Binop($1, Bxor, $3) }

```

```

/* Variable Assignment */
| VARIABLE ASSIGN expr { Asn($1, $3) }
/* | VARIABLE OPENSQ INT_LIT CLOSESQ { ArrayIndex($1, $3) } TODO: Arrays
*/

/*Functional for contract binding & calling */
/* | VARIABLE ASSIGN VARIABLE BIND VARIABLE { Bind($1, $3) } TODO: Allow
Contract variables */
| VARIABLE OPENPAR actuals_opt CLOSEPAR { Call($1, $3) }

/*parentheses*/
| OPENPAR expr CLOSEPAR { InnerExpr($2) }

actuals_opt:
    /* nothing */ { [] }
| actuals_list { List.rev $1 }

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

/*variable and array types*/
vartype:
| datatype { Primitive($1) }
/* | datatype OPENSQ INT_LIT CLOSESQ { Array($1, $3) } TODO: Arrays */

datatype:
    INT { Int }
| FLOAT { Float }
| STRING { String }
| BOOL { Bool }
| LIS { Listener }
| ACT { Action }
| CON { Contract }
| VOID { Void }

```

Ast.ml

```

type op = Lt | Lte | Gt | Gte | Eq | NotEq | And | Or |
          Plus | Minus | Times | Divide | Modulo |
          Band | Bor | Bright | Bleft | Bxor

```

```

type uop = Not
type primitive = Int | Float | String | Bool | Listener | Contract | Action
| Void

type typ =
  | Primitive of primitive
  (* | Array of primitive * int TODO: Arrays *)

type bind = typ * string

type expr =
  IntLit of int
  | BoolLit of bool
  | FloatLit of float
  | StringLit of string
  | Var of string
  | Binop of expr * op * expr
  | Unop of uop * expr
  | Asn of string * expr
  | Call of string * expr list
  (* | ArrayIndex of string * int TODO: Arrays*)
  | InnerExpr of expr
  | NoExpr

type statement =
  Expr of expr
  | For of expr * expr * expr * statement
  | While of expr * statement
  | If of expr * statement * statement
  | Block of statement list
  | VarInit of typ * string * expr
  | LisActInit of typ * string * statement list
  | Break
  | Continue
  | Bind of string * string
  | Return of expr
  | Resolve of expr
  | Reject

type func_decl =
  {

```

```
    typ: typ;
    fname: string;
    formals: bind list;
    body: statement list;
  }
```

```
type program = func_decl list * statement list
```

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

```
let string_of_op = function
```

```
| Plus -> "+"
| Minus -> "-"
| Times -> "*"
| Divide -> "/"
| Modulo -> "%"
| Eq -> "=="
| NotEq -> "!="
| Lt -> "<"
| Lte -> "<="
| Gt -> ">"
| Gte -> ">="
| And -> "&&"
| Or -> "||"
| Band -> "&"
| Bor -> "|"
| Bright -> ">>"
| Bleft -> "<<"
| Bxor -> "^"
```

```
let string_of_uop = function
```

```
| Not -> "!"
```

```
let string_of_primitive = function
```

```
Int -> "int"
| Bool -> "bool"
| Float -> "float"
| String -> "String"
| Listener -> "Listener"
| Contract -> "Contract"
| Action -> "Action"
```

```

| Void -> "Void"

let string_of_typ = function
  | Primitive(p) -> string_of_primitive p
  (* | Array(p, i) -> string_of_primitive p ^ "[" ^ string_of_int i ^ "]"
  TODO: Arrays *)

let rec string_of_expr = function
  | IntLit(l) -> string_of_int l
  | BoolLit(true) -> "true"
  | BoolLit(false) -> "false"
  | StringLit(s) -> s
  | Binop(e1,o,e2) -> string_of_expr e1 ^ " " ^ string_of_op o ^ " " ^
string_of_expr e2
  | Unop(o,e) -> string_of_uop o ^ string_of_expr e
  | Asn(v, e) -> v ^ " = " ^ string_of_expr e
  | NoExpr -> ""
  | FloatLit(f) -> string_of_float f
  | Call(f, el) ->
  f ^ "(" ^ String.concat ", " (List.map string_of_expr el) ^ ")"
  | Var(v) -> v
  (* | ArrayIndex(v, i) -> v ^ "[" ^ string_of_int i ^ "]" TODO: Arrays
  *)
  | InnerExpr(e) -> "(" ^ string_of_expr e ^ ")"

let rec string_of_stmt = function
  | Block(stmts) ->
  "{\n" ^ String.concat "" (List.map string_of_stmt stmts) ^ "}\n"
  | Expr(expr) -> string_of_expr expr ^ ";\n";
  | Return(expr) -> "return " ^ string_of_expr expr ^ ";\n";
  | If(e, s, Block([])) -> "if (" ^ string_of_expr e ^ ")\n" ^
string_of_stmt s
  | If(e, s1, s2) -> "if (" ^ string_of_expr e ^ ")\n" ^ string_of_stmt s1
^ "else\n" ^ string_of_stmt s2
  | While(e, s) -> "while (" ^ string_of_expr e ^ ") " ^ string_of_stmt s
  | VarInit(t, v, e) -> string_of_typ t ^ " " ^ v ^ " " ^ string_of_expr
e
  | LisActInit(t, v, s) -> string_of_typ t ^ " " ^ v ^ " " ^
String.concat "" (List.map string_of_stmt s)
  | Break -> "break\n"
  | Continue -> "continue\n"
  | Bind(v1, v2) -> v1 ^ " -> " ^ v2

```

```

    | For(e1, e2, e3, s) -> "for (" ^ string_of_expr e1 ^ "; " ^
string_of_expr e2 ^ "; " ^ string_of_expr e3 ^ ")\n{" ^ string_of_stmt s ^
"}\n"
    | Resolve (e) -> "resolve " ^ string_of_expr e ^ "\n"
    | Reject -> "reject\n"

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

let rec string_of_program stor = function
  | ([], []) -> String.concat "\n" (List.rev stor) ^ "\n"
  | ([], stmt :: tl) -> string_of_program (string_of_stmt stmt :: stor)
([], tl)
  | (func :: tl, []) -> string_of_program (string_of_fdecl func ::
stor) (tl, [])

(* return all functions first, then statements *)
| (func :: ftl, stmts) ->
  string_of_program (string_of_fdecl func :: stor) (ftl, stmts)

```

Semant.ml

```

open Ast
open Sast

module StringMap = Map.Make(String)

(* Semantic checking of the AST. Returns an SAST if successful,
throws an exception if something is wrong.

Check each global variable, then check each function *)

let check (functions, statements) =

  let prim_of_dt = function

```



```

    | Primitive(p) -> p
    (* | Array(p, _) -> p TODO: Arrays *)
in

(* Verify a list of bindings has no void types or duplicate names *)
let check_binds (kind : string) (binds : bind list) =
  let binds_converted =
    List.map (fun (t,b) -> (prim_of_dt t, b)) binds
  in
  List.iter (function
    (Void, b) -> raise (Failure ("illegal void " ^ kind ^ " " ^ b))
    | _ -> ()) binds_converted;
  let rec dups = function
    [] -> ()
    | ((_,n1) :: (_,n2) :: _) when n1 = n2 ->
      raise (Failure ("duplicate " ^ kind ^ " " ^ n1))
    | _ :: t -> dups t
  in dups (List.sort (fun (_,a) (_,b) -> compare a b) binds)
in

(**** Check global variables ****)

(* TODO: Allow global variables to work *)
(* check_binds "global" statements; Note this is something we'll need to
fix *)

(**** Check functions ****)

(* Collect function declarations for built-in functions: no bodies *)
let built_in_decls =
  let add_bind map (name, ty) =
    if (Void = prim_of_dt ty) then
      (StringMap.add name {
        typ = Primitive(Void);
        fname = name;
        formals = []; body = [] } map)
    else
      StringMap.add name {
        typ = Primitive(Void);
        fname = name;

```

```

        formals = [(ty, "x")]; body = [] } map
    in List.fold_left add_bind StringMap.empty [ ("print",
Primitive(String));
                                                ("printi", Primitive(Int));
                                                ("printf", Primitive(Float));
                                                ("webhook", Primitive(Int));
                                                ("query", Primitive(String));
                                                ("system_call", Primitive(String));
                                                ]
in

(* Add function name to symbol table *)
let add_func map fd =
    let built_in_err = "function " ^ fd.fname ^ " may not be defined"
    and dup_err = "duplicate 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

(* 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 then lvaluet else raise (Failure err)
in

(* Make sure the bind function works *)
let check_bind lvaluet rvaluet =

```

```

        if ((lvaluet = Listener || lvaluet = Contract) && rvaluet == Action)
then Contract
    else raise (Failure ("Illegal bind. You only have to listners and
contracts on left hand side." ^
    "Only actions are allowed on right hand side"))
in

(* Make sure the array is valid TODO: Arrays *)
(* let check_valid_array lt rt =
    if (lt != Int || lt != Float || lt != String ||
    lt != Bool || lt != Listener || lt != Contract ||
    lt != Action) || rt != Int then raise (Failure ("Illegal Array
access"))
    else lt
in *)

(* Get local variables *)
let locals statements =
    let rec get_locals mylocals = function
        | [] -> mylocals
        | [VarInit (t, s, _)] -> get_locals [(t, s)] []
        | VarInit (t, s, _) :: r -> get_locals ((t, s) :: mylocals) r
        | [LisActInit (t, s, _)] -> get_locals [(t, s)] []
        | LisActInit (t, s, _) :: r -> get_locals ((t, s) :: mylocals) r
        | _ :: r -> get_locals mylocals r
    in
    get_locals [] statements
in

(* Return a variable from our local symbol table *)
let type_of_identifer symbols s =
    try StringMap.find s symbols
    with Not_found -> raise (Failure ("undeclared identifier " ^ s))
in

(* Return the type when given a literal TODO: Arrays *)
(* let type_of_literal = function
    int -> Int
    | float -> Float
    | bool -> Bool
    | string -> String
in *)

```

```

(* Return a semantically-checked expression, i.e., with a type *)
let rec expr symbols = function
  IntLit l -> (Int, SIntLit l)
  | FloatLit l -> (Float, SFloatLit l)
  | BoolLit l -> (Bool, SBoolLit l)
  | StringLit l -> (String, SStringLit l)
  | NoExpr -> (Void, SNoexpr)
  | InnerExpr e -> let (t, e') = expr symbols e in (t, SInnerExpr (t,
e'))
  | Var v -> (prim_of_dt (type_of_identifer symbols v), SVar v)
  | Asn(var, e) as ex ->
    let lt = prim_of_dt (type_of_identifer symbols var) and (rt, e') =
expr symbols e in
    let err = "illegal assignment " ^ string_of_primitive lt ^ " = " ^
      string_of_primitive rt ^ " in " ^ string_of_expr ex
    in (check_assign lt rt err, SAsn(var, (rt, e')))
  | Unop(op, e) as ex ->
    let (t, e') = expr symbols e in
    let ty = match op with
      | Not when t = Bool -> Bool
      | _ -> raise (Failure ("illegal unary operator " ^
        string_of_uop op ^ string_of_primitive t ^
        " in " ^ string_of_expr ex))
    in (ty, SUnop(op, (t, e')))
  | Binop(e1, op, e2) as e ->
    let (t1, e1') = expr symbols e1
    and (t2, e2') = expr symbols e2 in
    (* All binary operators require operands of the same type *)
    let same = t1 = t2 in
    (* Determine expression type based on operator and operand types *)
    let ty = match op with
      Plus | Minus | Times | Divide when same && t1 = Int -> Int
    | Plus | Minus | Times | Divide when same && t1 = Float -> Float
    | Eq | NotEq when same -> Bool
    | Lt | Lte | Gt | Gte
      when same && (t1 = Int || t1 = Float) -> Bool
    | And | Or when same && t1 = Bool -> Bool
    | _ -> raise (
      Failure ("illegal binary operator " ^
        string_of_primitive t1 ^ " " ^ string_of_op op ^ "
" ^

```

```

                                string_of_primitive t2 ^ " in " ^ string_of_expr
e))
  in (ty, SBinop((t1, e1'), op, (t2, e2')))

  (* | ArrayIndex(v,i) -> let lt = prim_of_dt (type_of_identifer
symbols v) and rt = type_of_literal i in
  (check_valid_array lt rt, SArrayIndex(v, i)) TODO: Arrays *)

  | 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') = expr symbols e in
    let ftt = prim_of_dt ft in
    let err = "illegal argument found " ^ string_of_primitive et ^
" expected " ^ string_of_primitive ftt ^ " in " ^
string_of_expr e
    in (check_assign ftt et err, e')
    in
    let args' = List.map2 check_call fd.formals args
    in (prim_of_dt fd.typ, SCall(fname, args'))
  in

  (* Check semanticness of boolean expressions *)
  let check_bool_expr symbols e =
    let (t', e') = expr symbols e
    and err = "expected Boolean expression in " ^ string_of_expr e
    in if t' != Bool then raise (Failure err) else (t', e')
  in

  (* Return a semantically-checked statement i.e. containing sexprs *)
  let rec check_stmt symbols return_typ = function
    Expr e -> SExpr (expr symbols e)
    | If(p, b1, b2) -> SIf(check_bool_expr symbols p, check_stmt symbols
(Primitive(Void)) b1, check_stmt symbols (Primitive(Void)) b2)
    | For(e1, e2, e3, st) ->
    SFor(expr symbols e1, expr symbols e2, expr symbols e3, check_stmt
symbols (Primitive(Void)) st)
    | While(p, s) -> SWhile(check_bool_expr symbols p, check_stmt symbols

```

```

(Primitive(Void)) s)
  | Continue -> SContinue
  | Break -> SBreak
  | Return e -> let (t, e') = expr symbols e in
  if t = prim_of_dt return_typ then SReturn (t, e')
  else raise (Failure ("return gives " ^ string_of_primitive t ^ "
expected " ^
string_of_typ return_typ ^ " in " ^ string_of_expr e))

(* A block is correct if each statement is correct and nothing
follows any Return statement.  Nested blocks are flattened. *)
  | Block s1 ->
  let rec check_stmt_list = function
    [Return _ as s] -> [check_stmt symbols return_typ s]
    | Return _ :: _ -> raise (Failure "nothing may follow a
return")
    | Block s1 :: ss -> check_stmt_list (s1 @ ss) (* Flatten
blocks *)
    | s :: ss -> check_stmt symbols return_typ s ::
check_stmt_list ss
    | [] -> []
  in SBlock(check_stmt_list s1)
  | Bind(v1, v2) ->
  let lt = prim_of_dt (type_of_identifier symbols v1) and rt =
prim_of_dt (type_of_identifier symbols v2) in
  if Contract = check_bind lt rt then SBind((v1, v2))
  else raise (Failure("Bind syntax is not correct"))

  | VarInit (t, v, e) -> SVarInit(t, v, expr symbols e)
  | LisActInit (t, v, s) ->
  let rec check_stmt_list = function
    [Return _ as s] -> [check_stmt symbols return_typ s]
    | Return _ :: _ -> raise (Failure "nothing may follow a return")
    | Block s1 :: ss -> check_stmt_list (s1 @ ss) (* Flatten blocks *)
    | s :: ss -> check_stmt symbols return_typ s ::
check_stmt_list ss
    | [] -> []
  in
  let t_prim = prim_of_dt t in
  if ((t_prim = Listener) || (t_prim = Action)) then SLisActInit(t, v,
check_stmt_list s)
  else raise (Failure ("expected listener or action when assigning to "

```

```

^ v ^
  " but instead got " ^ string_of_primitive t_prim))
  | Resolve e -> let (lt, e') = expr symbols e in SResolve(lt, e')
  | Reject -> SReject

in

let check_function func =
  (* Make sure no formals or locals are void or duplicates *)
  check_binds "formal" func.formals;

  let localVars = (locals func.body) in
  check_binds "local" localVars;

  (* 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 (func.formals @ localVars )
  in

  (* body of check_function *)
  { styp = func.typ;
    sfname = func.fname;
    sformals = func.formals;
    sbody = match check_stmt symbols func.typ (Block func.body) with
    SBlock(s1) -> s1
    | _ -> raise (Failure ("internal error: block didn't become a
block?"))
  }
  in

let statement_vars = locals statements in
check_binds "global" statement_vars;

let statement_symbols = List.fold_left (fun m (ty, name) -> StringMap.add
name ty m)
                          StringMap.empty ( statement_vars )
in

let apply_funs statement = check_stmt statement_symbols (Primitive(Void))
statement in

```

```
(List.map check_function functions, List.map apply_funs statements)
```

Sast.ml

```
open Ast

type sexpr = primitive * sx
and sx =
  | SIntLit of int
  | SBoolLit of bool
  | SFloatLit of float
  | SStringLit of string
  | SVar of string
  | SBinop of sexpr * op * sexpr
  | SUnop of uop * sexpr
  | SAsn of string * sexpr
  | SCall of string * sexpr list
  (* | SArrayIndex of string * int TODO: Arrays *)
  | SInnerExpr of sexpr
  | SNoexpr
```



```

type sstatement =
  SExpr of sexpr
| SFor of sexpr * sexpr * sexpr * sstatement
| SWhile of sexpr * sstatement
| SIf of sexpr * sstatement * sstatement
  | SBlock of sstatement list
  | SVarInit of typ * string * sexpr
  | SLisActInit of typ * string * sstatement list
| SBreak
| SContinue
  | SBind of string * string
| SReturn of sexpr
  | SResolve of sexpr
| SReject

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

type sprogram = sfunc_decl list

(* Pretty-printing functions *)

let rec string_of_sexpr (t, e) =
  "(" ^ string_of_primitive t ^ " : " ^ (match e with
    SIntLit(l) -> string_of_int l
  | SBoolLit(true) -> "true"
  | SBoolLit(false) -> "false"
  | SStringLit(s) -> s
  | SFloatLit(l) -> string_of_float l
  | SVar(s) -> s
  | SBinop(e1, o, e2) ->
    string_of_sexpr e1 ^ " " ^ string_of_op o ^ " " ^ string_of_sexpr e2
  | SUNop(o, e) -> string_of_uop o ^ string_of_sexpr e
  | SAsn(v, e) -> v ^ " = " ^ string_of_sexpr e
  | SCall(f, e1) ->

```

```

    f ^ "(" ^ String.concat ", " (List.map string_of_sexpr e1) ^ ")"
  | SNoexpr -> ""
  (* | SArrayIndex(v, i) -> v ^ "[" ^ string_of_int i ^ "]" TODO: Arrays
  *)
  | SInnerExpr(e) -> "(" ^ string_of_sexpr e ^ ")"
    ^ ")"

let rec string_of_sstmt = function
  SBlock(stmts) ->
    "{\n" ^ String.concat "" (List.map string_of_sstmt stmts) ^ "}\n"
  SExpr(expr) -> string_of_sexpr expr ^ ";\n";
  SReturn(expr) -> "return " ^ string_of_sexpr expr ^ ";\n";
  SIf(e, s, SBlock([])) ->
    "if (" ^ string_of_sexpr e ^ ")\n" ^ string_of_sstmt s
  SIf(e, s1, s2) -> "if (" ^ string_of_sexpr e ^ ")\n" ^
    string_of_sstmt s1 ^ "else\n" ^ string_of_sstmt s2
  SFor(e1, e2, e3, s) ->
    "for (" ^ string_of_sexpr e1 ^ " ; " ^ string_of_sexpr e2 ^ " ; " ^
    string_of_sexpr e3 ^ ") " ^ string_of_sstmt s
  SWhile(e, s) -> "while (" ^ string_of_sexpr e ^ ") " ^ string_of_sstmt
  s
  | SVarInit(t, v, e) -> string_of_typ t ^ " " ^ v ^ " " ^
string_of_sexpr e
  | SListActInit(t, v, s) -> string_of_typ t ^ " " ^ v ^ " " ^
(String.concat "" (List.map string_of_sstmt s))
  | SBind(v1, v2) -> v1 ^ " -> " ^ v2
  | SBreak -> "break\n"
  | SContinue -> "continue\n"
  | SReject -> "reject\n"
  | SResolve(e) -> "resolve" ^ string_of_sexpr e ^ "\n"

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 (functions, statements) =
  String.concat "" (List.map string_of_sfdecl functions) ^ "\n" ^
  String.concat "\n" (List.map string_of_sstmt statements)

```

Scanner.mll

```
{ open Parser }

let digit = ['0' - '9']
let digits = digit+
let ascii = ([' ' - '!' '#' - '[' ] - '~')
let esc = '\\\ ['\\\ ' ' ' ' ' 'n' 'r' 't']

rule token =
  parse
  (*white space*)
  [' ' '\t' '\r' '\n'] { token lexbuf }

  (*Punctuators*)
  | '[' { OPENSQ }
  | ']' { CLOSESQ }
  | '(' { OPENPAR }
```

```
| ')' { CLOSEPAR }
| '{' { OPENBLOCK }
| '}' { CLOSEBLOCK }
| ';' { SEMICOLON }
| ',' { COMMA }

(*arithmetic*)
| '+' { PLUS }
| '-' { MINUS }
| '*' { TIMES }
| '/' { DIVIDE }
| '%' { MODULO }

(*unary operators *)
| '!' { NOT }

(*Binary Operators*)
| '|' { BOR }
| '&' { BAND }
| ">>" { BRIGHT }
| "<<" { BLEFT }
| '^' { BXOR }
| "!=" { NOTEQ }
| "==" { EQ }

(*key words*)
| "resolve" { RESOLVE }
| "reject" { REJECT }
| "Action" { ACT }
| "if" { IF }
| "for" { FOR }
| "true" { BOOL_LIT (true) }
| "false" { BOOL_LIT (false) }
| "else" { ELSE }
| "while" { WHILE }
| "break" { BREAK }
| "continue" { CONTINUE }
| "return" { RETURN }

(* logical operators*)
| "&&" { AND }
| "||" { OR }
```

```

| '!' { NOT }
| '<' { LT }
| "<=" { LTE }
| '>' { GT }
| ">=" { GTE }

(*special operators*)
| "->" { BIND }
| "[" { ARRDEC }

(*Assignment*)
| "=" { ASSIGN }

(*data types*)
| "int" { INT }
| "float" { FLOAT }
| "bool" { BOOL }
| "String" { STRING }
| "void" { VOID }
| "Contract" { CON }
| "Listener" { LIS }

(*data values*)
| "null" { NULL }
| eof { EOF }

(*literals*)
| digits as lit { INT_LIT(int_of_string lit) }
| digits '.' digit* ( ['e' 'E'] ['+' '-']? digits )? as lit {
FLOAT_LIT(float_of_string lit)}
| '"' ((ascii | esc)* as s)'"' { STRING_LIT(s) }
| ['a'-'z' 'A'-'Z']['a'-'z' 'A'-'Z' '0'-'9' '_' ]* as id { VARIABLE(id) }
| eof { EOF }

```

Codegen.ml

```
open Sast
module L = Llvm
module A = Ast
module StringMap = Map.Make(String)

(*translate : Sast.program -> Llvm.module*)
let translate program =

  (*Decompose program*)
  let (fdecls, stmts) = program in (*List to store function decls,
```

```

statements, action decls and listener decls*)
  let context = L.global_context() in
    let the_module = L.create_module context "Scolang" in

      (*===== TYPES =====*)
      (*Get types from the context and assigning them to local variables*)
      let function_ptr_type = L.function_type (L.i32_type context)
(Array.of_list([]))
      and function_ptr2_type = L.function_type (L.void_type context)
(Array.of_list([]))
      and 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 void_t = L.void_type context in
      (*pointer types and contract*)
      let str_t = L.pointer_type (L.i8_type context)
      and listener_t = L.pointer_type function_ptr_type
      (* and contract_t = L.i32_type context TODO: Get contracts working *)
      and action_t = L.pointer_type function_ptr2_type
      in

      (*Return the LLVM type for Scolang type*)
      let ltype_of_primitive = function
A.Int -> i32_t
| A.Bool -> i1_t
| A.String -> str_t
| A.Float -> float_t
| A.Void -> void_t
| A.Contract -> i32_t
| A.Listener -> listener_t
| A.Action -> action_t
      in

      (* TODO!!!! FIGURE OUT TYPE HEIRARCHY AND FIX IT*)
      (*=====*)

      (* Function to convert from typ type to primitive type *)
      let prim_of_dt = function
      | A.Primitive(p) -> p
      (* | A.Array(p, _) -> p TODO: Arrays *)
      in

```

```

(*===== EXTERNAL FUNCTIONS =====*)

let print_t : L.lltype =
L.var_arg_function_type i32_t [| L.pointer_type i8_t |] in
let print_func : L.llvalue =
L.declare_function "print" print_t the_module 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 bindval_t : L.lltype =
L.function_type i32_t [|L.pointer_type function_ptr_type;
L.pointer_type function_ptr2_type|] in
let bindval_func : L.llvalue =
L.declare_function "bind_val" bindval_t the_module in

let query_t : L.lltype =
L.var_arg_function_type i32_t [| L.pointer_type i8_t |] in
let query_func : L.llvalue =
L.declare_function "query" query_t the_module in

let webhook_t : L.lltype =
L.var_arg_function_type i1_t [| i32_t |] in
let webhook_func : L.llvalue =
L.declare_function "webhook" webhook_t the_module in

let system_call_t : L.lltype =
L.var_arg_function_type i32_t [| L.pointer_type i8_t |] in
let system_call_func : L.llvalue =
L.declare_function "system_call" system_call_t the_module in

(*=====*)

(*===== GLOBAL VARIABLES =====*)
(* TODO Figure out this matching here*)
(* TODO Actually store global values in the global var map *)

(*Function to check whether something is an initialisation for global

```



```

var*)
  let is_global_initialization stmt = match stmt with
  SVarInit(_,_,_) -> true
  | _ -> false
  in

  let get_global_name stmt = match stmt with
  SVarInit(t,n,_) -> prim_of_dt t,n
  | _ -> failwith ("Globab extraction failing " ^ string_of_sstmt
stmt)
  in

  let globals_list = List.filter is_global_initialization stmts
  in
  let globals = List.map get_global_name globals_list
  in
  (*=====*)

  (*===== EXPOSED STATEMENTS =====*)
  let is_exposed_statement stmt = match stmt with
  SVarInit(_,_,_) -> true
  | SLisActInit(_,_,_) -> false
  | _ -> true
  in

  let pure_statements = List.filter is_exposed_statement stmts
  in
  (*=====*)

  (* ===== Create an array of Listeners and Action Functions =====
*)
  let all_actions =
  let pop_actions action_list stmt = match stmt with
  SLisActInit(t, n, s) ->
    if A.Action = prim_of_dt t then
    {
    styp = A.Primitive(A.Void);
    sfname = n;
    sformals = [];
    sbody = s
    } :: action_list
    else (action_list)

```

```

| _ -> action_list
in List.fold_left pop_actions [] stmts
in

let all_listeners =
let pop_listeners listener_list stmt = match stmt with
SLisActInit(t, n, s) ->
    if A.Listener = prim_of_dt t then
    {
    styp = A.Primitive(A.Int);
    sfname = n;
    sformals = [];
    sbody = s
    } :: listener_list
    else (listener_list)
| _ -> listener_list
in List.fold_left pop_listeners [] stmts
in

(* TODO: Scan for actions and listeners inside of functions *)

(* Create a string map of actions *)
let action_decls: (L.llvalue * sfunc_decl) StringMap.t =
let function_decl map action =
let name = action.sfname in
let ftype = L.function_type (ltype_of_primitive A.Void)
(Array.of_list([])) in
StringMap.add name (L.define_function name ftype the_module, action)
map
in
List.fold_left function_decl StringMap.empty all_actions
in

(* Create a string map of listeners *)
let listener_decls: (L.llvalue * sfunc_decl) StringMap.t =
let function_decl map listener =
let name = listener.sfname in
let ftype = L.function_type (ltype_of_primitive A.Int)
(Array.of_list([])) in
StringMap.add name (L.define_function name ftype the_module,
listener) map
in

```

```

List.fold_left function_decl StringMap.empty all_listeners
in

(*===== CREATING MAIN FUNCTION =====*)
(*We're now going to create a list of functions from our fdecls and
build our faux main with pure statements*)
let functions =
let main_fdecl = {
    styp = A.Primitive(A.Int);
    sfname = "main";
    sformals = [];
    sbody = List.rev(SReturn(A.Int, SIntLit(0)) ::
List.rev(pure_statements))
}
in main_fdecl :: fdecls
in
(*=====*)

(*===== GLOBAL VARS =====*)
(* Store the global variables in a string map *)
(*TODO LOOK AT LLVALUE STRINGMAP*)
let global_vars : L.llvalue StringMap.t =
let global_var map (t, n) =
let init = match t with
    A.Float -> L.const_float (ltype_of_primitive t) 0.0
    | A.String -> L.const_null (ltype_of_primitive t)
    | A.Void -> L.const_null (ltype_of_primitive t)
    | A.Action -> L.const_null (ltype_of_primitive t)
    | A.Listener -> L.const_null (ltype_of_primitive t)
    | _ -> L.const_int (ltype_of_primitive t) 0
in StringMap.add n (L.define_global n init the_module) map
in List.fold_left global_var StringMap.empty globals
in
(*=====*)

(*===== FUNCTIONS AND FUNC. POPULATION =====*)
let function_decls : (L.llvalue * sfunc_decl) StringMap.t =
let function_decl map fdecl =
let name = fdecl.sfname
and formal_types = Array.of_list (List.map (fun (t, _) ->
ltype_of_primitive (prim_of_dt t)) fdecl.sformals)

```

```

        in
        let ftype = L.function_type (ltype_of_primitive (prim_of_dt
fdecl.styp)) formal_types in
        StringMap.add name (L.define_function name ftype the_module, fdecl)
map
    in
    List.fold_left function_decl StringMap.empty functions
    in

    let build_function_body decl_list fdecl =
    let (the_function, _) = StringMap.find fdecl.sfname decl_list 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
in

    let local_vars =
    let add_formal var_map (formal_type, formal_name) param =
L.set_value_name formal_name param;
    let local = L.build_alloca (ltype_of_primitive (prim_of_dt
formal_type)) formal_name builder in
        ignore(L.build_store param local builder);
        StringMap.add formal_name local var_map
and add_local map (formal_type, formal_name) =
    let local_var = L.build_alloca (ltype_of_primitive (prim_of_dt
formal_type)) formal_name builder in
        StringMap.add formal_name local_var map
    in

    let formals = List.fold_left2 add_formal StringMap.empty
fdecl.sformals
(Array.to_list (L.params the_function)) in

    let function_locals =
    let get_locals_from_fbody function_body =
        let get_vdecl locals_list stmt = match stmt with
SVarInit(typ, string, _) ->
        if( fdecl.sfname = "main")
        then (typ, string) :: locals_list
        else (typ, string) :: locals_list
        | _ -> locals_list

```

```

        in List.fold_left get_vdecl [] function_body
in get_locals_from_fbody fdecl.sbody
in List.fold_left add_local formals function_locals
in

let lookup n = try StringMap.find n local_vars
with Not_found -> StringMap.find n global_vars
in

let lookup_actions n = try StringMap.find n action_decls
with Not_found -> raise (Failure ("searched for an undefined action "
^ n))
in

let lookup_listeners n = try StringMap.find n listener_decls
with Not_found -> raise (Failure ("searched for an undefined listener
" ^ n))
in

(*=====*)

let rec expr builder ((_, e) : sexpr) = match e with
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 float_t l
| SStringLit s -> L.build_global_stringptr s "str" builder
| SVar s -> L.build_load (lookup s) s builder
| SBinop (e1, op, e2) ->
let e1' = expr builder e1 and
e2' = expr builder e2 in
(match op with
A.Lt -> L.build_icmp L.Icmp.Slt
| A.Lte -> L.build_icmp L.Icmp.Sle
| A.Gt -> L.build_icmp L.Icmp.Sgt
| A.Gte -> L.build_icmp L.Icmp.Sge
| A.Eq -> L.build_icmp L.Icmp.Eq
| A.NotEq -> L.build_icmp L.Icmp.Ne
| A.And -> L.build_and
| A.Or -> L.build_or
| A.Plus -> L.build_add
| A.Minus -> L.build_sub

```

```

| A.Times -> L.build_mul
| A.Divide -> L.build_sdiv
| A.Modulo -> L.build_srem
| A.Band -> L.build_and
| A.Bor -> L.build_or
| A.Bright -> L.build_lshr
| A.Bleft -> L.build_shl
| A.Bxor -> L.build_xor
) e1' e2' "tmp" builder
| SUnop (op, e) -> let e' = expr builder e in
( match op with A.Not -> L.build_not) e' "tmp" builder
| SNoexpr -> L.const_int i32_t 1
| SAsn (s,e) -> let e' = expr builder e in
ignore(L.build_store e' (lookup s) builder); e'
(* | SArrayIndex(v, i) TODO: Arrays *)
| SInnerExpr(e) -> expr builder e
| SCall("print", [e]) -> L.build_call print_func [| (expr builder e)
]|] "print" builder
| SCall("printi", [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 ("webhook", [e]) -> L.build_call webhook_func [| (expr
builder e) |] "webhook" builder
| SCall ("system_call", [e]) -> L.build_call system_call_func [|
(expr builder e) |] "system_call" builder
| SCall("query", [e]) -> L.build_call query_func [| (expr builder
e)|] "query" builder
| SCall(f, args) ->
let (fdef, fdecl) = StringMap.find f function_decls in
let llargs = List.rev (List.map (expr builder)(List.rev args)) in
let result = (match fdecl.styp with
A.Primitive(A.Void) -> "" | _ -> f^"_result") in
L.build_call fdef (Array.of_list llargs) result builder 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
| SVarInit (_, v, e) -> let e' = expr builder e in

```

```

                                ignore (L.build_store e' (lookup v) builder);
builder
  | SBlock s1 -> List.fold_left stmt builder s1
  | SExpr e -> ignore(expr builder e); builder
  | SBind (v1, v2) ->
    let (action_llvalue, _) = lookup_actions v2
    and (listener_llvalue, _) = lookup_listeners v1 in
    ignore(L.build_call bindval_func [|listener_llvalue; action_llvalue|]
"bind_val" builder); builder
  | SReturn e -> ignore(match fdecl.styp with
    A.Primitive(A.Void) -> L.build_ret_void builder |
    _ -> 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

| SFor (e1, e2, e3, body) -> stmt builder ( SBlock [SEExpr e1;
SWhile(e2, SBlock [body; SEExpr e3])])
| SResolve e -> ignore(L.build_ret (expr builder e) builder); builder
| SReject -> ignore(L.build_ret_void builder); builder
| SBreak -> ignore(L.build_ret_void builder); builder
| SContinue -> builder
| SLisActInit(_, _, _) -> builder (* This case is already handled *)
in
let builder = stmt builder (SBlock fdecl.sbody) in

(* Add a return if the last block falls off the end *)
add_terminal builder (match fdecl.styp with
A.Primitive(A.Void) -> L.build_ret_void
| A.Primitive(A.Float) -> L.build_ret (L.const_float float_t 0.0)
| t -> L.build_ret (L.const_int (ltype_of_primitive (prim_of_dt t))
0))
in
List.iter (build_function_body function_decls) functions;
List.iter (build_function_body action_decls) all_actions;
List.iter (build_function_body listener_decls) all_listeners;
the_module

```

Compile.sh

```

#!/bin/bash

make scolang.native
./scolang.native $1.sav > $1.ll
llc $1.ll > $1.s
gcc -o $1 $1.s stdlib/scolang_stdlib.o -lcurl -no-pie
rm -rf $1.s $1.ll

```


Dockerfile

```
# Based on 18.04 LTS
FROM ubuntu:bionic

RUN apt-get -yq update && \
    apt-get -y upgrade && \
    apt-get -yq --no-install-suggests --no-install-recommends install \
    ocaml \
    menhir \
    llvm-6.0 \
    llvm-6.0-dev \
    m4 \
    git \
    aspcud \
    ca-certificates \
    python2.7 \
    pkg-config \
    vim \
    nano \
    cmake \
    opam

RUN ln -s /usr/bin/lli-6.0 /usr/bin/lli
RUN ln -s /usr/bin/llc-6.0 /usr/bin/llc

RUN opam init
RUN opam install \
    llvm.6.0.0 \
    ocamlfind

WORKDIR /root

ENTRYPOINT ["opam", "config", "exec", "--"]

CMD ["bash"]
```

.travis.yml

```
language: c

services:
  - docker

before_install:
  - docker build -t ./
  - docker run -d -p 127.0.0.1:80:4567 -c
  -

script: bash -ex testing.sh
env:
  - OCAML_VERSION=4.05.0 OPAM_VERSION=2.0.0
notifications:
  email:
    recipients:
      - kanishk.vashisht+travisci@gmail.com
    on_success: change
    on_failure: always

addons:
  apt:
    sources:
      - ubuntu-toolchain-r-test
```

Test Suite

Fail-action1.sav

```
//resolve or reject present in action
Action action;

action = {
  print("I make even less sense");
  resolve();
}
```

```
}
```

Fail-action2.sav

```
//no return present in action  
Action action;  
  
action = {  
  print("action");  
}
```

Fail-assign1.sav

```
//assign boolean to int  
int a;  
a = true;
```

Fail-assign2.sav

```
//assign Listener to int  
Listener a;  
bool b;  
b = a;
```

Fail-assign3.sav

```
//action assigned to integer  
int a;  
Action b;  
a = b;
```

Fail-assign4.sav

```
//assign contract to int  
Contract a;  
int b;
```

```
b = a;
```

Fail-assign5.sav

```
//assign string to int  
  
String a;  
int b;  
b = a;
```

Fail-assign6.sav

```
//assigning non string to string  
String a;  
int b;  
a = b;
```

Fail-assign7.sav

```
//assigning non-listener to listener  
Contract c;  
Listener a;  
a = c;
```

Fail-assign8.sav

```
//assigning non-action to action  
String a;  
Action c;  
c = a;
```

Fail-assign9.sav

```
//assigning non boolean to boolean
```

```
bool c;  
Contract a;  
c = a;
```

Fail-assign10.sav

```
//assigning function to anything  
Listener c;  
  
int func(int a){  
    return a  
}  
  
c = func
```

Fail-assign11.sav

```
//assigning non listener->action to contract  
  
Contract c;  
int dd;  
c = dd;
```

Fail-assign12.sav

```
//assigning non char to char  
char a;  
int b;  
a = b;
```

Fail-Contract1.sav

```
//contract bound wrongly listener -> listener  
Listener a = {resolve()}  
Listener b = {resolve()}  
Contract c;
```

```
c = a -> b;
```

Fail-Contract2.sav

```
//contract bound wrongly action -> action  
Action a = {return 1;}  
Action b = {return 1;}  
Contract c;  
c = a -> b;
```

Fail-Contract3.sav

```
//contract bound wrongly contract -> contract  
Listener a = {resolve();}  
Action b = {return 1;}  
  
Contract c;  
Contract d;  
Contract e;  
  
c = a -> b;  
d = a -> b;  
e = c -> d;
```

Fail-dead1.sav

```
//code after resolve reject  
  
Listener listen;  
Action action;  
Contract contract;  
  
listen = {  
  resolve();  
  print("error!");  
}
```

```
action = {
  print("Worked!");
}

contract = listen -> action;
```

Fail-dead2.sav

```
//Declarations after listener/actions

Listener listen;
Action action;
Contract contract;

listen = {
  print("listen!");
  resolve();
}

action = {
  print("Worked!");
}

Listener illegal;

contract = listen -> action;
```

Fail-dead3.sav

```
//Else without if

Listener listen;
Action action;
Contract contract;

listen = {
  else{
    print("error")
  }
  resolve();
}
```

```
action = {
  print("Worked!");
}

contract = listen -> action;
```

Fail-dead4.sav

```
//elif without if

Listener listen;
Action action;
Contract contract;

listen = {
  elif(True){
    print("error")
  }
  resolve();
}

action = {
  print("Worked!");
}

contract = listen -> action;
```

Fail-dead5.sav

```
//Declarations after listener/actions

Listener listen;
Action action;
Listener listen
Contract contract;

listen = {
```



```
    print("listen!");
    resolve();
}

action = {
    print("Worked!");
}

contract = listen -> action;
```

Fail-dead6.sav

```
//call unexisting functions

foo()
```

Fail-expr1.sav

```
//bool op non-bool not allowed

bool a = True;
bool b;
int c = 1000;

b = a + c
```

Fail-expr2.sav

```
//float op non-float

bool a = false;
float b = 1.222;
float c;
c = b + a; //err
```

Fail-expr3.sav

```
//listener op non-listener
Listener a = { resolve(); }
int b = 100;

a = a + 100;
```

Fail-expr4.sav

```
//Action op anything
Action a = {print("action jack");}
int b = 100;
a = b * a;
```

Fail-expr5.sav

```
//String op non-string

String b = "ababba";
int a = 100;

b += a;
```

Fail-expr6.sav

```
//char op non-char

char a = 'v';
int b = 333;
a = b + a;
```

Fail-expr7.sav

```
//contract op non-contract

Listener a = {resolve();}
```

```
Action b = {print("test");}  
int d = 100;
```

```
Contract c = a->b;  
c = c*d;
```

Fail-expr8.sav

```
//unassigned expressions  
int d = 100;  
int c = 100;  
d*c;
```

Fail-expr9.sav

```
//no semi colon  
int a = 100
```

Fail-for1.sav

```
//unassigned variable initialization for  
for(c = 0; c < 10; c+=1) {}
```

Fail-for2.sav

```
//unassigned variable initialization in statement 2  
int c;  
for(c = 0; c < d; c+=1) {}
```

Fail-for3.sav

```
//unassigned variable initialization in statement 3  
int c;  
for(c = 0; c < 10; c+=d) {}
```

Fail-for4.sav

```
//statement 1 is not an initialization  
int c;  
for(c < 0; c < 10; c+=d) {}
```

Fail-for5.sav

```
//statement 2 is not a boolean  
int c;  
for(c = 0; c = 10; c+=d) {}
```

Fail-for6.sav

```
//statement 3 is not a statement  
int c;  
for(c = 0; c = 10; c < 4) {}
```

Fail-func1.sav

```
//duplicate functions  
  
int a(){  
    print("a");  
    return 0;  
}  
  
int a(){  
    print("c");  
    return 0;  
}
```

Fail-func2.sav

```
//returning type mismatch  
  
int a(){  
    print("a");  
    return "error";  
}  
a();
```

Fail-func3.sav

```
//duplicate formals  
  
int foo(int a, int a){  
    return a + a;  
}  
  
foo(2,2);
```

Fail-func4.sav

```
//illegal formals  
int foo(int a, int b){  
    return a + b;  
}  
foo(3, "err")
```

Fail-func5.sav

```
//expressions in formals  
  
int foo(int a, int b + 5){  
    return a+b;  
}  
  
foo(4,4);
```

Fail-func6.sav

```
//overloading inbuilt functions  
  
String print(String a){  
    return a;  
}  
  
print("error");
```

Fail-func7.sav

```
//wrong number of arguments

int foo(int a){
    return a;
}

foo(4,4);
```

Fail-while.out

```
int a;
a = 10;
while(a = a + 30){
    print("what???");
}
```

Test-Arithmetic.sav

```
int add(int x, int y)
{
    return x + y;
}

printi(add(17, 25));
printi(17+25);

int a = 10;
int b = 5;

printi(10-5);
printi(a-b);
printi(a*b);
printi(5*10);
printi(a/b);
```

```
printi(10/2);  
printi((10*5)/2-5*5);
```

Test-for.sav

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

Test-func.sav

```
int add(int a, int b)  
{  
    return a + b;  
}  
  
int fun(int x, int y)  
{  
    return 0;  
}  
  
int foo(int a)  
{  
    return a;  
}  
  
int bar(int a, bool b, int c) { return a + c; }  
  
int a;  
a = add(39, 3);  
printi(a);  
  
int i;  
i = 1;  
  
fun(i = 2, i = i+1);
```

```
printi(i);  
  
printi(bar(17, false, 25));
```

Test-if.sav

```
int i = 10;  
  
if(i == 20)  
{  
    print("check1");  
    i = i+ 10;  
}  
  
if(i == 10)  
{  
    print("check2");  
    i = i+10;  
}  
  
if(i == 10)  
{  
    print("check3");  
}  
else  
{  
  
    print("check4");  
}  
  
if(i == 30)  
{  
  
    print("check5");  
}  
else{  
    print("check7");  
}
```



```
if(i == 10)
{
    print("check8");
}
else
{
    print("check9");
}
```

Test-prim-assignment.sav

```
int a = 10;
int b = 5;
printi(a);
printi(b);
b = a;
printi(a);
printi(b);

float f1 = 5.5;
float f2 = 3.30;
printf(f1);
printf(f2);
f2 = f1;
printf(f1);
printf(f2);

String s1 = "Hello";
String s2 = "World";

print(s1);
print(s2);

s2 = s1;

print(s1);
print(s2);
```

Test-print.sav

```
int i = 0;
float f = 1.4;
String out = "hello";

printi(i);
printf(f);
print(out);
```

Test-return.sav

```
String strfun()
{
    return "hello";
}
int intfun()
{
    return 1;
}

float floatfun()
{
    return 2.4;
}
print(strfun());

printi(intfun());

printf(floatfun());
```

Test-while1.sav

```
int i;
i = 5;
while (i > 0) {
    printi(i);
    i = i - 1;
}
```

```
printi(42);
```

Test-while2.sav

```
int foo(int a)
{
    int j;
    j = 0;
    while (a > 0) {
        j = j + 2;
        a = a - 1;
    }
    return j;
}

printi(foo(7));
```

Test-world.sav

```
print("Hello world");
```

Git Commits

commit b9640bb4edbb32fcc9a3b0ed62e4a30556a3935f
Author: vnv-varun <vnv-varun@users.noreply.github.com>
Date: Wed Dec 19 20:51:54 2018 -0500

Final Commit Changes -- Cleanup

commit 6a3d3c568845eb83f114823f6c26b31b17e37dd9
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Wed Dec 19 19:00:52 2018 +0000

Working demo

commit 375e1f0242e9d2c12617779749d21801297cbf0d
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Wed Dec 19 18:43:46 2018 +0000

Working demo

commit df33272807ed0db52e69086c7242b0b692594cee
Author: vnv-varun <vnv-varun@users.noreply.github.com>
Date: Wed Dec 19 13:32:47 2018 -0500

Removed List

commit f1e74c7010bc01d7b09f1bee166e02a46f26ed08
Merge: 3aecc45 13aba6d
Author: vnv-varun <vnv-varun@users.noreply.github.com>
Date: Wed Dec 19 13:31:32 2018 -0500

Merge branch 'master' of <https://github.com/KanishkVashisht/scolang>

commit e8836391cb1cd5fce1d3387d6f2e97f5f5e8ec52
Author: vnv-varun <vnv-varun@users.noreply.github.com>
Date: Tue Dec 18 22:10:46 2018 -0500

Update world.sav

commit 415ecd214fedde31196b3ffb9a568decc1660eb0
Author: vnv-varun <vnv-varun@users.noreply.github.com>
Date: Tue Dec 18 14:04:09 2018 -0500

Output files for previous commit

commit 9cd34214bb6af4a20d534cc54a342ede4fa5a985
Author: vnv-varun <vnv-varun@users.noreply.github.com>
Date: Tue Dec 18 14:02:33 2018 -0500

More Fail Tests w/o output files

commit 2898e08ff2de6d881452d3f53e00c623bda57795
Author: vnv-varun <vnv-varun@users.noreply.github.com>
Date: Tue Dec 18 14:01:04 2018 -0500

Added More Fail Tests

commit 7694bb29d47d13e7d270d6966ab056706caa01df
Author: vnv-varun <vnv-varun@users.noreply.github.com>
Date: Tue Dec 18 13:58:16 2018 -0500

Removed Empty Function Test

commit a76b7d4eee71fb890669eafac000856a33d6ee51
Author: vnv-varun <vnv-varun@users.noreply.github.com>
Date: Tue Dec 18 13:56:21 2018 -0500

Testing is life

commit 5831ccc3d2bbb01f60a31d3e8e398f16a3001df2
Author: vnv-varun <vnv-varun@users.noreply.github.com>
Date: Tue Dec 18 13:52:59 2018 -0500

Tests

commit aedc249db1b44b9e4b818e90fc7233f5a4239b9d
Author: vnv-varun <vnv-varun@users.noreply.github.com>
Date: Tue Dec 18 13:51:08 2018 -0500

Added Fail-Expr Into Testing

commit a4ab080421eb450fb19777ddca8c6b408cd8d603
Author: vnv-varun <vnv-varun@users.noreply.github.com>
Date: Tue Dec 18 13:46:11 2018 -0500

Added Output file and More Tests

commit 80bf54608b6a1e87bf2266993fad4466e7dc27ea
Author: vnv-varun <vnv-varun@users.noreply.github.com>
Date: Tue Dec 18 13:42:54 2018 -0500

Added a fail test

commit 0b405f4af1d6cb4820782a5b3cc8dd3a0c31d16e
Author: vnv-varun <vnv-varun@users.noreply.github.com>
Date: Tue Dec 18 13:41:01 2018 -0500

Changed test directory

commit 254fe93df46657310561f3e54e1b46fdfca219e9
Author: vnv-varun <vnv-varun@users.noreply.github.com>
Date: Tue Dec 18 13:38:07 2018 -0500

Added Tests & Curl To ENV -- Need to compile files for test

commit 724e0b32ab611c25a3331d0ff78f87b4c3cd3f2b
Author: vnv-varun <vnm-varun@users.noreply.github.com>
Date: Tue Dec 18 13:13:15 2018 -0500

Minor Test Changes

commit 04025d15b055c313fe53bcb20dc01d1ac1a77a68
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Mon Dec 17 02:24:26 2018 -0500

readme adn notes

commit d944b100d6f9a04759f9fa28c15422d1b3597a10
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Mon Dec 17 05:11:54 2018 +0000

renamed

commit 1e3b7cdd2f09a4ab6b2a5a86822af22a45c4e993
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Sun Dec 16 23:59:57 2018 -0500

merge

commit 43094c7eb661372ad13ceac119470aba1e98cce8
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Mon Dec 17 04:55:46 2018 +0000

updating testing

commit 4ded4a7e418224bce65efa22af2df88876ba5bbb
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Mon Dec 17 04:55:29 2018 +0000

updated testing

commit 8e76a6386706f3d88ea19a31d1e2315fec0c4f6e
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Sun Dec 16 23:41:21 2018 -0500

returned continue

commit 298d7f2c87eed4a001926a18da376efb3bbe2f33
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Sun Dec 16 23:40:29 2018 -0500

updated continue interpet

commit ad0d9014641a10f8d133572c4ee8370d6123fd03
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Sun Dec 16 21:25:50 2018 -0500

testing lcurl

commit fb4a0196245118ebe8a8090036071551baae3841
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Sun Dec 16 21:15:22 2018 -0500

stdlibby

commit 4a212d17a8636c21884bdbc4d69ed0f713d9db8a
Merge: b05e892 ab168de
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Sun Dec 16 21:09:33 2018 -0500

Merge branch 'master' of <https://github.com/KanishkVashisht/scolang>

commit b05e892df5bea2cbe18ed465390865cad97ecc25
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Sun Dec 16 21:09:27 2018 -0500

fixes to world.exe

commit ab168dee4aa6a5fd60600354eed84c5e46710e40
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Mon Dec 17 00:57:27 2018 +0000

world exe

commit f6c84e60a155d7000825e695b08e088a639ebcd0
Merge: 5e10a9b f20a898
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>

Date: Sun Dec 16 19:55:21 2018 -0500

Merge branch 'master' of <https://github.com/KanishkVashisht/scolang>

commit 5e10a9bf470b46e29ade1034aaa1bad1062f01ae
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Sun Dec 16 19:55:14 2018 -0500

demo

commit f20a8987515f60eeedf51c89152fb21ea431991f
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Mon Dec 17 00:53:07 2018 +0000

porting the executable

commit 7bce37ae85d6df84e3ad7c46482b2d69dbef9274
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Sun Dec 16 19:46:56 2018 -0500

semant wav

commit 1f5c2b8a59da78bbf64bbda575cca032bde2e3f6
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Sun Dec 16 19:45:07 2018 -0500

changes to semant

commit b10fb94792e394ef84be2ce1422e56f39af85b9b
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Sun Dec 16 19:42:09 2018 -0500

codegen

commit 7dda5da9a40447bce1c0e6f83b255d3f1b2a835e
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Sun Dec 16 19:37:20 2018 -0500

work it

commit e5914399f204774d71db23f1b288ad0bda471440
Merge: 445d09a f371bb2
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Sun Dec 16 19:36:09 2018 -0500

Merge branch 'master' of <https://github.com/KanishkVashisht/scolang>

commit f371bb272d1faf01e90d029e066878d727956a40
Author: Sambhav Anand <sambhav182@gmail.com>
Date: Sun Dec 16 19:36:00 2018 -0500

freed up

commit 02bd7bbd145c4f3f1cf1cb04faa9a5a189880fdf
Merge: 59bd3a9 24e6d74
Author: Sambhav Anand <sambhav182@gmail.com>
Date: Sun Dec 16 19:35:24 2018 -0500

Merge branch 'master' of <https://github.com/KanishkVashisht/scolang>

commit 59bd3a9987c481e811d571bef6930b0da391166b
Author: Sambhav Anand <sambhav182@gmail.com>
Date: Sun Dec 16 19:35:07 2018 -0500

Made query to accept one string

commit 24e6d744f28b609e0adbd213206bde4bd5b1c63d
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Sun Dec 16 19:34:34 2018 -0500

exception

commit c73b649057423940a5b88e9f765d78f49f246941
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Sun Dec 16 19:31:34 2018 -0500

lame semant

commit 9e94e7951d1e49646f7eaec956668a70f91f3630
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Sun Dec 16 19:28:26 2018 -0500

updated stdlib to include query

commit d6a8d506042b0d41d5f6459d1289e5eb66505e19
Author: Sambhav Anand <sambhav182@gmail.com>
Date: Sun Dec 16 18:59:19 2018 -0500

query function signature changed

commit d1ae00478ddf86ce33da2a206987ec83d13b0178
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Sun Dec 16 23:57:44 2018 +0000

new testing script

commit 73f95e95b7c2f4049bf40ee95fe64aca9c4fd0e8
Merge: 892e9e9 27912e7
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Sun Dec 16 23:50:13 2018 +0000

Merge branch 'master' of <https://github.com/KanishkVashisht/scolang>

commit 27912e7fd6f236c7c6d87b2bb8038c737ac99f5b
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Sun Dec 16 18:50:02 2018 -0500

bugz

commit 892e9e9f403096e1ab20986be4acddab88579d6a
Merge: a0a5c3e d721567
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Sun Dec 16 23:49:36 2018 +0000

Merge branch 'master' of <https://github.com/KanishkVashisht/scolang>

commit d72156720d2410b9f3f1def121b22306ce77924f
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Sun Dec 16 18:49:08 2018 -0500

adding print functionality

commit e39d88aa91e8bbd7091688095f7b46597d541022
Author: vnv-varun <vnv-varun@users.noreply.github.com>

Date: Sun Dec 16 14:46:32 2018 -0500

added -no-pie into compile script

commit 617849ca531663b4e4b6387febe5b1f4dac1022a
Author: vnv-varun <vnm-varun@users.noreply.github.com>
Date: Sat Dec 15 22:43:50 2018 -0500

Removed Array Test Files

commit 8632ad5002c702ebbc9030f54acad1b0e56c18a8
Author: vnv-varun <vnm-varun@users.noreply.github.com>
Date: Sat Dec 15 22:39:23 2018 -0500

Removed char from tests

commit b3909cf16a6f33031a9bc1c2f9037ed6bf513d86
Author: vnv-varun <vnm-varun@users.noreply.github.com>
Date: Sat Dec 15 22:35:25 2018 -0500

Removed Main from Testfiles

commit 23b753ee7cc7397f83472d5a34d08185fc77cbca
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Sat Dec 15 21:55:31 2018 -0500

ok next

commit 1f2268cb8d782804adacbc025f63865ee04c8194
Merge: 18e8131 b34f88a
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Sat Dec 15 21:52:03 2018 -0500

Merge branch 'master' of <https://github.com/KanishkVashisht/scolang>

commit 18e81316478306f68b32ce40ea80d236f54e10e6
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Sat Dec 15 21:51:54 2018 -0500

check

commit b34f88ab5aa65e1807752863af270439393a3e1a

Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>

Date: Sun Dec 16 02:36:20 2018 +0000

compile script

commit d86f3682bbb1246b027aef925d5e8ca097f8fc58

Merge: 33bd214 4ccc2e0

Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>

Date: Sat Dec 15 19:45:09 2018 +0000

Merge branch 'master' of <https://github.com/KanishkVashisht/scolang>

commit 33bd21436188adcb1359e82659db125f8dfea490

Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>

Date: Sat Dec 15 19:44:48 2018 +0000

Testall fix

commit 4ccc2e0d8c915fbb346d5e8e594dc1f0ae99bfae

Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>

Date: Sat Dec 15 01:13:50 2018 -0500

new testscript

commit 5e583655bf1b5d689aa2fc9eda2255cd5475ac44

Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>

Date: Sat Dec 15 01:09:24 2018 -0500

SCall

commit 6fd75cbee5d62b99002f9148a924f651a6f1a51c

Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>

Date: Sat Dec 15 01:08:41 2018 -0500

integrated system library into stdlib

commit 4b685272ba06fda585db8c483b0a1d35cae779f2

Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>

Date: Fri Dec 14 23:50:52 2018 -0500

added style to the test suite

commit 78e588782c3ef8751d059fa5c1abe7aef8fafc8a

Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Fri Dec 14 23:24:39 2018 -0500

cleanup

commit ad68242b4e087e1abb6b5c4b4bb37133905782b5
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Fri Dec 14 22:14:36 2018 -0500

mayb

commit f0bb4a2e164c198c604d1f6b170b5c726929e00e
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Fri Dec 14 22:10:35 2018 -0500

testing

commit 5b3f3f907ba4249399447161d43cec0903fdc68b
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Fri Dec 14 22:07:07 2018 -0500

webhook now accepts port

commit 830304545b54981a5858213ac423fb05d27e89b2
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Fri Dec 14 21:45:59 2018 -0500

exit

commit 29a48918ebc989dba872bd35daba091d75f5c127
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Fri Dec 14 21:45:26 2018 -0500

lol what did jae really teach me?

commit 31f3346775e8254a18413c3d3d600c0ac1fca13c
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Fri Dec 14 21:39:13 2018 -0500

forkin hell

commit 3e36d53741568f1fbfb87a20b89bfccfd30832c8
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>

Date: Fri Dec 14 21:31:39 2018 -0500

print function fixed

commit 4bf181e84c1a832448812c8dda8441bdc8a28ced
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Fri Dec 14 21:09:49 2018 -0500

Merge branch 'master' of <https://github.com/KanishkVashisht/scolang>

commit f2beef23a2cbd666206879013f8b37e4da5264c5
Merge: 5c3fb17 daad68f
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Fri Dec 14 21:09:09 2018 -0500

Merge branch 'master' of <https://github.com/KanishkVashisht/scolang>

commit 5c3fb1775ca9176d3146676ea28a7aa85b2ffed3
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Fri Dec 14 21:09:02 2018 -0500

testing fork

commit daad68f2479a3645ce451de6df756530af893e9b
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Fri Dec 14 19:02:08 2018 -0500

Remove commented out lines in scanner

commit 7cab8ac7da780dd7c40ad14272f78ab35908edaa
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Fri Dec 14 18:43:49 2018 -0500

Include time lib in webhook

commit 07052f6e161d8c5fc6d9f76f02ac68f7e8f87180
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Fri Dec 14 18:42:36 2018 -0500

Fix unused vars in semant

commit 63186a6b2fceedde6f87b511f413bcf14bed494
Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Fri Dec 14 18:41:31 2018 -0500

Fully comment

commit 598618f31e418f4007612ca6f00999c134121c19

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Fri Dec 14 18:40:52 2018 -0500

Add semant check for bind

commit 279346e77030272f3f370212b327cf6a2a51216f

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Fri Dec 14 18:39:19 2018 -0500

typ

commit 1eeca47e761f9570dc6353a2283f7fa29bff019d

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Fri Dec 14 18:38:53 2018 -0500

Match sinnerexpr

commit fafe62368f41207e55c259a15fb32d4cc9cec798

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Fri Dec 14 18:37:58 2018 -0500

typ instead of prim

commit 5c81d2411819ceb15774d26820daf3be534c9fa3

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Fri Dec 14 18:37:33 2018 -0500

Complete expr in semant

commit 8606ae95b2d2cfc8d428ee4874ad7a85f8d804ac

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Fri Dec 14 18:33:43 2018 -0500

Multi line string in semant

commit 3e12003a01655c7017eb1b978b3f6e65e51fcc5c

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Fri Dec 14 18:31:34 2018 -0500

Remove unused vars in codegen

commit 9876cbfee099a5794708a5dc3a8f436eb6912f0e

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Fri Dec 14 18:27:53 2018 -0500

fix statement codegeng

commit c138418314dafd7e398da935449821e71133e638f

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Fri Dec 14 18:24:12 2018 -0500

move sinnerexpr up

commit 6a9f268132b4b4c42dfa35dfc29aebba4f7c02d5

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Fri Dec 14 18:23:24 2018 -0500

Deprecate arrays

commit b7392607aa9bb418da108047060948d0986ccde4

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Fri Dec 14 17:58:52 2018 -0500

Complete string_of_sstmt

commit 6dc205d5c8544c3e2796eec5e0ed8636b2d70e60

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Fri Dec 14 17:55:44 2018 -0500

Fix syntax in string of sexpr

commit e053d6802fc9745614638fd9cde78fd6a3f1f165

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Fri Dec 14 17:55:04 2018 -0500

Fix printing snoexpr

commit 1142be83b4434c1df6aee6df33f6eda3940bf9b9

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Fri Dec 14 17:54:04 2018 -0500

Complete string_of_expr

commit 8b24fdd29f215db4ee07458eabe5b1b33c0bc6cc
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Fri Dec 14 17:49:21 2018 -0500

Char is deprecated officially

commit 9d54857e1f3a4fb8435ce43431bbacc641948dce
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Fri Dec 14 17:39:26 2018 -0500

Change expr to innerexpr for type disambiguation

commit e01ebe7de6b7aa896ec7bd6da2dd1c4c1abb1fb6
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Fri Dec 14 17:32:35 2018 -0500

Add back expr into expr print

commit ed1dc3732a236844540b5b3bf23f49dc31215c50
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Fri Dec 14 17:30:41 2018 -0500

fix return type for reject

commit 7b568dd67c1bc48bad3fbe8ded10f88cabd60a45
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Fri Dec 14 17:30:03 2018 -0500

Fix return type in resolve

commit ba837f4b2875eb66eae08a89ff5cdb0e0bd73c18
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Fri Dec 14 17:26:49 2018 -0500

move resolve and reject under statement

commit 9223db42922dfa8ef98951761f367f89f22cd5d7
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Fri Dec 14 17:25:14 2018 -0500

Move resolve and reject under statement

commit 6035990802fc4a1d90ab430a0fab951f33ec41d4
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Fri Dec 14 17:17:41 2018 -0500

AST should not have expr -> expr in print

commit 9837b610cb3ef52587a496e6db76215e1a9f9a69
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Fri Dec 14 17:14:03 2018 -0500

Attach all primitive to A

commit 44932807823af056e985c417855885227cce676f
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Fri Dec 14 17:12:43 2018 -0500

Primitive . A

commit b2324e6dc85a5ddcd2c232debcdbf6d1c0a6060c
Merge: 3b75203 6070a8b
Author: Sambhav Anand <sambhav182@gmail.com>
Date: Fri Dec 14 16:21:40 2018 -0500

removed main

commit 3b7520356e2e42091faf8cd6b599ba462bf39932
Author: Sambhav Anand <sambhav182@gmail.com>
Date: Fri Dec 14 16:20:59 2018 -0500

working query

commit 6070a8b3330cbb5bfc225c89db31bbe2fc4e3722
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Fri Dec 14 16:12:27 2018 -0500

Remove logging in codegen

commit 6eb6a6c51f23f44673648deb706c782bd02d30d4
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Fri Dec 14 16:04:13 2018 -0500

query fix

commit 1fa42479ee84046e56b2adecac884ef3d571bc24
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Fri Dec 14 16:02:30 2018 -0500

wurld

commit 325551a53fcfe74a1c6f34dd1ed6a8355d5e5fff
Merge: 779120c 04c8b98
Author: Sambhav Anand <sambhav182@gmail.com>
Date: Fri Dec 14 15:47:49 2018 -0500

Merge branch 'master' of <https://github.com/KanishkVashisht/scolang>

commit 779120c7240f88fb51dc6a6c2d4eb4038bf443a0
Author: Sambhav Anand <sambhav182@gmail.com>
Date: Fri Dec 14 15:47:44 2018 -0500

finished query

commit 04c8b98a06353e83956f5aeb8ec76ef0f8e2ca8
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Fri Dec 14 15:37:02 2018 -0500

query

commit 9e013ac0df6b22857d465cf3c3b70d64f3bed1f9
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Fri Dec 14 15:30:43 2018 -0500

why tho

commit 72278721dcab360d5eb55eb1d0724c1fa987c9e1
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Fri Dec 14 15:30:36 2018 -0500

void

commit c5e031729ea1fd037a4b21fb54453325784ff17e
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Fri Dec 14 15:28:16 2018 -0500

world.sav

commit 12032d2ea9d99d7d17f9c2f71818be5e7a253bc2
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Fri Dec 14 14:50:29 2018 -0500

query integrated

commit 43fd00dbf69e31fcc55e76da5c92c52efc5b10b1
Author: Sambhav Anand <sambhav182@gmail.com>
Date: Fri Dec 14 01:48:20 2018 -0500

Working on integrating query

commit 2320d7893135adb9da3dff3dc18c99c1ac2b26b5
Author: KanishkVashisht <kanishk.vashisht@gmail.com>
Date: Fri Dec 14 04:55:51 2018 +0000

working version

commit aae3b08a4298e4a11afe59c47db9b962c7873490
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Thu Dec 13 21:23:43 2018 -0500

webhook

commit e069501dffbeabf504f60c6dbd5f6c7fe08a827
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Thu Dec 13 21:21:49 2018 -0500

webhook returns bool

commit 247ee76e536889acd791f403abac085dee28713a
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Thu Dec 13 20:51:24 2018 -0500

yeye

commit fbe6b8385dc097ee04fc6f9fabdf1be8782cb7df
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Thu Dec 13 20:50:24 2018 -0500

builder

commit 0d41a1ea235fe64451b9723b73c2d8680f5872fc
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Thu Dec 13 20:40:34 2018 -0500

blocking

commit 6ee91ccdcffaa98d76bef3f273416a711198a26a
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Thu Dec 13 20:15:52 2018 -0500

blocking

commit 7ec56877a6acf8fad494f0295382924ce4489092
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Thu Dec 13 20:09:51 2018 -0500

erring

commit dcaa67c97502fe99c20b3a5ee8b24b5c83969a4d
Merge: b85ee19 787ae74
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Thu Dec 13 20:05:26 2018 -0500

Merge branch 'master' of <https://github.com/KanishkVashisht/scolang>

commit b85ee19360c60d5138281ffbacdc735abb247984
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Thu Dec 13 20:05:08 2018 -0500

codegen

commit 787ae74ea2242dca63aa112c28fb8d823a1fe117
Author: Sambhav Anand <sambhav182@gmail.com>
Date: Thu Dec 13 20:03:22 2018 -0500

Working version

commit 8e06aee7c2780f5e9ad3f4e76fdf7ed4aa464252
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Thu Dec 13 20:00:33 2018 -0500

Merge branch 'master' of <https://github.com/KanishkVashisht/scolang>

commit f0527ee3818c72709f7f9e352dae8befc94ff5b4
Merge: 670e3b8 1ac3499
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Thu Dec 13 19:58:05 2018 -0500

Merge branch 'master' of <https://github.com/KanishkVashisht/scolang>

commit 670e3b8c4a5fd04e2c421f82435587622ac10312
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Thu Dec 13 19:57:56 2018 -0500

Merge branch 'master' of <https://github.com/KanishkVashisht/scolang>

commit 1ac34999508becfac21ac4457336f57096753241
Merge: 89d2087 e7036c0
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Thu Dec 13 19:44:18 2018 -0500

Merge branch 'master' of [github.com:KanishkVashisht/scolang](https://github.com/KanishkVashisht/scolang)

commit 89d2087edddcfc7a0978490fbaaf481fadfe5da4
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Thu Dec 13 19:44:01 2018 -0500

link webhook in

commit e7036c0e42ebf4e899150cac1ca0548aa4ec2b22
Merge: 47ea834 2e677d0
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Thu Dec 13 19:40:28 2018 -0500

Merge branch 'master' of <https://github.com/KanishkVashisht/scolang>

commit 2e677d0649aea41edb9c6369cb9bd4a84318813e
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Thu Dec 13 19:18:46 2018 -0500

webhook takes nothing

commit 771ab27269bbb40a8398eee8cee3f8aa63b8e3ee

Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Thu Dec 13 19:15:21 2018 -0500

add webhook call to world.sav

commit 79e35c7be6370c81fb8b761641374413aa629434
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Thu Dec 13 19:14:40 2018 -0500

hmm didn't fix shift reduce

commit c6dd4b43963fd31d21c377803b0b8f0b4b600e20
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Thu Dec 13 19:13:49 2018 -0500

test shift reduce

commit 60f0b12404fda7a4f5edec49fe22f61340b936fa
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Thu Dec 13 19:12:09 2018 -0500

fix shift reduce

commit 367d90a36286e88149b222d57a618fb562fcf239
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Thu Dec 13 19:08:19 2018 -0500

wrap

commit d59d9b5397a92ffa6af6236f2f2f0b2f69f3fce7
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Thu Dec 13 18:58:24 2018 -0500

add paren

commit 11dc8a745fd2f475006a0e2db05f41f8ab6e19a5
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Thu Dec 13 18:42:57 2018 -0500

d

commit 103728dc90423f2b70c70418f38e50646cc6f7c2
Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Thu Dec 13 18:41:52 2018 -0500

add semi

commit 2b4797242104dfcf7633eb390f43a32be509b107

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Thu Dec 13 18:41:00 2018 -0500

change lgo

commit 9a06330bf76f6588a7db75c784bfcaad82a7a7fe

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Thu Dec 13 18:40:17 2018 -0500

add semicolon

commit 8651b466f7b63f9fd493aa67ebf7328b53d37ea7

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Thu Dec 13 18:39:50 2018 -0500

add log

commit 1fe80b59b3a432b585b5b5f91925d17290052bb5

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Thu Dec 13 18:38:04 2018 -0500

add_bind conditional

commit fd61a7a71c0d9413122ab92e8702e11ca42b8248

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Thu Dec 13 18:30:41 2018 -0500

none

commit 721f841384b570b872a851b6c5d2ca4b590df90e

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Thu Dec 13 18:30:00 2018 -0500

no param in webhook

commit a45fd487cf2f37a058b28a33c69ad6dedd4f3310

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Thu Dec 13 18:29:24 2018 -0500

add webhok to semant

commit 043c9ac7babe5e0d31ad67cb85c0a8693d38f1b4

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Thu Dec 13 18:25:06 2018 -0500

bind -> bind_val

commit e6584c0fd36c498e6f82a0c6931f2220e87d2066

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Thu Dec 13 18:24:17 2018 -0500

add webhook

commit 3d18ad895eca8b086247c7ea21d5599bb00896ea

Author: Sushanth Raman <s.raman@columbia.edu>

Date: Thu Dec 13 18:14:12 2018 -0500

fad

commit d6927edcf4324bca6033fe4b041a13a6fb18b10b

Author: Sushanth Raman <s.raman@columbia.edu>

Date: Thu Dec 13 18:04:24 2018 -0500

world.sav

commit cbd27cdd35e840de1b400c50bfd6eb3280a5c6ea

Author: Sushanth Raman <s.raman@columbia.edu>

Date: Thu Dec 13 17:54:21 2018 -0500

updated codegen

commit 934152d08225095e18604fcea1de4aa4bdd34344

Author: Sushanth Raman <s.raman@columbia.edu>

Date: Thu Dec 13 17:53:00 2018 -0500

updated codegen

commit 71ef9063299cedcaa58b98561b170e023cbbe259

Author: Sushanth Raman <s.raman@columbia.edu>

Date: Thu Dec 13 17:52:13 2018 -0500

updated codegen

commit f1fa4cd5257474cfd7f50da91f273b11bcba6bb4
Author: Sushanth Raman <s.raman@columbia.edu>
Date: Thu Dec 13 17:42:45 2018 -0500

updated codegen

commit b375e415f492c721a81e05885fed5e5e31c50a2
Author: Sushanth Raman <s.raman@columbia.edu>
Date: Thu Dec 13 17:34:14 2018 -0500

updated codegen

commit 7e14d97950481fcc5431d299316d590257c079cf
Author: Sushanth Raman <s.raman@columbia.edu>
Date: Thu Dec 13 17:30:06 2018 -0500

updated codegen

commit 8c069b89299b0035b50a71560046e7d976618674
Author: Sushanth Raman <s.raman@columbia.edu>
Date: Thu Dec 13 17:28:39 2018 -0500

updated codegen

commit d36f42c6c52793314b36e26b3e38ccbde76a4940
Author: Sushanth Raman <s.raman@columbia.edu>
Date: Thu Dec 13 17:28:21 2018 -0500

updated codegen

commit 3fd095df8833d3d08ae346915a92ea417f74c6f0
Author: Sushanth Raman <s.raman@columbia.edu>
Date: Thu Dec 13 17:27:29 2018 -0500

updated codegen

commit d4349470dd8ad695f81060dbb8d424587e7e37cc
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Thu Dec 13 17:18:32 2018 -0500

change bak

commit 64f89ab03e3afd82850f08ac15eb4ae8f08769cd
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Thu Dec 13 17:16:54 2018 -0500

i32 -> i8

commit 3431a18c1398b1dfcfb3687b4aad264a82313556
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Thu Dec 13 17:14:47 2018 -0500

remove it

commit e350164471d519b1901a101e2275bb360e9622a5
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Thu Dec 13 17:14:26 2018 -0500

remove ignore

commit 512a99f1a6c2cea68533c3b80059e285c98a05db
Author: Sushanth Raman <s.raman@columbia.edu>
Date: Thu Dec 13 17:13:26 2018 -0500

pushed

commit ac9ee51153068013dadf3dd4a2dfba03af9be4a5
Merge: 3dd6542 5d47475
Author: Sushanth Raman <s.raman@columbia.edu>
Date: Thu Dec 13 17:12:51 2018 -0500

Merge branch 'master' of <https://github.com/KanishkVashisht/scolang>

commit 3dd6542f277901cd386b4d490cd57f7b9e925c09
Author: Sushanth Raman <s.raman@columbia.edu>
Date: Thu Dec 13 17:12:46 2018 -0500

pushed

commit 5d4747541c89a9fa9518f0c610005320f5a30fcd
Author: vnv-varun <vnv-varun@users.noreply.github.com>
Date: Thu Dec 13 17:12:09 2018 -0500

Moved files from callbacks to main direc

commit aa97a5d436f305a7e26cb7d8c3f8b91c6c252e33
Author: Sushanth Raman <s.raman@columbia.edu>
Date: Thu Dec 13 17:11:53 2018 -0500

pushed

commit c40c3a02c98ca7a4c7664223f57c4721f1c4cccf
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Thu Dec 13 17:00:02 2018 -0500

fix misspelling

commit a06e0a96ab0473301b36a1f6a776dbf101cd0340
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Thu Dec 13 16:59:40 2018 -0500

remove context

commit 80ad9175ec688f3c18854a4bb7bf49e7a59e1e32
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Thu Dec 13 16:57:19 2018 -0500

try this

commit cc99d4cbdebc4b15f5a46aa3f0996135272bedff
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Thu Dec 13 16:44:23 2018 -0500

remove in

commit 692cd56749db69cb8e59d70310213c9e111bc3c2
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Thu Dec 13 16:41:58 2018 -0500

asd

commit 7039cece2d063efe7290dfe88bcda94f74dd560b
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Thu Dec 13 09:59:21 2018 -0500

refactored

commit b988d483d7c9dec1611e5dd8cccad9331149b330
Author: Sushanth Raman <s.raman@columbia.edu>
Date: Wed Dec 12 23:28:47 2018 -0500

updated Makefile

commit 8d9e54e4700e0b31c232083c3b14ec9f5fc465ad
Author: Sushanth Raman <s.raman@columbia.edu>
Date: Wed Dec 12 23:27:52 2018 -0500

updated Makefile

commit f68da82db8e34a49f93b21004c3afd4f01296075
Author: Sushanth Raman <s.raman@columbia.edu>
Date: Wed Dec 12 22:29:40 2018 -0500

updated webhook

commit 152be55d44947ff9f0036f383816b6166bb28249
Author: Sushanth Raman <s.raman@columbia.edu>
Date: Wed Dec 12 22:24:29 2018 -0500

add c files

commit eb500b6f6d9cd714d3e256243b945d8374c28996
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Wed Dec 12 22:17:29 2018 -0500

pass in llvalue

commit b51410ead12ac1c8d06af2e638820cd75a7e22af
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Wed Dec 12 22:12:45 2018 -0500

remove print in return

commit 641649bd9dce2aa92b94e96a7e1f77b15adeea6b
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Wed Dec 12 22:11:31 2018 -0500

fix action type

commit f93b95850338e5e466116808f867fa7a218b6072

Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Wed Dec 12 22:10:57 2018 -0500

remove context

commit 47ee302d25cf08cf75967f3f7931c67f7ddc5ac3
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Wed Dec 12 22:10:06 2018 -0500

move it up

commit d24528703630b426bafd3dc5f23e7c3526f492e8
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Wed Dec 12 22:09:08 2018 -0500

break into vars

commit 8e274833d7c3717570c515eea8637763667c4fb0
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Wed Dec 12 22:07:15 2018 -0500

add contextrt

commit d6b2df226d753e3b2d7fe02a1712c1e900c72ac7
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Wed Dec 12 21:58:04 2018 -0500

l prefix

commit c0a56a51ed904c50a2b6ba77fc1585e1753147a0
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Wed Dec 12 21:57:15 2018 -0500

get rid of ltype call in type

commit 5265d713251f43b217c59ffbb4529cfe6562d3c6
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Wed Dec 12 21:56:18 2018 -0500

move ltype up

commit e01fce3adeae5aeb996c50b43c1a2d00756b98e9

Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Wed Dec 12 21:55:13 2018 -0500

better paren

commit a8347310f28148ed3aadb3fbec5853917e9bc75
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Wed Dec 12 21:53:53 2018 -0500

add paren

commit 269e2382f5f0c6ed11ae6512925f1700d991b0db
Merge: 2b74433 5f23413
Author: Sushanth Raman <s.raman@columbia.edu>
Date: Wed Dec 12 20:53:45 2018 -0500

Merge branch 'master' of <https://github.com/KanishkVashisht/scolang>

commit 2b74433f323c393c66aee31dd70c063a39f41096
Author: Sushanth Raman <s.raman@columbia.edu>
Date: Wed Dec 12 20:52:49 2018 -0500

updated codegen

commit 5f23413d47470d868657830e30d5ff2d33acc9f8
Author: Sambhav Anand <sambhav182@gmail.com>
Date: Wed Dec 12 20:52:46 2018 -0500

trying to fix globals

commit 554ca92b7a2bc4b54f4c25ce1c9a3c89032d52dc
Author: Sambhav Anand <sambhav182@gmail.com>
Date: Wed Dec 12 20:42:59 2018 -0500

Trying to fix locals lsit

commit 27b54568b4f15902d96ae9997ec758ee9f35fae4
Author: Sambhav Anand <sambhav182@gmail.com>
Date: Wed Dec 12 20:33:31 2018 -0500

Working locals

commit 41b0465b15564c490a701f3ee0ebe0ece1cd7d21

Author: vnv-varun <vnv-varun@users.noreply.github.com>
Date: Wed Dec 12 20:31:55 2018 -0500

semi

commit cbf23ec260bfe9b226494c161e8fcbad7f219fbf
Author: vnv-varun <vnv-varun@users.noreply.github.com>
Date: Wed Dec 12 20:31:10 2018 -0500

globals

commit 09aaa8cfafa8d80a89fbc3c8d04589025af06e01
Merge: bbd7c0c aeea121
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Wed Dec 12 20:24:32 2018 -0500

Merge branch 'master' of github.com:KanishkVashisht/scolang

commit bbd7c0c9ae4ddb052e0b2bba791e050772f4a58a
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Wed Dec 12 20:24:27 2018 -0500

test 1 of fn ptr

commit aeea1215a0c62f30c32732c284408831849f17c2
Author: vnv-varun <vnv-varun@users.noreply.github.com>
Date: Wed Dec 12 20:23:54 2018 -0500

Changed main vars to global

commit 77852aadcf66b283e3aa7bb60f26b8ab176c44bb
Author: KanishkVashisht <kanishk.vashisht@gmail.com>
Date: Thu Dec 13 00:56:41 2018 +0000

Got printi working for integers

commit 7e4250cd99ce7acca91ce742b067da0f1d6bf4e8
Author: vnv-varun <vnv-varun@users.noreply.github.com>
Date: Wed Dec 12 19:54:26 2018 -0500

Added printi to semant

commit 5a998035238be2ebdeecb2a23ed353e96e07e482

Author: vnv-varun <vnm-varun@users.noreply.github.com>
Date: Wed Dec 12 19:52:30 2018 -0500

Added print int features

commit 5f9ef99ae450830f73e2607e552bebc5a8c26102
Author: vnv-varun <vnm-varun@users.noreply.github.com>
Date: Wed Dec 12 19:45:33 2018 -0500

Changes to l.doubletype

commit 976499c332c35168a5d4500b1feb9a92335881fe
Merge: 0253c90 3c2b7fa
Author: vnv-varun <vnm-varun@users.noreply.github.com>
Date: Wed Dec 12 19:43:36 2018 -0500

Merge branch 'master' of <https://github.com/KanishkVashisht/scolang>

commit 3c2b7fa2ed478ec806fce897b3fb2183915a17e3
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Wed Dec 12 19:43:29 2018 -0500

test v2

commit 744a00adb84ed2c619006dc5de6395f7a10a7274
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Wed Dec 12 19:42:08 2018 -0500

forgot semicolon

commit 170583bf21332038d77523df3c66a5da8c15c221
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Wed Dec 12 19:41:35 2018 -0500

print out bind

commit 0c9cc2da3ec9a79858ee9c35a72c2cf19ed6bada
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Wed Dec 12 19:39:21 2018 -0500

change world.sav

commit 33c7c47278181348b082074f6579e61d131486ff

Merge: 2488db2 1c5db82
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Wed Dec 12 19:37:44 2018 -0500

Merge branch 'master' of github.com:KanishkVashisht/scolang

commit 2488db2ccbfd1317a3f0454535874dbb098d4142
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Wed Dec 12 19:37:38 2018 -0500

ban SLisActInit from statements

commit 0253c901d8a5413a03e09da843e8413ff5f2214d
Author: vnv-varun <vnv-varun@users.noreply.github.com>
Date: Wed Dec 12 19:37:36 2018 -0500

changed printf type from pointer to float

commit 1c5db82cc35afabb912cdeb72eb9f8949f14b471
Author: KanishkVashisht <kanishk.vashisht@gmail.com>
Date: Thu Dec 13 00:15:01 2018 +0000

Working commit

commit bd79daabbbbee935f4ca3fe0661504fc1f6b19f56
Author: KanishkVashisht <kanishk.vashisht@gmail.com>
Date: Wed Dec 12 23:52:50 2018 +0000

Working Commit

commit ce192bd997961dc050741811ce318a13bf48a8d7
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Wed Dec 12 17:08:50 2018 -0500

add semi to bind

commit 2d5271133c55c09af20058ae93c5a6f25508a8f7
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Wed Dec 12 17:07:34 2018 -0500

strings

commit ed2996df61adfcf6312bb4aa5904af72cfbee57b

Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Wed Dec 12 17:06:50 2018 -0500

remove bind check temp

commit 6f5e0daf5b799af838d459b8bc84da6ed81668db
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Wed Dec 12 17:01:43 2018 -0500

varun is here

commit 6fa144cef5cd16039d13850e93c5fb871b81c124
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Wed Dec 12 17:00:07 2018 -0500

goldman sachs

commit 98c631776b4ad21f67744a37d91aec0da8bf8a7f
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Wed Dec 12 16:30:38 2018 -0500

retrain

commit ec761868a48c4af2719c5ae48430b5a4dea03bca
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Wed Dec 12 16:29:12 2018 -0500

waymo

commit 4aed5219db57b32b34ea95a794079d66c267d293
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Wed Dec 12 16:24:11 2018 -0500

lol

commit fef434a208cb78dbdaf43c04502275fa055bc5fb
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Wed Dec 12 16:23:24 2018 -0500

omg shut up dude lol

commit 71bd0e6473569337b5a34933f5c6e98ccb497f3d
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Wed Dec 12 16:22:33 2018 -0500

this dude needs to shut up

commit 63796de145915ba3c35e4e9ae1667e13ac7d3dcf
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Wed Dec 12 16:19:22 2018 -0500

#top4

commit 07ab579383d08d9aeb1ca0e7cb1152874f962d29
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Wed Dec 12 16:15:19 2018 -0500

trying some weird replacement strategy

commit 14b0d48592e324c49cfe62fc4cd77ba503e2eb25
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Wed Dec 12 15:54:40 2018 -0500

add

commit bac816931bc88d38597f2f6b8efbae8e2334eaec
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Wed Dec 12 15:50:54 2018 -0500

log all

commit 7d6f93bc1dc5146f4db06c7560e0ce983113fb4b
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Wed Dec 12 15:47:33 2018 -0500

move in ignore

commit 62ea028693a78f953ae254e3a4124ceef60ecd72
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Wed Dec 12 15:39:42 2018 -0500

ignore

commit 66ccf589067c465197ef38def148ea7f72ad39ac
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Wed Dec 12 15:37:53 2018 -0500

printing

commit 8af552bd449d8c287de5060e1ee81254272b41a8
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Wed Dec 12 15:21:06 2018 -0500

accessing

commit a7d6f6f6496ee0c80298decab564e02b300d1d11
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Wed Dec 12 14:58:20 2018 -0500

arg

commit 554ce58809a46c05205336dbb68bd7ce4058351f
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Wed Dec 12 14:57:58 2018 -0500

Where is

commit 4df6bdd5fb1e445ac5aa0422e29df7a6adbf1e66
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Wed Dec 12 14:57:18 2018 -0500

A

commit cec8ed07a59d3e8ef2caf11d070432e17c6671af
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Wed Dec 12 14:52:36 2018 -0500

printvar

commit 5d9f7c5aaf3c6ef17af28dffdf7c76590b5f184
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Wed Dec 12 14:43:47 2018 -0500

uppercase string

commit fb6e71d8920896ccd82d0cd2a710ce20a196ee52
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Wed Dec 12 14:29:38 2018 -0500

WEEE ARE THE CHAMPIONS MY FRIENDS

commit 1eb70c1350bfbc5253ec043b5a5f52cba8a966ba
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Wed Dec 12 14:28:00 2018 -0500

Exhaustive lists

commit 6961dc267da063b07502c5679770963359d3edb9
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Wed Dec 12 14:26:00 2018 -0500

Remove bar on matchg

commit 683f93277187c0089034c5c72da90b1cb3fcb7c
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Wed Dec 12 14:24:10 2018 -0500

makefile errors

commit a981235e38be26d0b9cfd06f12e2cf3e113b2a93
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Wed Dec 12 14:19:19 2018 -0500

attempt to fix this language, and my heart...

commit 924a4fa7eebaa5d61904ecbd8f8d25ab28808287
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Wed Dec 12 14:16:22 2018 -0500

I want to know what love is, I want you to show me

commit 8c7ca2d1891239970b1e6bd382f98f34aa4235d6
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Wed Dec 12 14:09:48 2018 -0500

I know you can show me

commit 2283c45245260dc000b99dce94bd1cd5b9ae4051
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Wed Dec 12 14:08:07 2018 -0500

do you know what love is?

commit dff2ee28ab542b5b46be5232723465f127ba0d3e
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Wed Dec 12 14:05:34 2018 -0500

revert

commit ccf5a90cf6fec7a8c3fd5212698c2bbbefb16d62
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Wed Dec 12 14:04:20 2018 -0500

Translate wants sast

commit 4f8c7002fae56a69073a95e34db8e07e6541d994
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Wed Dec 12 14:03:11 2018 -0500

Lol redefine twice

commit aa359c22c5f5dd2faccd715ee9a40810570fb228
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Wed Dec 12 13:58:29 2018 -0500

Remove return type entirely

commit febf7e93204c74ea94c70873a03968f6eb75861a
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Wed Dec 12 13:57:27 2018 -0500

Check statement list

commit 5d9f576eb1981c49b19e839d2da6dd87b809cce6
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Wed Dec 12 13:54:57 2018 -0500

Fix resolve in semant

commit 5b7946324ee49555cb789e1b259815952ab0bc8c
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Wed Dec 12 13:53:24 2018 -0500

Add arg to resolve

commit 55953181c49330cbf078f4ff15fa2858c1ab1864
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Wed Dec 12 13:52:25 2018 -0500

Change print for resolve

commit e8d5823540e4bd41f75f7b7c2bf6a6b9acf1f016
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Wed Dec 12 13:51:43 2018 -0500

pass arg into sresolve

commit 1398d8942983e531976d7d1c817f81d1a53eb521
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Wed Dec 12 13:48:04 2018 -0500

Fix args

commit 41c2afb021113f771e67211490efbd18da85fbdd
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Wed Dec 12 13:46:45 2018 -0500

Force block statement

commit 9478984cf33288a9d40149437e8d7d33e162809e
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Wed Dec 12 05:09:18 2018 -0500

all_actions

commit 354acfb6dfc2cf05b0ef75cd7e32de7ce7696cbc
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Wed Dec 12 05:08:02 2018 -0500

uber

commit 88449272155e23677ab06854b8c61a276c0ad5e5
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Wed Dec 12 05:04:07 2018 -0500

ls

commit 33d6e552b5be5f96587bc623ff3f983629d29fa4
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Wed Dec 12 05:02:59 2018 -0500

sast

commit 65a56ff32bac1a06d5398e40dedfe9c1e1815245
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Wed Dec 12 04:57:24 2018 -0500

aprim

commit 49cc26e7495580d8f75c89c35ef9c6be8a455884
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Wed Dec 12 04:56:07 2018 -0500

now reject returns void

commit b378dd5931822a82600e22e1c21146430ebb5edc
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Wed Dec 12 04:51:53 2018 -0500

making resolve a return properly

commit 2bf6d89e781181e73f05411a126b9dc7dc97b7d4
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Wed Dec 12 04:48:08 2018 -0500

moved SVarInIt in builder

commit f3fb5ba17a99f2d26907c1b3af4d23ba58002bbb
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>

Date: Wed Dec 12 04:43:47 2018 -0500

()

commit c4484f9318b2d64b5d9346c99fd5839b484f07cd
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Wed Dec 12 04:42:57 2018 -0500

moved SBind to sstatement, and SResolve,Sreject,Snoexpr to expr builder

commit aa80b2b093270858b64207e729e83295524303ef
Merge: b8dff81 a4092b0
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Wed Dec 12 04:40:41 2018 -0500

Merge branch 'master' of <https://github.com/KanishkVashisht/scolang>

commit b8dff814dfa616fa9bf925b55f22ab72743 added0b
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Wed Dec 12 04:40:34 2018 -0500

moved the Sbind and Sresolve, Sreject to the sstatement variant

commit a4092b066b9e9d9c3d7a5b22d209a456e2b6c861
Merge: 403ad1d dbc5461
Author: vnv-varun <vnm-varun@users.noreply.github.com>
Date: Wed Dec 12 04:39:58 2018 -0500

Merge branch 'master' of <https://github.com/KanishkVashisht/scolang>

commit 46b78bab67ef0e18441c16094b68e8ee5410e3f3
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Wed Dec 12 04:34:21 2018 -0500

SCall

commit 4b1c68756448c2444559bd86 added9feae1eb16872
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Wed Dec 12 04:33:52 2018 -0500

dcaf

commit 03c4d02d753d17a21c0d7d7c33d3e353f3dbf58b

Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Wed Dec 12 04:28:10 2018 -0500

commas yeah

commit 29ac7a25a51414a037c1dbdfc281e67a3544f867
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Wed Dec 12 04:27:34 2018 -0500

some commas

commit a383728809f4048de671e8571147eaacb76bd75e
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Wed Dec 12 04:26:59 2018 -0500

burje

commit 1017e36eff9e03acc50e0a84156aad0e4b77eb54
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Wed Dec 12 04:26:10 2018 -0500

lol

commit e2a79f005058ab23c6aa65313e4315b63decfc88
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Wed Dec 12 04:25:07 2018 -0500

no comma

commit f9894ba57268f6572b8353ea951fdf63695fbdd9
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Wed Dec 12 04:24:04 2018 -0500

s

commit c7c877add7f33d1ffff7dcce36ac5c5ffc9afd3
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Wed Dec 12 04:23:37 2018 -0500

s

commit cf57904f0d65d6a9abe28a581f8ec236f3e52702
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>

Date: Wed Dec 12 04:23:11 2018 -0500

move)

commit 227ce1af4327607cf90bf13955cf0a56196ca1dc
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Wed Dec 12 04:22:37 2018 -0500

prim of dt

commit 6285ae3eeb423449723c5d441136435fac15a499
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Wed Dec 12 04:22:03 2018 -0500

tuple

commit a241d16a2980a721a830c52cf7342bec7434c850
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Wed Dec 12 04:18:12 2018 -0500

TODO

commit 7922a458744782368b15b36c3bf61c4367b60c1d
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Wed Dec 12 04:18:05 2018 -0500

char

commit f4acf70e0be5fd31da5e75d8f806bd16e3546030
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Wed Dec 12 04:17:18 2018 -0500

zero

commit d68c2cd6f86b8b3d35281e6682cc489ff8c85844
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Wed Dec 12 04:11:02 2018 -0500

add_local

commit 3195b486decc7076cdfc952c900f96399804c018

Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Wed Dec 12 04:10:32 2018 -0500

sbody

commit 0f009fe44c13a041d9159ac6e1b5686bd9802578
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Wed Dec 12 04:10:05 2018 -0500

sfame

commit 10daa5591a97c26a45e2ec8d5bc952fc9e2c8d59
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Wed Dec 12 04:09:37 2018 -0500

[prm of dt

commit 25dfd83bc9ce31f6d8dc106bb47fb4759182312b
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Wed Dec 12 04:08:01 2018 -0500

other way

commit 5d70b966f998795698b60b09d5b7744c8467e484
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Wed Dec 12 04:03:08 2018 -0500

Primitive

commit 351362f4846ca550f7974f0c72038d1f1c780c50
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Wed Dec 12 04:01:47 2018 -0500

matching

commit 7b8ff7ff6f39eadb7c4ace848ff11a55eae98d18
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Wed Dec 12 03:59:53 2018 -0500

add prim

commit 1b8dd2ad640d1e1430180f05d0ce08a6dcd27923
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Wed Dec 12 03:59:06 2018 -0500

fixing type errors

commit 7a698dc90b6a6b29fbce263972eae62844867ae7
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Wed Dec 12 03:56:53 2018 -0500

rename

commit 816954e27d7a56ecd37fde32d67753472e24e8b4
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Wed Dec 12 03:55:50 2018 -0500

param stay to itsefl

commit 9bd5def0ce6c13407cba37c145c82ea89439ff03
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Wed Dec 12 03:55:23 2018 -0500

array not list

commit 4c42d0335c476763b0ba9e7d22625ad6675ff267
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Wed Dec 12 03:53:16 2018 -0500

old code

commit 2e52a27cd3b202cdef5aa701c22f522c89a77166
Merge: b23540a 840e3ca
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Wed Dec 12 03:52:22 2018 -0500

Merge branch 'master' of <https://github.com/KanishkVashisht/scolang>

commit b23540a3c50e37aa12a0000d631846a35ae259cc
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Wed Dec 12 03:51:54 2018 -0500

thanks

commit 840e3ca3885af0bce4c41cc3f019c23d7611e327
Author: vnv-varun <vnv-varun@users.noreply.github.com>
Date: Wed Dec 12 03:38:32 2018 -0500

sast added to Makefile

commit 3e4879775844d8ae0e3525e19c7c24037f972097
Merge: f5b9e45 e437b48
Author: vnv-varun <vnv-varun@users.noreply.github.com>
Date: Wed Dec 12 03:36:31 2018 -0500

Merge branch 'master' of <https://github.com/KanishkVashisht/scolang>

commit e437b486c44afca6c9032fe6e4d1fab2181dad11
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Wed Dec 12 03:36:07 2018 -0500

change order of execution

commit f5b9e45e6e0aad24099434e7aaf41a801b2cbe1e
Author: vnv-varun <vnv-varun@users.noreply.github.com>
Date: Wed Dec 12 03:36:06 2018 -0500

Added sast in Makefile

commit 4ebc6031c205cfae060b45b1103ab21e2d11e7af
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Wed Dec 12 03:35:28 2018 -0500

make lisactinit a statement list

commit 07e4e2c54d0ece3f325a825ad6e1afaa2d6e1ece
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Wed Dec 12 03:32:36 2018 -0500

add s

commit 7162866dc6ab6d86d79b62d9b100428f3d3fe9fc
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Wed Dec 12 03:31:49 2018 -0500

prim

commit e19e381b6affa03bbf4d2f2d75a2536e5c214bfb
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Wed Dec 12 03:28:20 2018 -0500

remove useless

commit 9043bdbd4cedfddd7bf0ec246ec0c81eb295305c
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Wed Dec 12 03:27:31 2018 -0500

add action and listener decls

commit 9ecd1f121aa6bba0a595c1c04e90d80febe17984
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Wed Dec 12 02:39:08 2018 -0500

Revert "debug"

This reverts commit f2e73e9410f30849a64434ad553b2aa2ca87a8c6.

commit ba3bbdb6cbc41b235f74c777f3712ae383c703d4
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Wed Dec 12 02:39:07 2018 -0500

Revert "TODO"

This reverts commit 8fcd1d0d6b8c3d7a7757c8f6467cc42682002e01.

commit a7ac24c59673a89d6fbe69a2dd74cb3dd5a2e2a3
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Wed Dec 12 02:39:07 2018 -0500

Revert "Semantic"

This reverts commit 221c241d1ecccff6db4a0c7e26fc13cfa1d7538a.

commit e591325d54f99820f0ddae43469ec57527ae16ca
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Wed Dec 12 02:39:05 2018 -0500

Revert "yeh yeh yeh"

This reverts commit 0e1d9285bd94460016f0fb9a52bce79ab78e27f4.

commit d63a9f8c28382e0e2fafa74e1fb022564bb0fd9e
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Wed Dec 12 02:39:05 2018 -0500

Revert "jackiechen"

This reverts commit 417b9d5a8ff1ed985827108733d9c9598db58f33.

commit 34d302fc5c9b85dc50cc2e5c4da1ccd46d42c2bd
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Wed Dec 12 02:39:04 2018 -0500

Revert "sa"

This reverts commit 11f20efe373f13dbc82ba60a0072d14d04afaa2a.

commit 5c087fa573c99a10d0fb1e20302e37b15668137e
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Wed Dec 12 02:39:03 2018 -0500

Revert "listofstatements"

This reverts commit 9cb297b498db1f25043ff2ed961cf7eb182b01ef.

commit 3c06ac489030d3c3f1fa550289e1fd51935657ed
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Wed Dec 12 02:39:01 2018 -0500

Revert "sas"

This reverts commit cdd73c983649d917e52793ac0a8956c8fda91b5d.

commit d987fbfa65a571a451e43cbf0c474917912240c9
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Wed Dec 12 02:39:00 2018 -0500

Revert "err"

This reverts commit 882846e1f29fc3a67f319f0658f81f2db7806c74.

commit 3a7fa044d87410d53c32c9ebee87d1798fcd6c
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Wed Dec 12 02:38:59 2018 -0500

Revert "uncommented"

This reverts commit f490588d83356da550ca58ab43486bcdb37b4dfb.

commit 1998082ebe2aa5c56266919a72056552ea3dd9ae
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Wed Dec 12 02:38:58 2018 -0500

Revert "sblock"

This reverts commit e0e8049e5ab8dca1473a39d5247ff4645b385aac.

commit 04542c5e55046a0da0cc4976f79f973514515242
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Wed Dec 12 02:38:58 2018 -0500

Revert "la"

This reverts commit c86951fbbe3f96c6506ca5f163579189ac39bbea.

commit 6233edadd70c3b3732fff25f42da76f75c5c74e6
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Wed Dec 12 02:38:57 2018 -0500

Revert "statement list"

This reverts commit beee2a359f20957de6fb1b4ee76f7f1117baf207.

commit 266bdb6cc978021ba7820ff62ce428991cc1c9ea
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Wed Dec 12 02:38:56 2018 -0500

Revert "fix sprogram"

This reverts commit a2006d0ce28b77a8202d1c306e2ef4ce218254c6.

commit 8c89ca7b70cf49c0b544413f01ad3b58810e27be

Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>

Date: Wed Dec 12 02:38:55 2018 -0500

Revert "test"

This reverts commit d26284024fb38b79ac2fbab319738499a87d6b50.

commit a67b991ef355d0880bfd898656cd7a02d07cc449

Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>

Date: Wed Dec 12 02:38:54 2018 -0500

Revert "change str"

This reverts commit e4ff17d67746feb2c676ef3755a42280d9de3819.

commit b97f86e916e9bc961f7d3bfabb836142b026eff1

Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>

Date: Wed Dec 12 02:38:52 2018 -0500

Revert "open sast"

This reverts commit 1c2d5a2bfac6bede0d788e30c428bac0762266e6.

commit 1fe205bc6a327baf2b56a955829451bedbde4f77

Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>

Date: Wed Dec 12 02:38:42 2018 -0500

Revert "asdf"

This reverts commit 17f1ef6542cdb13371f6b55a2bd7920edf148a1e.

commit 1c2d5a2bfac6bede0d788e30c428bac0762266e6

Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>

Date: Wed Dec 12 02:15:23 2018 -0500

open sast

commit e4ff17d67746feb2c676ef3755a42280d9de3819

Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>

Date: Wed Dec 12 02:10:02 2018 -0500

change str

commit d26284024fb38b79ac2fbab319738499a87d6b50
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Wed Dec 12 02:08:18 2018 -0500

test

commit a2006d0ce28b77a8202d1c306e2ef4ce218254c6
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Wed Dec 12 02:03:22 2018 -0500

fix sprogram

commit beee2a359f20957de6fb1b4ee76f7f1117baf207
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Wed Dec 12 02:01:34 2018 -0500

statement list

commit c86951fbbe3f96c6506ca5f163579189ac39bba
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Wed Dec 12 01:52:43 2018 -0500

la

commit e0e8049e5ab8dca1473a39d5247ff4645b385aac
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Wed Dec 12 01:51:03 2018 -0500

sblock

commit f490588d83356da550ca58ab43486bcdb37b4dfb
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Wed Dec 12 01:29:26 2018 -0500

uncommented

commit 98da2ac0741fa3cf8d16dade1a54b04b73bbcbff
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Wed Dec 12 01:20:39 2018 -0500

fixing type

commit 9cb297b498db1f25043ff2ed961cf7eb182b01ef
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Wed Dec 12 01:18:40 2018 -0500

listofstatements

Semantic

commit 8fcd1d0d6b8c3d7a7757c8f6467cc42682002e01
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Wed Dec 12 01:11:04 2018 -0500

TODO

commit f2e73e9410f30849a64434ad553b2aa2ca87a8c6
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Wed Dec 12 01:10:41 2018 -0500

debug

commit d9c78df782e4bdc00fef91b3abc968b9db91dad8
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Wed Dec 12 01:10:06 2018 -0500

semi colon

commit 63d335c4378fa2ba79a3b6686aef994b8d33f88c
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Wed Dec 12 00:22:50 2018 -0500

mainfdecls

commit fff5c3569bdf91bfe20869b51e4304f81e921e3f
Merge: 8834250 0891451
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Tue Dec 11 20:50:35 2018 -0500

Merge branch 'master' of <https://github.com/KanishkVashisht/scolang>

commit 089145170b7baa059bfabacafa5e89322219bc46
Author: vnv-varun <vnv-varun@users.noreply.github.com>
Date: Tue Dec 11 20:37:50 2018 -0500

typos

commit 3000f889dd8d97e462b6a34ce075d28b8038ad70
Author: vnv-varun <vnv-varun@users.noreply.github.com>
Date: Tue Dec 11 20:35:27 2018 -0500

Added in webhook into Makefile

commit ecaf99451c5f74916cf03b65fa52632c20f456c5
Merge: fe554d4 9f78e66
Author: Sushanth Raman <s.raman@columbia.edu>
Date: Tue Dec 11 20:21:26 2018 -0500

Merge branch 'master' of <https://github.com/KanishkVashisht/scolang>

commit fe554d407a080042b2ae6ebc3eebfd271b833f65
Author: Sushanth Raman <s.raman@columbia.edu>
Date: Tue Dec 11 20:21:19 2018 -0500

added webhook

commit 9f78e66f0f2dc2d112dea15ab8b17bb2463649d3
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Tue Dec 11 20:17:47 2018 -0500

gavin belson authorized this commit

commit 890b0205c63b93edb7ac35e0a9c83bc8bd5dfa42
Merge: dc646ab d63acc1
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Tue Dec 11 19:23:12 2018 -0500

Merge branch 'master' of <https://github.com/KanishkVashisht/scolang>

commit dc646abfb48cd57325d46904e3c9f2a061f3230a
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Tue Dec 11 19:22:56 2018 -0500

hellagenna

commit d63acc17c679ca87d89fc9899dd3e150cc4836e2
Author: Sushanth Raman <s.raman@columbia.edu>
Date: Tue Dec 11 19:11:23 2018 -0500

updated webhook library

commit 04116a5a711846af3c370c2be729b44e38599e43
Merge: 3a30105 8d7cf76
Author: Sushanth Raman <s.raman@columbia.edu>
Date: Tue Dec 11 17:36:16 2018 -0500

Merge branch 'master' of <https://github.com/KanishkVashisht/scolang>

commit 3a30105d75a06baca86c5df93d0fe44acc89d71f
Author: Sushanth Raman <s.raman@columbia.edu>
Date: Tue Dec 11 17:36:12 2018 -0500

added updated library

commit 8d7cf764d1e3a3d7e0427dde98de09eb67936ebe
Author: KanishkVashisht <kanishk.vashisht@gmail.com>
Date: Tue Dec 11 21:38:48 2018 +0000

done well

commit ee7a5b8db3a4510a6677e9f4d364b53f51b9796b
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Tue Dec 11 16:18:08 2018 -0500

lowercase llvm

commit 69bbdc05cd2fc3782698779fed47598530beaeefe
Merge: 68c8923 d371cd5
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Tue Dec 11 16:14:34 2018 -0500

Merge branch 'master' of github.com:KanishkVashisht/scolang

commit 68c89233d5f0bcf59fc455885cf576917e5699dc
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Tue Dec 11 16:14:31 2018 -0500

Lowercase module and open

commit d371cd50a2e2aeca50c2b110d6afc5e26d991485

Merge: 18aa191 deebcca

Author: Sushanth Raman <s.raman@columbia.edu>

Date: Tue Dec 11 15:17:57 2018 -0500

merge branch 'master' of <https://github.com/KanishkVashisht/scolang>

commit 18aa1915067c18922f394727f061cc89b9e0d27c

Author: Sushanth Raman <s.raman@columbia.edu>

Date: Tue Dec 11 15:17:53 2018 -0500

add updated webhook

commit deebcca6e789831f9b0d078fbd9cf034f570f55c

Merge: 2e993e4 9010201

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Tue Dec 11 14:49:08 2018 -0500

Merge branch 'master' of github.com:KanishkVashisht/scolang

commit 2e993e4a7ff05b8e1b951218a7ce40f051abc6bb

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Tue Dec 11 14:48:24 2018 -0500

Remove useless comments

commit 9010201730457c4a01954989c12f59d59b08a8e2

Merge: 0718d5c 352e0a3

Author: Sushanth Raman <s.raman@columbia.edu>

Date: Tue Dec 11 14:45:42 2018 -0500

merge branch 'master' of <https://github.com/KanishkVashisht/scolang>

commit 352e0a3ad738e3320770f181c6fda95e5c8ece82

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Tue Dec 11 14:45:37 2018 -0500

Update string_of_sprogram

commit 0718d5c526c1104c7b5a7d80e308951cc7787974

Author: Sushanth Raman <s.raman@columbia.edu>

Date: Tue Dec 11 14:45:24 2018 -0500

added webhook library

commit b13d65ae518d2559be809ea11beedec1e50e734f

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Tue Dec 11 14:42:50 2018 -0500

Wow did that work

commit 1a9a156b0e880e41a4ae086e88f48b8bc1e9a016

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Tue Dec 11 14:41:32 2018 -0500

paren?

commit 52ae51946151005336fbefc16f104bcffd70d20c

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Tue Dec 11 14:40:57 2018 -0500

Purposely miss one

commit 64621dd4d628053acb001562c8e8d780a026e7d6

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Tue Dec 11 14:36:49 2018 -0500

Add match

commit 3132a533405b8cce94c30f015db3e0d75d66f6fb

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Tue Dec 11 14:36:00 2018 -0500

Try copy

commit f7c42bc4ca709db69ec280d1d3e351cb2dd46cdc

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Tue Dec 11 14:31:32 2018 -0500

Remove not wokrin

commit 1b90efed3387aaad414696cecc24ed2e46b02f4f

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Tue Dec 11 14:31:15 2018 -0500

Tets3g

commit f39f89ca7264aeb111f9c0241269a49a646dbd82

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Tue Dec 11 14:30:19 2018 -0500

move out fn

commit cac2f51c18ed77ef3812a786d02511546f4e16ad

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Tue Dec 11 14:27:56 2018 -0500

Add paren

commit 840324d7d0b171f624701ef14cc15d5b3bd20771

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Tue Dec 11 14:25:49 2018 -0500

typ

commit a49c5c752d25367aa6a82bb15eeae7e375f9ee9f

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Tue Dec 11 14:24:58 2018 -0500

Try to chain statement fn

commit 0e0fb4194cfccca479566e7413d00f5006881892

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Tue Dec 11 14:22:35 2018 -0500

Build statement symbol table

commit 45ad70d8bef8e69651e906303c934a91ee68ca1c

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Tue Dec 11 14:20:15 2018 -0500

Introduce 2nd element of tuple

commit 231f835ec3b6f42bc73fcd21072d5561a71766e6

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Tue Dec 11 14:19:53 2018 -0500

Make unenclosed statements have void return type

commit 3bf8e313b946f87dcfe9f3a48c1faaf625a1fa0c
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Tue Dec 11 14:17:02 2018 -0500

Change prim of dt

commit 44b5e19017a03c01eefb9210e4435dc1bc176e27
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Tue Dec 11 14:15:31 2018 -0500

Test 2

commit 1166cceda3b94d206e2edaee8edcf0a1a17995be
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Tue Dec 11 14:13:32 2018 -0500

test

commit cc62dcf9bff7f1b5ce4a72cec7be14e508edf8cf
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Tue Dec 11 14:10:00 2018 -0500

Remove func.typ from error mess

commit 4cbe8d0bfc7cf14470fb08a4090aee0d957c2054
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Tue Dec 11 14:06:57 2018 -0500

Forgot to add return_typ

commit 7a44fb00d9183b017533618ad1bccb7c130cb4f8
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Tue Dec 11 14:05:54 2018 -0500

Check return typ

commit 6339fb320baaf26415faf6c8fa9dd43efc9624aa
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Tue Dec 11 14:00:28 2018 -0500

Move check_stmt out

commit 82ed92886e2b9d7063dbd72ce505b0671b0c848e
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Tue Dec 11 13:59:20 2018 -0500

Move check_bool_expr up

commit 877f22f32a491409f9068b4e88a72ee059241e56
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Tue Dec 11 13:57:41 2018 -0500

Move expr up

commit aa68b5fbf78311a51c190ba9f4b07061c8c761ca
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Tue Dec 11 13:51:05 2018 -0500

Move type_of_literal out

commit 73321b76905164ffc72bde394fc82394422b6336
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Tue Dec 11 13:50:08 2018 -0500

Oops wrong order

commit 5d7d7ce15e4346c62a372c3341903ba8d751b071
Merge: e6e4fdf 53bc7bc
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Tue Dec 11 13:46:31 2018 -0500

Merge branch 'master' of github.com:KanishkVashisht/scolang

commit e6e4fdf41a381ed69889fdf876283c29496bea37
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Tue Dec 11 13:46:25 2018 -0500

Move type of identifier

commit 53bc7bc90e25f8e7e7258a484ab5567af9421f93
Merge: d9ec57c b6951d8
Author: Sushanth Raman <s.raman@columbia.edu>
Date: Tue Dec 11 13:45:40 2018 -0500

Merge branch 'master' of <https://github.com/KanishkVashisht/scolang>

commit d9ec57cdfa0c9314028496a4a500b772ef7ea7f7

Author: Sushanth Raman <s.raman@columbia.edu>

Date: Tue Dec 11 13:45:35 2018 -0500

added contract

commit b6951d8b1f5d473b9b71ffa0dea703faf2a41736

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Tue Dec 11 13:44:55 2018 -0500

Test if it is a scoping issue

commit 4c944aa52116e3c2e44b24ea6de5261f82ceaba8

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Tue Dec 11 13:43:00 2018 -0500

Remove in

commit aeee2e44b2024322ae15cb9c24e6e14616bdccfb

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Tue Dec 11 13:37:28 2018 -0500

Move locals out'

commit 80751f9fa2b450bbc036e6f9549e7143b4a06713

Author: Sushanth Raman <s.raman@columbia.edu>

Date: Tue Dec 11 02:18:43 2018 -0500

added contract

commit 8917f1248272023d7e9ba474315bfc8ee5475455

Author: vnv-varun <vnm-varun@users.noreply.github.com>

Date: Tue Dec 11 01:48:04 2018 -0500

Callbacks

commit 1c074532015e55f500770524f2b37915b4308b98

Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>

Date: Tue Dec 11 01:30:04 2018 -0500

Merge branch 'master' of <https://github.com/KanishkVashisht/scolang>

commit c2e3e7ff73645da74fbd63a0a08bd9e59835fa5a
Merge: c8c9547 62fd729
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Tue Dec 11 01:18:33 2018 -0500

Merge branch 'master' of <https://github.com/KanishkVashisht/scolang>

commit c8c95479214d875632ab02e2e5bf429e45eae664
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Tue Dec 11 01:18:26 2018 -0500

codegen

commit 62fd729b8100424a1519fcf9cc8a9d4202fc5e68
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Sat Dec 8 19:42:17 2018 -0500

They should actually be strings

commit d529bd1a892ba62fa33ba4e29007a95ca40ee3eb
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Sat Dec 8 19:41:49 2018 -0500

Not string

commit 97cac12d4e604a73277ca84f4fe04c3002b63ad9
Merge: 0baf11c 0bf5680
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Sat Dec 8 19:40:56 2018 -0500

Merge branch 'master' of [github.com:KanishkVashisht/scolang](https://github.com/KanishkVashisht/scolang)

commit 0baf11c9e5ea9885a299915441e843ffe2368c81
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Sat Dec 8 19:40:54 2018 -0500

Phase 2

commit 0bf5680b4c7be1fe2b3ab456594adb84b9c96aa7
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Sat Dec 8 19:31:29 2018 -0500

I

commit dd324655a0db6772553cbada34e3bd06489018d3
Merge: 316330c 1893481
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Sat Dec 8 19:31:07 2018 -0500

Merge branch 'master' of <https://github.com/KanishkVashisht/scolang>

commit 316330c4409e4fc661c238010e1ee20f5524d2c8
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Sat Dec 8 19:30:50 2018 -0500

S

commit 1893481c45ca7e515857cf9a44ea3cf9f2fef09b
Merge: 2e16f76 e98cc7c
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Sat Dec 8 19:30:15 2018 -0500

Merge branch 'master' of [github.com:KanishkVashisht/scolang](https://github.com/KanishkVashisht/scolang)

commit 2e16f769b339629e0bebadb1c818441da7157406
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Sat Dec 8 19:30:08 2018 -0500

missed one

commit e98cc7c127b10e7fb3e7ecd11680b59f68cd3eee
Merge: cc59ed9 de63749
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Sat Dec 8 19:28:11 2018 -0500

Merge branch 'master' of <https://github.com/KanishkVashisht/scolang>

commit cc59ed96f382a0c86538491680e40762e9b85cb0
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Sat Dec 8 19:28:04 2018 -0500

L

commit de6374927d20b4927d7923b6d9ae5725352c4c09

Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Sat Dec 8 19:23:03 2018 -0500

kanishk is bad a coding

commit 564e192cdccaefe979ad78cdcdf9d9a4b64da722
Merge: d9232a6 8fcc046
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Sat Dec 8 19:22:21 2018 -0500

Merge branch 'master' of github.com:KanishkVashisht/scolang

commit d9232a6e3b54f11235a353c76a70d55a1e0c9c56
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Sat Dec 8 19:22:16 2018 -0500

typ

commit 8fcc046a12ea75a43e088317e4a43c234fcfb289
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Sat Dec 8 19:21:32 2018 -0500

au

commit ae2d72d9f8765be126eb41525b87218c99d877b1
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Sat Dec 8 19:19:48 2018 -0500

Add paren

commit 80e74e6d66829b054a5a605901e2c07444c506a5
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Sat Dec 8 19:19:28 2018 -0500

Comment out all str of prim

commit e8241b2d30f7e953ae3f6df676a3a25f27e709dd
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Sat Dec 8 19:16:44 2018 -0500

LOL

commit 6d8ba55e693202dc0842bb55a7ea55933d3c44e8

Merge: 062ba5c e173ac7
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Sat Dec 8 19:12:06 2018 -0500

Merge branch 'master' of <https://github.com/KanishkVashisht/scolang>

commit 062ba5cbbd856c93226ef1333db6788bf577999a
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Sat Dec 8 19:11:56 2018 -0500

LOL

commit e173ac7c78ac592862e19cc03102795136c93c02
Merge: 6bfa60c ad10553
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Sat Dec 8 19:11:34 2018 -0500

Merge branch 'master' of github.com:KanishkVashisht/scolang

commit 6bfa60cf2dbdbc9764f3cda9a2f06b8e3ea0324b
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Sat Dec 8 19:11:31 2018 -0500

Test 3

commit ad105535fec8852b1836038d1e0dd06138fb7e45
Merge: fb82f44 e4cde1f
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Sat Dec 8 19:11:12 2018 -0500

Merge branch 'master' of <https://github.com/KanishkVashisht/scolang>

commit fb82f442f3be2e9f95116281a24c9e2126bc71de
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Sat Dec 8 19:11:01 2018 -0500

generic

commit e4cde1f045e60a8e627c1df8fe5997f3c0c971d4
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Sat Dec 8 19:08:54 2018 -0500

Test 2

commit 37f7f114fee4856ef62977dbdedc0575c85d6590
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Sat Dec 8 19:07:22 2018 -0500

Remove in

commit 621c5d2ed40101c8edd8dc121ba0ffdf8a0b670d
Merge: 4f9176d 827ff55
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Sat Dec 8 19:07:01 2018 -0500

Merge branch 'master' of github.com:KanishkVashisht/scolang

commit 4f9176daf838b27e964b4ebafd178cfd34de4024
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Sat Dec 8 19:06:56 2018 -0500

Test changing str

commit 827ff5500f217b37d8ae338c96ab94759fb19f02
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Sat Dec 8 19:04:43 2018 -0500

meg wife

commit 591378f6190701beee50caf143f0ab55251b0373
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Sat Dec 8 19:04:22 2018 -0500

L

commit 569ce7127ac8612202ff24a12cbc9ff9d9f227f8
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Sat Dec 8 19:02:53 2018 -0500

:

commit d2f4fcc59afa20e1779a4a78209de499d5bac435
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Sat Dec 8 19:02:28 2018 -0500

tri

commit 6c82ff5be52a07d1a7591eb1dd37b2812ba56fba
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Sat Dec 8 19:01:49 2018 -0500

LOL

commit f5d49dbe58652b69b4b90cb5c647633b409c5f4b
Merge: ad58920 577bfb5
Author: Sushanth Raman <s.raman@columbia.edu>
Date: Sat Dec 8 18:57:04 2018 -0500

Merge branch 'master' of <https://github.com/KanishkVashisht/scolang>

commit ad589209fa0942de4284fd3149887191838ddb0b
Author: Sushanth Raman <s.raman@columbia.edu>
Date: Sat Dec 8 18:56:52 2018 -0500

dsd

commit 577bfb53b43e104b8e44949ea2729df3f7f5d5f5
Merge: 5c71165 1820108
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Sat Dec 8 18:50:55 2018 -0500

Merge branch 'master' of github.com:KanishkVashisht/scolang

commit 5c711653f0d77ce36867bd39c71fd6fa5dfe5dff
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Sat Dec 8 18:50:54 2018 -0500

Add to sast

commit 182010868e50d75c4bd35e90281728394b44b2ef
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Sat Dec 8 18:48:07 2018 -0500

jackson messed up

commit 2d2c3027a9b67761fbfa8b73d8ec43da2b2c4577
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Sat Dec 8 18:46:52 2018 -0500

semantically checked ast

commit 484672aef2f1aa379c24e2ff6c403ad5beb5a95f
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Sat Dec 8 18:42:30 2018 -0500

L

commit 8e2d1b1a64af87279282c875eaa8e5269d81eec7
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Sat Dec 8 18:41:45 2018 -0500

jeff

commit 180501c287f6b8685b28ae2072762b86efa7c873
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Sat Dec 8 18:40:28 2018 -0500

scan

commit 0919eed4087e301ff21523cc1a1c944face7fd85
Merge: 52f8a13 ceb5768
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Sat Dec 8 18:39:31 2018 -0500

Merge branch 'master' of <https://github.com/KanishkVashisht/scolang>

commit 52f8a131e853a37e5196662e0f621c4a97f1a076
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Sat Dec 8 18:39:20 2018 -0500

L

commit ceb57687379e7860ab37afaadabedcac287f5a1f
Merge: dbe2840 bfed477
Author: Sushanth Raman <s.raman@columbia.edu>
Date: Sat Dec 8 18:39:12 2018 -0500

Merge branch 'master' of <https://github.com/KanishkVashisht/scolang>

the commit.

commit dbe28401f2b1f7fa470bf90e7048b4d2a0116fd5

Author: Sushanth Raman <s.raman@columbia.edu>

Date: Sat Dec 8 18:39:07 2018 -0500

updated scnaer

commit bfed477b18b7c1f60abdb2726895446ce875a789

Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>

Date: Sat Dec 8 18:38:52 2018 -0500

L

commit 1b3690adffa7b4845816873f7f463fbd304a8d03

Author: Sushanth Raman <s.raman@columbia.edu>

Date: Sat Dec 8 18:37:47 2018 -0500

updated scanner

commit ca5af63516f7ea293bd4219a99356a1520247bbb

Author: Sushanth Raman <s.raman@columbia.edu>

Date: Sat Dec 8 18:36:18 2018 -0500

added sscanner

commit de03e6e86b1b292896e98346c76488460f70942d

Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>

Date: Sat Dec 8 18:30:34 2018 -0500

L

commit b78e788b0b1746223577cf680c64d6c5281a7a66

Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>

Date: Sat Dec 8 18:26:48 2018 -0500

L

commit d6ab8bbbd8d97ce2668a1bbed5dbd4659266fb08

Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>

Date: Sat Dec 8 18:26:25 2018 -0500

L

commit 8d5d4c578705242d35a229d6338dce2521ee97f1

Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>

Date: Sat Dec 8 18:25:06 2018 -0500

L

commit d868c3a3e429e10b12d029842cd69348510f0160

Merge: d4825ca a44700a

Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>

Date: Sat Dec 8 18:21:54 2018 -0500

Merge branch 'master' of <https://github.com/KanishkVashisht/scolang>

commit d4825ca5058629c7cc17fb893401343d342db261

Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>

Date: Sat Dec 8 18:21:47 2018 -0500

scan

commit a44700a5070d2a6903d8824bd92a86a3af418689

Merge: 85d866f f128f67

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Sat Dec 8 18:19:44 2018 -0500

Merge branch 'master' of github.com:KanishkVashisht/scolang

commit 85d866fd62d4f45d8691b74a0e21f3b14ea9236f

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Sat Dec 8 18:19:41 2018 -0500

Remove main req in semant

commit f128f671b604b3895ae47ba1e126ef3e576fdd68

Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>

Date: Sat Dec 8 18:17:08 2018 -0500

w

commit ecbeba92c07092f8bb077e55e99377495d6bda23

Merge: 83853be 2d251e8

Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>

Date: Sat Dec 8 18:16:52 2018 -0500

Merge branch 'master' of <https://github.com/KanishkVashisht/scolangf>

commit 83853be57f1068615e303ee3a4706237ce2ff517
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Sat Dec 8 18:16:29 2018 -0500

scanner

commit 2d251e886b7b381bf744ca46f340b4941d404eb6
Author: Sushanth Raman <s.raman@columbia.edu>
Date: Sat Dec 8 18:13:42 2018 -0500

updated scanner

commit 4d6faf7275869e9355ff9c78437aeafdd0a7246e
Author: Sushanth Raman <s.raman@columbia.edu>
Date: Sat Dec 8 18:04:45 2018 -0500

a

commit 88241533995c283bd05a7ee58d2cacade0f5ed70
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Sat Dec 8 17:57:29 2018 -0500

Paren

commit 413a07a6d38835bd30d3ef2ca6ee35e86b385562
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Sat Dec 8 17:55:08 2018 -0500

Add back in

commit 2a9b359bebf40941df805d3ee85694fa754e1ea7
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Sat Dec 8 17:54:50 2018 -0500

Semant outputs fn

commit 38fa7df52b2deae5a20e3fec27a73f873892c6f6
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Sat Dec 8 17:42:51 2018 -0500

Indent correctly

commit 3b0cd7789bc46ba6549ef4c609343b26177972e4

Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Sat Dec 8 17:36:23 2018 -0500

remove in

commit 1882ff8db0818f0767d0ba07fb3fdce6a2d0ae7b
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Sat Dec 8 17:35:59 2018 -0500

Have semant output correct

commit 12bd62e1c72238db1a7a6c7c28add439ba12d709
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Sat Dec 8 17:34:10 2018 -0500

Add let

commit 939e977f45929a70ea540d424c4ae211582a1547
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Sat Dec 8 17:33:31 2018 -0500

No fn locals

commit 95a21b477dc163837050dd54db120e5197b027e2
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Sat Dec 8 17:32:55 2018 -0500

No var

commit 0215cff7a79c0317b1dbe5f460e241ff4201fa62
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Sat Dec 8 17:32:06 2018 -0500

sexpr type prim

commit 5a0ab63e5ac9b474d3332ce03e649f27640e441a
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Sat Dec 8 17:31:08 2018 -0500

String of float

commit 92c12879fa38e9f0098fc3a0ac14406bba6671dd
Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Sat Dec 8 17:29:41 2018 -0500

Remap names in pretty print

commit 9723ba393325df3adb7ed5b8a8ed35cf6d902ecd

Merge: c4ff496 500033a

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Sat Dec 8 17:27:20 2018 -0500

Merge branch 'master' of github.com:KanishkVashisht/scolang

commit c4ff496f23380480599be9351e533c8906fe2616

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Sat Dec 8 17:27:17 2018 -0500

string_of_sprogram

commit 500033afe8b56ba68b5a9b76897bc8315e082bed

Author: Sushanth Raman <s.raman@columbia.edu>

Date: Sat Dec 8 17:23:41 2018 -0500

updated scolang

commit 4f03fae47d8c099e85d00a23452c6ec0bf6bebfd

Merge: 51585fe b3dbd8a

Author: Sushanth Raman <s.raman@columbia.edu>

Date: Sat Dec 8 17:22:12 2018 -0500

Merge branch 'master' of https://github.com/KanishkVashisht/scolang

commit 51585fea5c681a7db7a8b8427aa181f56ffa104d

Author: Sushanth Raman <s.raman@columbia.edu>

Date: Sat Dec 8 17:21:51 2018 -0500

updated scolang

commit b3dbd8a4fb0e97c2bed4ca45051ac0de8b8d806c

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Sat Dec 8 17:09:51 2018 -0500

Uncomment

commit cbe59d4dca6ceff00320e9b1b6a862a75ef0a9ba

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Sat Dec 8 17:01:03 2018 -0500

Temp remove float

commit 840d2fd93b39c288aa968a53e3d941b8d7d78cb4

Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>

Date: Fri Dec 7 22:54:44 2018 -0500

TESTALL

commit b728aa09545ede51427c8b0143b8d3b1e9dc00d5

Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>

Date: Fri Dec 7 22:53:03 2018 -0500

yello world

commit 5f61122d6cfbd469f475fa436f5cd757c9200385

Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>

Date: Fri Dec 7 22:52:13 2018 -0500

helloworld

commit d694f54801aca146e3724ea81b1661a89dac4dc7

Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>

Date: Fri Dec 7 22:51:04 2018 -0500

updating gitignore

commit 96ae09803c4e1a169dfe2ab15cba7276204b22e7

Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>

Date: Fri Dec 7 22:49:25 2018 -0500

testing

commit b66e23d53579e417eb038216b66a4cf1f1528603

Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>

Date: Fri Dec 7 22:45:59 2018 -0500

Test script

commit 4756880599806f607059adf12b265aa2cea83bb4

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Thu Dec 6 21:33:55 2018 -0500

string of prim in check stmt

commit 692a317404f452516d97bb0eb4f92642555197c1

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Thu Dec 6 21:33:09 2018 -0500

typ -> prim in statement

commit 4ca4005eaea075f262b950edcf94cbd2d5280def

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Thu Dec 6 21:32:12 2018 -0500

Remove more prim

commit 763f623eca8054d3eca3a35ad527a954f3a0608b

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Thu Dec 6 21:31:44 2018 -0500

Remove more primitive constructor

commit 8af15add6d0a7611760903744e24bfc11bd3bc26

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Thu Dec 6 21:31:14 2018 -0500

Remove prim const

commit c57d6687ef65f1023206d22947abac16a20a1f00

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Thu Dec 6 21:30:34 2018 -0500

change sexpr

commit c050865fc00acf01fc38c5f5bbaf87474e5a680c

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Thu Dec 6 21:28:27 2018 -0500

sast add s to everything in sstatement

commit faf7b5a64ff02781923d0e2825079c5eaa299cac

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Thu Dec 6 21:22:56 2018 -0500

typ in call

commit 7cd64c1bd436edcb2a6f251f3ff7667935c41da6
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Thu Dec 6 21:18:40 2018 -0500

Sast misspelling

commit 76c5edd757d436906d80a1cd653bdea31f553bc0
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Thu Dec 6 21:16:25 2018 -0500

Call output prim first tuple

commit d29c1aba05ed889d6ce6f0fea008cb3e3525686a
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Thu Dec 6 21:15:31 2018 -0500

Restructure fn

commit 18b103988d25fdb91038ce5216b7300727504136
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Thu Dec 6 21:02:53 2018 -0500

string of prim

commit 3e2c89597a2bd5390e2d30a4c429c9f2796b835a
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Thu Dec 6 21:02:14 2018 -0500

pass correctly into sarrayindex

commit e7c5739e3db555b03c6cd2ce332c00d6817b795b
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Thu Dec 6 21:01:07 2018 -0500

Fn takes in lit

commit ff2926f9cda5dd083df7ffa68986e72fd645b057
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Thu Dec 6 20:59:15 2018 -0500

Arg conf

commit 6887bf7dcce60d03e71c790176bd52c0f8562896
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Thu Dec 6 20:57:24 2018 -0500

arrayindex lt fix

commit d6f727fec2030bc57df11b76e4a0d9ff4c6ec708
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Thu Dec 6 20:56:31 2018 -0500

Toss that

commit f6b2036daba0d89ae101863c0a3a9a3c3f8f8d31
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Thu Dec 6 20:54:37 2018 -0500

Lit arg takes more arg

commit fb5209d6be9ccff7439888850a92b4995c97e5b6
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Thu Dec 6 20:53:39 2018 -0500

Create type_of_lit fn

commit cfb82612de144fbae4e855fccd1a9f7ff90455f6
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Thu Dec 6 20:43:59 2018 -0500

Fix mismatch paren

commit 821f6643206c561dc2a29abb1bce1192e2969a99
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Thu Dec 6 20:43:00 2018 -0500

Fix arrayindex

commit c3ac8dd61fd5da9422d8595449a9b508f1408d1d
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Thu Dec 6 20:32:29 2018 -0500

Arrayindex set var correctly

commit 7dc57102c3f0e36059c44777aa39a9f5d4805223
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Thu Dec 6 20:24:23 2018 -0500

Add primitive to binop

commit 83ed24253bde77c8107110dea34b9a5051a0989c
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Thu Dec 6 16:46:01 2018 -0500

a

commit 79546224bf77c160e6aa2810136df2309c37353f
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Thu Dec 6 16:45:36 2018 -0500

Fix binop

commit 79e7394636359a3b05e9cfa2b747b56988ecddb0
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Thu Dec 6 16:44:10 2018 -0500

a

commit 3ece6f419ec78407a30d77835911f0006daa18d4
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Thu Dec 6 16:43:30 2018 -0500

a

commit 5ffa0c7b03d8a6e4abc82a3acc00be84d73bcfde
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Thu Dec 6 16:43:09 2018 -0500

remove neg uop

commit e7c32fc24b150e8b1e89a4f1aca83dcb325cc48e
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Thu Dec 6 15:43:31 2018 -0500

Cast

commit 71d566699ae0392425f7dcae918baa837d69215a
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Thu Dec 6 02:04:29 2018 -0500

update print

commit 7943d8ec9373ede986500bded62c30870b193f09
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Thu Dec 6 02:03:25 2018 -0500

prim_of_dt

commit 462953929de10b928d8787e05c96079bb3638c94
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Thu Dec 6 01:54:52 2018 -0500

String of prim

commit 6abf86ee73533e0eb6f52c9e97115043080e5b1e
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Thu Dec 6 01:54:23 2018 -0500

Remove unnecessary casting

commit 8315a56187af15ca303764eb45749271b4a3a04e
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Thu Dec 6 01:53:30 2018 -0500

cast to prim

commit a338e4d7632a7cf4d365e0ea33b2d04cbf674f3a
Author: Sushanth Raman <s.raman@columbia.edu>
Date: Thu Dec 6 01:49:23 2018 -0500

updated sast

commit e41fb5fca8e7cdca76c8b5238e6017da354ec38b
Author: Sushanth Raman <s.raman@columbia.edu>
Date: Thu Dec 6 01:46:17 2018 -0500

updated sast

commit 18fb170f42165ace5b3fe1e4a1e3391b8c745c0f

Author: Sushanth Raman <s.raman@columbia.edu>
Date: Thu Dec 6 01:41:44 2018 -0500

updated sast

commit f2e67fe173f730554c0fe5574739cc4f25278942
Author: Sushanth Raman <s.raman@columbia.edu>
Date: Thu Dec 6 01:41:11 2018 -0500

updated sast

commit ddb99e35b8f1e1a1d45104d23b55431f5933ca6a
Merge: 991b08f a8db7dc
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Thu Dec 6 01:11:46 2018 -0500

Merge branch 'master' of github.com:KanishkVashisht/scolang

commit 991b08f5649bf5929de2d09d32991e5143afb88d
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Thu Dec 6 01:11:44 2018 -0500

Fix bind

commit a8db7dc34ac92d67ccd5cc2fcd15f188a8587506
Author: Sushanth Raman <s.raman@columbia.edu>
Date: Thu Dec 6 01:07:06 2018 -0500

updated sast

commit e2ce9861929de2d583144369ed85b3ca0ea1e241
Merge: 63710e2 a820ca3
Author: Sushanth Raman <s.raman@columbia.edu>
Date: Thu Dec 6 01:02:46 2018 -0500

Merge branch 'master' of https://github.com/KanishkVashisht/scolang

commit 63710e2382e6d88cd2db3f1271fae7589bd4cf12
Author: Sushanth Raman <s.raman@columbia.edu>
Date: Thu Dec 6 01:02:38 2018 -0500

updated semant

commit a820ca3511704e592e909b74acf0174a5f15a98a
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Thu Dec 6 01:01:39 2018 -0500

Remove

commit 20904edde3eba65433fcdf2229c183c3e5c1d5b0
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Thu Dec 6 01:01:21 2018 -0500

pass in typ

commit 63680f7acb80fabc64cefcddde712f7d31ef7dafb
Author: Sushanth Raman <s.raman@columbia.edu>
Date: Thu Dec 6 00:58:35 2018 -0500

updated sast

commit abc5dc9ca1597624f2b13ad45a0d6ce59828a86f
Merge: 5dbd718 a644387
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Thu Dec 6 00:53:43 2018 -0500

Merge branch 'master' of github.com:KanishkVashisht/scolang

commit 5dbd718ebe121ecaeeef1d06d60878700f97e5f50
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Thu Dec 6 00:53:41 2018 -0500

string of prim

commit a6443876dc62a5d3bc561e4ee0717b3a13132312
Author: Sushanth Raman <s.raman@columbia.edu>
Date: Thu Dec 6 00:52:48 2018 -0500

updated semant

commit 7aaccfe36cf5053c95ea3d8dbc2b0a4b65b8c500
Author: Sushanth Raman <s.raman@columbia.edu>
Date: Thu Dec 6 00:52:17 2018 -0500

updated semant

commit 77f5d4a736a12bac3cf0aaa7f39857558cc87810
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Thu Dec 6 00:48:08 2018 -0500

typ

commit 874be37d573a1ea5209c4f23e1cd5e5b5ff284fb
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Thu Dec 6 00:46:57 2018 -0500

Temp comment out bind

commit 1757413415ca31f327c92f9ae842086f480f530b
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Thu Dec 6 00:45:13 2018 -0500

Remove cast

commit df0e130ce293e1345fd4612e926fbec4fd788834
Merge: 9713813 a9ca32e
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Thu Dec 6 00:44:14 2018 -0500

Merge branch 'master' of github.com:KanishkVashisht/scolang

commit 971381385ed4e8c5c1de0a3b2cd5f070678dc7df
Author: Jacksn <jc@dyn-160-39-197-156.dyn.columbia.edu>
Date: Thu Dec 6 00:43:34 2018 -0500

Sexpr

commit a9ca32e58e871e7cee13bd8ab5f5b9feace08d3f
Author: Sushanth Raman <s.raman@columbia.edu>
Date: Thu Dec 6 00:41:13 2018 -0500

added cast

commit dab674c6d9a6400c8d3c7256fa242b830ffad3c8
Author: Sushanth Raman <s.raman@columbia.edu>
Date: Thu Dec 6 00:34:22 2018 -0500

updated semant

commit bc2a08dac2dbd8d494f003d2c9d66b243b4b2136
Author: Sushanth Raman <s.raman@columbia.edu>
Date: Thu Dec 6 00:31:19 2018 -0500

updated semant

commit 776e21466e517e979cc6c26b02b9c124be52bf2f
Author: Sushanth Raman <s.raman@columbia.edu>
Date: Thu Dec 6 00:24:46 2018 -0500

updated sast

commit 9ec80db5f5654aa48a04d53373732adf01ce38c1
Author: Sushanth Raman <s.raman@columbia.edu>
Date: Thu Dec 6 00:19:55 2018 -0500

a

commit 2f9fbcfb24bea5799c41553ee2150601d9db89b2
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Mon Dec 3 15:59:27 2018 -0500

opam

commit 54d1329b4c314db8df843445bc2fe5ab5086a515
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Sun Dec 2 22:39:40 2018 -0500

test

commit 434931116040f9f80592ef7d79511f08313bcd5a
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Sun Dec 2 22:37:40 2018 -0500

revert

commit d0a48bb9177a25772ac1895f6f42292a187ab6b5
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Sun Dec 2 22:37:03 2018 -0500

t

commit 4fbb8f2abca11abd136a15b87a824cb69f564a8e

Author: Sushanth Raman <s.raman@columbia.edu>

Date: Sun Dec 2 22:34:07 2018 -0500

updated semant

commit 345c214558ab0b057464420baece1315550e62b5

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Sun Dec 2 22:32:57 2018 -0500

Get rid of that

commit d92594bed87605e33584c67e0e6a4fd5aae1cffa

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Sun Dec 2 22:30:12 2018 -0500

lol

commit aeedb03215b302edf52bb1552546a47c49a1017d

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Sun Dec 2 22:28:04 2018 -0500

prim_of_dt

commit 19e860f49ba9c0e6d3f8e0c545e43feaa3b40aff

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Sun Dec 2 22:25:59 2018 -0500

Add par

commit 3a8f41bfec60f0a9e46943b436e45390faa931a1

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Sun Dec 2 22:25:22 2018 -0500

type of ident -> prim

commit 2cd63e48a60d77b27ae497de6feb8038584027ae

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Sun Dec 2 22:20:52 2018 -0500

a

commit a9a8fdedc4800d811c0dd401e506d0fc8d175c36

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Sun Dec 2 22:19:50 2018 -0500

a

commit 60439ad686bf2bef6abf067606555298f07fc201

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Sun Dec 2 22:19:19 2018 -0500

test

commit b9adf0a81b8a118db49da378cfac55299394bcef

Author: Sushanth Raman <s.raman@columbia.edu>

Date: Sun Dec 2 22:15:51 2018 -0500

updated semant

commit 928b4eead213e7cfb3a3b11ef7d520efdf4fb9b7

Author: Sushanth Raman <s.raman@columbia.edu>

Date: Sun Dec 2 22:15:17 2018 -0500

updated semant

commit fbd75f49257b0f6aad9a6662dc683cc245a573a6

Author: Sushanth Raman <s.raman@columbia.edu>

Date: Sun Dec 2 22:14:35 2018 -0500

a

commit 6d73214b3bc82725de338ad9b0c4edf4a230b166

Author: Sushanth Raman <s.raman@columbia.edu>

Date: Sun Dec 2 22:11:47 2018 -0500

a

commit de49f2237d55a9db5ced140696d2d72a7e6f543d

Author: Sushanth Raman <s.raman@columbia.edu>

Date: Sun Dec 2 22:08:43 2018 -0500

a'

commit 777fa913be664b53634ece074790fba836d75380

Author: Sushanth Raman <s.raman@columbia.edu>

Date: Sun Dec 2 22:05:15 2018 -0500

fixed error

commit 93409d711f57ce9f4de919767d226142ef48d3d6

Author: Sushanth Raman <s.raman@columbia.edu>

Date: Sun Dec 2 22:03:34 2018 -0500

updated sast

commit f4e5209ab07ee704c1f95ffa60b0bbe3a5e53008

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Sun Dec 2 21:57:50 2018 -0500

Misspelled in sast

commit edceaf4e738fd04918a05b0126e6ab954ffd792a

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Sun Dec 2 21:57:16 2018 -0500

Char

commit 9497dded4b87ea39212a993ce9a751eded5b0a78

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Sun Dec 2 21:56:41 2018 -0500

Pasas argu to s

commit ae6db9eb73dd3651f7b606230cc9ed7f07d8222b

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Sun Dec 2 21:56:12 2018 -0500

Cap

commit c0e9ee49362a2b8828528d2ebf03070780678734

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Sun Dec 2 21:55:30 2018 -0500

Spell statement correct

commit 000514a8a2f41206013d93042741fa255fb44e8d

Author: Sushanth Raman <s.raman@columbia.edu>

Date: Sun Dec 2 21:52:12 2018 -0500

updated semant

commit 1075a71992ee2710a3f12e999a235db4bc96bdee
Merge: 066f660 eb704e0
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Sun Dec 2 21:50:13 2018 -0500

Merge branch 'master' of <https://github.com/KanishkVashisht/scolang>

commit 066f6603532907911dae03dfb43e7e9f4bbb9dbd
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Sun Dec 2 21:50:08 2018 -0500

sick

commit eb704e06f51f1d110dc792b9946c58ea8004e465
Author: Sushanth Raman <s.raman@columbia.edu>
Date: Sun Dec 2 21:47:27 2018 -0500

semant.ml

commit aa61902ca4cf9561f2063a78a4be0f81f874b380
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Sun Dec 2 21:40:52 2018 -0500

Cast all this to prim

commit ff763205127b4672f5fcc05cef8dc5588f1745b5
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Sun Dec 2 21:39:44 2018 -0500

lol

commit d963d5a463546c5c27973abb78216980c4cfcfb9
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Sun Dec 2 21:38:57 2018 -0500

Nvm

commit becee5ee4c6985049eea35030c0401415df7ef4c
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Sun Dec 2 21:37:19 2018 -0500

Change primitive

commit b8ebdc560249e131f1854cacafd26e953e20de7a
Merge: 33223cd 945f873
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Sun Dec 2 21:33:32 2018 -0500

Merge branch 'master' of github.com:KanishkVashisht/scolang

commit 33223cdbdcc592336466abe53fcbfa8412ae9286
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Sun Dec 2 21:33:22 2018 -0500

Uncomment out check_binds call

commit 945f873cf36a688d25383a921afcafb7ea8ba4c3
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Sun Dec 2 21:31:01 2018 -0500

;

commit 9b7b77a0f7bec5757f973de285bb3ab7cc768e71
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Sun Dec 2 21:30:49 2018 -0500

arry -> list

commit 0ae0d10bad8b5e4bb85302e57c249db7aed7906a
Merge: 948a757 c3360fb
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Sun Dec 2 21:30:13 2018 -0500

Merge branch 'master' of github.com:KanishkVashisht/scolang

commit 948a757ff93b60ae2122aae8ec357ad17997c9e0
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Sun Dec 2 21:30:07 2018 -0500

Try converting type

commit c3360fbf6651438e31825d758012c18b2bb1ea00
Merge: 0c1133a bd1f813
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>

Date: Sun Dec 2 21:26:40 2018 -0500

Merge branch 'master' of <https://github.com/KanishkVashisht/scolang>

commit 0c1133aca714f061f3393ff464b638e5a36b1f8f
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Sun Dec 2 21:26:33 2018 -0500

llvm

commit bd1f8133bea50d711782585115a3eada08edf7a5
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Sun Dec 2 21:23:23 2018 -0500

Comment everything out

commit 1580e4fdd9dd5540e6344960c853d0a19e3d9acf
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Sun Dec 2 21:22:04 2018 -0500

Mispell arr

commit 039c0e957fb916c205b5f40856b2568bd50805da
Merge: c65e0be d32fb6d
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Sun Dec 2 21:21:18 2018 -0500

Merge branch 'master' of github.com:KanishkVashisht/scolang

commit c65e0be5447842c789a1725f75653b15a22c45ca
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Sun Dec 2 21:20:56 2018 -0500

a

commit d32fb6daf26436c172d21bd54c3e0a8ceebf1e9b
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Sun Dec 2 21:19:41 2018 -0500

jackson loves incest

commit 548716e16a8f60a787b60e84255af43a7636af51
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>

Date: Sun Dec 2 21:17:36 2018 -0500

a

commit 0db816e01c0836ba874095daee10564e2269e7ee

Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>

Date: Sun Dec 2 21:12:28 2018 -0500

elev

commit 79e2c8842b3c41ae217b28668c2b7f0b26e5347f

Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>

Date: Sun Dec 2 21:10:03 2018 -0500

Merge branch 'master' of <https://github.com/KanishkVashisht/scolang>

commit 2539affa6b0cbe1d9037ae51ae6e067898c48d11

Merge: f053e92 76765c0

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Sun Dec 2 21:05:24 2018 -0500

Merge branch 'master' of github.com:KanishkVashisht/scolang

commit f053e92dc5bf06d82118f487ddeb196a47991be4

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Sun Dec 2 21:05:02 2018 -0500

Extra in

commit 76765c015f329b6c72c0806c221e9e2f8689aee3

Merge: 97d1631 1e32b34

Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>

Date: Sun Dec 2 21:04:51 2018 -0500

Merge branch 'master' of <https://github.com/KanishkVashisht/scolang>

commit 97d1631f3587d320f0c3f3aeda7b4f12492cd058

Merge: f48fdcc 0457ac1

Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>

Date: Sun Dec 2 21:04:31 2018 -0500

Merge branch 'master' of <https://github.com/KanishkVashisht/scolang>

commit 1e32b34f4629f40d94290312b7581fa25e7d7b81
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Sun Dec 2 21:04:29 2018 -0500

Fix indent

commit f48fdccbce355785585b487105d6025f9c89f3bd
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Sun Dec 2 21:04:03 2018 -0500

llvm

commit 0457ac186ee6fd78ec17fa0c45a0c0e62acff020
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Sun Dec 2 21:03:05 2018 -0500

Mismatched paren

commit dff6d20c4c0cff0b410a2291f4988d21171b8c87
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Sun Dec 2 21:02:36 2018 -0500

Mismatched paren

commit 235b1b431e175aced26d9445dee0723fc042e03a
Merge: 2b6ac6b 55b1747
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Sun Dec 2 21:01:57 2018 -0500

Merge branch 'master' of github.com:KanishkVashisht/scolang

commit 2b6ac6ba38afabda32461a854989687747454de4
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Sun Dec 2 21:01:55 2018 -0500

Try wrapping statements in main

commit 55b174792c84c3dcdffbc66896ec029f8877e3dfa
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Sun Dec 2 20:58:49 2018 -0500

./testall

commit afe7da9097e19005b306180ddbef8413deddf19d
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Sun Dec 2 20:41:06 2018 -0500

stall

commit 9a756513cd30a182911ff110445b7f57946e75c4
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Sun Dec 2 20:38:51 2018 -0500

tst

commit af9249f5ff11aa1325168907ca2dfb8f07bebb77
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Sun Dec 2 20:32:43 2018 -0500

lli

commit ebb08b8c5b94502462ad12ec44895c3112715e9b
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Sun Dec 2 20:29:09 2018 -0500

removed llvm directly

commit 6bc209e33816c4e08002ca03e4a1d9564da6639e
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Sun Dec 2 20:26:00 2018 -0500

cl

commit ec399e23b27f1e98c7c96bd7a79614a7e3f680ff
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Sun Dec 2 20:14:25 2018 -0500

Remove second_statement

commit 9c7a36e19142ccbad586267ecc7e42ada9e82a1b
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Sun Dec 2 20:13:42 2018 -0500

updated

commit bfb21d983c74642df1818576efe5233fbb7c5d46
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Sun Dec 2 20:10:42 2018 -0500

commit 5d7689404e5e92a7f755f791b7b99e7185195df6
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Sun Dec 2 19:54:42 2018 -0500

scott

commit 9d071f4d89f5b2136517511fe9b1b335bbb84c51
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Sun Dec 2 19:45:29 2018 -0500

travis

commit b8eb1d54425947a231509622a7d592ab5488a9d6
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Sun Dec 2 17:27:19 2018 -0500

llvm keys keys keys keys

commit 509731e5bbb7fd1f1264b26929f738b51f145be1
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Sun Dec 2 17:22:43 2018 -0500

new llvm repo

commit a0153958a6362f692a883debec63d079a0052aca
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Sun Dec 2 17:17:54 2018 -0500

flags

commit ad84e10af933c7bd68133c0fcd1000c199e32b26
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Sun Dec 2 17:11:22 2018 -0500

lad

commit d770ad0aa685406ed13cff9a8ac2cd0fbdd8862
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>

Date: Sun Dec 2 17:09:57 2018 -0500

updated keys

commit 2ac5c09343e6978788f3581abe930125a2082351
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Sun Dec 2 17:06:19 2018 -0500

testing

commit 3a26648093e0c0e8e194d33dd74479d708e479c0
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Sun Dec 2 16:55:13 2018 -0500

add repo

commit 6c7fde60f19f45f5ba9f5d608cf9be2182ba7a21
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Sun Dec 2 16:50:49 2018 -0500

upgradation

commit e14b802137e6758bfdac97b7bd4ba8f474a47e04
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Sun Dec 2 16:45:17 2018 -0500

reducing dependencies

commit 51867b329579816fe586622854a710d7a0be52cd
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Sun Dec 2 16:35:10 2018 -0500

rewrote testing.sh file

commit cd0886bc5629601965444df5c9778f790aa8949e
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Sun Dec 2 16:19:15 2018 -0500

fixing opam bugs

commit 3542481ef2cc71d68deea62f67ddc62763886efc
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Sun Dec 2 16:17:15 2018 -0500

testing email notifs

commit 46f292da242114b26cc6f30d0a146c369204d5ca
Author: Kanishk Vashisht <kanishk.vashisht@gmail.com>
Date: Sun Dec 2 16:08:39 2018 -0500

email notifs

commit a3be3fc678b69b38659e1a4bae1aac952202249f
Author: Kanishk Vashisht <kv@dyn-129-236-215-158.dyn.columbia.edu>
Date: Sun Dec 2 15:58:40 2018 -0500

testing

commit 17e0e1f360c564807a0a84a61c7cf9c8c1cf0ffc
Author: Kanishk Vashisht <kv@dyn-129-236-215-158.dyn.columbia.edu>
Date: Sun Dec 2 15:51:59 2018 -0500

testing

commit 55a4086d7eb45d4cad642d1b7afba22a3d11e0dc
Author: Kanishk Vashisht <kv@dyn-129-236-215-158.dyn.columbia.edu>
Date: Sun Dec 2 15:43:56 2018 -0500

Updating tests

commit fe248c707816c01ad31b289b72d4661ceced81dd
Author: Kanishk Vashisht <kv@dyn-129-236-215-158.dyn.columbia.edu>
Date: Sun Dec 2 15:39:58 2018 -0500

correcting environment vars

commit 01d4db284a3d75a6cedafe233872b7119a2d55b4
Author: Kanishk Vashisht <kv@dyn-129-236-215-158.dyn.columbia.edu>
Date: Sun Dec 2 15:37:03 2018 -0500

Travis CI Check

commit bdef32c717776fb3aa4590d9de4ed34a609ac8bc
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Sun Dec 2 01:46:37 2018 -0500

Compile sast

commit 229663ed6a6d5c1fedff4fceb91b08117619bfc
Merge: 737aa20 ce5ff76
Author: Sushanth Raman <s.raman@columbia.edu>
Date: Sun Dec 2 01:42:01 2018 -0500

Merge branch 'master' of <https://github.com/KanishkVashisht/scolang>

commit 737aa20f2e99a59f0908084874c91064edeb22c5
Author: Sushanth Raman <s.raman@columbia.edu>
Date: Sun Dec 2 01:41:49 2018 -0500

sast.ml

commit ce5ff76121a2aa8663f9b2dc96958b8ed649dc13
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Sun Dec 2 01:32:27 2018 -0500

Compile semant

commit 7c2c88fcd7dcec6c6926ffd613721c860edc57b8
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Sun Dec 2 01:16:50 2018 -0500

import semant

commit 623494fb26c5e2bf223c8ec9e781f5289ac1665f
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Sun Dec 2 01:15:45 2018 -0500

Add back semant to scolang

commit bb2c2e1eff81954a126663af1a2cf2a929d739af
Author: Sushanth Raman <s.raman@columbia.edu>
Date: Sun Dec 2 01:13:41 2018 -0500

updated semant to consider local variables

commit dd90bab30488ca1b710bc745a2935a23488a7397
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Sun Dec 2 00:50:03 2018 -0500

Move varinit

commit 6c970c21e3e7e930c6a94e8421a951c0493e5206
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Sun Dec 2 00:45:21 2018 -0500

Remove decompose from scanner.mll

commit f95165c4df9a130e38b26a7df81bcfd4ab3d938b
Merge: 55b7467 bb30fd8
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Sun Dec 2 00:44:23 2018 -0500

Merge branch 'master' of github.com:KanishkVashisht/scolang

commit 55b7467132e29ae7c82d3a1e46e1d701e9f98953
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Sun Dec 2 00:44:03 2018 -0500

Move var instant + declar into statement

commit bb30fd8e66bbe23b11856e47b63fe9269f0711ae
Author: Sushanth Raman <sushanthraman@Sushanths-MacBook-Pro.local>
Date: Sun Dec 2 00:21:07 2018 -0500

Updated Semant

commit 0ed3b645c358e73645093150d421efdfbc458ecc
Author: Sushanth Raman <sushanthraman@Sushanths-MacBook-Pro.local>
Date: Sat Dec 1 23:15:20 2018 -0500

semant.ml

commit 2199c87288024370728db7877767ddfc4827b86d
Merge: 5005e95 6b2dbdf
Author: Sushanth Raman <sushanthraman@Sushanths-MacBook-Pro.local>
Date: Sat Dec 1 23:01:06 2018 -0500

Merge branch 'master' of https://github.com/KanishkVashisht/scolang

commit 5005e9586aa14400acb3a6f61f6dcbf53ba4bf33
Author: Sushanth Raman <sushanthraman@Sushanths-MacBook-Pro.local>
Date: Sat Dec 1 23:00:44 2018 -0500

Adding first version of semant

commit 6b2dbdfbb040bb3738b5c0247fc523b4922b6eeb

Merge: 7025adc b981815

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Sat Dec 1 22:49:08 2018 -0500

Merge branch 'master' of github.com:KanishkVashisht/scolang

commit 7025adcd0eeda3c81f4789b0270bdcc172bd0a16

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Sat Dec 1 22:49:03 2018 -0500

Remove fluff

commit b9818150620674461f3c606b50c84eb4efddb91a

Author: vnv-varun <vnv-varun@users.noreply.github.com>

Date: Tue Nov 27 19:44:12 2018 -0500

Added in Vim and Nano

commit c3250c18a77a59b784a13cd1e9ad6b4927528021

Author: vnv-varun <vnv-varun@users.noreply.github.com>

Date: Wed Nov 14 22:48:02 2018 -0500

Added to Readme

commit a6b893b486ebf39678d8a3f76958c003b3a6eacd

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Wed Nov 14 21:58:00 2018 -0500

run -> testall

commit e9ff23370407a50e0061d7cd23c5f892693f6da1

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Wed Nov 14 21:55:57 2018 -0500

make shell executable

commit 847b62e89a8578214c2f473bf8e2f51acf093698

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Wed Nov 14 21:54:54 2018 -0500

FINAL VERSION

commit ad098a4446710675a15bf8ed7b860a39d7650ffa
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Wed Nov 14 21:37:08 2018 -0500

kanishk do not touch anything on this commit or after

commit c6d95f6e189d893aac84170eb78140ce779df9ea
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Wed Nov 14 21:33:26 2018 -0500

default state

commit 98e68f5ef15b7854dbde7f16fa4ea49715778457
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Wed Nov 14 20:43:06 2018 -0500

simple statement

commit d3b4128b5a8b7c8fbfcfc4934ea3752b5b68c618
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Wed Nov 14 20:16:00 2018 -0500

stacktrace

commit 66088b190a73f07c8bf0e1791ba96bd8328aea0f
Merge: f925b62 80a626c
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Wed Nov 14 19:44:40 2018 -0500

Merge branch 'master' of github.com:KanishkVashisht/scolang

commit f925b62f32f5bc43dab08e227fc9dbded307f5e9
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Wed Nov 14 19:44:22 2018 -0500

indent

commit 80a626c38b37190e139b68cbd42523cb113c374c
Author: vnv-varun <vnv-varun@users.noreply.github.com>
Date: Wed Nov 14 19:43:49 2018 -0500

fixing more

commit a14e920732ba96a749e628714ce89a24c4b78d8d
Merge: bd37829 15d0835
Author: vnv-varun <vnv-varun@users.noreply.github.com>
Date: Wed Nov 14 19:43:16 2018 -0500

Merge branch 'master' of <https://github.com/KanishkVashisht/scolang>

commit bd37829ed9e8fb9d2416449c0112556150b399b4
Author: vnv-varun <vnv-varun@users.noreply.github.com>
Date: Wed Nov 14 19:43:13 2018 -0500

small fix

commit 15d08350cb86d4e9dbb94cb8c7afa6cbb4c41f13
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Wed Nov 14 19:39:19 2018 -0500

microc -> scolang

commit c21c4003b0958221dc74a2fda4f317a216f26d01
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Wed Nov 14 19:31:54 2018 -0500

move as s out

commit 15f101db491a6533afe7e4e7ce29d1c5eb046a47
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Wed Nov 14 19:30:49 2018 -0500

string quote

commit 5514eec7c658f39a11ee7e3b05010676bd9437c5
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Wed Nov 14 19:16:06 2018 -0500

type compat

commit 786f88ce17da4fa250444ed5d013119948a2c3ea
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Wed Nov 14 19:14:45 2018 -0500

No semant

commit 8979f934562a9b6cc2aa2e857267050404b52f6c
Merge: 4eedc00 154c394
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Wed Nov 14 19:14:11 2018 -0500

Merge branch 'master' of github.com:KanishkVashisht/scolang

commit 4eedc00f256d3eebbbe4546fb247b85f2263c049
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Wed Nov 14 19:14:07 2018 -0500

Change to start_state

commit 154c39446ac0555be793199040d20378ba430826
Author: vnv-varun <vnv-varun@users.noreply.github.com>
Date: Wed Nov 14 19:09:00 2018 -0500

Solid

commit e03064907504e66274a19743abfde7fbcac9ae66
Author: vnv-varun <vnv-varun@users.noreply.github.com>
Date: Wed Nov 14 19:07:44 2018 -0500

semant.cmx removed

commit 28bd3645a908f5b10f12b8b6962fa1b9a7995b2d
Author: Sushanth Raman <sushanthraman@Sushanths-MacBook-Pro.local>
Date: Wed Nov 14 19:01:29 2018 -0500

fixed error

commit b496e3fdb77e7519895a33ed0d31316a298d278b
Merge: 0f0ed94 a0f3fc6
Author: Sushanth Raman <sushanthraman@Sushanths-MacBook-Pro.local>
Date: Wed Nov 14 18:59:26 2018 -0500

Merge branch 'master' of <https://github.com/KanishkVashisht/solang>

the commit.

commit 0f0ed94b56269bbf2541945374d738828b2cc9c0

Author: Sushanth Raman <sushanthraman@Sushanths-MacBook-Pro.local>

Date: Wed Nov 14 18:58:19 2018 -0500

fixed error

commit a0f3fc645e55803e6330679eb1210eec99f63f1d

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Wed Nov 14 18:54:58 2018 -0500

unindent dependency tree

commit 61861927261c28d47d1a327c83c6fe188779569c

Author: Sushanth Raman <sushanthraman@Sushanths-MacBook-Pro.local>

Date: Wed Nov 14 18:51:09 2018 -0500

Added makefile

commit b8052e2867635bde14ea0d45574a56431e4be646

Author: vnv-varun <vnv-varun@users.noreply.github.com>

Date: Wed Nov 14 18:42:34 2018 -0500

solang.ml

commit da5c8b1bb0deae39593fc06841d37967948f1cb1

Author: vnv-varun <vnv-varun@users.noreply.github.com>

Date: Wed Nov 14 18:05:50 2018 -0500

Revert "microc"

This reverts commit [ac10d38688c2959dedd2cb4b915d37a0ae4834b2](https://github.com/KanishkVashisht/solang/commit/ac10d38688c2959dedd2cb4b915d37a0ae4834b2).

commit dcfa1fb577865f8f6b5d39df7ae86dc49cbb6ff6

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Wed Nov 14 05:25:34 2018 -0500

microc test

commit ac10d38688c2959dedd2cb4b915d37a0ae4834b2
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Wed Nov 14 05:23:35 2018 -0500

microc

commit a4cb90005b5147821e4a78343840ec5319b1b3a9
Merge: b3648ff ce6abe6
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Wed Nov 14 04:53:53 2018 -0500

Merge branch 'master' of github.com:KanishkVashisht/scolang

commit ce6abe61357f1951d48c3b78bd67ef9e3da2c605
Author: vnv-varun <vnv-varun@users.noreply.github.com>
Date: Wed Nov 14 04:34:54 2018 -0500

Deleted unneeded/duplicate test

commit 345174b3d0d581baff33d72f910eda8a04874d56
Author: vnv-varun <vnv-varun@users.noreply.github.com>
Date: Wed Nov 14 04:33:51 2018 -0500

Added in Readme as Jennifer wanted

commit 7a531c74905183dd1072c526d8ae6c5067fe58c4
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Wed Nov 14 04:26:37 2018 -0500

More Minor Changes

commit 92e8b2e5dae1cf56023d2cbc94736891b2371b3f
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Wed Nov 14 04:25:09 2018 -0500

Trying Something

commit d58797fbeb213cca471f2b004ab9a8218280205e
Merge: b8241d9 48029f1
Author: vnv-varun <vnv-varun@users.noreply.github.com>

Date: Wed Nov 14 04:22:29 2018 -0500

Merge branch 'master' of <https://github.com/KanishkVashisht/scolang>

commit 5acf13feb5e51ca6d3e40438fb4704f6d1191c50
Author: vnv-varun <vnv-varun@users.noreply.github.com>
Date: Wed Nov 14 04:19:49 2018 -0500

Revert "Merge branch 'master' of <https://github.com/KanishkVashisht/scolang>"

This reverts commit 2369e45e3c2820a890c882e1217c904d85369ddb, reversing changes made to 10fcafb0a78603c22d0265a157a7674f2712967d.

commit 2369e45e3c2820a890c882e1217c904d85369ddb
Merge: 10fcafb b66c813
Author: vnv-varun <vnv-varun@users.noreply.github.com>
Date: Wed Nov 14 04:19:19 2018 -0500

Merge branch 'master' of <https://github.com/KanishkVashisht/scolang>

commit 646394461f50a72da1d72d97fe02a3352b8df30c
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Wed Nov 14 02:56:29 2018 -0500

Trying different things

commit 1704ec0414689920031bb5e9812585e214bf72ef
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Wed Nov 14 02:39:30 2018 -0500

sim

commit b0ac564e1ff0b761b38ed5373830c39fde975320
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Wed Nov 14 02:28:15 2018 -0500

lol

commit 49aa69f59127886636f4a3f6102f073e424617b7
Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Wed Nov 14 02:26:20 2018 -0500

No void:

commit fbd98b399267878ecfb5fd841884ba2cb7851d83

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Wed Nov 14 02:24:45 2018 -0500

Plan b

commit 7acaffc959ea54eb6395a335858358457f3d78b6

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Wed Nov 14 02:01:26 2018 -0500

s

commit 02824c94af306f36b50a09f6b067ef9bf5f94f04

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Wed Nov 14 02:01:04 2018 -0500

builder

commit d106b3ae26ef2d0cf84522471e0caf7a532aa5fb

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Wed Nov 14 02:00:31 2018 -0500

int_format_str

commit 2a140cea26427d7f1acd87af1a4e7ab9d6efd241

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Wed Nov 14 01:59:11 2018 -0500

+ -> -

commit a2ace012c95cc8752e69d80ded17b8598958176f

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Wed Nov 14 01:44:25 2018 -0500

add dep

commit f2517ca34a45db27251cd4963c9b4b38b67faf3b

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Wed Nov 14 01:41:02 2018 -0500

Make work

commit 1549af4a409fcdccd214a08efe3b9d2049297fd5
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Wed Nov 14 01:29:32 2018 -0500

makefile

commit 56aa774c2ee95d7b7fb718b376fd4050ee1cc58e
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Wed Nov 14 01:24:34 2018 -0500

Swap ast and llvm

commit 1e4f850475a32d657f5ca7f371717a9bfc4208a7
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Wed Nov 14 01:23:48 2018 -0500

add in

commit 5743337b5b74f68a0e2a3cb93b3af61c9c809da6
Merge: 65397d4 2ced452
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Wed Nov 14 00:07:20 2018 -0500

Merge branch 'master' of github.com:KanishkVashisht/scolang

commit 65397d4f592a74e50a35cfa9e26dc96a60396e7a
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Tue Nov 13 23:45:43 2018 -0500

Everything works in parser

commit 2ced452a17ae793eb6beb98d255049685823c40f
Author: vnv-varun <vnv-varun@users.noreply.github.com>
Date: Tue Nov 13 23:26:25 2018 -0500

Fixed typo

commit e211c91222402242a721ce8f1ee28ca9ae8aa061
Merge: f9bcb7f 2e23135
Author: Sushanth Raman <sushanthraman@Sushanths-MacBook-Pro.local>

Date: Tue Nov 13 23:23:58 2018 -0500

Merge branch 'master' of <https://github.com/KanishkVashisht/scolang>
the commit.

commit f9bcb7f7b939e28a9101776edfbd49db653a4afc
Author: Sushanth Raman <sushanthraman@Sushanths-MacBook-Pro.local>
Date: Tue Nov 13 23:23:04 2018 -0500

Code generation file

commit 2e23135e3483bd732cd3defd507ee4859b51417b
Merge: 649d2e6 f826cda
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Tue Nov 13 23:18:16 2018 -0500

Merge branch 'master' of github.com:KanishkVashisht/scolang

commit 649d2e624f61aa8021add81769f721479a3f7afc
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Tue Nov 13 23:18:12 2018 -0500

Remove eof dup token

commit f826cda05d36f2fb02070ea08397270fc8c6730b
Author: vnv-varun <vnv-varun@users.noreply.github.com>
Date: Tue Nov 13 23:12:19 2018 -0500

Dockerfile

commit 21486a4bebbac7cd7f332d2394f8df06bfd9acc7
Merge: ee69af6 f4137be
Author: Sushanth Raman <sushanthraman@Sushanths-MacBook-Pro.local>
Date: Tue Nov 13 21:38:14 2018 -0500

Merge branch 'master' of <https://github.com/KanishkVashisht/scolang>

commit f4137be89e4354d470b9fb942a93446443e240d5
Author: vnv-varun <vnv-varun@users.noreply.github.com>
Date: Tue Nov 13 17:07:40 2018 -0500

Added working tests

Added tests that compile with their respective output files to compare. Only tests remaining are Contracts and Listeners

commit ee69af67cfbd3e42d8cd1ad2c9ad3ecb8219585a
Author: Sushanth Raman <sushanthraman@Sushanths-MacBook-Pro.local>
Date: Mon Nov 12 00:33:11 2018 -0500

Semantic checker

commit bf9fab1263286cc63825e1e080926a90731938ce
Author: vnv-varun <vnv-varun@users.noreply.github.com>
Date: Mon Nov 12 00:24:34 2018 -0500

Success Test Cases

commit 2017e75fc1aac24ee8e085bd28c36564e0da2de9
Author: Sushanth Raman <sushanthraman@Sushanths-MacBook-Pro.local>
Date: Mon Nov 12 00:23:54 2018 -0500

Semantic checker

commit fadd1e360db38bb9b42dab4b79b6da05047f454e
Merge: 56da26a 34343ff
Author: Kanishk Vashisht <kv@dyn-129-236-214-97.dyn.columbia.edu>
Date: Sun Nov 11 23:37:34 2018 -0500

Merge branch 'master' of <https://github.com/KanishkVashisht/scolang>

commit 56da26ae33557f3896f869116a486a5430f84b08
Author: Kanishk Vashisht <kv@dyn-129-236-214-97.dyn.columbia.edu>
Date: Sun Nov 11 23:36:45 2018 -0500

Failing test cases

commit 34343ffe0b2b5f96b1d20d7865208fc96ae06b5c
Merge: ab26831 ba81659
Author: Jackson Chen <jackson.chen@columbia.edu>
Date: Sun Nov 11 23:30:24 2018 -0500

Merge branch 'master' of github.com:KanishkVashisht/scolang

commit ab26831775f04cef97a9d80be2998dde63fe347f

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Sun Nov 11 23:30:07 2018 -0500

got a working prototype

commit ba816598cb2d5a3b17f7f95c1868f9ea74aae00f

Author: vnv-varun <vnv-varun@users.noreply.github.com>

Date: Sun Nov 11 22:12:52 2018 -0500

Test While Output 1

commit 5348d6cab36690867396d68de1b20d7da5340ca1

Author: Sambhav Anand <sambhav182@gmail.com>

Date: Sun Nov 11 20:57:53 2018 -0500

Fixed shift reduce errors

commit 66f9007316abee8516b6c24348c35eb7e71877fe

Author: Sambhav Anand <sambhav182@gmail.com>

Date: Sun Nov 11 20:35:30 2018 -0500

Fixed Error with building. Need to deal with reduce conflicts now

commit d5be7e0cd9e18b481db0970c09b6ecacca3afa2

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Sun Nov 11 20:01:02 2018 -0500

Fix parser scanner etc

commit 8cbe3f057bc7fd01c43dc0e46ba70413d04ab6b7

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Sun Nov 11 19:01:55 2018 -0500

Fix issues in parser and ast

commit 57f79b770af992f84be185a6d2332db7c164e9c0

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Sun Oct 28 22:06:22 2018 -0400

Add ast

commit 70c7c0656f8cd39a48ae3be5b26b7c48c058a5ee

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Sun Oct 28 21:10:26 2018 -0400

Remove elif

commit c3a4271a58db93da0a59f54a54fc8478e7e688d6

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Sun Oct 28 21:01:33 2018 -0400

Get rid of elif

commit e7bd9c18013859d83a16dedf55940e61a45c46bc

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Sun Oct 28 20:35:54 2018 -0400

Add return & action, remove decompose, restructure bind

commit 8c0680b7dc38bde98362663454c2e0b6b33c606

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Sun Oct 28 19:32:19 2018 -0400

Make sure ints start with nonzero

commit d3dc83dd407377869e51fdb89725b075a0df73c0

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Sun Oct 28 19:18:21 2018 -0400

Remove generated files

commit 67eeb5624575dd8846798092d7814ebb56951e2d

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Sun Oct 28 19:17:14 2018 -0400

long -> int for everything

commit 33d211de14c13dac5ff150427a5fd03a36663cd9

Author: Jackson Chen <jackson.chen@columbia.edu>

Date: Sun Oct 28 19:14:57 2018 -0400

long -> int

commit d5d88ed5eaf256e91738770f2a97108f32f10af6

Author: KanishkVashisht <kanishk.vashisht@gmail.com>

Date: Sun Oct 14 18:31:09 2018 -0400

Scanner and Parser

commit 1d914d29cba8cd1187ae521657f0a7d3b5bf893a
Author: KanishkVashisht <kanishk.vashisht@gmail.com>
Date: Sun Oct 14 15:42:44 2018 -0400

Fixed errors

commit d6c16d225ddff7c26b8c6b10d48fa46d5953d2c0
Author: KanishkVashisht <kanishk.vashisht@gmail.com>
Date: Sun Oct 14 15:42:08 2018 -0400

Fixed errors

commit 5b363a178dcb9f38205010394b82322e14708468
Author: KanishkVashisht <kanishk.vashisht@gmail.com>
Date: Sun Oct 14 15:40:07 2018 -0400

Fixed errors

commit d91ad22716d4736e5bace8e49942abde186c8bca
Author: KanishkVashisht <kanishk.vashisht@gmail.com>
Date: Sun Oct 14 15:16:25 2018 -0400

Scanner

commit 941408769cd4b12fe147882b6a0c9fc1fbe8d443
Author: Kanishk Vashisht <kv2295@columbia.edu>
Date: Sun Oct 14 14:07:53 2018 -0400

Initial commit