

Extend Language Final Report

Ishaan Kolluri
isk2108
Project Manager

Jared Samet
jss2272
Language Guru

Nigel Schuster
ns3158
System Architect

Kevin Ye
ky2294
Tester

December 17, 2016

Contents

| | | |
|----------|---|----------|
| 1 | Introduction | 4 |
| 1.1 | Inspiration & Use Cases | 4 |
| | Inspiration | 4 |
| | Complex Calculations Across Many Inputs | 4 |
| | Flexibility | 4 |
| 2 | Language Usage Tutorial | 6 |
| 2.1 | Setup | 6 |
| 2.2 | Compiling and Running Extend Code | 6 |
| 2.3 | Writing Extend Code - The Basics | 6 |
| | Functions | 7 |
| | Function Parameters | 7 |
| | Primitive Data Types | 7 |
| | The type() function | 7 |
| | Operators | 7 |
| | Arithmetic Operators | 7 |
| | Bitwise Operators | 8 |
| | Boolean Operators | 8 |
| | String Concatenation | 8 |
| | Conditionals | 8 |
| | The Ternary Conditional | 8 |
| | The Switch Expression | 8 |
| | Ranges | 8 |
| | Range Slicing & Selection | 9 |
| | The Hash Operator | 9 |
| | Application on Ranges | 9 |
| | Range Attribute Functions | 9 |
| | Precedence Expressions | 9 |
| | Import Statements | 9 |
| 2.4 | Standard Library Functions | 10 |
| | Basic Functions | 10 |
| | The toString() Function | 10 |
| | Print Function | 10 |
| | Math Functions | 10 |
| | File I/O | 10 |
| | Plotting | 11 |

| | | |
|----------|--|-----------|
| 3 | Language Reference Manual | 12 |
| 3.1 | Introduction to Extend | 12 |
| 3.2 | Structure of an Extend Program | 12 |
| | Import Statements | 12 |
| | Function Definitions | 12 |
| | Global Variables | 13 |
| | External Library Declarations | 13 |
| | main function | 13 |
| | Scoping and Namespace | 13 |
| 3.3 | Types and Literals | 14 |
| | Primitive Data Types | 14 |
| | Ranges | 14 |
| | Range Literals | 14 |
| 3.4 | Expressions | 15 |
| | Arithmetic Operators | 16 |
| | Boolean Operators | 16 |
| | Conditional Expressions | 18 |
| | Ternary Expressions | 18 |
| | Switch Expressions | 18 |
| | Additional Operators | 19 |
| | Function Calls | 19 |
| | Range Expressions | 19 |
| | Slices | 19 |
| | Selections | 20 |
| | Corresponding Cell | 20 |
| | Selection Examples | 20 |
| | Precedence Expressions | 21 |
| 3.5 | Functions | 21 |
| | Format | 21 |
| | Variable Declarations | 21 |
| | Formula Assignment | 21 |
| | Combined Variable Declaration and Formula Assignment | 22 |
| | Formula Assignment Errors | 22 |
| | Parameter Declarations | 22 |
| | Application on Ranges | 23 |
| | Lazy Evaluation and Circular References | 23 |
| | External Libraries | 24 |
| 3.6 | Introduction to Extend | 24 |
| 3.7 | Structure of an Extend Program | 24 |
| | Import Statements | 24 |
| | Function Definitions | 24 |
| | Global Variables | 25 |
| | External Library Declarations | 25 |
| | main function | 25 |
| | Scoping and Namespace | 26 |
| 3.8 | Standard Library Reference | 26 |
| 3.9 | Example Program | 26 |

| | | |
|----------|---|------------|
| 4 | Project Plan | 27 |
| 4.1 | Meetings | 27 |
| 4.2 | Development Workflow | 27 |
| | Github & Travis CI | 27 |
| 4.3 | Team Member Responsibilities | 28 |
| 5 | Extend's Internal Architecture | 29 |
| 5.1 | The Extend Compiler | 30 |
| | The Scanner | 30 |
| | The Parser and Abstract Syntax Tree | 30 |
| | The Transformer | 30 |
| | The Semantic Analyzer | 30 |
| | The Code Generator | 30 |
| | The Linker | 31 |
| 6 | Testing | 32 |
| 6.1 | Feature Integration & Testing | 32 |
| 6.2 | Regression Test Suite | 32 |
| | Integration with Travis CI | 33 |
| 7 | Extend Code Listing | 34 |
| 7.1 | scanner.mll | 34 |
| 7.2 | parser.mly | 36 |
| 7.3 | ast.ml | 41 |
| 7.4 | transform.ml | 46 |
| 7.5 | codegen.ml | 57 |
| 7.6 | linker.ml | 80 |
| 7.7 | main.ml | 80 |
| 7.8 | lib.c | 81 |
| 7.9 | runtime.c | 85 |
| 7.10 | stdlib.xtnd | 94 |
| 8 | Tests and Output | 95 |
| 9 | Git Logs | 127 |

1. Introduction

Extend is a declarative programming language meant to support spreadsheet-like functionality. It contains features such as complex calculations on large inputs of data, side-effect free and immutable values, and automatic formula adjustments relative to rows and columns. Extend is compiled to the LLVM (Low Level Virtual Machine) intermediate representation, which in turn is reduced to machine assembly. Extend takes inspiration from software such as Microsoft Excel, which allows users to link several formulae on dependent groups of data together, but takes this technology a step further by allowing users to encapsulate such calculations as functions.

1.1 Inspiration & Use Cases

Inspiration

The design goal of our language was to be "a spreadsheet you can compile". Extend was conceptualized to address the limitations that prevented the spreadsheet environment from evolving into a compiled, flexible programming language. To create this, there were three main things that needed to be changed about the way interactive spreadsheets work:

- The language needs reusable functions as opposed to having to copy & paste a block of cells.
- Cell ranges need to be created with dynamic runtime-determined decisions.
- In order to support more complex data structures, we must allow complex content within cells as opposed to single numbers or strings.

With these changes in mind, we attempted to keep the semantics as similar as possible to traditional spreadsheet programs; this meant implementing a dynamically typed language that is fairly forgiving to the user.

Microsoft Excel and Google Sheets find issues at scale when users need to process more and more complicated calculations on several different sets of data, sometimes at scheduled intervals. Extend was conceptualized as a standalone application that removes the manual element of entering new inputs. It brings the best of spreadsheets and computation into one product.

Complex Calculations Across Many Inputs

Extend is spiritually closer in behavior to Microsoft Excel than other conventional programming languages. Extend can nest basic operations in cells of a range, Extend's proprietary data type, and apply mathematical operations on both individual cells of a range and the entire range itself. A dependency graph is evaluated at runtime to optimize and execute these calculations.

Flexibility

Extend, as a declarative language, allows you to keep dimensional values potentially variable until runtime, and handles some errors with the "empty" keyword, which persists throughout calculation instead of

crashing the program. It supports a range of standard library functions that are additionally supported in conventional spreadsheet technology as well.

2. Language Usage Tutorial

This will cover the configuration of the user's environment and the usage of Extend's features.

2.1 Setup

The Extend compiler requires that the OCaml Language and LLVM be installed on the host machine. Development was done in a virtual machine running the 64-bit Ubuntu 16.04 operating system.

```
1 sudo apt-get install -y ocaml m4 llvm opam oasis ounit
2 opam init
3 opam install llvm.3.8 ocamlfind
4 eval `opam config env`
```

After setting up the environment, clone the Extend git repository:

```
1 git clone https://github.com/ExtendLang/Extend.git
```

2.2 Compiling and Running Extend Code

To build the Extend compiler, the first steps are the following.

```
1 cd Extend/
2 make
```

If this does not successfully build, run `eval `opam config env``, which should configure the environment to use OPAM packages. Alternatively, add this command to your bash profile.

After running `make`, you should see a `main.byte` file. To compile and run an Extend program, perform the following:

```
1 ./compile.sh example_source_file.xtnd
```

This should produce an `out` file. Running `./out` should successfully run the program.

2.3 Writing Extend Code - The Basics

Extend code can be written in a file that contains the conventional `.xtnd` extension. It consists of optional import statements and global variables, and one or more series of functions. Runnable Extend programs must contain a `main` function, and other functions can be written into the program as well.

Below is a short tour of the features of Extend. More detail can be found in the next chapter - the Language Reference Manual. Please note Extend uses the `range` data type to represent multiple cells of data, composed of primitive data types or another `range`.

Functions

Functions are commonplace in Extend. They are declared with the syntax `function_name([optional_dimensions] function_arguments){...}`. Below is the syntax of the `main` function, which is needed to run Extend code, within a simple Extend program.

```
1  main(args){
2      return 0;
3  }
```

The return type of a function is a **range** of flexible size. **Ranges** will be discussed in a later section of this chapter. A function is composed of a series of statements, concluding with the return statement. Note that the **return** statement is always the last statement in the function.

Function Parameters

Function parameters consist of zero or more ranges, signed with an optional dimension. If the arguments have been written with dimensions, those dimensions will be verified at runtime.

```
1      foo([m,n] arg){
2          return m * n; \\ m and n initialized through arg1
3      }
4
5      bar([1,1] arg){
6          return 0; \\ 1 by 1 ranges should be primitive data types. If arg is not, a
              runtime error will be thrown
7      }
```

Primitive Data Types

Extend has three primitive data types: Numbers, Strings, and `empty`. An example of each is shown below.

```
1  bar := 5; \\ Assigns bar to a number
2  foo := "Hello World"; \\ Assigns foo to a string
3  bar = empty; \\ Reassigns bar to empty
```

Note that we declare a variable and assign it a value together with `:=`. If the variable already exists, `=` will suffice. More information about Extend's primitive data types can be found in the Language Reference Manual.

The `type()` function

Extend offers a `type(expr)` function, which takes an expression and returns Number, String, Range, or Empty.

Operators

Extend includes a comprehensive set of operators. Each category is listed in order of precedence. A more detailed explanation of each operator can be found in the Language Reference Manual.

Arithmetic Operators

- Unary Operations: -
- Binary Operations: **, *, /, %, +, -

Bitwise Operators

- Unary Operations: `~`
- Binary Operations: `<<`, `>>`, `&`, `|`, `^`

Boolean Operators

- Unary Operations: `!`
- Binary Operations: `==`, `!=`, `<`, `>`, `<=`, `>=`, `&&`, `||`

String Concatenation

Note that the `+` symbol can be used to perform concatenation between two strings.

```
1 "Hello " + "World\n"
```

Conditionals

There are two types of conditionals: the ternary conditional and a `switch` expression.

The Ternary Conditional

The two ways to write the ternary condition are as follows:

```
condition ? expr_if_true : expr_if_false
```

or:

```
if(condition, expr_if_true, expr_if_false)
```

The Switch Expression

Below is an example of the switch case used in a function:

```
1 switch_func(foo){
2     return switch(foo){
3         case 1: "Hello, one!";
4         case 2: "Hello, two!";
5         default: "Hello?";
6     };
7 }
```

Ranges

Ranges are an incredibly important aspect of Extend. A range is a composite data type composed of rows and columns of cells. Each cell contains a formula that can evaluate to another range or one of the aforementioned three data types - number, string, or `empty`. Note that **1 by 1** ranges are automatically dereferenced to their contents in Extend.

```
1 [1,2] foo; \\ Declares range of dimensions 1 by 2
2 [1,3] bar := 4; \\ Declares range of dimensions 1 by 3, and assigns each cell to 4
```

Ranges can also be declared as range literals. Rows are delineated by `;` and columns by `,`.

```
1 foo := {"Hello", 1, 2; 3, 4, 5; 6, 7, 8} \\ Three rows, three columns.
```

Range Slicing & Selection

Python-style array-slicing syntax can be applied to ranges, which will return a subrange based on either absolute or relative indexing. All indices are zero-based.

```
1     foo[0,2] \\ The cell in the first row, third column
2     foo[0,:] \\ The range of cells in the first row
3     foo[0,[1]] \\ The range in the column that is 1 column right of the left-hand-
        side cell.
4     foo[,] \\ Cell in first row, first column if 1 by 1. If not, then relative first
        row and relative first column
```

More examples and detail can be found in the next chapter.

The Hash Operator

A common case for ranges in Excel is to perform calculations on specific cells. For example, `foo[,]` is commonly used to retrieve the cell that is being calculated on. Since this is a popular use case, the `#` operator will perform the same functionality.

Application on Ranges

Extend, in the vein of spreadsheets, allows the programmer to apply functions cell-wise on a range. Using the `#` operator, we can perform cell-wise multiplication across two ranges.

```
1     foo([m,n] arg1, [m,n] arg2){
2         [m,n] bar := #arg1 * #arg2; \\ Multiplies the cell in arg1 with the
        corresponding cell in arg2.
3         return bar;
4     }
```

This is an incredibly powerful aspect of Extend. Make sure to study it well!

Range Attribute Functions

Extend has the `row()` and `column()` functions, which respectively return the row and column of the cell that is being calculated at that point in time. There is also a `size(expr)` function, which returns a 1 by 2 range; the first cell contains the number of rows, and the second cell contains the number of columns.

Precedence Expressions

Precedence Expressions are used to force the evaluation of one expression before another. This is used when the order of operation is required for functions with side effects, such as the ones housed in Extend's standard library. The `->` operator is used to denote precedence.

```
1     return print_func(4) -> 0; \\ print_func prints 4 to stdout before returning 0
```

Import Statements

In Extend, you can import other Extend files at the top of your program via relative directory path. The use case is below:

```
1     import "../programs/helloworld.xtnd"
```

2.4 Standard Library Functions

Extend offers an assortment of standard library functions. In order to use them, an **extern** statement must be added at the top of the file, which tells the program where to look for the standard library function you want.

```
1  extern "stdlib.o" {
2      toString(a);
3      extend_open(a,b);
4      extend_sin(a);
5      print(a,b);
6  }
```

The letters **a** and **b** represent how many parameters Extend should expect that function to have.

While their usage will be covered in more length in the Language Reference Manual, here are some of the more useful standard library functions to remember.

Basic Functions

The toString() Function

The **toString()** function takes a 1 by 1 range and renders its value as a string. This will return one of the primitive data types.

```
1      return "Hello " + toString(14); \\ "Hello 14"
```

Print Function

Extend allows the reader to easily print to stdout. It is called with the **print** signature, and takes two parameters - the location to print to, and the string to print.

```
1      print(1, "Hello World\n") \\ Prints Hello World to stdout
```

Math Functions

Borrowing from C's standard library math functions, Extend offers: **sin**, **cos**, **tan**, **acos**, **asin**, **atan**, **sinh**, **cosh**, **tanh**, **exp**, **log**, **log10**, **sqrt**, **ceil**, **fabs** and **floor**. To use them, each function is called with the **extend_** prefix, and must be declared similarly in the **extern** statement.

```
1      extern "stdlib.o" {
2          extend_sqrt(b);
3          print(a,b);
4          toString(a);
5      }
6      main(args){
7          bar := extend_sqrt(16);
8          return print(1, toString(bar)) -> 0; \\ Prints 4 to stdout
9      }
```

File I/O

Extend has **extend_open**, **extend_close**, **extend_read**, and **extend_write** functions to interact with files. Usage is as follows:

```
1  extern "stdlib.o" {
2      extend_open(a,b);
3      extend_read(a,b);
4      extend_close(a);
5      print(a,b);
6  }
7
8  [1,1] main(args){
9      return print(1, extend_read(extend_open("test_file.txt", "r"),5)) -> 0;
10 }
```

Plotting

Extend also offers the ability to export plots to a PNG file with the `extend_plot` function.

```
1  TODO: Demonstrate usage here.
```

3. Language Reference Manual

3.1 Introduction to Extend

Extend is a domain-specific programming language used to designate ranges of cells as reusable functions. It is a dynamically-typed, statically-scoped, declarative language that uses lazy evaluation to carry out computations. Once computed, all values are immutable. In order to offer the best performance, Extend compiles down to LLVM.

Extend's syntax is meant to provide clear punctuation and easily understandable cell range access specifications, while borrowing elements from languages with C-style syntax for ease of development. Despite these syntactic similarities, the semantics of an Extend program have more in common with a spreadsheet such as Microsoft Excel than imperative languages such as C, Java or Python.

3.2 Structure of an Extend Program

An Extend program consists of one or more source files. A source file can contain any number of import directives, function definitions, global variable declarations, and external library declarations, in any order.

Import Statements

Import statements in Extend are written with `import`, followed by the name of a file in double quotes, and terminated with a semicolon. The syntax is as follows:

```
1 import "string.xtnd";
```

Extend imports act like `#include` in C, except that multiple imports of the same file are ignored. The imports are all aggregated into a single namespace.

Function Definitions

Function definitions comprise the bulk of an Extend program. In short, a function consists of a set of variable declarations, formula assignments, and a return expression. Each variable consists of cells; the values of each cell are, if necessary, calculated according to formulas which each apply to a specified subset of the cells. Each cell value, once calculated, is immutable. A couple examples follow for context; functions are described in detail in section 3.5.

```
1 isNumber(x) {
2     return type(x) == "Number";
3 }
4
5 sum_column([m,1] rng) {
6     /* Returns the sum of the values in the column, skipping any values that are non-
7        numeric */
7     [m,1] running_sum;
8     running_sum[0,0] = #rng;
```

```

9   running_sum[1:,0] = running_sum[[-1],] + (isNumber(#rng) ? #rng : 0);
10  return running_sum[-1];
11 }

```

Global Variables

In essence, global variable declarations function as constants in Extend. They are written with the keyword **global**, followed by a variable declaration in the combined variable declaration and assignment format described in section 3.5. As with local variables, the cell values of a global variable, once computed, are immutable. A few examples follow:

```

1  global pi := 3.14159265359;
2  global num_points := 24;
3  global [num_points,1]
4     circle_x_vals := cos(2 * pi * row() / num_points),
5     circle_y_vals := sin(2 * pi * row() / num_points);

```

External Library Declarations

An external library is declared with the **extern** keyword, followed by the name of an object file in double quotes, followed by a semicolon-delimited list of external function declarations enclosed by curly braces. A library declaration informs the compiler of the functions' names and signatures and instructs the compiler to link the object file when producing an executable. An external function declared as **foo** will call an appropriately written C function **extend_foo**. An example follows:

```

1  extern "mylib.o" {
2     foo(arg1, arg2);
3     bar();
4 }

```

This declaration would cause the compiler to link **mylib.o** and would make the C functions **extend_foo** and **extend_bar** available to Extend programs as **foo** and **bar** respectively. The required signature and format of the external functions is specified precisely in section 3.5.

main function

When a compiled Extend program is executed, the **main** function is evaluated. All computations necessary to calculate the return value of the function are performed, after which the program terminates. The **main** function must be a function of a single argument, conventionally denoted **args**, which is guaranteed to be a 1-by-n range containing the command line arguments.

Scoping and Namespace

For functions and for global variables, there is a single namespace that is shared between all files composing an Extend program, and they are visible throughout the entire program. Functions declared in external libraries share this namespace as well. For a local variable, the scope is the entire body of the function in which it is defined. Functions may declare local variables sharing a name with a global variable; inside that function, the name will refer to the local variable.

```

1  global x := "I'm a global";
2
3  foo() {
4     y := x; // Scope of x is entire function
5     x := "In here I'm a local";
6     return y; // Returns "In here I'm a local"

```

```

7 }
8
9 bar(x) {
10     return x; // Parameters mask globals; returns argument
11 }
12
13 baz() {
14     return x; // Returns "I'm a global"
15 }

```

3.3 Types and Literals

Extend has three primitive data types, **Number**, **String**, and **Empty**, and one composite type, **Range**.

Primitive Data Types

A **Number** is an immutable primitive value corresponding to a double-precision 64-bit binary format IEEE 754 value. Numbers can be written in an Extend source file as either integer or floating point constants; both are represented internally as floating-point values. There is no separate type representing an integer.

A **String** is an immutable primitive value that is internally represented a C-style null-terminated byte array corresponding to ASCII values. A String can be written in an Extend source file as a sequence of characters enclosed in double quotes, with the usual escaping conventions. Extend does not allow for slicing of strings to access specific characters; access to the contents of a string will only be available through standard library functions.

The **Empty** type can be written as the keyword **empty**, and serves a similar function to NULL in SQL; it represents the absence of a value.

| Primitive Data Types | Examples |
|----------------------|----------------------------------|
| Number | 42 or -5 or 2.71828 or 314159e-5 |
| String | "Hello, World!\n" or "foo" or "" |
| Empty | empty |

Ranges

Extend has one composite type, **Range**. A range borrows conceptually from spreadsheets; it is a group of cells with two dimensions, described as rows and columns. Each cell is assigned a formula that either evaluates to a Number, a String, **empty**, or another Range. Cell formulas are described in detail in section 3.5. A range can either be declared as described in section 3.5 or with a range literal expression. Ranges can be nested arbitrarily deeply and can be used to represent (immutable) lists, matrices, or more complicated data structures.

Range Literals

A range literal is a semicolon-delimited list of rows, enclosed in curly brackets. Each row is a comma-delimited list of numbers, strings, or range literals. A few examples follow:

```

1 legal_ranges() {
2     r1 := {"Don't"; "Panic"}; // two rows, one column
3     r2 := {"Don't", "Think", "Twice"}; // one row, three columns
4     r3 := {1,2,3;4,5,6;7,8,9}; // three rows, three columns
5     r4 := {"Hello";0,1,2,3,4}; // two rows, five columns

```

```
6   r5 := {{{{{1}}}}}; // one row, one column
7   r7 := {-1.5,-2.5,{-2,"nested"},-3.5}; // one row, four columns
8   return 0;
9 }
```

3.4 Expressions

Expressions in Extend allow for arithmetic and boolean operations, function calls, conditional branching, extraction of contents of other variables, string concatenation, and determination of the location of the cell containing the expression. The sections for boolean and conditional operators refer to **truthy** and **falsey** values: the **Number** 0 is the only falsey value; all other values are **truthy**. As **empty** represents the absence of a value, it is neither **truthy** nor **falsey**.

Arithmetic Operators

The arithmetic operators listed below take one or two expressions and return a number, if both expressions are Numbers, or **empty** otherwise. Operators grouped within the same inner box have the same level of precedence, and are listed from highest precedence to lowest precedence. All of the binary operators are infix operators, and, with the exception of exponentiation, are left-associative. Exponentiation, bitwise negation, and unary negation are right-associative. All of the unary operators are prefix operators. The bitwise operators round their operands to the nearest signed 32-bit integer (rounding half to even) before performing the operation and evaluate to a Number.

| Operator | Description | Definition |
|-----------------------|----------------|---|
| <code>~</code> | Bitwise NOT | Performs a bitwise negation on the binary representation of an expression. |
| <code>-</code> | Unary negation | A simple negative sign to negate expressions. |
| <code>**</code> | Power | Returns the first expression raised to the power of the second expression |
| <code>*</code> | Multiplication | Multiplies two expressions |
| <code>/</code> | Division | Divides first expression by second. |
| <code>%</code> | Modulo | Finds the remainder by dividing the expression on the left side of the modulo by the right side expression. |
| <code><<</code> | Left Shift | Performs a bitwise left shift on the binary representation of an expression. |
| <code>>></code> | Right Shift | Performs a sign-propagating bitwise right shift on the binary representation of an expression. |
| <code>&</code> | Bitwise AND | Performs a bitwise AND between two expressions. |
| <code>+</code> | Addition | Adds two expressions together. |
| <code>-</code> | Subtraction | Subtracts second expression from first. |
| <code> </code> | Bitwise OR | Performs a bitwise OR between two expressions. |
| <code>^</code> | Bitwise XOR | Performs a bitwise exclusive OR between two expressions. |

```
1 easy() {
2   return 3 - -3 ** 2 %5; //-1
3 }
4 g_eazy() {
5   return (((1 << 2 | 1) << 2) | 1) << 1; //42
6 }
```

Boolean Operators

These operators take one or two expressions and evaluate to **empty**, 0 or 1. Operators grouped within the same inner box have the same level of precedence and are listed from highest precedence to lowest precedence. All of these operators besides logical negation are infix, left-associative operators. The logical AND and OR operators feature short-circuit evaluation. Logical NOT is a prefix, right-associative operator. Besides logical NOT, all boolean operators have lower precedence than all arithmetic operators. For Strings, the boolean operators `<`, `<=`, `>`, and `>=` implement case-sensitive lexicographic comparison.

| Operator | Description | Definition |
|----------|---------------------------|---|
| ! | Logical NOT | Evaluates to 0 or 1 given a truthy or falsey value respectively. <code>!empty</code> evaluates to <code>empty</code> . It has equal precedence with <code>and</code> and unary minus. |
| == | Equals | Always evaluates to 0 if the two expressions have different types. If both expressions are primitive values, evaluates to 1 if they have the same type and the same value, or 0 otherwise. If both expressions are ranges, evaluates to 1 if the two ranges have the same dimensions and each cell of the first expression == the corresponding cell of the second expression. <code>empty == empty</code> evaluates to 1. Strings are compared by value. |
| != | Not equals | <code>x != y</code> is equivalent to <code>!(x == y)</code> . |
| < | Less than | If the expressions are both Numbers or both Strings and the first expression is less than the second, evaluates to 1. If the expressions are both Numbers or both Strings and the first expression is greater than or equal to the second, evaluates to 0. Otherwise, evaluates to <code>empty</code> . |
| > | Greater than | Equivalent rules about typing as for <. |
| <= | Less than or equal to | Equivalent rules about typing as for <. |
| >= | Greater than or equal to | Equivalent rules about typing as for <. |
| && | Short-circuit Logical AND | If the first expression is falsey or <code>empty</code> , evaluates to 0 or <code>empty</code> respectively. Otherwise, if the second expression is truthy, falsey, or <code>empty</code> , evaluates to 1, 0, or <code>empty</code> respectively. |
| | Short-circuit Logical OR | If the first expression is truthy or <code>empty</code> , evaluates to 1 or <code>empty</code> respectively. Otherwise, if the second expression is truthy, falsey, or <code>empty</code> , evaluates to 1, 0, or <code>empty</code> respectively. |

```

1 somethings_false() {
2   return !1 != !1 || 4 <= 3;
3 }
4 somethings_empty() {
5   return empty || empty <= !3 || 5 > 3;
6 }
7 somethings_true() {
8   return 6 > 2 && !(1 == !1);
9 }

```

Conditional Expressions

There are two types of conditional expressions: a simple ternary if-then-else expression and a **switch** expression which can represent more complex logic.

Ternary Expressions

A ternary expression, written either as `cond-expr ? expr-if-true : expr-if-false` or, equivalently, `if(cond-expr, expr-if-true, expr-if-false)` evaluates to `expr-if-true` if `cond-expr` is truthy, or `expr-if-false` if `cond-expr` is falsey. If `cond-expr` is empty, the expression evaluates to `empty`. Both `expr-if-true` and `expr-if-false` are mandatory. `expr-if-true` is only evaluated if `cond-expr` is truthy, and `expr-if-false` is only evaluated if `cond-expr` is falsey. If `cond-expr` is empty, neither expression is evaluated. The ternary operator `? :` has the lowest precedence level of all operators.

Switch Expressions

A **switch** expression takes a optional condition, and a list of cases and expressions that the overall expression should evaluate to if the case applies. In the event that multiple cases are true, the expression of the first matching case encountered will be evaluated. An example is provided below:

```
1 switch_example(foo) {
2   return switch (foo) {
3     case 2: "foo is 2";
4     case 3,4: "foo is 3 or 4";
5     default: "none of the above";
6   };
7 }
8
9 alternate_format(foo) {
10  return switch {
11    case foo == 2:
12      "foo is 2";
13    case foo == 3, foo == 4:
14      "foo is 3 or 4";
15    default:
16      "none of the above";
17  };
18 }
```

The format for a **switch** statement is the keyword **switch**, optionally followed by pair of parentheses containing an expression **switch-expr**, followed by a list of case clauses enclosed in curly braces and delimited by semicolons. A case clause consists of the keyword **case** followed by a comma-separated list of expressions **case-expr1** [, **case-expr2**, [...]], a colon, and an expression **match-expr**, or the keyword **default**, a colon, and an expression **default-expr**. If **switch-expr** is omitted, the **switch** expression evaluates to the **match-expr** for the first case where one of the **case-exprs** is truthy, or **default-expr** if none of the **case-exprs** apply. If **switch-expr** is present, the **switch** expression evaluates to the **match-expr** for the first case where one of the **case-exprs** is equal (with equality defined as for the `==` operator) to **switch-expr**, or **default-expr** if none of the **case-exprs** apply.

The **switch** expression can be used to compactly represent what in most imperative languages would require a long string such as `if (cond1) {...} else if (cond2) {...}`. The **switch** operator is internally converted to an equivalent (possibly nested) ternary expression; as a result, it features short-circuit evaluation throughout.

Additional Operators

There are four additional operators available to determine the size and type of other expressions. In addition, the infix `+` operator is overloaded to perform string concatenation.

| Operator | Description | Definition |
|-------------------------|----------------------|---|
| <code>size(expr)</code> | Dimensions | Evaluates to a Range consisting of one row and two columns; the first cell contains the number of rows of <code>expr</code> and the second contains the number of columns. If <code>expr</code> is a Number, a String, or Empty, both cells will contain 1. |
| <code>type(expr)</code> | Value Type | Evaluates to "Number", "String", "Range", or "Empty". |
| <code>row()</code> | Row Location | No arguments; returns the row of the cell that is being calculated |
| <code>column()</code> | Column Location | No arguments; returns the column of the cell that is being calculated |
| <code>+</code> | String concatenation | "Hello, " + "World!\n" == "Hello, World!\n" |

Given `[5,5]foo`, then `foo[1,4] = row() * 2 + col()` will evaluate to 6.

Function Calls

A function expression consists of an identifier and an optional list of expressions enclosed in parentheses and separated by commas. The value of the expression is the result of applying the function to the arguments passed in as expressions. The arguments are evaluated from left to right before the function is called. For more detail, see section 3.5.

Range Expressions

Range expressions are used to select part or all of a range. A range expression consists of a bare identifier, a bare range literal, or an expression and a selector. If a range expression has exactly 1 row and 1 column, the value of the expression is the value of the single cell of the range. If it has more than 1 row or more than 1 column, the value of the expression is the selected range. If the range has zero or fewer rows or zero or fewer columns, the value of the expression is `empty`. If a range expression with a selector would access a row index or column index greater than the number of rows or columns of the range, or a negative row or column index, the value of the expression is `empty`.

Slices

A slice consists of an optional integer literal or expression `start`, a colon, and an optional integer literal or expression `end`, or a single integer literal or expression `index`. If `start` is omitted, it defaults to 0. If `end` is omitted, it defaults to the length of the dimension. A single `index` with no colon is equivalent to `index:index+1`. Enclosing `start` or `end` in square brackets is equivalent to the expression `row() + start` or `row() + end`, for a row slice, or `column() + start` or `column() + end` for a column slice. The slice includes `start` and excludes `end`, so the length of a slice is `end - start`. A negative value is interpreted as the length of the dimension minus the value. As mentioned above, the value of a range that is not 1 by 1 is a range, but the value of a 1 by 1 range is essentially dereferenced to the result of the cell formula.

Selections

A selection expression consists of an expression and a pair of slices separated by a comma and enclosed in square brackets, i.e. `[row_slice, column_slice]`. If one of the dimensions of the range has length 1, the comma and the slice for that dimension can be omitted. If the comma is present but a slice is omitted, that slice defaults to `[0]` for a slice corresponding to a dimension of length greater than one, or `0` for a slice corresponding to a dimension of length one.

Corresponding Cell

A very common selection to make is the cell in the "corresponding location" of a different variable. Since this case is so common, `#var` is syntactic sugar for `var[,]`. As a result, if `var` has more than column and more than one row, `#var` is equivalent to `var[row()],column()]`. If `var` has multiple rows and one column, it is equivalent to `var[row(),0]`. If `var` has one row and multiple columns, it is equivalent to `var[0,column()]`; and if `var` has one row and one column, it is equal to `var[0,0]`.

Selection Examples

```
1 selection_examples() {
2   foo[0,2] /* This evaluates to the cell value in the first row and third column. */
3   foo[0,:] /* Evaluates to the range of cells in the first row of foo. */
4   foo[:,2] /* Evaluates to the range of cells in the third column of foo. */
5   foo[:,[1]] /* The internal brackets denote RELATIVE notation.
6   In this case, 1 column right of the column of the left-hand-side cell. */
7
8   foo[3,] /* Equivalent to foo[3,[0]] if foo has more than one column
9   or foo[3,0] if foo has one column */
10
11  foo[5:, 7:] /* All cells starting from the 6th row and 8th column to the bottom
12             right */
13
14  foo[[1]:[2], 0:[7]]
15  /* Selects the rows between the 1st and 2nd row after LHS row, and
16     all the columns up to the 7th column to the right of the LHS column */
17
18  /* In this example, each cell of bar would be equal to the cell
19     * in foo in the equivalent location plus 1. */
20  [5,5] foo;
21  [5,5] bar := #foo + 1; // #foo = foo[[0],[0]]
22
23  /* In this example, bar would be a 3x5 range where in each row,
24     * the value in bar is equal to the value in foo in the same column.
25     * In other words, each row of bar would be a copy of foo. */
26  [1,5] foo; // foo has 1 row, 5 columns
27  [3,5] bar := #foo; // #foo = foo[0,[0]]
28
29  /* In this example, the values of baz would be
30     * 11, 12, 13 in the first row;
31     * 21, 22, 23 in the second row;
32     * 31, 32, 33 in the third row. */
33  foo := {1,2,3}; // 1 row, 3 columns
34  bar := {10;20;30}; // 3 rows, 1 column
35  [3,3] baz := #foo + #bar; // Equivalent to foo[0,[0]] + bar[[0],0]
36 }
```

Precedence Expressions

A precedence expression is used to force the evaluation of one expression before another, when that order of operation is required for functions with side-effects. It consists of an expression `prec-expr`, the precedence operator `->`, and an expression `succ-expr`. The value of the expression is `succ-expr`, but the value of `prec-expr` will be calculated first and the result ignored. All functions written purely in Extend are free of side effects. However, some of the external functions provided by the standard library, such as for file I/O and plotting, do have side effects. The precedence operator has the second-lowest grammatical precedence of all operators, higher only than the ternary operator.

3.5 Functions

The bulk of an Extend program consists of functions. Although Extend has some features, such as immutability and lazy evaluation, that are inspired by functional languages, its functions are not *first class objects*. By default, the standard library is automatically compiled and linked with a program, but there are no functions built into the language itself.

Format

As in most programming languages, the header of the function declares the parameters it accepts. The body of the function consists of an optional set of variable declarations and formula assignments, which can occur in any order, and a return statement, which must be the last statement in the function body. All variable declarations and formula assignments, in addition to the return statement, must be terminated by a semicolon. This very simple function returns whatever value is passed into it:

```
1  foo(arg) {  
2      return arg;  
3  }
```

Variable Declarations

A variable declaration associates an identifier with a range of cells of the specified dimensions, which are listed in square brackets before the identifier. For convenience, if the square brackets and dimensions are omitted, the identifier will be associated with a single cell. In addition, multiple identifiers, separated by commas, can be listed after the dimensions; all of these identifiers will be separate ranges, but with equal dimension sizes. The dimensions can be specified as any valid expression that evaluates to a Number, which will be rounded to the nearest signed 32-bit integer. If either dimension is zero or negative, or if the expression does not evaluate to a Number, a runtime error causing the program to halt will occur.

```
1  [2, 5] foo; // Declares foo as a range with 2 rows and 5 columns  
2  [m, n] bar; // Declares bar as a range with m rows and n columns  
3  [3, 3] ham, eggs, spam; // Declares ham, eggs and spam as distinct 3x3 ranges  
4  baz; // Declares baz as a single cell
```

Formula Assignment

A formula assignment assigns an expression to a subset of the cells of a variable. Unlike most imperative languages, this expression is not immediately evaluated, but is instead only evaluated if and when it is needed to calculate the return value of the function. A formula assignment consists of an identifier, an optional pair of slices enclosed in square brackets specifying the subset of the cells that the assignment applies to, an `=`, and an expression, followed by a semicolon. As with the expressions specifying the dimensions of a range, these slices specifying the cell subset can contain arbitrary expressions, as long as the expression taken as a

whole evaluates to a Number, which will be rounded to the nearest signed 32-bit integer. Negative numbers are legal in these slices, and correspond to (dimension length + value).

```
1 [5, 2] foo, bar, baz; // Declares foo, bar, and baz as distinct 5x2 ranges
2 foo[0,0] = 42; // Assigns the expression 42 to the first cell of the first row of foo
3 foo[0,1] = foo[0,0] * 2; // Assigns (foo[0,0] * 2) to the 2nd cell of the 1st row of
   foo
4 bar = 3.14159; // Assigns pi to every cell of every row of bar
5 baz[1:-1,0:1] = 2.71828; // Assigns e to cells (1,0) through (3,1), inclusive, of baz
6
7 /* The next line assigns foo[[-1],0] + 2 to every cell in
8    both columns of foo, besides the first row */
9 foo[1:,:] = foo[[-1],0] + 2;
```

The last line of the source snippet above demonstrates the idiomatic Extend way of simulating an imperative language's loop; `foo[4,0]` would evaluate to $42+2+2+2+2 = 50$ and `foo[4,1]` would evaluate to $(42*2)+2+2+2+2 = 92$.

Combined Variable Declaration and Formula Assignment

For convenience, a variable declaration and a formula assignment to all cells of that variable can be combined on a single line by inserting a `:=` and an expression after the identifier. Multiple variables and assignments, separated by commas, can be declared on a single line as well. All global variables must be defined using the combined declaration and formula assignment syntax.

```
1 /* Creates two 2x2 ranges; every cell of foo evaluates to 1 and every cell of
2    bar evaluates to 2. */
3 [2,2] foo := 1, bar := 2;
```

Formula Assignment Errors

If the developer writes code in such a way that more than one formula applies to a cell, a runtime error will occur if the cell's value is required to compute the return expression. If there is no formula assigned to a cell, the cell will evaluate to `empty`.

Parameter Declarations

Parameters can be declared with or without dimensions. If dimensions are declared, they can either be specified as integer literals or as identifiers. If a dimension is specified as an integer literal, the program will verify the dimension of the argument before beginning to evaluate the return expression; if it does not match, a runtime error will occur causing the program to halt. If it is specified as an identifier, that variable will contain the dimension size and will be available inside the function body. If the same identifier is repeated in the function declaration, the program will verify that every parameter dimension with that identifier has equal dimension size; if they differ, a runtime error will occur causing the program to halt. A few examples follow:

```
1 number_of_cells([m,n] arg) {
2   return m*n; // m and n are initialized with the dimensions of arg
3 }
4
5 die_unless_primitive([1,1] arg) {
6   return 0; // If arg is not a primitive value, a runtime error will occur
7 }
8
9 num_cells_if_column_vector([m,1] arg) {
10  // If arg has one column, return number of cells; otherwise runtime error
```

```

11     return m;
12 }
13
14 die_unless_square([m,m] arg) {
15     return 0; // Runtime error if number of rows != number of columns
16 }
17
18 num_cells_if_same_size([m,n] arg1, [m,n] arg2) {
19     // If arguments are the same size, return # of cells, otherwise runtime error
20     return m*n;
21 }

```

Application on Ranges

Extend gives the developer the power to easily apply operations in a functional style on ranges. For example, the following function performs cell wise addition:

```

1 foo([m,n] arg1, [m,n] arg2) {
2     [m,n] bar := #arg1 + #arg2;
3     return bar;
4 }

```

This function normalizes a column vector to have unit norm:

```

1 normalize_column_vector([m,1] arg) {
2     [m,1] squared_lengths := #arg * #arg, normalized := #arg / vector_norm;
3     vector_norm := sqrt(sum(squared_lengths));
4     return normalized;
5 }

```

Lazy Evaluation and Circular References

All cell values and variable dimensions are evaluated lazily if and when they are needed to calculate the return expression. Using lazy evaluation ensures that the cell values are calculated in a valid topological sort order and allows for detection of circular references; internally this is accomplished by constructing a function for each formula which is called the first time the cell's value is needed, and marking the cell as "in-progress" once it starts being evaluated and as "complete" once the value has been calculated. The only guarantees the language places on the order of cell evaluation are: (1) It will be a valid topological ordering; (2) In conditional expressions and in short-circuiting operator expressions, only the relevant conditional branches will be evaluated; and (3) In an expression using the precedence operator, the preceeding expression will be evaluated before the succeeding expression. A range selection consisting of multiple cells will not cause the constituent cells to be evaluated; however, selection of a single cell will cause that cell's value to be evaluated. If a program is written in such a way as to cause a circular dependency of one cell on another, and the return expression is dependent on that cell's value, a runtime error will occur. For example, in the following function:

```

1 maybeCircular(truth_value) {
2     x := x;
3     return truth_value ? x : 0;
4 }

```

A runtime error will occur if maybeCircular(1) is called; but if maybeCircular(0) is called, the function will simply return 0.

External Libraries

Using the following library declaration:

```
1 extern "mylib.o" {  
2     foo(arg1, arg2);  
3     bar();  
4 }
```

will make the functions `foo` (taking two arguments) and `bar` (taking zero arguments) available within Extend. In LLVM, the compiler will declare external functions `extend_foo` and `extend_bar` as functions of two and zero arguments respectively. All arguments must have the type `value_p`, and the function must have return type `value_p`, declared in the Extend standard library header file. In other words, the C file compiled to generate the library must have defined:

```
1 value_p extend_foo(value_p arg1, value_p arg2) {  
2     /* function body here; */  
3 }  
4  
5 value_p extend_bar() {  
6     /* function body here; */  
7 }
```

3.6 Introduction to Extend

Extend is a domain-specific programming language used to designate ranges of cells as reusable functions. It is a dynamically-typed, statically-scoped, declarative language that uses lazy evaluation to carry out computations. Once computed, all values are immutable. In order to offer the best performance, Extend compiles down to LLVM.

Extend's syntax is meant to provide clear punctuation and easily understandable cell range access specifications, while borrowing elements from languages with C-style syntax for ease of development. Despite these syntactic similarities, the semantics of an Extend program have more in common with a spreadsheet such as Microsoft Excel than imperative languages such as C, Java or Python.

3.7 Structure of an Extend Program

An Extend program consists of one or more source files. A source file can contain any number of import directives, function definitions, global variable declarations, and external library declarations, in any order.

Import Statements

Import statements in Extend are written with `import`, followed by the name of a file in double quotes, and terminated with a semicolon. The syntax is as follows:

```
1 import "string.xtnd";
```

Extend imports act like `#include` in C, except that multiple imports of the same file are ignored. The imports are all aggregated into a single namespace.

Function Definitions

Function definitions comprise the bulk of an Extend program. In short, a function consists of a set of variable declarations, formula assignments, and a return expression. Each variable consists of cells; the values of each cell are, if necessary, calculated according to formulas which each apply to a specified subset

of the cells. Each cell value, once calculated, is immutable. A couple examples follow for context; functions are described in detail in section 3.5.

```
1 isNumber(x) {
2     return type(x) == "Number";
3 }
4
5 sum_column([m,1] rng) {
6     /* Returns the sum of the values in the column, skipping any values that are non-
7        numeric */
8     [m,1] running_sum;
9     running_sum[0,0] = #rng;
10    running_sum[1:,0] = running_sum[[-1],] + (isNumber(#rng) ? #rng : 0);
11    return running_sum[-1];
12 }
```

Global Variables

In essence, global variable declarations function as constants in Extend. They are written with the keyword `global`, followed by a variable declaration in the combined variable declaration and assignment format described in section 3.5. As with local variables, the cell values of a global variable, once computed, are immutable. A few examples follow:

```
1 global pi := 3.14159265359;
2 global num_points := 24;
3 global [num_points,1]
4     circle_x_vals := cos(2 * pi * row() / num_points),
5     circle_y_vals := sin(2 * pi * row() / num_points);
```

External Library Declarations

An external library is declared with the `extern` keyword, followed by the name of an object file in double quotes, followed by a semicolon-delimited list of external function declarations enclosed by curly braces. A library declaration informs the compiler of the functions' names and signatures and instructs the compiler to link the object file when producing an executable. An external function declared as `foo` will call an appropriately written C function `extend_foo`. An example follows:

```
1 extern "mylib.o" {
2     foo(arg1, arg2);
3     bar();
4 }
```

This declaration would cause the compiler to link `mylib.o` and would make the C functions `extend_foo` and `extend_bar` available to Extend programs as `foo` and `bar` respectively. The required signature and format of the external functions is specified precisely in section 3.5.

main function

When a compiled Extend program is executed, the `main` function is evaluated. All computations necessary to calculate the return value of the function are performed, after which the program terminates. The `main` function must be a function of a single argument, conventionally denoted `args`, which is guaranteed to be a 1-by-n range containing the command line arguments.

Scoping and Namespace

For functions and for global variables, there is a single namespace that is shared between all files composing an Extend program, and they are visible throughout the entire program. Functions declared in external libraries share this namespace as well. For a local variable, the scope is the entire body of the function in which it is defined. Functions may declare local variables sharing a name with a global variable; inside that function, the name will refer to the local variable.

```
1 global x := "I'm a global";
2
3 foo() {
4     y := x; // Scope of x is entire function
5     x := "In here I'm a local";
6     return y; // Returns "In here I'm a local"
7 }
8
9 bar(x) {
10     return x; // Parameters mask globals; returns argument
11 }
12
13 baz() {
14     return x; // Returns "I'm a global"
15 }
```

3.8 Standard Library Reference

3.9 Example Program

```
1 import "./samples/stdlib.xtnd";
2
3 main([1,n] args) {
4     /* Get a working copy */
5     return 0;
6 }
```

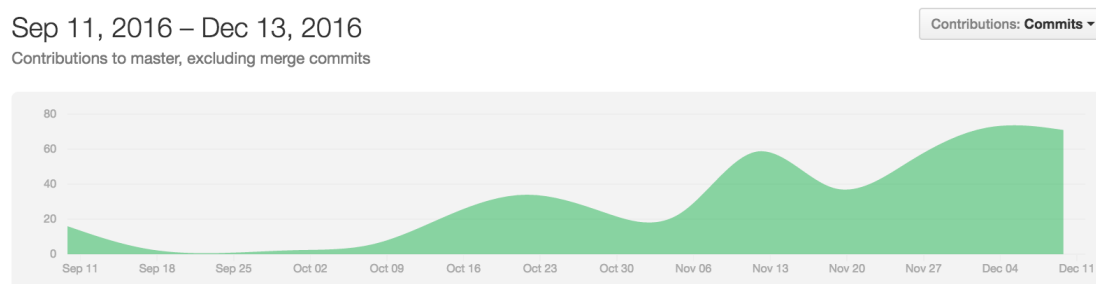
4. Project Plan

4.1 Meetings

Our goals were outlined by weekly meetings. We regularly met with Jacob Graff, our advisor throughout the development of Extend. Jacob served as a sounding board whenever Extend’s fundamental design philosophy was debated, and as a guide as we determined whether we were on track. We used any leftover time on those days to set goals for the upcoming week and pair program if time permitted.

Our team also met weekly on Fridays to further discuss the progression of Extend. In the first half of the semester, the discussions were primarily philosophical, as decisions had to be made about the language grammar and behavior of certain Extend artifacts prior to development. In the second half, time was devoted to ironing out the development timeline, discussing bugs, and making compiler implementation decisions.

4.2 Development Workflow



Github & Travis CI

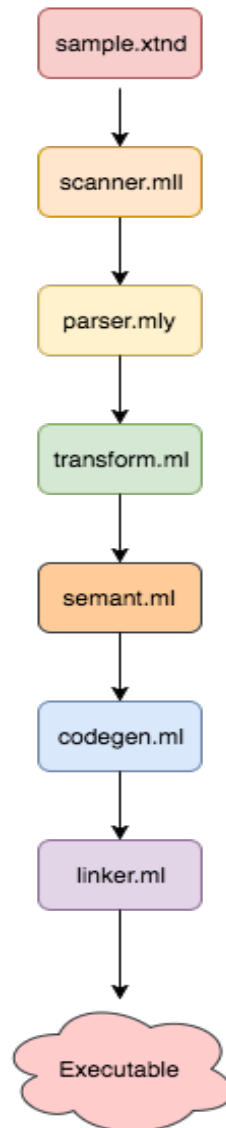
Our development and documentation were all done entirely through version control to maximize independent productivity. New features were introduced to the master branch through pull requests, and the team used this as a platform to peer review code to maximize code quality before such features entered production.

An important aspect of development for us was continuous integration. We used Travis CI to trigger project builds on each pull request, which kept us informed regarding unexpected hiccups that sometimes arose during development. Travis CI ensured that new features were implemented with protecting the code base in mind, and provided quick visibility as to whether a new feature would break the existing build.

4.3 Team Member Responsibilities

| Team Member | Responsibilities | GitHub Profile |
|----------------|--|-------------------------------|
| Jared Samet | design philosophy, semantic transformations, code generation | oracleofnj |
| Nigel Schuster | development protocol, code generation, scripting | Neitsch |
| Ishaan Kolluri | initial LRM, Final Report, regression tests, stdlib functions, scripting | ishaankolluri |
| Kevin Ye | initial scanner, regression tests, stdlib functions | kevinye1 |

5. Extend's Internal Architecture



5.1 The Extend Compiler

The Extend compilation process consists of several source files, each of which performs a different function in the compilation pipeline.

- `scanner.mll`: OCamllex scanner - consumes tokens.
- `parser.mly`: OCamllyacc parser - represents the Extend grammar.
- `ast.ml`: Abstract Syntax Tree, created from the output of the parser and representing the structure of an Extend program.
- `transform.ml`: Performs syntactic desugaring for easier compilation.
- `semant.ml`: Analyzes the semantics of the program to ensure that the program adheres to the rules of the language.
- `codegen.ml`: The LLVM IR code generator.
- `linker.ml`: Calls intermediary compilation steps on the generated `.ll`, including external functions if needed.

The Scanner

The function of `scanner.mll` is to parse a text stream into various tokens to be used in an Extend program. Only the tokens that are valid in Extend are to be given to the parser; all others will return a syntax error marked by the line and character number.

The Parser and Abstract Syntax Tree

The parser converts the tokens read by the scanner into a syntax tree deemed acceptable grammar within the Extend Language. This is converted into an Abstract Syntax Tree, which has nodes that can be consumed by the back end of the Extend compiler.

The Transformer

The transformer expands compact statements in the Extend syntax tree into statements with equivalent functionality, but reduced breadth. This step is done to preserve the convenience for the user, but revert the code later into a form that is easier for the compiler to chew on.

The Semantic Analyzer

The semantic analyzer ensures that Extend functions, variables, expressions, and more are being used properly at compile time, and throws flavorful exceptions to the user so that they may better understand why their program was illegal.

The Code Generator

Provided that the program was deemed legal by `transform.ml`, the code generator will take the program definition in the abstract syntax tree and generate the appropriate LLVM IR to turn it into a functional program. Instructions to allocate memory, interact with external functions, and platform optimization can be found here.

The Linker

If successful LLVM IR is generated, the linker will adopt the role of building an executable object from the .ll file. This includes compiling it to an object file and linking the runtime environment along with other imported libraries.

6. Testing

Due to Extend being a large undertaking, we took steps to ensure that all features were working as the design of the language intended.

This was done through implementing test cases that isolated specific aspects of the Extend language to ensure that each feature worked correctly. For basic components, we wrote a plethora of tests to illustrate functionality. For undertakings that required more debate on the design of the language, other tests were created and modified throughout development.

6.1 Feature Integration & Testing

Development of new features naturally means that they must be deemed legal by the scanner, parser, semantic analyzer, and code generator. As we developed new features, the process was roughly as follows:

1. Write a simple test that illustrated the feature to test.
2. Write the expected output of the aforementioned test to a text file.
3. Confirm that the scanner consumes the tokens related to the feature.
4. Confirm that the parser grammar has been adjusted to accomodate the new feature.
5. Confirm that the semantic analyzer and transformer can properly identify and check the new feature code.
6. Confirm that code generation generates the appropriate LLVM IR for the new features - such as allocating memory, building calls, and more.
7. Ensure that the test written can write its output to `stdout`, to be compared with expected output.
8. Compile and test the code to ensure that the code has worked to the team's expectations.

Earlier in the development process, we tested the front end of our compiler by JSON-ifying the abstract syntax tree, printing it, and examinining it. As we settled into full-fledged development, we would test with a full-feature regression test suite. Later in the semester, JSON-ifying still proved to be useful, as it gave us the option to print debug statements if needed.

6.2 Regression Test Suite

Extend's test suite is executable through the `testscript.sh` script at the top level of the project. There are over 100 integration test files for various features of the Extend language, and a corresponding file with their expected output to `stdout`. This is to ensure that the successful implementation of one feature does not impact that of others.

Regression tests were placed in the `testcases/inputs_regression` directory. Tests that did not pass at the time were placed in the `testcases/inputs` directory. The test script compiles and executes each test, and compares it with the corresponding expected output file, living in the `testcases/expected` directory. Whenever a test passed in `inputs`, it was automatically moved over to `inputs_regression`.

Note: We have added a full test listing at the end of this document. Please refer to the chapter titled "Test Listing" for more detail.

Integration with Travis CI

The aforementioned test suite is run by Travis CI in the event that the Extend compiler is successfully built; otherwise, the build will fail and exit. In our development workflow, checking the logs during build failures sometimes revealed that tests in the regression test suite did not succeed as expected. This integration kept the far-reaching effects of newly introduced features entirely transparent throughout the process.

Using Travis CI allowed us to maintain the working ability of our compiler, as it ensured that every new feature pushed to the master branch would still result in a successful build. This proved to be invaluable when testing the compiler at a macro-level, or providing Jacob, our TA, with up-to-date demonstrations.

7. Extend Code Listing

7.1 scanner.mll

```
1 {
2   open Lexing
3   open Parser
4   open String
5
6   exception SyntaxError of string
7   let syntax_error lexbuf = raise (SyntaxError("Invalid character: " ^ Lexing.lexeme
8     lexbuf))
9 }
10
11 let digit = ['0'-'9']
12 let exp = 'e' ('+'|'-'|'|'/'|'*')? ['0'-'9']+
13 let flt = (digit)+ ('.' (digit)* exp?|exp)
14 let id = ['a'-'z' 'A'-'Z'] ['a'-'z' 'A'-'Z' '0'-'9' '_' ]*
15
16 rule token = parse
17   ['\n']           { new_line lexbuf; token lexbuf }
18 | [' ' '\t' '\r']   { token lexbuf } (* Whitespace *)
19 | "/*"             { multiline_comment lexbuf }
20 | "//"             { oneline_comment lexbuf }
21 | '"'              { read_string (Buffer.create 17) lexbuf }
22 | '['              { LSQBRACK }
23 | ']'              { RSQBRACK }
24 | '('              { LPAREN }
25 | ')'              { RPAREN }
26 | '{'              { LBRACE }
27 | '}'              { RBRACE }
28 | ":@"             { GETS }
29 | '='              { ASN }
30 | ':'              { COLON }
31 | ','              { COMMA }
32 | "<->"             { PRECEDES }
33 | '?'              { QUESTION }
34 | "=="             { EQ }
35 | "!="             { NOTEQ }
36 | '<'              { LT }
37 | '>'              { GT }
38 | "<="             { LTEQ }
39 | ">="             { GTEQ }
40 | ';'              { SEMI }
```

```

41 | '!'          { LOGNOT }
42 | "&&"        { LOGAND }
43 | "||"        { LOGOR }
44 | '~'         { BITNOT }
45 | '&'         { BITAND }
46 | '|'         { BITOR }
47 | '^'         { BITXOR }
48 | '_'         { UNDERSCORE }
49 | '+'         { PLUS }
50 | '-'         { MINUS }
51 | '*'         { TIMES }
52 | '/'         { DIVIDE }
53 | '%'         { MOD }
54 | "**"         { POWER }
55 | "<<"         { LSHIFT }
56 | ">>"         { RSHIFT }
57 | '#'         { HASH }
58 | "if"        { IF }
59 | "empty"     { EMPTY }
60 | "size"      { SIZE }
61 | "type"      { TYPE }
62 | "row"       { ROW }
63 | "column"    { COLUMN }
64 | "switch"    { SWITCH }
65 | "case"      { CASE }
66 | "default"   { DEFAULT }
67 | "return"    { RETURN }
68 | "import"    { IMPORT }
69 | "global"    { GLOBAL }
70 | "extern"    { EXTERN }
71 | "debug"     { DEBUG }
72 | digit+ as lit { LIT_INT(int_of_string lit) }
73 | flt as lit   { LIT_FLOAT(float_of_string lit) }
74 | id as lit    { ID(lit) }
75 | eof          { EOF }
76 | _           { syntax_error lexbuf }
77
78 and multiline_comment = parse
79   "*/" { token lexbuf }
80 | '\n' { new_line lexbuf; multiline_comment lexbuf }
81 | _    { multiline_comment lexbuf }
82
83 and oneline_comment = parse
84   '\n' { new_line lexbuf; token lexbuf }
85 | _    { oneline_comment lexbuf }
86
87 (* read_string mostly taken from:
88 https://realworldocaml.org/v1/en/html/parsing-with-ocamllex-and-menhir.html *)
89 and read_string buf =
90   parse
91   | '"'        { LIT_STRING (Buffer.contents buf) }
92   | '\n'       { new_line lexbuf; Buffer.add_char buf '\n'; read_string buf lexbuf }
93   | '\\\' 'n'   { Buffer.add_char buf '\n'; read_string buf lexbuf }
94   | '\\\' 'r'   { Buffer.add_char buf '\r'; read_string buf lexbuf }
95   | '\\\' 't'   { Buffer.add_char buf '\t'; read_string buf lexbuf }
96   | '\\\' ([^\\\' 'n' 'r' 't'] as lxm)

```

```

97   { Buffer.add_char buf lxm; read_string buf lexbuf }
98   | [^ "'" '\\\']+
99   { Buffer.add_string buf (Lexing.lexeme lexbuf);
100     read_string buf lexbuf
101   }
102   | _ { syntax_error lexbuf }
103   | eof { raise (Failure("unterminated string")) }

```

7.2 parser.mly

```

1  /* Ocamlyacc parser for Extend */
2
3  %{
4  open Ast
5  %}
6
7  %token LSQBRACK RSQBRACK LPAREN RPAREN LBRACE RBRACE HASH
8  %token COLON COMMA QUESTION IF GETS ASN SEMI PRECEDES UNDERSCORE
9  %token SWITCH CASE DEFAULT SIZE TYPE ROW COLUMN
10 %token PLUS MINUS TIMES DIVIDE MOD POWER LSHIFT RSHIFT
11 %token EQ NOTEQ GT LT GTEQ LTEQ
12 %token LOGNOT LOGAND LOGOR
13 %token BITNOT BITXOR BITAND BITOR
14 %token EMPTY RETURN IMPORT GLOBAL EXTERN
15 %token DEBUG
16 %token <int> LIT_INT
17 %token <float> LIT_FLOAT
18 %token <string> LIT_STRING
19 %token <string> ID
20 %token EOF
21
22 %right QUESTION
23 %left PRECEDES
24 %left LOGOR
25 %left LOGAND
26 %left EQ NOTEQ LT GT LTEQ GTEQ
27 %left PLUS MINUS BITOR BITXOR
28 %left TIMES DIVIDE MOD LSHIFT RSHIFT BITAND
29 %right POWER
30 %right BITNOT LOGNOT NEG
31 %left LSQBRACK
32
33 %start program
34 %type <Ast.raw_program> program
35
36 %%
37
38 program:
39     program_piece EOF { let (imp, glob, fnc, ext) = $1 in (List.rev imp, List.rev
40       glob, List.rev fnc, List.rev ext) }
41
42 program_piece:
43     /* nothing */ { ([], [], [], []) }
44     | program_piece import { let (imp, glob, fnc, ext) = $1 in ($2 :: imp, glob,
45       fnc, ext) }

```

```

44 | program_piece global      { let (imp, glob, fnc, ext) = $1 in (imp, $2 :: glob,
    fnc, ext) }
45 | program_piece func_decl  { let (imp, glob, fnc, ext) = $1 in (imp, glob, $2 ::
    fnc, ext) }
46 | program_piece extern    { let (imp, glob, fnc, ext) = $1 in (imp, glob, fnc, $2
    :: ext) }
47
48 import:
49     IMPORT LIT_STRING SEMI {$2}
50
51 global:
52     GLOBAL varinit {$2}
53
54 extern:
55     EXTERN LIT_STRING LBRACE opt_extern_list RBRACE {(Library($2, $4))}
56
57 opt_extern_list:
58     /* nothing */ { [] }
59 | extern_list { List.rev $1 }
60
61 extern_list:
62     extern_fn { [$1] }
63 | extern_list extern_fn { $2 :: $1 }
64
65 extern_fn:
66     ID LPAREN func_param_list RPAREN SEMI
67     { {
68         extern_fn_name = $1;
69         extern_fn_params = $3;
70         extern_fn_libname = "";
71         extern_ret_val = (None, None);
72     } }
73 | ret_dim ID LPAREN func_param_list RPAREN SEMI
74     { {
75         extern_fn_name = $2;
76         extern_fn_params = $4;
77         extern_fn_libname = "";
78         extern_ret_val = $1;
79     } }
80
81 func_decl:
82     ID LPAREN func_param_list RPAREN LBRACE opt_stmt_list ret_stmt RBRACE
83     { {
84         name = $1;
85         params = $3;
86         body = $6;
87         raw_asserts = [];
88         ret_val = ((None, None), $7)
89     } }
90 | ret_dim ID LPAREN func_param_list RPAREN LBRACE opt_stmt_list ret_stmt RBRACE
91     { {
92         name = $2;
93         params = $4;
94         body = $7;
95         raw_asserts = [];
96         ret_val = ($1, $8);

```

```

97     } }
98
99 opt_stmt_list:
100     /* nothing */ { [] }
101     | stmt_list { List.rev $1 }
102
103 stmt_list:
104     stmt { [$1] }
105     | stmt_list stmt { $2 :: $1 }
106
107 stmt:
108     varinit { $1 } | assign { $1 }
109
110 ret_stmt:
111     RETURN expr SEMI { $2 }
112
113 varinit:
114     var_list SEMI { Varinit((None, None), List.rev $1) }
115     | dim var_list SEMI { Varinit($1, List.rev $2) }
116
117 var_list:
118     ID varassign { [ ($1, $2)] }
119     | var_list COMMA ID varassign { ($3, $4) :: $1 }
120
121 varassign:
122     /* nothing */ { None }
123     | GETS expr { Some $2 }
124
125 assign:
126     ID lhs_sel ASN expr SEMI { Assign($1, $2, Some $4) }
127
128 expr:
129     expr rhs_sel          { Selection($1, $2) }
130     | HASH ID              { Selection(Id($2), (None, None)) }
131     | op_expr              { $1 }
132     | ternary_expr         { $1 }
133     | switch_expr          { $1 }
134     | func_expr            { $1 }
135     | range_expr           { $1 }
136     | DEBUG LPAREN expr RPAREN { Debug($3) }
137     | expr PRECEDES expr   { Precedence($1, $3) }
138     | LPAREN expr RPAREN   { $2 }
139     | ID                   { Id($1) }
140     | LIT_INT              { LitInt($1) }
141     | LIT_FLOAT             { LitFlt($1) }
142     | LIT_STRING           { LitString($1) }
143     | EMPTY                { Empty }
144
145 op_expr:
146     expr PLUS expr        { BinOp($1, Plus, $3) }
147     | expr MINUS expr     { BinOp($1, Minus, $3) }
148     | expr TIMES expr     { BinOp($1, Times, $3) }
149     | expr DIVIDE expr    { BinOp($1, Divide, $3) }
150     | expr MOD expr       { BinOp($1, Mod, $3) }
151     | expr POWER expr     { BinOp($1, Pow, $3) }
152     | expr LSHIFT expr    { BinOp($1, LShift, $3) }

```

```

153 | expr RSHIFT expr    { BinOp($1, RShift, $3) }
154 | expr LOGAND expr    { BinOp($1, LogAnd, $3) }
155 | expr LOGOR expr     { BinOp($1, LogOr, $3) }
156 | expr BITXOR expr    { BinOp($1, BitXor, $3) }
157 | expr BITAND expr    { BinOp($1, BitAnd, $3) }
158 | expr BITOR expr     { BinOp($1, BitOr, $3) }
159 | expr EQ expr        { BinOp($1, Eq, $3) }
160 | expr NOTEQ expr     { UnOp(LogNot, (BinOp($1, Eq, $3))) }
161 | expr GT expr        { BinOp($1, Gt, $3) }
162 | expr LT expr        { BinOp($1, Lt, $3) }
163 | expr GTEQ expr      { BinOp($1, GtEq, $3) }
164 | expr LTEQ expr      { BinOp($1, LtEq, $3) }
165 | SIZE LPAREN expr RPAREN { UnOp(SizeOf, $3) }
166 | TYPE LPAREN expr RPAREN { UnOp(TypeOf, $3) }
167 | ROW LPAREN RPAREN      { UnOp(Row, Empty)}
168 | COLUMN LPAREN RPAREN  { UnOp(Column, Empty)}
169 | MINUS expr %prec NEG   { UnOp(Neg, $2) }
170 | LOGNOT expr            { UnOp(LogNot, $2) }
171 | BITNOT expr            { UnOp(BitNot, $2) }
172
173 ternary_expr:
174     IF LPAREN expr COMMA expr COMMA expr RPAREN { Ternary($3, $5, $7) }
175     | expr QUESTION expr COLON expr %prec QUESTION { Ternary($1, $3, $5) }
176
177 switch_expr:
178     SWITCH LPAREN switch_cond RPAREN LBRACE default_case_list RBRACE { Switch($3, fst
179         $6, snd $6) }
180     | SWITCH LBRACE default_case_list RBRACE { Switch(None, fst $3, snd $3) }
181
182 switch_cond:
183     /* nothing */ { None }
184     | expr { Some $1 }
185
186 default_case_list:
187     case_list {(List.rev $1, Empty)}
188     | case_list default_expr {(List.rev $1, $2)}
189
190 case_list:
191     case_stmt { [$1] }
192     | case_list case_stmt { $2 :: $1 }
193
194 case_stmt:
195     CASE case_expr_list COLON expr SEMI { (List.rev $2, $4) }
196
197 default_expr:
198     DEFAULT COLON expr SEMI { $3 }
199
200 case_expr_list:
201     expr { [$1] }
202     | case_expr_list COMMA expr { $3 :: $1 }
203
204 func_expr:
205     ID LPAREN opt_arg_list RPAREN { Call($1, $3) }
206
207 range_expr:
208     LBRACE row_list RBRACE { allow_range_literal (LitRange(List.rev $2)) }

```



```

208
209 row_list:
210     col_list {[List.rev $1]}
211   | row_list SEMI col_list {List.rev $3 :: $1}
212
213 col_list:
214     expr {[ $1]}
215   | col_list COMMA expr {$3 :: $1}
216
217 opt_arg_list:
218     /* nothing */ {[ ]}
219   | arg_list { List.rev $1 }
220
221 arg_list:
222     expr {[ $1]}
223   | arg_list COMMA expr {$3 :: $1}
224
225 lhs_sel:
226     /* nothing */ { (None, None) }
227 /* commented out: LSQBRACK lslice RSQBRACK { (Some $2, None) } */
228   | LSQBRACK lslice COMMA lslice RSQBRACK { (Some $2, Some $4) }
229
230 rhs_sel:
231     LSQBRACK rslice RSQBRACK { (Some $2, None) }
232   | LSQBRACK rslice COMMA rslice RSQBRACK { (Some $2, Some $4) }
233
234 lslice:
235     /* commented out: nothing production { (None, None) } */
236     lslice_val { (Some $1, None) }
237   | lslice_val COLON lslice_val { (Some $1, Some $3) }
238   | lslice_val COLON { (Some $1, Some DimensionEnd) }
239   | COLON lslice_val { (Some DimensionStart, Some $2) }
240   | COLON { (Some DimensionStart, Some DimensionEnd) }
241
242 rslice:
243     /* nothing */ { (None, None) }
244   | rslice_val { (Some $1, None) }
245   | rslice_val COLON rslice_val { (Some $1, Some $3) }
246   | rslice_val COLON { (Some $1, Some DimensionEnd) }
247   | COLON rslice_val { (Some DimensionStart, Some $2) }
248   | COLON { (Some DimensionStart, Some DimensionEnd) }
249
250 lslice_val:
251     expr { Abs($1) }
252
253 rslice_val:
254     expr { Abs($1) }
255   | LSQBRACK expr RSQBRACK { Rel($2) }
256
257 func_param_list:
258     /* nothing */ { [ ] }
259   | func_param_int_list { List.rev $1 }
260
261 func_param_int_list:
262     func_sin_param { [ $1] }
263   | func_param_int_list COMMA func_sin_param { $3 :: $1 }

```

```

264
265 func_sin_param:
266     ID { ((None, None), $1) }
267     | dim ID { ($1, $2) }
268
269 dim:
270     LSQBRACK expr RSQBRACK { (Some $2, None) }
271     | LSQBRACK expr COMMA expr RSQBRACK { (Some $2, Some $4) }
272
273 ret_dim:
274     LSQBRACK ret_sin RSQBRACK { ($2, None) }
275     | LSQBRACK ret_sin COMMA ret_sin RSQBRACK { ($2,$4) }
276
277 ret_sin:
278     expr { Some $1 }
279     | UNDERSCORE { Some Wild }

```

7.3 ast.ml

```

1 type op          = Plus | Minus | Times | Divide | Mod | Pow |
2                  LShift | RShift | BitOr | BitAnd | BitXor |
3                  Eq | Gt | GtEq | Lt | LtEq | LogAnd | LogOr
4 type unop        = Neg | LogNot | BitNot | SizeOf | TypeOf | Row | Column | Truthy
5
6 type expr        = LitInt of int |
7                  LitFlt of float |
8                  LitString of string |
9                  LitRange of (expr list) list |
10                 Id of string |
11                 Empty |
12                 Wild |
13                 BinOp of expr * op * expr |
14                 UnOp of unop * expr |
15                 Ternary of expr * expr * expr |
16                 Switch of expr option * case list * expr |
17                 Call of string * expr list |
18                 Selection of expr * sel |
19                 ReducedTernary of string * string * string |
20                 Precedence of expr * expr |
21                 Debug of expr
22 and index        = Abs of expr |
23                 Rel of expr |
24                 DimensionStart |
25                 DimensionEnd
26 and slice        = index option * index option
27 and sel          = slice option * slice option
28 and case         = expr list * expr
29
30 type dim          = expr option * expr option
31 type var          = dim * string
32 type assign       = string * sel * expr option
33 type init         = string * expr option
34 type stmt         = Assign of assign |
35                 Varinit of dim * init list
36

```

```

37 type raw_func = {
38     name: string;
39     params: var list;
40     body: stmt list;
41     raw_asserts: expr list;
42     ret_val: dim * expr;
43 }
44
45 type extern_func = {
46     extern_fn_name: string;
47     extern_fn_params: var list;
48     extern_fn_libname: string;
49     extern_ret_val: dim;
50 }
51
52 type library = Library of string * extern_func list
53 type raw_program = string list * stmt list * raw_func list * library list
54
55 (* Desugared types below *)
56 module StringMap = Map.Make(String)
57 type formula = {
58     formula_row_start: index;
59     formula_row_end: index option;
60     formula_col_start: index;
61     formula_col_end: index option;
62     formula_expr: expr;
63 }
64
65 type dim_expr = DimInt of int
66               | DimId of string
67
68 type variable = {
69     var_rows: dim_expr;
70     var_cols: dim_expr;
71     var_formulas: formula list;
72 }
73
74 type func_decl = {
75     func_params: var list;
76     func_body: variable StringMap.t;
77     func_asserts: expr list;
78     func_ret_val: dim * expr;
79 }
80
81 type program = (variable StringMap.t) * (func_decl StringMap.t) * (extern_func
    StringMap.t)
82
83 type listable = Inits of init list |
84               Vars of var list |
85               Stmts of stmt list |
86               RawFuncs of raw_func list |
87               Externs of extern_func list |
88               Libraries of library list |
89               Exprs of expr list |
90               Rows of (expr list) list |
91               Strings of string list |

```

```

92         Cases of case list |
93         Formulas of formula list
94
95 exception IllegalRangeLiteral of string
96 exception TransformedAway of string
97
98 let quote_string str =
99     let escape_characters = Str.regexp "[\n \t \r \\ \"]" in
100     let replace_fn s = match Str.matched_string s with
101     | "\n" -> "\\n" |
102     | "\t" -> "\\t" |
103     | "\r" -> "\\r" |
104     | "\"" -> "\\\"" |
105     | "\\" -> "\\\" |
106     | _ -> Str.matched_string s in
107     "\"" ^ Str.global_substitute escape_characters replace_fn str ^ "\""
108
109 let string_of_op o = "\"" ^ (match o with
110     Plus -> "+" | Minus -> "-" | Times -> "*" | Divide -> "/" | Mod -> "%" | Pow ->
111     "**" |
112     LShift -> "<<" | RShift -> ">>" | BitOr -> "|" | BitAnd -> "&" | BitXor -> "^" |
113     Eq -> "==" | Gt -> ">" | GtEq -> ">=" | Lt -> "<" | LtEq -> "<=" |
114     LogAnd -> "&&" | LogOr -> "||" ) ^ "\""
115
116 let string_of_unop = function
117     Neg -> "\"-\"" | LogNot -> "\"!\"" | BitNot -> "\"~\"" | Truthy -> "\"truthy\"" |
118     SizeOf -> "\"size\"" | TypeOf -> "\"type\"" | Row -> "\"row\"" | Column -> "\"
119     column\""
120
121 let rec string_of_expr = function
122     LitInt(l) -> "{\\"LitInt\\": " ^ string_of_int l ^ "}"
123     | LitFlt(l) -> "{\\"LitFlt\\": " ^ string_of_float l ^ "}"
124     | LitString(s) -> "{\\"LitString\\": " ^ quote_string s ^ "}"
125     | LitRange(rowlist) -> "{\\"LitRange\\": " ^ string_of_list (Rows rowlist) ^ "}"
126     | Id(s) -> "{\\"Id\\": " ^ quote_string s ^ "}"
127     | Empty -> "\"Empty\""
128     | Wild -> "\"Wild\""
129     | BinOp(e1, o, e2) -> "{\\"BinOp\\": { " ^
130     | UnOp(o, e) -> "{\\"UnOp\\": { " ^
131     | Ternary(c, e1, e2) -> "{\\"Ternary\\": { " ^
132     | ReducedTernary(s1, s2, s3) -> "{\\"ReducedTernary\\": { " ^
133     | Switch(eo, cases, dflt) -> "{\\"Switch\\": { " ^
134     (match eo with None -> "null" | Some e ->
135     string_of_expr e) ^ ", " ^

```

```

145         "\"cases\": " ^ string_of_list (Cases cases) ^ ", " ^
146         "\"defaultExpr\": " ^ string_of_expr dflt ^ "}"
147 | Call(f, arguments) -> "{\\"Call\\": { " ^
148         "\"function\": " ^ quote_string f ^ ", " ^
149         "\"arguments\": " ^ string_of_list (Exprs arguments) ^
150         "}}"
151 | Selection(e, s) -> "{\\"Selection\\": { " ^
152         "\"expr\": " ^ string_of_expr e ^ ", " ^
153         "\"slices\": " ^ string_of_sel s ^ "}"
154 | Precedence(e1, e2) -> "{\\"Precedence\\": { " ^
155         "\"prior_expr\": " ^ string_of_expr e1 ^ ", " ^
156         "\"dependent_expr\": " ^ string_of_expr e2 ^ "}"
157 | Debug(e) -> string_of_expr e
158 and string_of_case (el, e) =
159     "{\\"Cases\\": " ^ string_of_list (Exprs el) ^ ", " ^
160     "\"expr\": " ^ string_of_expr e ^ "}"
161
162 and string_of_sel (s1, s2) =
163     "{\\"slice1\\": " ^ string_of_slice s1 ^ ", \\"slice2\\": " ^ string_of_slice s2 ^ "}"
164
165 and string_of_slice = function
166     None -> "null"
167 | Some (start_idx, end_idx) -> "{\\"start\\": " ^ string_of_index start_idx ^ ", \\"end
168     \": " ^ string_of_index end_idx ^ "}"
169
170 and string_of_index = function
171     None -> "null"
172 | Some (Abs(e)) -> "{\\"Absolute\\": " ^ string_of_expr e ^ "}"
173 | Some (Rel(e)) -> "{\\"Relative\\": " ^ string_of_expr e ^ "}"
174 | Some (DimensionStart) -> "\\"DimensionStart\\"
175 | Some (DimensionEnd) -> "\\"DimensionEnd\\"
176
177 and string_of_dim (d1,d2) = "{\\"d1\\": " ^ (match d1 with None -> "null" | Some e ->
178     string_of_expr e) ^ ", " ^
179     "\"d2\\": " ^ (match d2 with None -> "null" | Some e ->
180     string_of_expr e) ^ "}"
181
182 and string_of_var (d, s) = "{\\"Dimensions\\": " ^ string_of_dim d ^ ", " ^
183     "\"VarName\\": " ^ quote_string s ^ "}"
184
185 and string_of_assign (s, selection, eo) =
186     "{\\"VarName\\": " ^ quote_string s ^ ", " ^
187     "\"Selection\\": " ^ string_of_sel selection ^ ", " ^
188     "\"expr\\": " ^ (match eo with None -> "null" | Some e -> string_of_expr e) ^ "}"
189
190 and string_of_varinit (d, inits) =
191     "{\\"Dimensions\\": " ^ string_of_dim d ^
192     ",\\"Initializations\\": " ^ string_of_list (Inits inits) ^ "}"
193
194 and string_of_init (s, eo) =
195     "{\\"VarName\\": " ^ quote_string s ^ ", " ^
196     "\"expr\\": " ^ (match eo with None -> "null" | Some e -> string_of_expr e) ^ "}"
197
198 and string_of_stmt = function
199     Assign(a) -> "{\\"Assign\\": " ^ string_of_assign a ^ "}"

```

```

197 | Varinit(d, inits) -> "{\"Varinit\": \" ^ string_of_varinit (d, inits) ^ \"}"
198
199 and string_of_range (d, e) = "{\"Dimensions\": \" ^ string_of_dim d ^ \", \" ^
200     \"expr\": \" ^ string_of_expr e ^ \"}"
201
202 and string_of_raw_func fd =
203     "{\"Name\": \" ^ quote_string fd.name ^ \", \" ^
204     \"Params\": \" ^ string_of_list (Vars fd.params) ^ \", \" ^
205     \"Stmts\": \" ^ string_of_list (Stmts fd.body) ^ \", \" ^
206     \"Assertions\": \" ^ string_of_list (Exprs fd.raw_asserts) ^ \", \" ^
207     \"ReturnVal\": \" ^ string_of_range fd.ret_val ^ \"}"
208
209 and string_of_extern_func fd =
210     "{\"Name\": \" ^ quote_string fd.extern_fn_name ^ \", \" ^
211     \"Params\": \" ^ string_of_list (Vars fd.extern_fn_params) ^ \", \" ^
212     \"Library\": \" ^ quote_string fd.extern_fn_libname ^ \", \" ^
213     \"ReturnDim\": \" ^ string_of_dim fd.extern_ret_val ^ \"}"
214
215 and string_of_library (Library(lib_name, lib_fns)) =
216     "{\"LibraryName\": \" ^ quote_string lib_name ^ \", \" ^
217     \"ExternalFunctions\": \" ^ string_of_list (Externs lib_fns) ^ \"}"
218
219 and string_of_dimexpr = function
220     DimInt(i) -> string_of_int i
221     | DimId(s) -> quote_string s
222
223 and string_of_formula f =
224     "{\"RowStart\": \" ^ string_of_index (Some f.formula_row_start) ^ \", \" ^
225     \"RowEnd\": \" ^ string_of_index (f.formula_row_end) ^ \", \" ^
226     \"ColumnStart\": \" ^ string_of_index (Some f.formula_col_start) ^ \", \" ^
227     \"ColumnEnd\": \" ^ string_of_index (f.formula_col_end) ^ \", \" ^
228     \"Formula\": \" ^ string_of_expr f.formula_expr ^ \"}"
229
230 and string_of_list l =
231     let stringrep = (match l with
232         Inits (il) -> List.map string_of_init il
233         | Vars (vl) -> List.map string_of_var vl
234         | Stmts (sl) -> List.map string_of_stmt sl
235         | RawFuncs (fl) -> List.map string_of_raw_func fl
236         | Externs (efl) -> List.map string_of_extern_func efl
237         | Libraries (libl) -> List.map string_of_library libl
238         | Exprs (el) -> List.map string_of_expr el
239         | Rows (rl) -> List.map (fun (el : expr list) -> string_of_list (Exprs el)) rl
240         | Strings (sl) -> List.map quote_string sl
241         | Cases (cl) -> List.map string_of_case cl
242         | Formulas (fl) -> List.map string_of_formula fl)
243     in "[" ^ String.concat ", " stringrep ^ "]"
244
245 let string_of_raw_program (imp, glb, fs, exts) =
246     "{\"Program\": { \" ^
247     \"Imports\": \" ^ string_of_list (Strings imp) ^ \", \" ^
248     \"Globals\": \" ^ string_of_list (Stmts glb) ^ \", \" ^
249     \"ExternalLibraries\": \" ^ string_of_list (Libraries exts) ^ \", \" ^
250     \"Functions\": \" ^ string_of_list (RawFuncs fs) ^ \"}}}"
251
252 let string_of_variable v =

```

```

253   "{"Rows\": " ^ string_of_dimexpr v.var_rows ^ "," ^
254   "\"Columns\": " ^ string_of_dimexpr v.var_cols ^ "," ^
255   "\"Formulas\": " ^ string_of_list (Formulas v.var_formulas) ^ "}"
256
257 let string_of_map value_desc val_printing_fn m =
258   let f_key_val_list k v l = (
259     "{" ^ value_desc ^ "Name\": " ^ quote_string k ^ "," ^
260     "\" ^ value_desc ^ "Def\": " ^ val_printing_fn v ^ "}"
261   ) :: l in
262   "[" ^ String.concat ", " (List.rev (StringMap.fold f_key_val_list m [])) ^ "]"
263
264 let string_of_funcdecl f =
265   "{"Params\": " ^ string_of_list (Vars f.func_params) ^ "," ^
266   "\"Variables\": " ^ string_of_map "Variable" string_of_variable f.func_body ^ "," ^
267   "\"Assertions\": " ^ string_of_list (Exprs f.func_asserts) ^ "," ^
268   "\"ReturnVal\": " ^ string_of_range f.func_ret_val ^ "}"
269
270 let string_of_program (glb, fs, exts) =
271   "{"Program\": {" ^
272     "\"Globals\": " ^ string_of_map "Variable" string_of_variable glb ^ "," ^
273     "\"Functions\": " ^ string_of_map "Function" string_of_funcdecl fs ^ "," ^
274     "\"ExternalFunctions\": " ^ string_of_map "ExternalFunctions"
275       string_of_extern_func exts ^ "}"
276
277 let allow_range_literal = function
278   LitRange(rowlist) ->
279     let rec check_range_literal rl =
280       List.for_all (fun exprs -> List.for_all check_basic_expr exprs) rl
281     and check_basic_expr = function
282       LitInt(_) | UnOp(Neg, LitInt(_)) | LitFlt(_) | UnOp(Neg, LitFlt(_)) |
283       LitString(_) | Empty -> true
284     | LitRange(rl) -> check_range_literal rl
285     | _ -> false in
286     if check_range_literal rowlist then LitRange(rowlist)
287     else raise (IllegalRangeLiteral(string_of_expr (LitRange(rowlist))))
288   | e -> raise (IllegalRangeLiteral(string_of_expr e))

```

7.4 transform.ml

```

1  open Ast
2  open Lexing
3  open Parsing
4
5  exception IllegalExpression of string;;
6  exception DuplicateDefinition of string;;
7  exception UnknownVariable of string;;
8  exception UnknownFunction of string;;
9  exception WrongNumberArgs of string;;
10 exception LogicError of string;;
11
12 module StringSet = Set.Make (String);;
13 let importSet = StringSet.empty;;
14
15 let builtin_signatures = [("cos", 1); ("column", 0); (*("printf", 2);*) ("toString",

```

```

15     1)]
16
17 let idgen =
18   (* from http://stackoverflow.com/questions/10459363/side-effects-and-top-level-
      expressions-in-ocaml*)
19   let count = ref (-1) in
20   fun prefix -> incr count; "_tmp_" ^ prefix ^ string_of_int !count;;
21
22 let expand_file filename =
23   let print_error_location filename msg lexbuf =
24     let pos = lexbuf.lex_curr_p in
25     prerr_endline ("Syntax error in \"^ filename ^ "\": " ^ msg) ;
26     prerr_endline ("Line " ^ (string_of_int pos.pos_lnum) ^ " at character " ^ (
      string_of_int (pos.pos_cnum - pos.pos_bol))) in
27
28   let rec expand_imports processed_imports globals fns exts dir = function
29     [] -> ([], globals, fns, exts)
30   | import :: imports ->
31     (* print_endline "-----";
32      print_endline ("Working on: " ^ import) ;
33      print_endline ("Already processed:"); *)
34     (* StringSet.iter (fun a -> print_endline a) processed_imports; *)
35     let in_chan = open_in import in
36     let lexbuf = (Lexing.from_channel (in_chan)) in
37     let (file_imports, file_globals, file_functions, file_extens) =
38       try Parser.program Scanner.token lexbuf
39       with
40         Parsing.Parse_error -> print_error_location import "" lexbuf ; exit(-1)
41       | Scanner.SyntaxError(s) -> print_error_location import s lexbuf ; exit(-1)
42     in
43     let file_imports = List.map (fun file -> dir ^ "/" ^ file) file_imports in
44     let new_proc = StringSet.add import processed_imports and _ = close_in in_chan
45     in
46     (* print_endline ("Now I'm done with: ") ; *)
47     (* StringSet.iter (fun a -> print_endline a) new_proc; *)
48     let first_im_hearing_about imp = not (StringSet.mem imp new_proc || List.mem imp
49       imports) in
50     let new_imports = StringSet.elements (StringSet.of_list (List.filter
51       first_im_hearing_about file_imports)) in
52     (* print_endline ("First I'm hearing about:"); *)
53     (* List.iter print_endline new_imports; *)
54     expand_imports new_proc (globals @ file_globals) (fns @ file_functions) (exts @
55       file_extens) (Filename.dirname import) (imports @ new_imports) in
56   expand_imports StringSet.empty [] [] [] (Filename.dirname filename) [filename]
57
58 let expand_expressions (imports, globals, functions, externs) =
59   let lit_zero = LitInt(0) in let abs_zero = Abs(lit_zero) in
60   let lit_one = LitInt(1) in let abs_one = Abs(lit_one) in
61   let one_by_one = (Some lit_one, Some lit_one) in
62   let zero_comma_zero = (Some (Some abs_zero, Some abs_one),
63     Some (Some abs_zero, Some abs_one)) in
64   let entire_dimension = (Some DimensionStart, Some DimensionEnd) in
65   let entire_range = (Some entire_dimension, Some entire_dimension) in
66
67   let expand_expr expr_loc = function
68     (* Create a new variable for all expressions on the LHS to hold the result;

```



```

65     return the new expression and whatever new statements are necessary to create
        the new variable *)
66     Empty      -> raise (IllegalExpression("Empty not allowed in " ^ expr_loc))
67 | Wild        -> raise (IllegalExpression("wild - this shouldn't be possible"))
68 | LitString(s) -> raise (IllegalExpression("String literal " ^ quote_string s ^ "
        not allowed in " ^ expr_loc))
69 | LitRange(rl) -> raise (IllegalExpression("Range literal " ^ string_of_list (Rows
        rl) ^ " not allowed in " ^ expr_loc))
70 | e           -> let new_id = idgen expr_loc in (
71     Id(new_id),
72     [Varinit (one_by_one, [(new_id, None)]];
73     Assign (new_id, zero_comma_zero, Some e)]) in
74
75 let expand_index index_loc = function
76   (* Expand one index of a slice if necessary. *)
77   Abs(e) -> let (new_e, new_stmts) = expand_expr index_loc e in
78   (Abs(new_e), new_stmts)
79 | DimensionStart -> (DimensionStart, [])
80 | DimensionEnd -> (DimensionEnd, [])
81 | Rel(_) -> raise (IllegalExpression("relative - this shouldn't be possible")) in
82
83 let expand_slice slice_loc = function
84   (* Expand one or both sides as necessary. *)
85   None -> (entire_dimension, [])
86 | Some (Some (Abs(e)), None) ->
87   let (start_e, start_stmts) = expand_expr (slice_loc ^ "_start") e in
88   ((Some (Abs(start_e)), None), start_stmts)
89 | Some (Some idx_start, Some idx_end) ->
90   let (new_start, new_start_exprs) = expand_index (slice_loc ^ "_start") idx_start
      in
91   let (new_end, new_end_exprs) = expand_index (slice_loc ^ "_end") idx_end in
92   ((Some new_start, Some new_end), new_start_exprs @ new_end_exprs)
93 | Some (Some _, None) | Some (None, _) -> raise (IllegalExpression("Illegal slice
      - this shouldn't be possible")) in
94
95 let expand_assign asgn_loc (var_name, (row_slice, col_slice), formula) =
96   (* expand_assign: Take an Assign and return a list of more
97   atomic statements, with new variables replacing any
98   complex expressions in the selection slices and with single
99   index values desugared to expr:expr+1. *)
100   try
101     let (new_row_slice, row_exprs) = expand_slice (asgn_loc ^ "_" ^ var_name ^ "_row
        ") row_slice in
102     let (new_col_slice, col_exprs) = expand_slice (asgn_loc ^ "_" ^ var_name ^ "_col
        ") col_slice in
103     Assign(var_name, (Some new_row_slice, Some new_col_slice), formula) :: (
        row_exprs @ col_exprs)
104   with IllegalExpression(s) ->
105     raise (IllegalExpression("Illegal expression (" ^ s ^ ") in " ^
106       string_of_assign (var_name, (row_slice, col_slice),
        formula))) in
107
108 let expand_init (r, c) (v, e) =
109   Varinit((Some r, Some c), [(v, None)]) ::
110   match e with
111   None -> []

```

```

112 | Some e -> [Assign (v, entire_range, Some e)] in
113
114 let expand_dimension dim_loc = function
115   None -> expand_expr dim_loc (LitInt(1))
116   | Some e -> expand_expr dim_loc e in
117
118 let expand_varinit fname ((row_dim, col_dim), inits) =
119   (* expand_varinit: Take a Varinit and return a list of more atomic
120    statements. Each dimension will be given a temporary ID, which
121    will be declared as [1,1] _tmpXXX; the formula for tmpXXX will be
122    set as a separate assignment; the original variable will be
123    declared as [_tmpXXX, _tmpYYY] var; and the formula assignment
124    will be applied to[:,:]. *)
125   try
126     let (row_e, row_stmts) = expand_dimension (fname ^ "_" ^ (String.concat "_" (
127       List.map fst inits)) ^ "_row_dim") row_dim in
128     let (col_e, col_stmts) = expand_dimension (fname ^ "_" ^ (String.concat "_" (
129       List.map fst inits)) ^ "_col_dim") col_dim in
130     row_stmts @ col_stmts @ List.concat (List.map (expand_init (row_e, col_e)) inits
131     )
132   with IllegalExpression(s) ->
133     raise (IllegalExpression("Illegal expression (" ^ s ^ ") in " ^
134       string_of_varinit ((row_dim, col_dim), inits))) in
135
136 let expand_stmt fname = function
137   Assign(a) -> expand_assign fname a
138   | Varinit(d, inits) -> expand_varinit fname (d, inits) in
139
140 let expand_stmt_list fname stmts = List.concat (List.map (expand_stmt fname) stmts)
141   in
142
143 let expand_params fname params =
144   let needs_sizevar = function
145     ((None, None), _) -> false
146     | _ -> true in
147   let params_with_sizevar = List.map (fun x -> (idgen (fname ^ "_" ^ (snd x) ^ "
148     _size"), x)) (List.filter needs_sizevar params) in
149   let expanded_args = List.map (fun (sv, ((rv, cv), s)) -> ((sv, s), [((sv, abs_zero
150     ), rv); ((sv, abs_one), cv)])) params_with_sizevar in
151   let (sizes, inits) = (List.map fst expanded_args, List.concat (List.map snd
152     expanded_args)) in
153   let add_item (varset, (assertlist, initlist)) ((sizevar, pos), var) =
154     (match var with
155      Some Id(s) ->
156        if StringSet.mem s varset then
157          (* We've seen this variable before; don't initialize it, just assert it *)
158          (varset, (BinOp(Id(s), Eq, Selection(Id(sizevar), (Some(Some(pos), None),
159            None))) :: assertlist, initlist))
160        else
161          (* We're seeing a string for the first time; don't assert it, just create
162            it *)
163          (StringSet.add s varset, (assertlist,
164            Assign(s, zero_comma_zero, Some (Selection(Id(
165              sizevar), (Some(Some(pos), None), None)))) ::
166            Varinit(one_by_one, [(s, None)]) ::
167            initlist))

```

```

158     | Some LitInt(i) -> (* Seeing a number; don't do anything besides create an
159       assertion *)
160       (varset, (BinOp(LitInt(i), Eq, Selection(Id(sizevar), (Some(Some(pos), None),
161         None))) :: assertlist, initlist))
162     | Some e -> raise (IllegalExpression("Illegal expression (" ^ string_of_expr e
163       ^ ") in function signature"))
164     | _ -> raise (IllegalExpression("Cannot supply a single dimension in function
165       signature")) in
166   let (rev_assertions, rev_inits) = snd (List.fold_left add_item (StringSet.empty,
167     ([], [])) inits) in
168   let create_sizevar (sizevar, arg) = [
169     Varinit(one_by_one, [(sizevar, None)]);
170     Assign(sizevar, entire_range, Some(UnOp(SizeOf, Id(arg))))] in
171   (List.concat (List.map create_sizevar sizes), List.rev rev_assertions, List.rev
172     rev_inits) in
173
174   let expand_function f =
175     let (new_sizevars, assertions, size_inits) = expand_params f.name f.params in
176     let new_retval_id = idgen (f.name ^ "_retval") in
177     let new_retval = Id(new_retval_id) in
178     let retval_inits = [Varinit (one_by_one, [(new_retval_id, None)]];
179     Assign (new_retval_id, zero_comma_zero, Some (snd f.ret_val))]
180     in
181     {
182       name = f.name;
183       params = f.params;
184       raw_asserts = assertions;
185       body = new_sizevars @ size_inits @ retval_inits @ expand_stmt_list f.name f.body
186       ;
187       ret_val = (fst f.ret_val, new_retval)
188     } in
189   (imports, expand_stmt_list "global" globals, List.map expand_function functions,
190     externs);;
191
192   let map_of_list list_of_tuples =
193     (* map_of_list: Take a list of the form [("foo", 2); ("bar", 3)]
194       and create a StringMap using the first value of the tuple as
195       the key and the second value of the tuple as the value. Raises
196       an exception if the key appears more than once in the list. *)
197     let rec aux acc = function
198       [] -> acc
199       | t :: ts ->
200         if (StringMap.mem (fst t) acc) then raise (DuplicateDefinition(fst t))
201         else aux (StringMap.add (fst t) (snd t) acc) ts in
202     aux StringMap.empty list_of_tuples
203
204   let create_maps (imports, globals, functions, externs) =
205     let vd_of_vi = function
206       (* vd_of_vi— Take a bare Varinit from the previous transformations
207         and return a (string, variable) pair *)
208       Varinit((Some r, Some c), [(v, None)]) -> (v, {
209         var_rows = (match r with
210           LitInt(i) -> DimInt(i)
211           | Id(s) -> DimId(s)
212           | _ -> raise (LogicError("Unrecognized expression for rows of " ^ v)));
213         var_cols = (match c with

```

```

205         LitInt(i) -> DimInt(i)
206         | Id(s) -> DimId(s)
207         | _ -> raise (LogicError("Unrecognized expression for rows of " ^ v));
208     var_formulas = [];
209 })
210 | _ -> raise (LogicError("Unrecognized format for post-desugaring Varinit")) in
211
212 let add_formula m = function
213     Varinit(_,_) -> m
214     | Assign(var_name, (Some (Some row_start, row_end), Some (Some col_start, col_end
215         )), Some e) ->
216         if StringMap.mem var_name m
217         then (let v = StringMap.find var_name m in
218             StringMap.add var_name {v with var_formulas = v.var_formulas @ [{
219                 formula_row_start = row_start;
220                 formula_row_end = row_end;
221                 formula_col_start = col_start;
222                 formula_col_end = col_end;
223                 formula_expr = e;
224             }]} m)
225         else raise (UnknownVariable(string_of_stmt (Assign(var_name, (Some (Some
226             row_start, row_end), Some (Some col_start, col_end)), Some e))))
227     | Assign(a) -> raise (LogicError("Unrecognized format for post-desugaring Assign:
228         " ^ string_of_stmt (Assign(a)))) in
229
230 let vds_of_stmts stmts =
231     let is_varinit = function Varinit(_,_) -> true | _ -> false in
232     let varinits = List.filter is_varinit stmts in
233     let vars_just_the_names = map_of_list (List.map vd_of_vi varinits) in
234     List.fold_left add_formula vars_just_the_names stmts in
235
236 let fd_of_raw_func f = (f.name, {
237     func_params = f.params;
238     func_body = vds_of_stmts f.body;
239     func_ret_val = f.ret_val;
240     func_asserts = f.raw_asserts;
241 }) in
242
243 let tupleize_library (Library(lib_name, lib_fns)) =
244     List.map (fun ext_fn -> (ext_fn.extern_fn_name, {ext_fn with extern_fn_libname =
245         lib_name})) lib_fns in
246
247 (vds_of_stmts globals,
248     map_of_list (List.map fd_of_raw_func functions),
249     map_of_list (List.concat (List.map tupleize_library externs)))
250
251 let single_formula e = {
252     formula_row_start = DimensionStart;
253     formula_row_end = Some DimensionEnd;
254     formula_col_start = DimensionStart;
255     formula_col_end = Some DimensionEnd;
256     formula_expr = e;
257 }
258
259 let ternarize_exprs (globals, functions, externs) =
260     let rec ternarize_expr lhs_var = function

```

```

257   BinOp(e1, LogAnd, e2) ->
258   let (new_e1, new_e1_vars) = ternarize_expr lhs_var e1 in
259   let (new_e2, new_e2_vars) = ternarize_expr lhs_var e2 in
260   (Ternary(UnOp(Truthy,new_e1), UnOp(Truthy,new_e2), LitInt(0)), new_e1_vars @
    new_e2_vars)
261 | BinOp(e1, LogOr, e2) ->
262   let (new_e1, new_e1_vars) = ternarize_expr lhs_var e1 in
263   let (new_e2, new_e2_vars) = ternarize_expr lhs_var e2 in
264   (Ternary(UnOp(Truthy,new_e1), LitInt(1), UnOp(Truthy,new_e2)), new_e1_vars @
    new_e2_vars)
265 | BinOp(e1, op, e2) ->
266   let (new_e1, new_e1_vars) = ternarize_expr lhs_var e1 in
267   let (new_e2, new_e2_vars) = ternarize_expr lhs_var e2 in
268   (BinOp(new_e1, op, new_e2), new_e1_vars @ new_e2_vars)
269 | UnOp(op, e) ->
270   let (new_e, new_e_vars) = ternarize_expr lhs_var e in
271   (UnOp(op, new_e), new_e_vars)
272 | Ternary(cond, e1, e2) ->
273   let (new_cond, new_cond_vars) = ternarize_expr lhs_var cond in
274   let (new_e1, new_e1_vars) = ternarize_expr lhs_var e1 in
275   let (new_e2, new_e2_vars) = ternarize_expr lhs_var e2 in
276   (Ternary(new_cond, new_e1, new_e2), new_cond_vars @ new_e1_vars @ new_e2_vars)
277 | Call(fname, args) ->
278   let new_args_and_vars = List.map (ternarize_expr lhs_var) args in
279   (Call(fname, (List.map fst new_args_and_vars)), List.concat (List.map snd
    new_args_and_vars))
280 | Selection(e, (sl1, sl2)) ->
281   let (new_e, new_e_vars) = ternarize_expr lhs_var e in
282   let (new_sl1, new_sl1_vars) = ternarize_slice lhs_var sl1 in
283   let (new_sl2, new_sl2_vars) = ternarize_slice lhs_var sl2 in
284   (Selection(new_e, (new_sl1, new_sl2)), new_e_vars @ new_sl1_vars @ new_sl2_vars)
285 | Precedence(e1, e2) ->
286   let (new_e1, new_e1_vars) = ternarize_expr lhs_var e1 in
287   let (new_e2, new_e2_vars) = ternarize_expr lhs_var e2 in
288   (Precedence(new_e1, new_e2), new_e1_vars @ new_e2_vars)
289 | Switch(cond, cases, dflt) ->
290   ternarize_switch lhs_var cases dflt cond
291 | LitRange(rowlist) -> let (lhs_varname, _) = lhs_var in formulize_litrage
    lhs_varname rowlist
292 | Debug(e) ->
293   let (new_e, new_e_vars) = ternarize_expr lhs_var e in
294   (Debug(new_e), new_e_vars)
295 | e -> (e, [])
296 and formulize_litrage lhs_varname rowlist =
297   let new_range_id = idgen (lhs_varname ^ "_litrage") in
298   let num_rows = List.length rowlist in
299   let num_cols = List.fold_left max 0 (List.map List.length rowlist) in
300   let formulize_expr r c = function
301     LitRange(rl) -> formulize_litrage (new_range_id ^ "_" ^ string_of_int r ^ "_"
    ^ string_of_int c) rl
302   | e -> (e, []) in
303   let formulize_row rownum col_exprs =
304     let col_formulas_and_vars = List.mapi (fun c e -> formulize_expr rownum c e)
    col_exprs in
305     let create_formula colnum e = {
306       formula_row_start = Abs(LitInt(rownum)); formula_row_end = None;

```

```

307     formula_col_start = Abs(LitInt(colnum)); formula_col_end = None;
308     formula_expr = e;
309   } in
310   (List.mapi create_formula (List.map fst col_formulas_and_vars), List.concat (
311     List.map snd col_formulas_and_vars)) in
312   let formulas_and_vars = List.mapi formulize_row rowlist in
313   let range_var = {
314     var_rows = DimInt(num_rows); var_cols = DimInt(num_cols);
315     var_formulas = List.concat (List.map fst formulas_and_vars);
316   } in
317   (Id(new_range_id),
318    (new_range_id, range_var) ::
319    List.concat (List.map snd formulas_and_vars))
320 and ternarize_switch lhs_var cases dflt cond =
321   let (new_cond_expr, new_cond_vars) = (match cond with
322     Some cond_expr ->
323       let (lhs_varname, lhs_vardef) = lhs_var in
324       let new_id = idgen (lhs_varname ^ "_switch_cond") in
325       let (new_e, new_e_vars) = ternarize_expr lhs_var cond_expr in
326       (Some (Selection (Id(new_id), (Some (Some (Rel (LitInt (0))), None), Some (Some (Rel (
327         LitInt (0))), None)))),
328       (new_id, {lhs_vardef with var_formulas = [single_formula new_e]})) ::
329       new_e_vars)
330   | None ->
331     (None, [])
332   ) in
333   let new_cases_and_vars = List.map (ternarize_case lhs_var new_cond_expr) cases in
334   let new_cases = List.map fst new_cases_and_vars in
335   let new_case_vars = List.concat (List.map snd new_cases_and_vars) in
336   let (new_dflt, new_dflt_vars) = ternarize_expr lhs_var dflt in
337   let rec combine_everything = function
338     [] -> new_dflt
339     | (combined_cases, e) :: more_cases -> Ternary(combined_cases, e,
340       combine_everything more_cases) in
341   (combine_everything new_cases, new_cond_vars @ new_case_vars @ new_dflt_vars)
342 and ternarize_case lhs_var cond (conds, e) =
343   let new_conds_and_vars = List.map (ternarize_expr lhs_var) conds in
344   let new_conds = List.map fst new_conds_and_vars in
345   let new_cond_vars = List.concat (List.map snd new_conds_and_vars) in
346   let (new_e, new_e_vars) = ternarize_expr lhs_var e in
347   let unify_case_cond_and_switch_cond case_cond = function
348     None -> case_cond
349     | Some switch_cond -> BinOp(switch_cond, Eq, case_cond) in
350   let rec unify_switch_cond_and_case_conds switch_cond = function
351     [case_cond] -> unify_case_cond_and_switch_cond case_cond switch_cond
352     | case_cond :: case_conds ->
353       let (combined_expr, _) = ternarize_expr lhs_var
354         (BinOp(unify_case_cond_and_switch_cond case_cond switch_cond, LogOr,
355           unify_switch_cond_and_case_conds switch_cond case_conds)) in
356       combined_expr
357     | [] -> raise (LogicError("Empty case condition list")) in
358   ((unify_switch_cond_and_case_conds cond new_conds, new_e), new_cond_vars @
359     new_e_vars)
360 and ternarize_slice lhs_var = function
361   None -> (None, [])
362   | Some (i1, i2) ->

```

```

358     let (new_i1, new_i1_vars) = ternarize_index lhs_var i1 in
359     let (new_i2, new_i2_vars) = ternarize_index lhs_var i2 in
360     (Some (new_i1, new_i2), new_i1_vars @ new_i2_vars)
361 and ternarize_index lhs_var = function
362     Some Abs(e) ->
363     let (new_e, new_e_vars) = ternarize_expr lhs_var e in
364     (Some (Abs(new_e)), new_e_vars)
365   | Some Rel(e) ->
366     let (new_e, new_e_vars) = ternarize_expr lhs_var e in
367     (Some (Rel(new_e)), new_e_vars)
368   | i -> (i, []) in
369 let ternarize_formula lhs_var f =
370     let (new_expr, new_vars) = ternarize_expr lhs_var f.formula_expr in
371     ({f with formula_expr = new_expr}, new_vars) in
372 let ternarize_variable varname vardef =
373     let new_formulas_and_vars = List.map (ternarize_formula (varname, vardef)) vardef.
374     var_formulas in
375     ({vardef with var_formulas = List.map fst new_formulas_and_vars}, List.concat (
376     List.map snd new_formulas_and_vars)) in
375 let ternarize_variables fn_name m =
376     let new_variables_and_maps = StringMap.mapi (fun varname vardef ->
377     ternarize_variable (fn_name ^ "_" ^ varname) vardef) m in
378     let add_item var_name (orig_var, new_vars) l = ((var_name, orig_var) :: fst l,
379     new_vars :: snd l) in
380     let combined_list = StringMap.fold add_item new_variables_and_maps ([],[]) in
381     map_of_list (List.rev (fst combined_list) @ List.concat (snd combined_list)) in
382 let ternarize_function fn_name fn_def = {fn_def with func_body = ternarize_variables
383     fn_name fn_def.func_body} in
384 (ternarize_variables "global" globals, StringMap.mapi ternarize_function functions,
385     externs)
386
387 let reduce_ternaries (globals, functions, externs) =
388     let rec reduce_expr lhs_var = function
389     | BinOp(e1, op, e2) ->
390     let (new_e1, new_e1_vars) = reduce_expr lhs_var e1 in
391     let (new_e2, new_e2_vars) = reduce_expr lhs_var e2 in
392     (BinOp(new_e1, op, new_e2), new_e1_vars @ new_e2_vars)
393   | UnOp(op, e) ->
394     let (new_e, new_e_vars) = reduce_expr lhs_var e in
395     (UnOp(op, new_e), new_e_vars)
396   | Ternary(cond, e1, e2) -> reduce_ternary lhs_var cond e1 e2
397   | Call(fname, args) ->
398     let new_args_and_vars = List.map (reduce_expr lhs_var) args in
399     (Call(fname, (List.map fst new_args_and_vars)), List.concat (List.map snd
400     new_args_and_vars))
401   | Selection(e, (s11, s12)) ->
402     let (new_e, new_e_vars) = reduce_expr lhs_var e in
403     let (new_s11, new_s11_vars) = reduce_slice lhs_var s11 in
404     let (new_s12, new_s12_vars) = reduce_slice lhs_var s12 in
405     (Selection(new_e, (new_s11, new_s12)), new_e_vars @ new_s11_vars @ new_s12_vars)
406   | Precedence(e1, e2) ->
407     let (new_e1, new_e1_vars) = reduce_expr lhs_var e1 in
408     let (new_e2, new_e2_vars) = reduce_expr lhs_var e2 in
409     (Precedence(new_e1, new_e2), new_e1_vars @ new_e2_vars)
410   | Debug(e) ->
411     let (new_e, new_e_vars) = reduce_expr lhs_var e in

```

```

407     (Debug(new_e), new_e_vars)
408   | e -> (e, [])
409 and reduce_ternary lhs_var cond e1 e2 =
410   let (new_cond, new_cond_vars) = reduce_expr lhs_var cond in
411   let (new_true_e, new_true_vars) = reduce_expr lhs_var e1 in
412   let (new_false_e, new_false_vars) = reduce_expr lhs_var e2 in
413   let (lhs_varname, lhs_vardef) = lhs_var in
414   let new_cond_id = idgen (lhs_varname ^ "_truthiness") in
415   let new_true_id = idgen (lhs_varname ^ "_values_if_true") in
416   let new_false_id = idgen (lhs_varname ^ "_values_if_false") in
417   (ReducedTernary(new_cond_id, new_true_id, new_false_id),
418    (new_cond_id, {lhs_vardef with var_formulas = [single_formula (UnOp(Truthy,
419      new_cond))]})) ::
420   (new_true_id, {lhs_vardef with var_formulas = [single_formula new_true_e]})) ::
421   (new_false_id, {lhs_vardef with var_formulas = [single_formula new_false_e]})) ::
422   (new_cond_vars @ new_true_vars @ new_false_vars))
423 and reduce_slice lhs_var = function
424   None -> (None, [])
425   | Some (i1, i2) ->
426     let (new_i1, new_i1_vars) = reduce_index lhs_var i1 in
427     let (new_i2, new_i2_vars) = reduce_index lhs_var i2 in
428     (Some (new_i1, new_i2), new_i1_vars @ new_i2_vars)
429 and reduce_index lhs_var = function
430   Some Abs(e) ->
431     let (new_e, new_e_vars) = reduce_expr lhs_var e in
432     (Some (Abs(new_e)), new_e_vars)
433   | Some Rel(e) ->
434     let (new_e, new_e_vars) = reduce_expr lhs_var e in
435     (Some (Rel(new_e)), new_e_vars)
436   | i -> (i, []) in
437 let reduce_formula lhs_var f =
438   let (new_expr, new_vars) = reduce_expr lhs_var f.formula_expr in
439   ({f with formula_expr = new_expr}, new_vars) in
440 let reduce_variable varname vardef =
441   let new_formulas_and_vars = List.map (reduce_formula (varname, vardef)) vardef.
442     var_formulas in
443   ({vardef with var_formulas = List.map fst new_formulas_and_vars}, List.concat (
444     List.map snd new_formulas_and_vars)) in
445 let reduce_variables fn_name m =
446   let new_variables_and_maps = StringMap.mapi (fun varname vardef -> reduce_variable
447     (fn_name ^ "_" ^ varname) vardef) m in
448   let add_item var_name (orig_var, new_vars) l = ((var_name, orig_var) :: fst l,
449     new_vars :: snd l) in
450   let combined_list = StringMap.fold add_item new_variables_and_maps ([],[]) in
451   map_of_list (List.rev (fst combined_list) @ List.concat (snd combined_list)) in
452 let reduce_function fn_name fn_def = {fn_def with func_body = reduce_variables
453   fn_name fn_def.func_body} in
454 (reduce_variables "global" globals, StringMap.mapi reduce_function functions,
455   externs)
456
457 let check_semantics (globals, functions, externs) =
458   let fn_signatures = map_of_list
459     (builtin_signatures @
460      (StringMap.fold (fun s f l -> (s, List.length f.func_params) :: l) functions
461        []) @
462      (StringMap.fold (fun s f l -> (s, List.length f.extern_fn_params) :: l) externs

```



```

    [])) in
455 let check_function fname f =
456   if StringMap.mem fname externs then raise(DuplicateDefinition(fname ^ "() is
      defined as both an external and local function")) else ();
457   let locals = f.func_body in
458   let params = List.map snd f.func_params in
459   List.iter
460     (fun param →
461       if StringMap.mem param locals then raise(DuplicateDefinition(param ^ " is
          defined multiple times in " ^ fname ^ "()))
462       else ())
463   params ;
464 let check_call called_fname num_args =
465   match called_fname with
466     "printf" → ()
467   | _ → if (not (StringMap.mem called_fname fn_signatures)) then raise(
      UnknownFunction(called_fname))
468       else let signature_args = StringMap.find called_fname fn_signatures in
469           if num_args != signature_args then raise(WrongNumberArgs(
470             "In " ^ fname ^ "(), the function " ^ called_fname ^ "() was called
              with " ^ string_of_int num_args ^ " arguments " ^
471              "but the signature specifies " ^ string_of_int signature_args
472              ))
473           else () in
474 let rec check_expr = function
475   BinOp(e1,_,e2) → check_expr e1 ; check_expr e2
476   | UnOp(_, e) → check_expr e
477   | Ternary(cond, e1, e2) → check_expr cond ; check_expr e1 ; check_expr e2
478   | ReducedTernary(s1, s2, s3) → check_expr (Id(s1)) ; check_expr (Id(s2)) ;
      check_expr (Id(s3))
479   | Id(s) → if (List.mem s params || StringMap.mem s locals || StringMap.mem s
      globals) then () else raise(UnknownVariable(fname ^ "(): " ^ s))
480   | Switch(Some e, cases, dflt) → check_expr e ; List.iter check_case cases ;
      check_expr dflt
481   | Switch(None, cases, dflt) → List.iter check_case cases ; check_expr dflt
482   | Call(called_fname, args) →
483     check_call called_fname (List.length args) ;
484     List.iter check_expr args
485   | Selection(e, (s11, s12)) → check_expr e ; check_slice s11 ; check_slice s12
486   | Precedence(e1, e2) → check_expr e1 ; check_expr e2
487   | Debug(e) → check_expr e;
488   | LitInt(_) | LitFlt(_) | LitRange(_) | LitString(_) | Empty | Wild → ()
489 and check_case (conds, e) = List.iter check_expr conds ; check_expr e
490 and check_slice = function
491   None → ()
492   | Some (i1, i2) → check_index i1 ; check_index i2
493 and check_index = function
494   Some Abs(e) → check_expr e
495   | Some Rel(e) → check_expr e
496   | _ → () in
497 let check_formula f =
498   check_index (Some f.formula_row_start) ;
499   check_index f.formula_row_end ;
500   check_index (Some f.formula_col_start) ;
501   check_index f.formula_col_end ;
502   check_expr f.formula_expr in

```

```

503   let check_dim = function
504       DimInt(1) -> ()
505       | DimInt(i) -> raise(IllegalExpression("This is not going to work right"))
506       | DimId(s) -> check_expr (Id(s)) in
507   let check_variable v =
508       check_dim v.var_rows ;
509       check_dim v.var_cols ;
510       List.iter check_formula v.var_formulas in
511
512   StringMap.iter (fun _ v -> check_variable v) f.func_body ;
513   check_expr (snd f.func_ret_val)
514
515   in StringMap.iter check_function functions
516
517 let create_ast filename =
518   let ast_imp_res = expand_file filename in
519   let ast_expanded = expand_expressions ast_imp_res in
520   let ast_mapped = create_maps ast_expanded in check_semantics ast_mapped ;
521   let ast_ternarized = ternarize_exprs ast_mapped in
522   let ast_reduced = reduce_ternaries ast_ternarized in check_semantics ast_reduced ;
523   ast_reduced

```

7.5 codegen.ml

```

1  (* Extend code generator *)
2
3  open Ast
4  open CodeGenTypes
5  exception NotImplemented
6  exception LogicError of string
7
8  type symbol = LocalVariable of int | GlobalVariable of int | FunctionParameter of int
9               | ExtendFunction of int
10
11 and symbolTable = symbol StringMap.t
12 and symbolTableType = Locals | Globals | ExtendFunctions
13
14 let helper_functions = Hashtbl.create 10
15 let runtime_functions = Hashtbl.create 20
16
17 let index_map table_type m =
18   let add_item key _ (accum_map, accum_idx) =
19       let index_val = match table_type with Locals -> LocalVariable(accum_idx) | Globals
20       -> GlobalVariable(accum_idx) | ExtendFunctions -> ExtendFunction(accum_idx) in
21       (StringMap.add key index_val accum_map, accum_idx + 1) in
22   StringMap.fold add_item m (StringMap.empty, 0)
23
24 let (=>) struct_ptr elem = (fun val_name builder ->
25   let the_pointer = Llvml.build_struct_gep struct_ptr elem "the_pointer" builder in
26   Llvml.build_load the_pointer val_name builder);;
27
28 (* from http://stackoverflow.com/questions/243864/what-is-the-ocaml-idiom-equivalent-
29 to-pythons-range-function without the infix *)
30 let zero_until i =
31   let rec aux n acc =
32       if n < 0 then acc else aux (n-1) (n :: acc)

```

```

29   in aux (i-1) []
30
31 let create_runtime_functions ctx bt the_module =
32   let add_runtime_func fname returntype arglist =
33     let the_func = Llvm.declare_function fname (Llvm.function_type returntype arglist)
34       the_module
35   in Hashtbl.add runtime_functions fname the_func in
36   add_runtime_func "strlen" bt.long_t [|bt.char_p|];
37   add_runtime_func "strcmp" bt.long_t [|bt.char_p; bt.char_p|];
38   add_runtime_func "pow" bt.float_t [|bt.float_t; bt.float_t|];
39   add_runtime_func "lrint" bt.int_t [|bt.float_t|];
40   add_runtime_func "llvm.memcpy.p0i8.p0i8.i64" bt.void_t [|bt.char_p; bt.char_p; bt.
41     long_t; bt.int_t; bt.bool_t|];
42   add_runtime_func "getVal" bt.value_p [|bt.var_instance_p; bt.int_t; bt.int_t|];
43   add_runtime_func "clone_value" bt.value_p [|bt.value_p|];
44   (* add_runtime_func "freeMe" (Llvm.void_type ctx) [|bt.extend_scope_p|]; *)
45   add_runtime_func "getSize" bt.value_p [|bt.var_instance_p|];
46   add_runtime_func "get_variable" bt.var_instance_p [|bt.extend_scope_p; bt.int_t|];
47   add_runtime_func "null_init" (Llvm.void_type ctx) [|bt.extend_scope_p|];
48   add_runtime_func "debug_print" (Llvm.void_type ctx) [|bt.value_p; bt.char_p|];
49   add_runtime_func "new_string_go_all_the_way" bt.value_p [|bt.char_p|];
50   ()
51
52 let create_helper_functions ctx bt the_module =
53   let create_def_bod fname rtype argtypes =
54     let fn_def = Llvm.define_function fname (Llvm.function_type rtype (Array.of_list
55       argtypes)) the_module in
56     let fn_bod = Llvm.builder_at_end ctx (Llvm.entry_block fn_def) in
57     (fn_def, fn_bod) in
58
59   (* let create_is_subrange_1x1 fname =
60     let is_index_one fn builder idx =
61       let the_value = (Llvm.param fn 0) => (subrange_field_index idx) "the_value"
62         builder in
63       let the_bool = Llvm.build_icmp Llvm.Icmp.Eq the_value (Llvm.const_int bt.int_t
64         1) "the_bool" builder in
65       the_bool in
66     let (fn_def, fn_bod) = create_def_bod fname bt.bool_t [bt.subrange_p] in
67     let one_row = is_index_one fn_def fn_bod SubrangeRows in
68     let one_col = is_index_one fn_def fn_bod SubrangeCols in
69     let one_by_one = Llvm.build_and one_row one_col "one_by_one" fn_bod in
70     let _ = Llvm.build_ret one_by_one fn_bod in
71     Hashtbl.add helper_functions fname fn_def in
72   *)
73
74 let create_new_string fname =
75   let (fn_def, fn_bod) = create_def_bod fname bt.string_p [bt.char_p] in
76   let the_string_ptr = Llvm.build_malloc bt.string_t "the_string_ptr" fn_bod in
77   let src_char_ptr = Llvm.param fn_def 0 in
78   let dst_char_ptr_ptr = Llvm.build_struct_gep the_string_ptr (string_field_index
79     StringCharPtr) "dst_char_ptr_ptr" fn_bod in
80   let string_len = Llvm.build_call (Hashtbl.find runtime_functions "strlen") [|
81     src_char_ptr|] "string_len" fn_bod in
82   let extra_byte = Llvm.build_add string_len (Llvm.const_int bt.long_t 1) "
83     extra_byte" fn_bod in
84   let strlen_ptr = Llvm.build_struct_gep the_string_ptr (string_field_index
85     StringLen) "strlen_ptr" fn_bod in

```

```

76   let refcount_ptr = LlvM.build_struct_gep the_string_ptr (string_field_index
    StringRefCount) "refcount" fn_bod in
77   let dst_char_ptr = LlvM.build_array_malloc bt.char_t extra_byte "dst_char_ptr"
    fn_bod in
78   let _ = LlvM.build_store dst_char_ptr dst_char_ptr_ptr fn_bod in
79   let _ = LlvM.build_call (Hashtbl.find runtime_functions "llvm.memcpy.p0i8.p0i8.i64
    ")
80     [| dst_char_ptr ; src_char_ptr ; extra_byte ; (LlvM.const_int bt.int_t 0) ; (
        LlvM.const_int bt.bool_t 0) |]
81     "" fn_bod in
82   let _ = LlvM.build_store string_len strlen_ptr fn_bod in
83   let _ = LlvM.build_store (LlvM.const_int bt.int_t 1) refcount_ptr fn_bod in
84   let _ = LlvM.build_ret the_string_ptr fn_bod in
85   Hashtbl.add helper_functions fname fn_def in
86
87   (* let create_box_native_string_list fname =
88     let (fn_def, fn_bod) = create_def_bod fname bt.string_p_p [bt.int_t; bt.char_p_p]
        in
89     let argc = LlvM.param fn_def 0 in
90     let argv = LlvM.param fn_def 1 in
91     let ret_val = LlvM.build_array_malloc bt.string_p argc "ret_val" fn_bod in
92     let i_ptr = LlvM.build_alloca bt.int_t "i_ptr" fn_bod in
93     let _ = LlvM.build_store (LlvM.const_int bt.int_t 0) i_ptr fn_bod in (* i = 0; *)
94     let pred_bb = LlvM.append_block ctx "while_pred" fn_def in
95     let body_bb = LlvM.append_block ctx "while_body" fn_def in
96     let merge_bb = LlvM.append_block ctx "merge" fn_def in
97     let _ = LlvM.build_br pred_bb fn_bod in
98     let pred_builder = LlvM.builder_at_end ctx pred_bb in
99     let i_val = LlvM.build_load i_ptr "i" pred_builder in
100    let pred_bool = LlvM.build_icmp LlvM.Icmp.Slt i_val argc "i_lt_argc" pred_builder
        in (* i < argc *)
101    let _ = LlvM.build_cond_br pred_bool body_bb merge_bb pred_builder in
102    let body_builder = LlvM.builder_at_end ctx body_bb in
103    let i_val = LlvM.build_load i_ptr "i" body_builder in
104    let argv_i_addr = LlvM.build_in_bounds_gep argv [|i_val|] "argv_i_addr"
        body_builder in
105    let argv_i = LlvM.build_load argv_i_addr "argv_i" body_builder in
106    let ns_ptr = LlvM.build_call (Hashtbl.find helper_functions "new_string") [|argv_i
        |] "ns_ptr" body_builder in
107    let dst = LlvM.build_in_bounds_gep ret_val [|i_val|] "dst" body_builder in
108    let _ = LlvM.build_store ns_ptr dst body_builder in
109    let i_plus_1 = LlvM.build_add i_val (LlvM.const_int bt.int_t 1) "i_plus_1"
        body_builder in
110    let _ = LlvM.build_store i_plus_1 i_ptr body_builder in
111    let _ = LlvM.build_br pred_bb body_builder in
112    let merge_builder = LlvM.builder_at_end ctx merge_bb in
113    let _ = LlvM.build_ret ret_val merge_builder in
114    Hashtbl.add helper_functions fname fn_def in *)
115
116   let create_box_value_string fname =
117     let (fn_def, fn_bod) = create_def_bod fname bt.value_p [bt.string_p] in
118     let str = LlvM.param fn_def 0 in
119     let ret_val = LlvM.build_malloc bt.value_t "" fn_bod in
120     let sp = LlvM.build_struct_gep ret_val (value_field_index String) "str_pointer"
        fn_bod in
121     let _ = LlvM.build_store (LlvM.const_int bt.char_t (value_field_flags_index String

```

```

    )) (Llvm.build_struct_gep ret_val (value_field_index Flags) "" fn_bod) fn_bod
    in
122   let _ = Llvm.build_store str sp fn_bod in
123   let _ = Llvm.build_ret ret_val fn_bod in
124   Hashtbl.add helper_functions fname fn_def in
125
126   (* let create_box_value_float fname =
127       let (fn_def, fn_bod) = create_def_bod fname bt.value_p [bt.float_t] in
128       let str = Llvm.param fn_def 0 in
129       let ret_val = Llvm.build_malloc bt.value_t "" fn_bod in
130       let sp = Llvm.build_struct_gep ret_val (value_field_index Number) "num_pointer"
131         fn_bod in
132       let _ = Llvm.build_store (Llvm.const_int bt.char_t (value_field_index Number)
133         ) (Llvm.build_struct_gep ret_val (value_field_index Flags) "" fn_bod) fn_bod
134       in
135       let _ = Llvm.build_store str sp fn_bod in
136       let _ = Llvm.build_ret ret_val fn_bod in
137       Hashtbl.add helper_functions fname fn_def in *)
138
139   (* let create_box_single_value fname =
140       let (fn_def, fn_bod) = create_def_bod fname bt.subrange_p [bt.value_p] in
141       let value = Llvm.param fn_def 0 in
142       let subrange = Llvm.build_malloc bt.subrange_t "" fn_bod in
143       let var_instance = Llvm.build_malloc bt.var_instance_t "" fn_bod in
144       let rp = Llvm.build_struct_gep subrange (subrange_field_index BaseRangePtr) "
145         range_p" fn_bod in
146       let vp = Llvm.build_struct_gep var_instance (var_instance_field_index Values) "
147         value_p" fn_bod in
148       let _ = Llvm.build_store value vp fn_bod in
149       let _ = Llvm.build_store var_instance rp fn_bod in
150       let _ = Llvm.build_store (Llvm.const_int bt.int_t 0) (Llvm.build_struct_gep
151         subrange (subrange_field_index BaseOffsetCol) "" fn_bod) fn_bod in
152       let _ = Llvm.build_store (Llvm.const_int bt.int_t 0) (Llvm.build_struct_gep
153         subrange (subrange_field_index BaseOffsetRow) "" fn_bod) fn_bod in
154       let _ = Llvm.build_store (Llvm.const_int bt.int_t 1) (Llvm.build_struct_gep
155         subrange (subrange_field_index SubrangeRows) "" fn_bod) fn_bod in
156       let _ = Llvm.build_store (Llvm.const_int bt.int_t 1) (Llvm.build_struct_gep
157         subrange (subrange_field_index SubrangeCols) "" fn_bod) fn_bod in
158       let _ = Llvm.build_store (Llvm.const_int bt.int_t 1) (Llvm.build_struct_gep
159         var_instance (var_instance_field_index Rows) "" fn_bod) fn_bod in
160       let _ = Llvm.build_store (Llvm.const_int bt.int_t 1) (Llvm.build_struct_gep
161         var_instance (var_instance_field_index Cols) "" fn_bod) fn_bod in
162       let _ = Llvm.build_ret subrange fn_bod in
163       Hashtbl.add helper_functions fname fn_def in *)
164
165   (* create_is_subrange_1x1 "is_subrange_1x1"; *)
166   (* create_get_val "get_val";
167   create_deref_subrange "deref_subrange"; *)
168   create_new_string "new_string";
169   (* create_box_native_string_list "box_native_string_list"; *)
170   create_box_value_string "box_value_string";
171   (* create_box_single_value "box_single_value"; *)
172   (* create_box_value_float "box_value_float"; *)
173   ()

```

```

165 let translate (globals, functions, externs) =
166
167   (* LLVM Boilerplate *)
168   let context = LlvM.global_context () in
169   let base_module = LlvM.create_module context "Extend" in
170   let base_types = setup_types context in
171
172   (* Declare the runtime functions that we need to call *)
173   create_runtime_functions context base_types base_module ;
174   create_helper_functions context base_types base_module ;
175
176   (* Build function_llvalues, which is a StringMap from function name to llvalue.
177    * It includes both functions from external libraries, such as the standard library,
178    * and functions declared within Extend. *)
179   let declare_library_function fname func accum_map =
180     let llvm_ftype = LlvM.function_type base_types.value_p (Array.of_list (List.map (
181       fun a -> base_types.value_p) func.extern_fn_params)) in
182     let llvm_fn = LlvM.declare_function fname llvm_ftype base_module in
183     StringMap.add fname llvm_fn accum_map in
184   let library_functions = StringMap.fold declare_library_function externs StringMap.
185     empty in
186   let define_user_function fname func =
187     let llvm_fname = "extend_" ^ fname in
188     let llvm_ftype = LlvM.function_type base_types.value_p (Array.of_list (List.map (
189       fun a -> base_types.value_p) func.func_params)) in
190     let llvm_fn = LlvM.define_function llvm_fname llvm_ftype base_module in
191     (func, llvm_fn) in
192   let extend_functions = StringMap.mapi define_user_function functions in
193   let function_llvalues = StringMap.fold StringMap.add (StringMap.map snd
194     extend_functions) library_functions in
195
196   (* Build the global symbol table *)
197   let (global_symbols, num_globals) = index_map Globals globals in
198   let (extend_fn_numbers, num_extend_fns) = index_map ExtendFunctions extend_functions
199     in
200
201   (* Create the global array that will hold each function's array of var_defns. *)
202   let vardefn_ptr = LlvM.const_pointer_null base_types.var_defn_p in
203   let vardefn_array = Array.make (StringMap.cardinal extend_functions) vardefn_ptr in
204   let array_of_vardefn_ptrs = LlvM.define_global "array_of_vardefn_ptrs" (LlvM.
205     const_array base_types.var_defn_p vardefn_array) base_module in
206
207   (* Create the pointer to the global scope object *)
208   let global_scope_loc = LlvM.define_global "global_scope_loc" (LlvM.
209     const_pointer_null base_types.extend_scope_p) base_module in
210
211   let main_def = LlvM.define_function "main" (LlvM.function_type base_types.int_t [|
212     base_types.int_t; base_types.char_p_p|]) base_module in
213   let main_bod = LlvM.builder_at_end context (LlvM.entry_block main_def) in
214
215   (* Create the array of value_ps that will contain the responses to TypeOf(val) *)
216   let null_val_ptr = LlvM.const_pointer_null base_types.value_p in
217   let null_val_array = Array.make (Array.length int_to_type_array) null_val_ptr in
218   let array_of_typeof_val_ptrs = LlvM.define_global "array_of_val_ptrs" (LlvM.
219     const_array base_types.value_p null_val_array) base_module in
220   let create_typeof_string i s =

```

```

212 let sp = LlvM.build_global_stringptr s "global_typeof_stringptr" main_bod in
213 let vp = LlvM.build_call (Hashtbl.find runtime_functions "
    new_string_go_all_the_way") [|sp|] "global_typeof_string" main_bod in
214 let vp_dst = LlvM.build_in_bounds_gep array_of_typeof_val_ptrs [|LlvM.const_int
    base_types.int_t 0; LlvM.const_int base_types.int_t i|] ("global_typeof_dst")
    main_bod in
215 let _ = LlvM.build_store vp vp_dst main_bod in
216 () in
217 Array.iteri create_typeof_string int_to_type_array ;
218
219 (* Look these two up once and for all *)
220 (* let deepCopy = Hashtbl.find runtime_functions "deepCopy" in *)
221 (* let freeMe = Hashtbl.find runtime_functions "freeMe" in *)
222 let getVal = Hashtbl.find runtime_functions "getVal" in (*getVal retrieves the value
    of a variable instance for a specific x and y*)
223 let getVar = Hashtbl.find runtime_functions "get_variable" in (*getVar retrieves a
    variable instance based on the offset. It instantiates the variable if it does
    not exist yet*)
224
225 (* build_formula_function takes a symbol table and an expression, builds the LLVM
    function, and returns the llvalue of the function *)
226 let build_formula_function (varname, formula_idx) symbols formula_expr =
227 let form_decl = LlvM.define_function ("formula_fn_" ^ varname ^ "_num_" ^ (
    string_of_int formula_idx)) base_types.formula_call_t base_module in
228 let builder_at_top = LlvM.builder_at_end context (LlvM.entry_block form_decl) in
229 let local_scope = LlvM.param form_decl 0 in
230 let global_scope = LlvM.build_load global_scope_loc "global_scope" builder_at_top
    in
231
232 (* Some repeated stuff to avoid cut & paste *)
233 let empty_type = (LlvM.const_int base_types.char_t (value_field_flags_index Empty)
    ) in
234 let number_type = (LlvM.const_int base_types.char_t (value_field_flags_index
    Number)) in
235 let string_type = (LlvM.const_int base_types.char_t (value_field_flags_index
    String)) in
236 let range_type = (LlvM.const_int base_types.char_t (value_field_flags_index Range)
    ) in
237 let make_block blockname =
238 let new_block = LlvM.append_block context blockname form_decl in
239 let new_builder = LlvM.builder_at_end context new_block in
240 (new_block, new_builder) in
241 let store_number value_ptr store_builder number_llvalue =
242 let sp = LlvM.build_struct_gep value_ptr (value_field_index Number) "num_pointer
    " store_builder in
243 let _ = LlvM.build_store number_type (LlvM.build_struct_gep value_ptr (
    value_field_index Flags) "" store_builder) store_builder in
244 ignore (LlvM.build_store number_llvalue sp store_builder) in
245 let store_empty value_ptr store_builder =
246 ignore (LlvM.build_store empty_type (LlvM.build_struct_gep value_ptr (
    value_field_index Flags) "" store_builder) store_builder) in
247
248 let make_truthiness_blocks blockprefix ret_val =
249 let (merge_bb, merge_builder) = make_block (blockprefix ^ "_merge") in
250
251 let (make_true_bb, make_true_builder) = make_block (blockprefix ^ "_true") in

```

```

252     let _ = store_number ret_val make_true_builder (Llvm.const_float base_types.
          float_t 1.0) in
253     let _ = Llvm.build_br merge_bb make_true_builder in
254
255     let (make_false_bb, make_false_builder) = make_block (blockprefix ^ "_false") in
256     let _ = store_number ret_val make_false_builder (Llvm.const_float base_types.
          float_t 0.0) in
257     let _ = Llvm.build_br merge_bb make_false_builder in
258
259     let (make_empty_bb, make_empty_builder) = make_block (blockprefix ^ "_empty") in
260     let _ = store_empty ret_val make_empty_builder in
261     let _ = Llvm.build_br merge_bb make_empty_builder in
262
263     (make_true_bb, make_false_bb, make_empty_bb, merge_builder) in
264
265   let rec build_expr old_builder exp = match exp with
266     LitInt(i) -> let vvv = Llvm.const_float base_types.float_t (float_of_int i) in
267     let ret_val = Llvm.build_malloc base_types.value_t "int_ret_val" old_builder
          in
268     let _ = store_number ret_val old_builder vvv in
269     (ret_val, old_builder)
270   | LitFlt(f) -> let vvv = Llvm.const_float base_types.float_t f in
271     let ret_val = Llvm.build_malloc base_types.value_t "flt_ret_val" old_builder
          in
272     let _ = store_number ret_val old_builder vvv in
273     (ret_val, old_builder)
274   | Empty ->
275     let ret_val = Llvm.build_malloc base_types.value_t "empty_ret_val" old_builder
          in
276     let _ = store_empty ret_val old_builder in
277     (ret_val, old_builder)
278   | Debug(e) ->
279     let (ret_val, new_builder) = build_expr old_builder e in
280     let _ = Llvm.build_call (Hashtbl.find runtime_functions "debug_print") [|
          ret_val; Llvm.const_pointer_null base_types.char_p|] "" new_builder in
281     (ret_val, new_builder)
282   | Id(name) ->
283     (
284       match (try StringMap.find name symbols with Not_found -> raise(LogicError("
          Something went wrong with your semantic analysis - " ^ name ^ " not found
          "))) with
285       LocalVariable(i) ->
286         let llvm_var = Llvm.build_call getVar [|local_scope; Llvm.const_int
          base_types.int_t i|] "llvm_var" old_builder in
287         (Llvm.build_call getVal [|llvm_var; Llvm.const_int base_types.int_t 0;
          Llvm.const_int base_types.int_t 0|] "local_id_ret_val" old_builder,
          old_builder)
288     | GlobalVariable(i) ->
289       let llvm_var = Llvm.build_call getVar [|global_scope; Llvm.const_int
          base_types.int_t i|] "llvm_var" old_builder in
290       (Llvm.build_call getVal [|llvm_var; Llvm.const_int base_types.int_t 0;
          Llvm.const_int base_types.int_t 0|] "global_id_ret_val" old_builder,
          old_builder)
291   | FunctionParameter(i) ->
292     let paramarray = (local_scope => (scope_field_type_index FunctionParams))
          "paramarray" old_builder in

```



```

293     let param_addr = LlvM.build_in_bounds_gep paramarray [|LlvM.const_int
294         base_type.int_t i|] "param_addr" old_builder in
295     let param = LlvM.build_load param_addr "param" old_builder in
296     (LlvM.build_call (Hashtbl.find runtime_functions "clone_value") [|param|]
297         "function_param_ret_val" old_builder, old_builder)
298 | ExtendFunction(i) -> raise (LogicError("Something went wrong with your
299     semantic analysis - function " ^ name ^ " used as variable in RHS for " ^
300     varname))
301 )
302 | Selection(expr, sel) -> build_expr old_builder expr
303 | Precedence(a,b) -> let (_, new_builder) = build_expr old_builder a in
304     build_expr new_builder b
305 | LitString(str) ->
306     let boxxxx = LlvM.build_call
307         (Hashtbl.find helper_functions "new_string")
308         (Array.of_list [
309             LlvM.build_global_stringptr str "glob_str" old_builder
310         ]) "boxed_str" old_builder in
311     let boxx = LlvM.build_call
312         (Hashtbl.find helper_functions "box_value_string")
313         (Array.of_list [boxxxx]) "box_value_str" old_builder
314     in (boxx, old_builder)
315 | Call(fn,exl) -> (*TODO: Call needs to be reviewed. Possibly switch call
316     arguments to value_p*)
317     let build_one_expr (arg_list, intermediate_builder) e =
318         let (arg_val, next_builder) = build_expr intermediate_builder e in
319         (arg_val :: arg_list, next_builder) in
320     let (reversed_arglist, call_builder) = List.fold_left build_one_expr ([],
321         old_builder) exl in
322     let args = Array.of_list (List.rev reversed_arglist) in
323     let result = LlvM.build_call (
324         StringMap.find fn function_llvalues
325     ) args "call_ret_val" call_builder in
326     (result, call_builder)
327 | BinOp(expr1,op,expr2) -> (
328     let (val1, builder1) = build_expr old_builder expr1 in
329     let (val2, int_builder) = build_expr builder1 expr2 in
330     let bit_shift = (LlvM.const_int base_types.char_t 4) in
331     let expr1_type = (val1 => (value_field_index Flags)) "expr1_type"
332         int_builder in
333     let expr2_type = (val2 => (value_field_index Flags)) "expr2_type"
334         int_builder in
335     let expr1_type_shifted = LlvM.build_shl expr1_type bit_shift "
336         expr_1_type_shifted" int_builder in
337     let combined_type = LlvM.build_add expr1_type_shifted expr2_type "
338         combined_type" int_builder in
339     let number_number = LlvM.const_add (LlvM.const_shl number_type bit_shift)
340         number_type in
341     let string_string = LlvM.const_add (LlvM.const_shl string_type bit_shift)
342         string_type in
343     let empty_empty = LlvM.const_add (LlvM.const_shl empty_type bit_shift)
344         empty_type in
345     let range_range = LlvM.const_add (LlvM.const_shl range_type bit_shift)
346         range_type in
347     let build_simple_binop oppp int_builder =
348         (let ret_val = LlvM.build_malloc base_types.value_t "binop_minus_ret_val"

```

```

    int_builder in
334 let _ = LlvM.build_store
335 (
336     LlvM.const_int
337     base_types.char_t
338     (value_field_flags_index Empty)
339 ) (
340     LlvM.build_struct_gep
341     ret_val
342     (value_field_index Flags)
343     ""
344     int_builder
345 )
346 int_builder
347 in
348 let bailout = (LlvM.append_block context "" form_decl) in
349 let bbailout = LlvM.builder_at_end context bailout in
350 let (numnum_bb, numnum_builder) = make_block "numnum" in
351 let numeric_val_1 = (val1 => (value_field_index Number)) "number_one"
    numnum_builder in
352 let numeric_val_2 = (val2 => (value_field_index Number)) "number_two"
    numnum_builder in
353 let numeric_res = oppp numeric_val_1 numeric_val_2 "numeric_res"
    numnum_builder in
354 let _ = LlvM.build_store
355     numeric_res (
356         LlvM.build_struct_gep
357         ret_val
358         (value_field_index Number)
359         ""
360         numnum_builder
361     )
362     numnum_builder in
363 let _ = LlvM.build_store
364 (
365     LlvM.const_int
366     base_types.char_t
367     (value_field_flags_index Number)
368 ) (
369     LlvM.build_struct_gep
370     ret_val
371     (value_field_index Flags)
372     ""
373     numnum_builder
374 )
375     numnum_builder in
376 let _ = LlvM.build_br bailout numnum_builder in
377 let _ = LlvM.build_cond_br (LlvM.build_icmp LlvM.Icmp.Eq combined_type
    number_number "" int_builder) numnum_bb bailout int_builder in
378     (ret_val, bbailout)
379 )
380 and build_simple_int_binop oppp int_builder =
381     (let ret_val = LlvM.build_malloc base_types.value_t "binop_minus_ret_val"
    int_builder in
382     let _ = LlvM.build_store
383     (

```

```

384         Lllvm.const_int
385         base_types.char_t
386         (value_field_flags_index Empty)
387     ) (
388         Lllvm.build_struct_gep
389         ret_val
390         (value_field_index Flags)
391         ""
392         int_builder
393     )
394     int_builder
395 in
396 let bailout = (Lllvm.append_block context "" form_decl) in
397 let bbailout = Lllvm.builder_at_end context bailout in
398 let (numnum_bb, numnum_builder) = make_block "numnum" in
399 let roundfl x = Lllvm.build_call (Hashtbl.find runtime_functions "lrint
400     ") [|x|] "" numnum_builder in
401 let numeric_val_1 = roundfl ((val1 => (value_field_index Number)) "
402     number_one" numnum_builder) in
403 let numeric_val_2 = roundfl ((val2 => (value_field_index Number)) "
404     number_two" numnum_builder) in
405 let numeric_res = oppp numeric_val_1 numeric_val_2 "numeric_res"
406     numnum_builder in
407 let _ = Lllvm.build_store
408     (Lllvm.build_sitofp numeric_res base_types.float_t "" numnum_builder
409     )
410     (
411         Lllvm.build_struct_gep
412         ret_val
413         (value_field_index Number)
414         ""
415         numnum_builder
416     )
417     numnum_builder in
418 let _ = Lllvm.build_store
419     (
420         Lllvm.const_int
421         base_types.char_t
422         (value_field_flags_index Number)
423     ) (
424         Lllvm.build_struct_gep
425         ret_val
426         (value_field_index Flags)
427         ""
428         numnum_builder
429     )
430     numnum_builder in
431 let _ = Lllvm.build_br bailout numnum_builder in
432 let _ = Lllvm.build_cond_br (Lllvm.build_icmp Lllvm.Icmp.Eq combined_type
433     number_number "" int_builder) numnum_bb bailout int_builder in
434     (ret_val, bbailout)
435 ) in
436 let build_boolean_op numeric_comparator string_comparator int_builder =
437     let ret_val = Lllvm.build_malloc base_types.value_t "binop_gt_ret_val"
438         int_builder in
439     let (make_true_bb, make_false_bb, make_empty_bb, merge_builder) =

```

```

433         make_truthiness_blocks "binop_eq" ret_val in
434
435     let (numnum_bb, numnum_builder) = make_block "numnum" in
436     let numeric_val_1 = (val1 => (value_field_index Number)) "number_one"
437         numnum_builder in
438     let numeric_val_2 = (val2 => (value_field_index Number)) "number_two"
439         numnum_builder in
440     let numeric_greater = LlvM.build_fcmp numeric_comparator numeric_val_1
441         numeric_val_2 "numeric_greater" numnum_builder in
442     let _ = LlvM.build_cond_br numeric_greater make_true_bb make_false_bb
443         numnum_builder in
444
445     let (strstr_bb, strstr_builder) = make_block "strstr" in
446     let str_p_1 = (val1 => (value_field_index String)) "string_one"
447         strstr_builder in
448     let str_p_2 = (val2 => (value_field_index String)) "string_two"
449         strstr_builder in
450     let char_p_1 = (str_p_1 => (string_field_index StringCharPtr)) "char_p_one"
451         strstr_builder in
452     let char_p_2 = (str_p_2 => (string_field_index StringCharPtr)) "char_p_two"
453         strstr_builder in
454     let strcmp_result = LlvM.build_call (Hashtbl.find runtime_functions "
455         strcmp") [|char_p_1; char_p_2|] "strcmp_result" strstr_builder in
456     let string_greater = LlvM.build_icmp string_comparator strcmp_result (LlvM
457         .const_null base_types.long_t) "string_greater" strstr_builder in
458     let _ = LlvM.build_cond_br string_greater make_true_bb make_false_bb
459         strstr_builder in
460
461     let switch_inst = LlvM.build_switch combined_type make_empty_bb 2
462         int_builder in (* Incompatible ==> default to empty *)
463     LlvM.add_case switch_inst number_number numnum_bb;
464     LlvM.add_case switch_inst string_string strstr_bb;
465     (ret_val, merge_builder) in
466 match op with
467 | Minus -> build_simple_binop LlvM.build_fsub int_builder
468 | Plus ->
469     let result = LlvM.build_malloc base_types.value_t "" int_builder
470     and stradd = (LlvM.append_block context "" form_decl)
471     and numadd = (LlvM.append_block context "" form_decl)
472     and bailout = (LlvM.append_block context "" form_decl)
473     and numorstrorother = (LlvM.append_block context "" form_decl)
474     and strorother = (LlvM.append_block context "" form_decl)
475     in
476     let bstradd = LlvM.builder_at_end context stradd
477     and bnumadd = LlvM.builder_at_end context numadd
478     and bnumorstrorother = LlvM.builder_at_end context numorstrorother
479     and bstrorother = LlvM.builder_at_end context strorother
480     and bbailout = LlvM.builder_at_end context bailout
481     and _ = LlvM.build_store
482         (
483             LlvM.const_int
484             base_types.char_t
485             (value_field_flags_index Empty)
486         ) (
487             LlvM.build_struct_gep
488             result

```

```

476         (value_field_index Flags)
477         ""
478         int_builder
479     )
480     int_builder
481 in
482 (*let _ = LlvM.build_cond_br pred_bool body_bb merge_bb pred_builder in
483 *)
484 let isnumber = LlvM.build_icmp
485     LlvM.Icmp.Eq
486     (
487         LlvM.build_load
488         (
489             LlvM.build_struct_gep
490             vail
491             (value_field_index Flags)
492             ""
493             bnumorstrother
494         ) "" bnumorstrother
495     ) (
496         LlvM.const_int
497         base_types.char_t
498         (value_field_flags_index Number)
499     )
500     ""
501     bnumorstrother
502 and isstring = LlvM.build_icmp
503     LlvM.Icmp.Eq
504     (
505         LlvM.build_load
506         (
507             LlvM.build_struct_gep
508             vail
509             (value_field_index Flags)
510             ""
511             bstrorother
512         ) ""
513         bstrorother
514     ) (
515         LlvM.const_int
516         base_types.char_t
517         (value_field_flags_index String)
518     )
519     ""
520     bstrorother
521 and isnumorstring = LlvM.build_icmp
522     LlvM.Icmp.Eq
523     (
524         LlvM.build_load
525         (
526             LlvM.build_struct_gep
527             vail
528             (value_field_index Flags)
529             ""
530             int_builder

```

```

531         )
532         ""
533         int_builder
534     ) (
535         Llvm.build_load
536     (
537         Llvm.build_struct_gep
538         val2
539         (value_field_index Flags)
540         ""
541         int_builder
542     )
543     ""
544     int_builder
545 )
546 ""
547 int_builder
548 and _ = Llvm.build_store (
549     Llvm.build_fadd
550     (
551         Llvm.build_load
552     (
553         Llvm.build_struct_gep
554         val1
555         (value_field_index Number)
556         ""
557         bnumadd
558     )
559     ""
560     bnumadd
561 ) (
562     Llvm.build_load
563     (
564         Llvm.build_struct_gep
565         val2
566         (value_field_index Number)
567         ""
568         bnumadd
569     )
570     ""
571     bnumadd
572 )
573 ""
574 bnumadd
575 ) (
576     Llvm.build_struct_gep
577     result
578     (value_field_index Number)
579     ""
580     bnumadd
581 )
582 bnumadd
583 and _ = Llvm.build_store (
584     Llvm.const_int base_types.char_t (value_field_flags_index Number)
585 ) (
586     Llvm.build_struct_gep

```

```

587         result
588         (value_field_index Flags)
589         ""
590         bnumadd
591     )
592     bnumadd
593 and str1 = LlvM.build_load
594 (
595     LlvM.build_struct_gep
596     val1
597     (value_field_index String)
598     ""
599     bstradd
600 ) "" bstradd
601 and str2 = LlvM.build_load
602 (
603     LlvM.build_struct_gep
604     val2
605     (value_field_index String)
606     ""
607     bstradd
608 ) "" bstradd
609 and newstr =
610 (
611     LlvM.build_malloc base_types.string_t "" bstradd
612 )
613 in
614 let len1 = LlvM.build_load (
615     LlvM.build_struct_gep
616     str1
617     (string_field_index StringLen)
618     ""
619     bstradd
620 ) "" bstradd
621 and len2 = LlvM.build_load (
622     LlvM.build_struct_gep
623     str2
624     (string_field_index StringLen)
625     ""
626     bstradd
627 ) "" bstradd
628 and p1 = LlvM.build_load (
629     LlvM.build_struct_gep
630     str1
631     (string_field_index StringCharPtr)
632     ""
633     bstradd
634 ) "" bstradd
635 and p2 = LlvM.build_load (
636     LlvM.build_struct_gep
637     str2
638     (string_field_index StringCharPtr)
639     ""
640     bstradd
641 ) "" bstradd
642 and dst_char_ptr_ptr = (

```

```

643         Lllvm.build_struct_gep
644         newstr
645         (string_field_index StringCharPtr)
646         ""
647         bstradd
648     )
649     and _ = Lllvm.build_store (
650         Lllvm.const_int base_types.char_t (value_field_flags_index String)
651     ) (
652         Lllvm.build_struct_gep
653         result
654         (value_field_index Flags)
655         ""
656         bstradd
657     ) bstradd
658     and _ = Lllvm.build_store newstr (
659         Lllvm.build_struct_gep
660         result
661         (value_field_index String)
662         ""
663         bstradd
664     )
665     bstradd in
666     let fullLen = Lllvm.build_nsw_add (Lllvm.build_nsw_add len1 len2 ""
667         bstradd) (Lllvm.const_int base_types.long_t 1) "" bstradd
668     and extra_byte2 = (Lllvm.build_add len2 (Lllvm.const_int base_types.long_t
669         1) "" bstradd) in
670     let dst_char = Lllvm.build_array_malloc base_types.char_t (Lllvm.
671         build_trunc fullLen base_types.int_t "" bstradd) "" bstradd in
672     let dst_char2 = Lllvm.build_in_bounds_gep dst_char [|len1|] "" bstradd in
673     let _ = Lllvm.build_call
674         (Hashtbl.find runtime_functions "llvm.memcpy.p0i8.p0i8.i64")
675         [|dst_char; p1; len1; (Lllvm.const_int base_types.int_t 0); (Lllvm.
676             const_int base_types.bool_t 0)|]
677         ""
678         bstradd
679     and _ = Lllvm.build_call
680         (Hashtbl.find runtime_functions "llvm.memcpy.p0i8.p0i8.i64")
681         [|dst_char2; p2; extra_byte2; (Lllvm.const_int base_types.int_t 0); (
682             Lllvm.const_int base_types.bool_t 0)|]
683         ""
684         bstradd
685     and _ = Lllvm.build_store dst_char dst_char_ptr_ptr bstradd
686     in
687     let _ = Lllvm.build_store (Lllvm.build_nsw_add fullLen (Lllvm.const_int
688         base_types.long_t (-1)) "" bstradd) (Lllvm.build_struct_gep newstr (
689             string_field_index StringLen) "" bstradd) bstradd
690     in
691     let _ = Lllvm.build_cond_br isnumorstring numorstrorother bailout
692         int_builder
693     and _ = Lllvm.build_cond_br isnumber numadd strorother bnumorstrorother
694     and _ = Lllvm.build_cond_br isstring stradd bailout bstrorother
695     and _ = Lllvm.build_br bailout bstradd
696     and _ = Lllvm.build_br bailout bnumadd
697     in
698     (result, bbailout)

```



```

691 | Times -> build_simple_binop Llvm.build_fmul int_builder
692 | Eq ->
693   (* let _ = Llvm.build_call (Hashtbl.find runtime_functions "debug_print")
    [|val1; Llvm.build_global_stringptr "Eq operator - value 1" ""
    old_builder|] "" int_builder in
694   let _ = Llvm.build_call (Hashtbl.find runtime_functions "debug_print") [|
    val2; Llvm.build_global_stringptr "Eq operator - value 2" ""
    old_builder|] "" int_builder in *)
695   let ret_val = Llvm.build_malloc base_types.value_t "binop_eq_ret_val"
    int_builder in
696   let (make_true_bb, make_false_bb, _, merge_builder) =
    make_truthiness_blocks "binop_eq" ret_val in
697
698   let (numnum_bb, numnum_builder) = make_block "numnum" in
699   let numeric_val_1 = (val1 => (value_field_index Number)) "number_one"
    numnum_builder in
700   let numeric_val_2 = (val2 => (value_field_index Number)) "number_two"
    numnum_builder in
701   let numeric_equality = Llvm.build_fcmp Llvm.Fcmp.Oeq numeric_val_1
    numeric_val_2 "numeric_equality" numnum_builder in
702   let _ = Llvm.build_cond_br numeric_equality make_true_bb make_false_bb
    numnum_builder in
703
704   let (strstr_bb, strstr_builder) = make_block "strstr" in
705   let str_p_1 = (val1 => (value_field_index String)) "string_one"
    strstr_builder in
706   let str_p_2 = (val2 => (value_field_index String)) "string_two"
    strstr_builder in
707   let char_p_1 = (str_p_1 => (string_field_index StringCharPtr)) "char_p_one"
    strstr_builder in
708   let char_p_2 = (str_p_2 => (string_field_index StringCharPtr)) "char_p_two"
    strstr_builder in
709   let strcmp_result = Llvm.build_call (Hashtbl.find runtime_functions "
    strcmp") [|char_p_1; char_p_2|] "strcmp_result" strstr_builder in
710   let string_equality = Llvm.build_icmp Llvm.Icmp.Eq strcmp_result (Llvm.
    const_null base_types.long_t) "string_equality" strstr_builder in
711   let _ = Llvm.build_cond_br string_equality make_true_bb make_false_bb
    strstr_builder in
712
713   let (rngrng_bb, rngrng_builder) = make_block "rngrng" in
714   (* TODO: Make this case work *)
715   let _ = Llvm.build_br make_false_bb rngrng_builder in
716
717   let switch_inst = Llvm.build_switch combined_type make_false_bb 4
    int_builder in (* Incompatible ==> default to false *)
718   Llvm.add_case switch_inst number_number numnum_bb;
719   Llvm.add_case switch_inst string_string strstr_bb;
720   Llvm.add_case switch_inst range_range rngrng_bb;
721   Llvm.add_case switch_inst empty_empty make_true_bb; (* Nothing to check in
    this case, just return true *)
722   (ret_val, merge_builder)
723 | Gt -> build_boolean_op Llvm.Fcmp.Ogt Llvm.Icmp.Sgt int_builder
724 | GtEq -> build_boolean_op Llvm.Fcmp.Oge Llvm.Icmp.Sge int_builder
725 | Lt -> build_boolean_op Llvm.Fcmp.Olt Llvm.Icmp.Slt int_builder
726 | LtEq -> build_boolean_op Llvm.Fcmp.Ole Llvm.Icmp.Sle int_builder
727 | LogAnd | LogOr -> raise (TransformedAway("&& and || should have been

```

```

        transformed into a short-circuit ternary expression! Error in the
        following expression:\n" ^ string_of_expr exp))
728 | Divide-> build_simple_binop Llvm.build_fdiv int_builder
729 | Mod-> build_simple_binop Llvm.build_frem int_builder
730 | Pow-> (
731     let powcall numeric_val_1 numeric_val_2 valname b =
732         Llvm.build_call (Hashtbl.find runtime_functions "pow") [|numeric_val_1;
            numeric_val_2|] "" b in
733     build_simple_binop powcall int_builder)
734 | LShift-> build_simple_int_binop Llvm.build_shl int_builder
735 | RShift-> build_simple_int_binop Llvm.build_lshr int_builder
736 | BitOr-> build_simple_int_binop Llvm.build_or int_builder
737 | BitAnd-> build_simple_int_binop Llvm.build_and int_builder
738 | BitXor-> build_simple_int_binop Llvm.build_xor int_builder
739 )
740 | UnOp(SizeOf,expr) -> let vvv = Llvm.const_float base_types.float_t 0.0 in
741     let ret_val = Llvm.build_malloc base_types.value_t "unop_size_ret_val"
        old_builder in
742     let sp = Llvm.build_struct_gep ret_val (value_field_index Number) "num_pointer
        " old_builder in
743     let _ = Llvm.build_store (Llvm.const_int base_types.char_t (
        value_field_flags_index Number)) (Llvm.build_struct_gep ret_val (
        value_field_index Flags) "" old_builder) old_builder in
744     let _ = Llvm.build_store vvv sp old_builder in
745     (ret_val, old_builder)
746 | UnOp(Truthy, expr) ->
747     let ret_val = Llvm.build_malloc base_types.value_t "unop_truthy_ret_val"
        old_builder in
748     let (expr_val, expr_builder) = build_expr old_builder expr in
749
750     let (truthy_bb, falsey_bb, empty_bb, merge_builder) = make_truthiness_blocks "
        binop_eq" ret_val in
751
752     let expr_flags = (expr_val => (value_field_index Flags)) "expr_flags"
        expr_builder in
753     let is_empty_bool = (Llvm.build_icmp Llvm.Icmp.Eq expr_flags (Llvm.const_int
        base_types.flags_t (value_field_flags_index Empty)) "is_empty_bool"
        expr_builder) in
754     let is_empty = Llvm.build_zext is_empty_bool base_types.char_t "is_empty"
        expr_builder in
755     let is_empty_two = Llvm.build_shl is_empty (Llvm.const_int base_types.char_t
        1) "is_empty_two" expr_builder in
756     let is_number = Llvm.build_icmp Llvm.Icmp.Eq expr_flags (Llvm.const_int
        base_types.flags_t (value_field_flags_index Number)) "is_number"
        expr_builder in
757     let the_number = (expr_val => (value_field_index Number)) "the_number"
        expr_builder in
758     let is_zero = Llvm.build_fcmp Llvm.Fcmp.Oeq the_number (Llvm.const_float
        base_types.number_t 0.0) "is_zero" expr_builder in
759     let is_numeric_zero_bool = Llvm.build_and is_zero is_number "
        is_numeric_zero_bool" expr_builder in
760     let is_numeric_zero = Llvm.build_zext is_numeric_zero_bool base_types.char_t "
        is_numeric_zero" expr_builder in
761     let switch_num = Llvm.build_add is_empty_two is_numeric_zero "switch_num"
        expr_builder in
762     let switch_inst = Llvm.build_switch switch_num empty_bb 2 expr_builder in

```

```

763     Llvm.add_case switch_inst (Llvm.const_int base_types.char_t 0) truthy_bb; (*
        empty << 1 + is_zero == 0 ==> truthy *)
764     Llvm.add_case switch_inst (Llvm.const_int base_types.char_t 1) falsey_bb; (*
        empty << 1 + is_zero == 1 ==> falsey *)
765     (ret_val, merge_builder)
766 | UnOp(LogNot, expr) ->
767     let (truth_val, truth_builder) = build_expr old_builder (UnOp(Truthy, expr))
        in
768     let the_number = (truth_val => (value_field_index Number)) "the_number"
        truth_builder in
769     let not_the_number = Llvm.build_fsub (Llvm.const_float base_types.float_t 1.0)
        the_number "not_the_number" truth_builder in
770     let sp = Llvm.build_struct_gep truth_val (value_field_index Number) "
        num_pointer" truth_builder in
771     let _ = Llvm.build_store not_the_number sp truth_builder in
772     (truth_val, truth_builder)
773 | UnOp(Neg, expr) ->
774     let ret_val = Llvm.build_malloc base_types.value_t "unop_truthy_ret_val"
        old_builder in
775     let _ = store_empty ret_val old_builder in
776     let (expr_val, expr_builder) = build_expr old_builder expr in
777     let expr_type = (expr_val => (value_field_index Flags)) "expr_type"
        expr_builder in
778     let is_number = Llvm.build_icmp Llvm.Icmp.Eq expr_type number_type "is_number"
        expr_builder in
779     let (finish_bb, finish_builder) = make_block "finish" in
780
781     let (number_bb, number_builder) = make_block "number" in
782     let the_number = (expr_val => (value_field_index Number)) "the_number"
        number_builder in
783     let minus_the_number = Llvm.build_fneg the_number "minus_the_number"
        number_builder in
784     let _ = store_number ret_val number_builder minus_the_number in
785     let _ = Llvm.build_br finish_bb number_builder in
786
787     let _ = Llvm.build_cond_br is_number number_bb finish_bb expr_builder in
788     (ret_val, finish_builder)
789 | UnOp(BitNot, expr) -> print_endline "Unsupported Unop" ; print_endline (Ast.
        string_of_expr exp); raise NotImplemented
790 | UnOp(TypeOf, expr) ->
791     let (expr_val, expr_builder) = build_expr old_builder expr in
792     let expr_type = (expr_val => (value_field_index Flags)) "expr_type"
        expr_builder in
793     let vp_to_clone_loc = Llvm.build_in_bounds_gep array_of_typeof_val_ptrs [|Llvm
        .const_int base_types.int_t 0; expr_type|] ("vp_to_clone_log") expr_builder
        in
794     let vp_to_clone = Llvm.build_load vp_to_clone_loc "vp_to_clone" expr_builder
        in
795     let ret_val = Llvm.build_call (Hashtbl.find runtime_functions "clone_value")
        [|vp_to_clone|] "typeof_ret_val" expr_builder in
796     (ret_val, expr_builder)
797 | UnOp(Row, expr) -> print_endline "Unsupported Unop" ; print_endline (Ast.
        string_of_expr exp); raise NotImplemented
798 | UnOp(Column, expr) -> print_endline "Unsupported Unop" ; print_endline (Ast.
        string_of_expr exp); raise NotImplemented
799 | ReducedTernary(cond_var, true_var, false_var) ->

```

```

800     let ret_val_addr = LlvM.build_alloca base_types.value_p "tern_ret_val_addr"
      old_builder in
801     let (cond_val, _) = build_expr old_builder (Id(cond_var)) in (* Relying here
      on the fact that Id() doesn't change the builder *)
802     let merge_bb = LlvM.append_block context "merge" form_decl in
803     let merge_builder = LlvM.builder_at_end context merge_bb in
804     let ret_val = LlvM.build_load ret_val_addr "tern_ret_val" merge_builder in
805
806     let truthy_bb = LlvM.append_block context "truthy" form_decl in
807     let truthy_builder = LlvM.builder_at_end context truthy_bb in
808     let (truthy_val, _) = build_expr truthy_builder (Id(true_var)) in (* Relying
      here on the fact that Id() doesn't change the builder *)
809     let _ = LlvM.build_store truthy_val ret_val_addr truthy_builder in
810     let _ = LlvM.build_br merge_bb truthy_builder in
811
812     let falsey_bb = LlvM.append_block context "falsey" form_decl in
813     let falsey_builder = LlvM.builder_at_end context falsey_bb in
814     let (falsey_val, _) = build_expr falsey_builder (Id(false_var)) in (* Relying
      here on the fact that Id() doesn't change the builder *)
815     let _ = LlvM.build_store falsey_val ret_val_addr falsey_builder in
816     let _ = LlvM.build_br merge_bb falsey_builder in
817
818     let empty_bb = LlvM.append_block context "empty" form_decl in
819     let empty_builder = LlvM.builder_at_end context empty_bb in
820     let ret_val_empty = LlvM.build_malloc base_types.value_t "tern_empty"
      empty_builder in
821     let _ = store_empty ret_val_empty empty_builder in
822     let _ = LlvM.build_store ret_val_empty ret_val_addr empty_builder in
823     let _ = LlvM.build_br merge_bb empty_builder in
824
825     let expr_flags = (cond_val => (value_field_index Flags)) "expr_flags"
      old_builder in
826     let is_empty_bool = (LlvM.build_icmp LlvM.Icmp.Eq expr_flags (LlvM.const_int
      base_types.flags_t (value_field_index Empty)) "is_empty_bool"
      old_builder) in
827     let is_empty = LlvM.build_zext is_empty_bool base_types.char_t "is_empty"
      old_builder in
828     let is_empty_two = LlvM.build_shl is_empty (LlvM.const_int base_types.char_t
      1) "is_empty_two" old_builder in
829     let is_number = LlvM.build_icmp LlvM.Icmp.Eq expr_flags (LlvM.const_int
      base_types.flags_t (value_field_index Number)) "is_number"
      old_builder in
830     let the_number = (cond_val => (value_field_index Number)) "the_number"
      old_builder in
831     let is_zero = LlvM.build_fcmp LlvM.Fcmp.Oeq the_number (LlvM.const_float
      base_types.number_t 0.0) "is_zero" old_builder in
832     let is_numeric_zero_bool = LlvM.build_and is_zero is_number "
      is_numeric_zero_bool" old_builder in
833     let is_numeric_zero = LlvM.build_zext is_numeric_zero_bool base_types.char_t "
      is_numeric_zero" old_builder in
834     let switch_num = LlvM.build_add is_empty_two is_numeric_zero "switch_num"
      old_builder in
835     let switch_inst = LlvM.build_switch switch_num empty_bb 2 old_builder in
836     LlvM.add_case switch_inst (LlvM.const_int base_types.char_t 0) truthy_bb; (*
      empty << 1 + is_zero == 0 ==> truthy *)
837     LlvM.add_case switch_inst (LlvM.const_int base_types.char_t 1) falsey_bb; (*

```

```

empty << 1 + is_zero == 1 ==> falsey *)
838   (ret_val, merge_builder)
839   | unknown_expr -> print_endline (string_of_expr unknown_expr);raise
      NotImplemented in
840   let (ret_value_p, final_builder) = build_expr builder_at_top formula_expr in
841   let _ = LlvM.build_ret ret_value_p final_builder in
842   form_decl in
843
844   (*build formula creates a formula declaration in a separate method from the function
      it belongs to*)
845   let build_formula (varname, idx) formula_array element symbols =
846     let storage_addr = LlvM.build_in_bounds_gep formula_array [|LlvM.const_int
      base_types.int_t idx|] "" main_bod in
847     let getStarts = function (* Not really just for starts *)
848       Abs(LitInt(1)) | Abs(LitInt(0)) | DimensionStart | DimensionEnd -> (1, -1)
849     | Abs(Id(s)) ->
850       (match StringMap.find s symbols with
851         LocalVariable(i) | GlobalVariable(i) -> (0, i)
852       | _ -> raise(TransformedAway("Error in " ^ varname ^ ": The LHS expressions
      should always either have dimension length 1 or be the name of a variable
      in their own scope.")))
853     | _ -> print_endline ("Error in " ^ varname ^ " formula number " ^ string_of_int
      idx); raise(LogicError("Something wrong with the index of formula: " ^
      string_of_formula element)) in
854     let getEnds = function
855       Some x -> let (b, c) = getStarts x in (b, c, 0)
856     | None -> (0, -1, 1) in
857     let (fromStartRow, rowStartVarnum) = getStarts element.formula_row_start in
858     let (fromStartCol, colStartVarnum) = getStarts element.formula_col_start in
859     let (toEndRow, rowEndVarnum, isSingleRow) = getEnds element.formula_row_end in
860     let (toEndCol, colEndVarnum, isSingleCol) = getEnds element.formula_col_end in
861
862     let _ = LlvM.build_store (LlvM.const_int base_types.char_t fromStartRow) (LlvM.
      build_struct_gep storage_addr (formula_field_index FromFirstRow) "" main_bod)
      main_bod in
863     let _ = LlvM.build_store (LlvM.const_int base_types.int_t rowStartVarnum) (LlvM.
      build_struct_gep storage_addr (formula_field_index RowStartNum) "" main_bod)
      main_bod in
864     let _ = LlvM.build_store (LlvM.const_int base_types.char_t toEndRow) (LlvM.
      build_struct_gep storage_addr (formula_field_index ToLastRow) "" main_bod)
      main_bod in
865     let _ = LlvM.build_store (LlvM.const_int base_types.int_t rowEndVarnum) (LlvM.
      build_struct_gep storage_addr (formula_field_index RowEndNum) "" main_bod)
      main_bod in
866     let _ = LlvM.build_store (LlvM.const_int base_types.char_t isSingleRow) (LlvM.
      build_struct_gep storage_addr (formula_field_index IsSingleRow) "" main_bod)
      main_bod in
867
868     let _ = LlvM.build_store (LlvM.const_int base_types.char_t fromStartCol) (LlvM.
      build_struct_gep storage_addr (formula_field_index FromFirstCols) "" main_bod)
      main_bod in
869     let _ = LlvM.build_store (LlvM.const_int base_types.int_t colStartVarnum) (LlvM.
      build_struct_gep storage_addr (formula_field_index ColStartNum) "" main_bod)
      main_bod in
870     let _ = LlvM.build_store (LlvM.const_int base_types.char_t toEndCol) (LlvM.
      build_struct_gep storage_addr (formula_field_index ToLastCol) "" main_bod)

```

```

      main_bod in
871   let _ = LlvM.build_store (LlvM.const_int base_types.int_t colEndVarnum) (LlvM.
      build_struct_gep storage_addr (formula_field_index ColEndNum) "" main_bod)
      main_bod in
872   let _ = LlvM.build_store (LlvM.const_int base_types.char_t isSingleCol) (LlvM.
      build_struct_gep storage_addr (formula_field_index IsSingleCol) "" main_bod)
      main_bod in
873
874   let form_decl = build_formula_function (varname, idx) symbols element.formula_expr
      in
875   let _ = LlvM.build_store form_decl (LlvM.build_struct_gep storage_addr (
      formula_field_index FormulaCall) "" main_bod) main_bod in
876   () in
877
878   (* Builds a var_defn struct for each variable *)
879   let build_var_defn defn varname va symbols =
880     let numForm = List.length va.var_formulas in
881     let formulas = LlvM.build_array_malloc base_types.formula_t (LlvM.const_int
      base_types.int_t numForm) "" main_bod in
882     (*getDefn simply looks up the correct definition for a dimension declaration of a
      variable. Note that currently it is ambiguous whether it is a variable or a
      literal. TOOD: consider negative numbers*)
883     let getDefn = function
884       DimId(a) -> (match StringMap.find a symbols with LocalVariable(i) -> i |
      GlobalVariable(i) -> i | _ -> raise(TransformedAway("Error in " ^ varname ^
      ": The LHS expressions should always either have dimension length 1 or be
      the name of a variable in their own scope.")))
885       | DimInt(1) -> 1
886       | DimInt(_) -> print_endline "Non1Dim" ; raise(NotImplemented) in
887     let _ = (match va.var_rows with
888       DimInt(1) -> LlvM.build_store (LlvM.const_int base_types.char_t 1) (LlvM.
      build_struct_gep defn (var_defn_field_index OneByOne) "" main_bod)
      main_bod
889       | DimInt(_) -> print_endline "Non1Dim" ; raise(NotImplemented)
890       | DimId(a) -> (
891         let _ = LlvM.build_store (LlvM.const_int base_types.char_t 0) (LlvM.
      build_struct_gep defn (var_defn_field_index OneByOne) "" main_bod)
      main_bod in ();
892         let _ = LlvM.build_store (LlvM.const_int base_types.int_t (getDefn va.
      var_rows)) (LlvM.build_struct_gep defn (var_defn_field_index Rows) ""
      main_bod) main_bod in ();
893         LlvM.build_store (LlvM.const_int base_types.int_t (getDefn va.var_cols)) (
      LlvM.build_struct_gep defn (var_defn_field_index Cols) "" main_bod)
      main_bod
894       )
895     ) in
896   let _ = LlvM.build_store (LlvM.const_int base_types.int_t numForm) (LlvM.
      build_struct_gep defn (var_defn_field_index NumFormulas) "" main_bod) main_bod
897   and _ = LlvM.build_store formulas (LlvM.build_struct_gep defn (
      var_defn_field_index Formulas) "" main_bod) main_bod
898   and _ = LlvM.build_store (LlvM.build_global_stringptr varname "" main_bod) (LlvM.
      build_struct_gep defn (var_defn_field_index VarName) "" main_bod) main_bod in
899   List.iteri (fun idx elem -> build_formula (varname, idx) formulas elem symbols) va
      .var_formulas in
900
901   (* Creates a scope object and inserts the necessary instructions into main to

```

```

    populate the var_defns, and
902 * into the function specified by builder to populate the scope object. *)
903 let build_scope_obj
904   fname (* The function name, or "globals" *)
905   symbols (* The symbols to use when creating the functions *)
906   vars (* The variables to build definitions and formula-functions for *)
907   static_location_ptr (* The copy of the global pointer used in main *)
908   var_defns_loc (* The copy of the global pointer used in the local function *)
909   num_params (* How many parameters the function takes *)
910   builder (* The LLVM builder for the local function *)
911   =
912   let cardinal = Llvm.const_int base_types.int_t (StringMap.cardinal vars) in
913   let build_var_defns =
914     let static_var_defns = Llvm.build_array_malloc base_types.var_defn_t cardinal (
915       fname ^ "_static_var_defns") main_bod in
916     let _ = Llvm.build_store static_var_defns static_location_ptr main_bod in
917     let add_variable varname va (sm, count) =
918       let fullname = fname ^ "_" ^ varname in
919       let defn = (Llvm.build_in_bounds_gep static_var_defns [|Llvm.const_int
920         base_types.int_t count|] (fullname ^ "_defn") main_bod) in
921       let _ = build_var_defn defn fullname va symbols in
922       (StringMap.add varname count sm, count + 1) in
923     ignore (StringMap.fold add_variable vars (StringMap.empty, 0)) in
924   let var_defns = Llvm.build_load var_defns_loc (fname ^ "_global_defn_ptr_loc")
925     builder in
926   let var_insts = Llvm.build_array_malloc base_types.var_instance_p cardinal "
927     var_insts" builder in
928   let scope_obj = Llvm.build_malloc base_types.extend_scope_t "scope_obj" builder in
929   (*Store variable definition and instance*)
930   let _ = Llvm.build_store var_defns (Llvm.build_struct_gep scope_obj (
931     scope_field_type_index VarDefn) "" builder) builder in
932   let _ = Llvm.build_store var_insts (Llvm.build_struct_gep scope_obj (
933     scope_field_type_index VarInst) "" builder) builder in
934   let _ = Llvm.build_store cardinal (Llvm.build_struct_gep scope_obj (
935     scope_field_type_index VarNum) "" builder) builder in
936   let _ = Llvm.build_store (Llvm.const_int base_types.int_t 0) (Llvm.
937     build_struct_gep scope_obj (scope_field_type_index ScopeRefCount) "" builder)
938     builder in
939   let paramarray = if num_params > 0 then Llvm.build_array_malloc base_types.value_p
940     (Llvm.const_int base_types.int_t num_params) "paramarray" builder else Llvm.
941     const_pointer_null (Llvm.pointer_type base_types.value_p) in
942   let _ = Llvm.build_store paramarray (Llvm.build_struct_gep scope_obj (
943     scope_field_type_index FunctionParams) "" builder) builder in
944   let copy_fn_arg i =
945     let param_addr = Llvm.build_in_bounds_gep paramarray [|Llvm.const_int base_types
946       .int_t i|] (fname ^ "_param_" ^ string_of_int i ^ "_loc") builder in
947     ignore (Llvm.build_store (Llvm.param (StringMap.find fname function_llvalues) i)
948       param_addr builder) in
949   List.iter copy_fn_arg (zero_until num_params);
950   let _ = Llvm.build_call (Hashtbl.find runtime_functions "null_init") [|scope_obj|]
951     "" builder in
952   build_var_defns ; scope_obj in
953 (* End of build_scope_obj *)
954

```

```

942 let build_function fname (fn_def, fn_llvalue) =
943   (* Build the symbol table for this function *)
944   let (local_indices, num_locals) = index_map Locals fn_def.func_body in
945   let add_param (st, idx) param_name =
946     let new_st = StringMap.add param_name (FunctionParameter(idx)) st in
947     (new_st, idx + 1) in
948   let (params_and_globals, _) = List.fold_left add_param (global_symbols, 0) (List.
     map snd fn_def.func_params) in
949   let symbols = StringMap.fold StringMap.add local_indices params_and_globals in
950   let fn_idx = match StringMap.find fname extend_fn_numbers with ExtendFunction(i)
     -> i | _ -> raise(LogicError(fname ^ " not in function table")) in
951   let builder = LlvM.builder_at_end context (LlvM.entry_block fn_llvalue) in
952   let static_location_ptr = LlvM.build_in_bounds_gep array_of_vardefn_ptrs [|LlvM.
     const_int base_types.int_t 0; LlvM.const_int base_types.int_t fn_idx|] (fname ^
     "_global_defn_ptr") main_bod in
953   let var_defns_loc = LlvM.build_in_bounds_gep array_of_vardefn_ptrs [|LlvM.
     const_int base_types.int_t 0; LlvM.const_int base_types.int_t fn_idx|] (fname ^
     "_local_defn_ptr") builder in
954
955   let scope_obj = build_scope_obj fname symbols fn_def.func_body static_location_ptr
     var_defns_loc (List.length fn_def.func_params) builder in
956
957   let ret = snd fn_def.func_ret_val in
958   match ret with
959   | Id(name) ->
960     (
961       match (try StringMap.find name symbols with Not_found -> raise(LogicError("
         Something went wrong with your semantic analysis - " ^ name ^ " not found")
         )) with
962       | LocalVariable(i) ->
963         let llvm_var = LlvM.build_call getVar [|scope_obj; LlvM.const_int base_types
           .int_t i|] "return_variable" builder in
964         let llvm_retval = LlvM.build_call getVal [|llvm_var; LlvM.const_int
           base_types.int_t 0; LlvM.const_int base_types.int_t 0|] "return_value"
           builder in
965         ignore (LlvM.build_ret llvm_retval builder)
966       | _ -> print_endline (string_of_expr ret); raise(TransformedAway("Error in " ^
         fname ^ ": The return value should always have been transformed into a
         local variable"))
967     )
968   | _ -> print_endline (string_of_expr ret); raise(TransformedAway("Error in " ^
     fname ^ ": The return value should always have been transformed into a local
     variable")) in
969 (* End of build_function *)
970
971 (* Build the global scope object *)
972 let vardefn_p_p = LlvM.build_alloca base_types.var_defn_p "v_p_p" main_bod in
973 let global_scope_obj = build_scope_obj "globals" global_symbols globals vardefn_p_p
     vardefn_p_p 0 main_bod in
974 let _ = LlvM.build_store global_scope_obj global_scope_loc main_bod in
975
976 (*iterates over function definitions*)
977 StringMap.iter build_function extend_functions ;
978
979 (* Define the LLVM entry point for the program *)
980 let extend_entry_point = StringMap.find "main" function_llvalues in

```



```

981   let inp = Llvml.build_alloca base_types.value_t "input_arg" main_bod in
982   let _ = Llvml.build_call extend_entry_point (Array.of_list [inp]) "" main_bod in
983   let _ = Llvml.build_ret (Llvml.const_int base_types.int_t 0) main_bod in
984
985   base_module
986
987 let build_this ast_mapped =
988   let modu = (translate ast_mapped) in
989   let _ = Llvml_analysis.assert_valid_module modu in
990   modu

```

7.6 linker.ml

```

1 module StringSet = Set.Make(String)
2 let link xtndOut ast compiler outputFile =
3   let tmpFilenameLL = Filename.temp_file "" ".ll"
4   and tmpFilenameC = Filename.temp_file "" ".o"
5   and getExterns (_,_,extern) =
6     StringSet.elements
7       (Ast.StringMap.fold
8         (fun key value store -> StringSet.add value.Ast.extern_fn_libname store)
9         extern
10        StringSet.empty) in
11   let tmpChan = open_out tmpFilenameLL in
12   output_string tmpChan xtndOut; close_out tmpChan;
13   let call1 = (String.concat " " ("llc-3.8" :: "-filetype=obj" :: tmpFilenameLL :: "-o"
14     " :: tmpFilenameC :: []))
15   and call2 = (String.concat " " (compiler :: "-O -o" :: outputFile :: tmpFilenameC ::
16     (getExterns ast) @ ["runtime.o"])) ^ " -lm" in
17   let rescl = Sys.command call1 in
18   if rescl == 0 then (
19     Sys.remove tmpFilenameLL;
20     let resc2 = Sys.command call2 in
21     Sys.remove tmpFilenameC;
22     if resc2 == 0 then () else raise Not_found
23   )
24   else (Sys.remove tmpFilenameC; raise Not_found)

```

7.7 main.ml

```

1 open Ast;;
2
3 let print_ast = ref false
4 let interpret_ast = ref false
5 let compile_ast = ref false
6 let link = ref false
7 let output = ref "./out"
8 let compiler = ref "gcc"
9
10 let the_ast = ref (StringMap.empty, StringMap.empty, StringMap.empty)
11 let just_one_please = ref false
12
13 let speclist = [
14   ("-p", Arg.Set print_ast, "Print the AST");
15   ("-i", Arg.Set interpret_ast, "Interpret the program");

```

```

16         ("-c", Arg.Set compile_ast, "Compile the program");
17         ("-l", Arg.Set link, "Link the program");
18     ]
19
20     let usage_message = "Welcome to Extend!\n\nUsage: extend <options> <source-file>\n\nOptions are:"
21
22     let parse_ast filename =
23         if !just_one_please
24         then print_endline "Any files after the first one are ignored."
25         else just_one_please := true ; the_ast := (Transform.create_ast filename);;
26
27     Arg.parse speclist parse_ast usage_message;
28     if not !just_one_please then Arg.usage speclist usage_message else ();
29     if !print_ast then print_endline (string_of_program !the_ast) else ();
30     if !interpret_ast then (Interpreter.interpret !the_ast; ()) else ();
31     if !compile_ast then
32         let compiled = (Llvm.string_of_llmodule (Codegen.translate !the_ast))
33         in
34             if not (!link) then print_endline compiled
35             else Linker.link compiled !the_ast !compiler !output
36     else ();

```

7.8 lib.c

```

1  #include<stdio.h>
2  #include<stdlib.h>
3  #include<math.h>
4  #include<string.h>
5  #include<stdbool.h>
6  /* #include <sys/time.h> */
7  #include <time.h>
8  #include "runtime.h"
9
10 #define MAX_FILES 255
11 FILE *open_files[1 + MAX_FILES] = {NULL};
12 int open_num_files = 0;
13
14 value_p print(value_p whatever, value_p text) {
15     if(!assertSingleString(text)) return new_val();
16     if(!assertText(text)) return new_val();
17     printf("%s", text->str->text);
18     return new_val();
19 }
20
21 value_p printv(value_p whatever, value_p text) {
22     printf("%s", text->str->text);
23     return new_val();
24 }
25
26 value_p printd(value_p whatever, value_p text) {
27     printf("%f\n", text->numericVal);
28     value_p result = malloc(sizeof(struct value_t));
29     return result;
30 }

```

```

31
32 value_p to_string(value_p val) {
33     if(assertSingleNumber(val)) {
34         double possible_num = val->numericVal;
35         int rounded_int = (int) lrint(possible_num);
36         char *converted_str;
37         if (fabs(possible_num - rounded_int) < FLOAT_CUTOFF) {
38             int size = snprintf(NULL, 0, "%d", rounded_int);
39             converted_str = malloc(size + 1);
40             sprintf(converted_str, "%d", rounded_int);
41         } else {
42             int size = snprintf(NULL, 0, "%f", possible_num);
43             converted_str = malloc(size + 1);
44             sprintf(converted_str, "%f", possible_num);
45         }
46         value_p result = box_value_string(new_string(converted_str));
47         return result;
48     }
49     else if(assertSingleString(val)) return val;
50     else if(val->flags == FLAG_EMPTY) {
51         value_p _new = new_val();
52         setString(_new, "empty", 5);
53         return _new;
54     }
55     // If the struct does not hold a string or number, return empty?
56     return new_val();
57 }
58
59 #define FUNC(name) value_p extend_##name(value_p a){if(!assertSingleNumber(a)) return
    new_val();double val = name(a->numericVal);return new_number(val);}
60 FUNC(sin)
61 FUNC(cos)
62 FUNC(tan)
63 FUNC(acos)
64 FUNC(asin)
65 FUNC(atan)
66 FUNC(sinh)
67 FUNC(cosh)
68 FUNC(tanh)
69 FUNC(exp)
70 FUNC(log)
71 FUNC(log10)
72 FUNC(sqrt)
73 FUNC(ceil)
74 FUNC(fabs)
75 FUNC(floor)
76
77 value_p extend_get_stdin() {
78     if (open_num_files + 1 > MAX_FILES) {
79         return new_val();
80     } else {
81         open_num_files++;
82         open_files[open_num_files] = stdin;
83         return new_number((double) open_num_files);
84     }
85 }

```

```

86
87 value_p extend_get_stdout() {
88     if (open_num_files + 1 > MAX_FILES) {
89         return new_val();
90     } else {
91         open_num_files++;
92         open_files[open_num_files] = stdout;
93         return new_number((double) open_num_files);
94     }
95 }
96
97 value_p extend_get_stderr() {
98     if (open_num_files + 1 > MAX_FILES) {
99         return new_val();
100     } else {
101         open_num_files++;
102         open_files[open_num_files] = stderr;
103         return new_number((double) open_num_files);
104     }
105 }
106
107 value_p extend_open(value_p filename, value_p mode){
108     FILE *val;
109     if ( !assertSingleString(filename)
110         || !assertSingleString(mode)
111         || open_num_files + 1 > MAX_FILES) {
112         return new_val();
113     }
114     val = fopen(filename->str->text, mode->str->text);
115     if(val == NULL) return new_val();
116     open_num_files++;
117     open_files[open_num_files] = val;
118     return new_number((double) open_num_files);
119 }
120
121 value_p extend_close(value_p file_handle) {
122     if(!assertSingleNumber(file_handle)) {
123         // Per the LRM this is actually supposed to crash the program.
124         fprintf(stderr, "EXITING - Attempted to close something that was not a valid file
125             pointer\n");
126         exit(-1);
127     }
128     int fileNum = (int) file_handle->numericVal;
129
130     if (fileNum > open_num_files || open_files[fileNum] == NULL) {
131         // Per the LRM this is actually supposed to crash the program.
132         fprintf(stderr, "EXITING - Attempted to close something that was not a valid file
133             pointer\n");
134         exit(-1);
135     }
136     fclose(open_files[fileNum]);
137     open_files[fileNum] = NULL; // Empty the container for the pointer.
138     return new_val(); // assuming it was an open valid handle, close() is just supposed
        to return empty

```

```

139 value_p extend_read(value_p file_handle, value_p num_bytes){
140     if(!assertSingleNumber(file_handle) || !assertSingleNumber(num_bytes)) return
        new_val();
141     int max_bytes;
142     int fileNum = (int) file_handle->numericVal;
143     if (fileNum > open_num_files || open_files[fileNum] == NULL) return new_val();
144     FILE *f = open_files[fileNum];
145     max_bytes = (int) num_bytes->numericVal;
146     if (max_bytes == 0) {
147         long cur_pos = ftell(f);
148         fseek(f, 0, SEEK_END);
149         long end_pos = ftell(f);
150         fseek(f, cur_pos, SEEK_SET);
151         max_bytes = end_pos - cur_pos;
152     }
153     char *buf = malloc(sizeof(char) * (max_bytes + 1));
154     int bytes_read = fread(buf, sizeof(char), max_bytes, f);
155     buf[bytes_read] = 0;
156     value_p result = box_value_string(new_string(buf));
157     free(buf);
158     return result;
159     //edge case: how to return the entire contents of the file if n == empty?
160 }
161
162 value_p extend_readline(value_p file_handle) {
163     int i=0, buf_size = 256;
164     char next_char;
165     if (!assertSingleNumber(file_handle)) return new_val();
166     int fileNum = (int) file_handle->numericVal;
167     FILE *f = open_files[fileNum];
168     if (fileNum > open_num_files || open_files[fileNum] == NULL) {
169         return new_val();
170     }
171     char *buf = (char *) malloc (buf_size * sizeof(char));
172     while ((next_char = fgetc(f)) != '\n') {
173         buf[i++] = next_char;
174         if (i == buf_size - 2) {
175             buf_size *= 2;
176             char *new_buf = (char *) malloc (buf_size * sizeof(char));
177             memcpy(new_buf, buf, i);
178             free(buf);
179             buf = new_buf;
180         }
181     }
182     buf[i] = '\0';
183     value_p result = box_value_string(new_string(buf));
184     free(buf);
185     return result;
186 }
187
188 value_p extend_write(value_p file_handle, value_p buffer){
189     if(!assertSingleNumber(file_handle) || !assertSingleString(buffer)) return new_val()
        ;
190     int fileNum = (int) file_handle->numericVal;
191     if (fileNum > open_num_files || open_files[fileNum] == NULL) {
192         // Per the LRM this is actually supposed to crash the program.

```

```

193     fprintf(stderr, "EXITING - Attempted to write to something that was not a valid
        file pointer\n");
194     exit(-1);
195 }
196 fwrite(buffer->str->text, 1, buffer->str->length, open_files[fileNum]);
197 // TODO: make this return empty once compiler handles Id(s)
198 // RN: Use the return value to close the file
199 return new_number((double) fileNum);
200 }
201
202 value_p extend_current_hour() {
203     time_t ltime;
204     struct tm info;
205     ltime = time(&ltime);
206     localtime_r(&ltime, &info);
207     return new_number((double) info.tm_hour);
208 }

```

7.9 runtime.c

```

1  #include<stdio.h>
2  #include<stdlib.h>
3  #include<math.h>
4  #include<string.h>
5  #include<stdbool.h>
6  #include "runtime.h"
7
8  void debug_print(value_p val, char *which_value) {
9      char *flag_meanings[4] = {"Empty", "Number", "String", "Subrange"};
10     fprintf(stderr, "-----Everything you ever wanted to know about %s:-----\n",
        which_value == NULL ? "some anonymous variable" : which_value);
11     fprintf(stderr, "Memory address: %p\n", val);
12     if (val == NULL) {
13         fprintf(stderr, "-----Nice try asking me to dereference a null pointer\n
            -----");
14         return;
15     }
16     fprintf(stderr, "Flags: %d (%s)\n", val->flags, flag_meanings[val->flags]);
17     fprintf(stderr, "NumericVal: %f\n", val->numericVal);
18     fprintf(stderr, "String contents: Probably safer not to check that pointer (%p)
        blindly\n", val->str);
19     if (val->flags == FLAG_STRING && val->str != NULL) {
20         printf(stderr, "It says it's a string and it's not a NULL pointer though, so here
            you go:\n");
21         fprintf(stderr, "String refcount: %d\n", val->str->refs);
22         fprintf(stderr, "String length: %ld\n", val->str->length);
23         fprintf(stderr, "String char* memory address: %p\n", val->str->text);
24         if (val->str->text == NULL) {
25             fprintf(stderr, "Not going to print the contents of NULL!\n");
26         } else {
27             fprintf(stderr, "String char* contents:\n%s\n", val->str->text);
28         }
29     }
30     fprintf(stderr, "Subrange contents: Probably safer not to check that pointer (%p)
        blindly either\n", val->subrange);

```

```

31     fprintf(stderr, "———That's all I've got to say about %s:———\n", which_value ==
        NULL ? "some anonymous variable" : which_value);
32 }
33
34 void debug_print_formula(struct ExtendFormula *fdef) {
35     fprintf(stderr, "———Everything you ever wanted to know about your favorite
        formula:———\n");
36     fprintf(stderr, "RowStart varnum: %d %d\n", fdef->rowStart_varnum, fdef->
        fromFirstRow);
37     fprintf(stderr, "RowEnd varnum: %d %d\n", fdef->rowEnd_varnum, fdef->toLastRow);
38     fprintf(stderr, "ColStart varnum: %d %d\n", fdef->colStart_varnum, fdef->
        fromFirstCol);
39     fprintf(stderr, "ColEnd varnum: %d %d\n", fdef->colEnd_varnum, fdef->toLastCol);
40 }
41
42 void debug_print_res_formula(struct ResolvedFormula *rdef) {
43     fprintf(stderr, "Some formula with function pointer %p applies to: [%d:%d,%d:%d]\n",
        rdef->formula, rdef->rowStart, rdef->rowEnd, rdef->colStart, rdef->colEnd);
44 }
45
46 void debug_print_vardefn(struct var_defn *pdef) {
47     fprintf(stderr, "———Everything you ever wanted to know about var defn %s:———\n
        ", pdef->name);
48     fprintf(stderr, "Row varnum: %d\n", pdef->rows_varnum);
49     fprintf(stderr, "Col varnum: %d\n", pdef->cols_varnum);
50     fprintf(stderr, "Num formulas: %d\n", pdef->numFormulas);
51     fprintf(stderr, "Formula defs: \n");
52     int i;
53     for (i=0; i < pdef->numFormulas; i++) {
54         debug_print_formula(pdef->formulas + i);
55     }
56     fprintf(stderr, "Is 1x1: %d\n", pdef->isOneByOne);
57 }
58
59 void debug_print_varinst(struct var_instance *inst) {
60     fprintf(stderr, "———Everything you ever wanted to know about var %s:———\n",
        inst->name);
61     fprintf(stderr, "Rows: %d\n", inst->rows);
62     fprintf(stderr, "Cols: %d\n", inst->cols);
63     fprintf(stderr, "Num formulas: %d\n", inst->numFormulas);
64     fprintf(stderr, "*****Formulas:*****\n");
65     int i;
66     for (i = 0; i < inst->numFormulas; i++) {
67         debug_print_res_formula(inst->formulas + i);
68     }
69     fprintf(stderr, "**** End of Formulas *** \n");
70     fprintf(stderr, "~~~~~Cells:~~~~~\n");
71     for (i = 0; i < inst->rows * inst->cols; i++) {
72         printf("%s[%d,%d]: Status=%d\n", inst->name, i / inst->cols, i % inst->cols, inst
            ->status[i]);
73         if (inst->status[i] == CALCULATED) {
74             printf("%s[%d,%d] Value:\n", inst->name, i / inst->cols, i % inst->cols);
75             debug_print(inst->values[i], inst->name);
76         }
77     }
78     fprintf(stderr, "~~~ End of Cells: ~~~\n");

```

```

79 }
80
81 double setNumeric(value_p result, double val) {
82     result->flags = FLAG_NUMBER;
83     return (result->numericVal = val);
84 }
85
86 char* setString(value_p result, char *str, int length) {
87     result->flags = FLAG_STRING;
88     result->str = malloc(sizeof(struct string_t));
89     result->str->length = length;
90     return (result->str->text = str);
91 }
92
93 double setFlag(value_p result, double flag_num) {
94     return (result->flags = flag_num);
95 }
96
97 int assertSingle(value_p value) {
98     /* TODO: dereference 1 by 1 subrange */
99     return !(value->flags == FLAG_SUBRANGE);
100 }
101
102 int assertSingleNumber(value_p p) {
103     if (!assertSingle(p)) {
104         return 0;
105     }
106     return (p->flags == FLAG_NUMBER);
107 }
108
109 int assertText(value_p my_val) {
110     return (my_val->flags == FLAG_STRING);
111 }
112
113 int assertSingleString(value_p p) {
114     if (!assertSingle(p)) {
115         return 0;
116     }
117     return (p->flags == FLAG_STRING);
118 }
119
120 int assertEmpty(value_p p) {
121     if (!assertSingle(p)) {
122         return 0;
123     }
124     return (p->flags == FLAG_EMPTY);
125 }
126
127 value_p new_val() {
128     value_p empty_val = malloc(sizeof(struct value_t));
129     setFlag(empty_val, FLAG_EMPTY);
130     return empty_val;
131 }
132
133 value_p new_number(double val) {
134     value_p new_v = malloc(sizeof(struct value_t));

```



```

135     setFlag(new_v, FLAG_NUMBER);
136     setNumeric(new_v, val);
137     return new_v;
138 }
139
140 value_p new_string_go_all_the_way(char *s) {
141     if (s == NULL) return new_val();
142     value_p new_v = malloc(sizeof(struct value_t));
143     setFlag(new_v, FLAG_STRING);
144     string_p new_str = malloc(sizeof(struct string_t));
145     long len = strlen(s);
146     new_str->text = malloc(len+1);
147     strcpy(new_str->text, s);
148     new_str->length = len;
149     new_str->refs = 1;
150     new_v->str = new_str;
151     return new_v;
152 }
153
154 struct ExtendScope *global_scope;
155
156 void null_init(struct ExtendScope *scope_ptr) {
157     int i;
158     for(i = 0; i < scope_ptr->numVars; i++)
159         scope_ptr->vars[i] = NULL;
160 }
161
162 int getIntFromOneByOne(struct ExtendScope *scope_ptr, int varnum) {
163     struct var_instance *inst = get_variable(scope_ptr, varnum);
164     if (inst->rows != 1 || inst->cols != 1) {
165         fprintf(stderr, "The variable you claimed (%s) was one by one is actually %d by %d\n",
166             inst->name, inst->rows, inst->cols);
167         debug_print_varinst(inst);
168         exit(-1);
169     }
170     value_p val = getVal(inst, 0, 0);
171     if (!assertSingleNumber(val)) {
172         fprintf(stderr, "The variable you claimed (%s) was a number isn't.\n", inst->name);
173         debug_print(val, inst->name);
174         exit(-1);
175     }
176     return (int) lrint(val->numericVal);
177 }
178
179 struct var_instance *instantiate_variable(struct ExtendScope *scope_ptr, struct
180     var_defn def) {
181     struct var_instance *inst = malloc(sizeof(struct var_instance));
182     if(def.isOneByOne) {
183         inst->rows = 1;
184         inst->cols = 1;
185     } else {
186         inst->rows = getIntFromOneByOne(scope_ptr, def.rows_varnum);
187         inst->cols = getIntFromOneByOne(scope_ptr, def.cols_varnum);
188     }
189     // TODO: do the same thing for each FormulaFP to turn an ExtendFormula into a

```

```

    ResolvedFormula
188 inst->numFormulas = def.numFormulas;
189 inst->closure = scope_ptr;
190 inst->name = def.name;
191 int size = inst->rows * inst->cols;
192 inst->values = malloc(sizeof(value_p) * size);
193 memset(inst->values, 0, sizeof(value_p) * size);
194 inst->status = malloc(sizeof(char) * size);
195 memset(inst->status, 0, sizeof(char) * size);
196 inst->formulas = malloc(sizeof(struct ResolvedFormula) * inst->numFormulas);
197 //debug_print_vardefn(&def);
198 //debug_print_varinst(inst);
199 int i;
200 for(i = 0; i < inst->numFormulas; i++) {
201
202     // Set the formula function pointer to the pointer from the definition
203     inst->formulas[i].formula = def.formulas[i].formula;
204
205     if (def.isOneByOne) {
206         inst->formulas[i].rowStart = 0;
207         inst->formulas[i].rowEnd = 1;
208         inst->formulas[i].colStart = 0;
209         inst->formulas[i].colEnd = 1;
210     } else {
211         if(def.formulas[i].fromFirstRow) {
212             inst->formulas[i].rowStart = 0;
213         } else {
214             inst->formulas[i].rowStart = getIntFromOneByOne(scope_ptr, def.formulas[i].
                rowStart_varnum);
215             if (inst->formulas[i].rowStart < 0) {
216                 inst->formulas[i].rowStart += inst->rows;
217             }
218             if (inst->formulas[i].rowStart < 0 || inst->formulas[i].rowStart >= inst->rows
                ) {
219                 //Doesn't matter, but will never get called
220             }
221         }
222         if (def.formulas[i].isSingleRow) {
223             inst->formulas[i].rowEnd = inst->formulas[i].rowStart + 1;
224         } else if (def.formulas[i].toLastRow) {
225             inst->formulas[i].rowEnd = inst->rows;
226         } else {
227             inst->formulas[i].rowEnd = getIntFromOneByOne(scope_ptr, def.formulas[i].
                rowEnd_varnum);
228         }
229         if(def.formulas[i].fromFirstCol) {
230             inst->formulas[i].colStart = 0;
231         } else {
232             inst->formulas[i].colStart = getIntFromOneByOne(scope_ptr, def.formulas[i].
                colStart_varnum);
233             if (inst->formulas[i].colStart < 0) {
234                 inst->formulas[i].colStart += inst->cols;
235             }
236             if (inst->formulas[i].colStart < 0 || inst->formulas[i].colStart >= inst->cols
                ) {
237                 //Doesn't matter, but will never get called

```

```

238     }
239 }
240 if (def.formulas[i].isSingleCol) {
241     inst->formulas[i].colEnd = inst->formulas[i].colStart + 1;
242 } else if (def.formulas[i].toLastCol) {
243     inst->formulas[i].colEnd = inst->cols;
244 } else {
245     inst->formulas[i].colEnd = getIntFromOneByOne(scope_ptr, def.formulas[i].
        colEnd_varnum);
246 }
247 }
248 }
249
250 scope_ptr->refcount++;
251 return inst;
252 }
253
254 struct var_instance *get_variable(struct ExtendScope *scope_ptr, int varnum) {
255     if (varnum >= scope_ptr->numVars) {
256         fprintf(stderr, "Runtime error: Asked for nonexistant variable number\n");
257         exit(-1);
258     }
259     if (scope_ptr->vars[varnum] == NULL) {
260         scope_ptr->vars[varnum] = instantiate_variable(scope_ptr, scope_ptr->defns[varnum]
            );
261     }
262     return scope_ptr->vars[varnum];
263 }
264
265 char assertInBounds(struct var_instance *defn, int r, int c) {
266     return (
267         r >= 0 && r < defn->rows &&
268         c >= 0 && c < defn->cols
269     );
270 }
271
272 value_p calcVal(struct var_instance *inst, int r, int c) {
273     int i;
274     for (i = 0; i < inst->numFormulas; i++) {
275         if (
276             r >= inst->formulas[i].rowStart && r < inst->formulas[i].rowEnd &&
277             c >= inst->formulas[i].colStart && c < inst->formulas[i].colEnd
278         ) {
279             return (inst->formulas[i].formula)(inst->closure, r, c);
280         }
281     }
282     return new_val();
283 }
284
285 void setRange(value_p val, struct var_instance *inst) {
286     subrange_p sr = malloc(sizeof(struct subrange_t));
287     sr->offsetCol = 0;
288     sr->offsetRow = 0;
289     sr->subrangeCol = inst->cols;
290     sr->subrangeRow = inst->rows;
291     sr->range = inst;

```

```

292     val->subrange = sr;
293     val->flags = FLAG_SUBRANGE;
294 }
295
296 value_p getSize(struct var_instance *inst) {
297     value_p res = malloc(sizeof(struct value_t));
298     setNumeric(res, 1); /*TODO*/
299     return res;
300 }
301
302 value_p deepCopy(value_p value) {
303     value_p _new = new_val();
304     if(value->flags == FLAG_EMPTY) {}
305     else if(value->flags == FLAG_STRING) {
306         _new->flags = FLAG_STRING;
307         _new->str = malloc(sizeof(struct string_t));
308         memcpy(_new->str->text, value->str->text, value->str->length);
309         _new->str->length = value->str->length;
310     }
311     else if(value->flags == FLAG_NUMBER) {
312         _new->flags = FLAG_NUMBER;
313         _new->numericVal = value->numericVal;
314     }
315     else if(value->flags == FLAG_SUBRANGE) {
316         struct var_instance *v = malloc(sizeof(struct subrange_t));
317         int cols = value->subrange->subrangeCol;
318         int rows = value->subrange->subrangeRow;
319         v->name = "COPYCAT";
320         v->formulas = NULL;
321         v->status = malloc(sizeof(char *) * rows * cols);
322         v->values = malloc(sizeof(value_p) * rows * cols);
323         v->closure = NULL;
324         int i, j;
325         for(i = 0; i < rows; i++) {
326             for(j = 0; j < cols; j++) {
327                 int offset = i * rows + j;
328                 *(v->status + offset) = CALCULATED;
329                 /*TODO: eval lazzzy*/
330                 *(v->values + offset) = getVal(value->subrange->range, i + value->subrange->
                    offsetRow, j + value->subrange->offsetCol);
331             }
332         }
333         setRange(_new, v);
334     }
335     return _new;
336 }
337
338 value_p clone_value(value_p old_value) {
339     value_p new_value = (value_p) malloc(sizeof(struct value_t));
340     new_value->flags = old_value->flags;
341     switch (new_value->flags) {
342         case FLAG_EMPTY:
343             break;
344         case FLAG_NUMBER:
345             new_value->numericVal = old_value->numericVal;
346             break;

```

```

347     case FLAG_STRING:
348         new_value->str = old_value->str;
349         new_value->str->refs++;
350         break;
351     case FLAG_SUBRANGE:
352         new_value->subrange = (subrange_p) malloc(sizeof(struct subrange_t));
353         memcpy(new_value->subrange, old_value->subrange, sizeof(struct subrange_t));
354         new_value->subrange->range->closure->refcount++; /* Not sure about this one */
355         break;
356     default:
357         fprintf(stderr, "clone_value(%p): Illegal value of flags: %c\n", old_value,
358                 new_value->flags);
359         exit(-1);
360         break;
361 }
362 return new_value;
363 }
364 void delete_string_p(string_p old_string) {
365     old_string->refs--;
366     if (old_string->refs == 0) {
367         /* free(old_string); */
368     }
369 }
370
371 void delete_subrange_p(subrange_p old_subrange) {
372     old_subrange->range->closure->refcount--;
373     free(old_subrange);
374 }
375
376 void delete_value(value_p old_value) {
377     switch (old_value->flags) {
378     case FLAG_EMPTY:
379         break;
380     case FLAG_NUMBER:
381         break;
382     case FLAG_STRING:
383         delete_string_p(old_value->str); /* doesn't do anything besides decrement the
384             ref count now */
385         break;
386     case FLAG_SUBRANGE:
387         delete_subrange_p(old_value->subrange);
388         break;
389     default:
390         fprintf(stderr, "delete_value(%p): Illegal value of flags: %c\n", old_value,
391                 old_value->flags);
392         exit(-1);
393         break;
394     }
395 }
396
397 value_p getVal(struct var_instance *inst, int r, int c) {
398     /* If we're going to return new_val() then we have to
399     * do clone_value(). Otherwise the receiver won't know
400     * whether or not they can free the value_p they get back.
401     * I think this should return, dangerously, return NULL if it's

```

```

400     * invalid, and the callers will have to be careful to check the value.
401     * The alternative is to always clone_value - safer, but much slower
402     * and makes our memory issues even bigger.
403     * Right now there are only a few places that call this. */
404
405     if(!assertInBounds(inst, r, c)) return NULL;
406     int cell_number = r * inst->cols + c;
407     char cell_status = inst->status[cell_number];
408     switch(cell_status) {
409         case NEVER_EXAMINED:
410             inst->status[cell_number] = IN_PROGRESS;
411             inst->values[cell_number] = calcVal(inst, r, c);
412             inst->status[cell_number] = CALCULATED;
413             break;
414         case IN_PROGRESS:
415             fprintf(stderr, "EXITING - Circular reference in %s[%d,%d]\n", inst->name, r, c)
416                 ;
417             exit(-1);
418             break;
419         case CALCULATED:
420             if (inst->values[cell_number] == NULL) {
421                 fprintf(stderr, "Supposedly, %s[%d,%d] was already calculated, but there is a
422                     null pointer there.\n", inst->name, r, c);
423                 fprintf(stderr, "Attempting to print contents of the variable instance where
424                     this occurred:\n");
425                 fflush(stdout);
426                 debug_print_varinst(inst);
427                 exit(-1);
428             }
429             break;
430         default:
431             fprintf(stderr, "Unrecognized cell status %d (row %d, col %d)!\n", cell_status,
432                 r, c);
433             fprintf(stderr, "Attempting to print contents of the variable instance where
434                 this occurred:\n");
435             fflush(stdout);
436             debug_print_varinst(inst);
437             exit(-1);
438             break;
439     }
440     return inst->values[cell_number];
441 // char *status = inst->status + offset;
442 // value_p return_val;
443 // if(*status & IN_PROGRESS) {
444 //     /* TODO: Circular dependency. Possibly throw? */
445 //     return_val = new_val();
446 // } else if ((~(*status)) & CALCULATED) { /* value not calculated */
447 //     value_p val = calcVal(inst, x, y);
448 //     inst->values[offset] = val;
449 //     *status = (*status && !IN_PROGRESS) | CALCULATED;
450 //     return_val = val;
451 // } else {
452 //     return_val = inst->values[offset];
453 // }
454 // while(return_val->flags == FLAG_SUBRANGE && return_val->subrange->subrangeRow == 1
455 //     && return_val->subrange->subrangeCol == 1) {

```

```

450 //      return_val = getVal(return_val->subrange->range, return_val->subrange->offsetRow
      , return_val->subrange->offsetCol);
451 //  }
452 //  // debug_print_varinst(inst);
453 //  return return_val;
454 }

```

7.10 stdlib.xtnd

```

1  extern "stdlib.o" {
2      extend_get_stdin();
3      extend_get_stdout();
4      extend_get_stderr();
5      extend_readline(file_handle);
6      extend_write(file_handle, str);
7      extend_current_hour();
8      to_string(x);
9  }
10
11 global STDIN := extend_get_stdin();
12 global STDOUT := extend_get_stdout();
13 global STDERR := extend_get_stderr();

```

8. Tests and Output

helloworld.xtnd

```
1 extern "stdlib.o" {
2   print(a,b);
3   printv(a,b);
4 }
5
6 [1,1] main(args) {
7   foo := printv(1,"Hello World\n") -> 0;
8   return foo;
9 }
```

helloworld.xtnd - Expected Output

```
1 Hello World
```

test-access-cell.xtnd

```
1 extern "stdlib.o" {
2   print(a,b);
3   to_string(a);
4 }
5
6 main([1,n] args) {
7   [2,2] foo := "string";
8   bar := foo[1,1];
9   return print(1,to_string(bar)) -> print(1, "\n") -> 0;
10 }
```

test-access-cell.xtnd - Expected Output

```
1 string
```

test-access-column-cell.xtnd

```
1 extern "stdlib.o" {
2   print(a,b);
3   to_string(a);
4 }
5
6 main([1,n] args) {
7   [4,1] foo := "string";
8   return print(1,to_string( foo[1,0]))+"\n") -> 0;
9 }
```

test-access-column-cell.xtnd - Expected Output


```
1 string
```

test-access-hashtag-multi-dim.xtnd

```
1 extern "stdlib.o" {
2     print(a,b);
3     to_string(a);
4 }
5
6 main([1,n] args) {
7     [4,4] foo := "string";
8     return print(1,to_string( #foo)+"\n") -> 0;
9 }
```

test-access-hashtag-multi-dim.xtnd - Expected Output

```
1 string
```

test-access-hashtag-single-dim.xtnd

```
1 extern "stdlib.o" {
2     print(a,b);
3     to_string(a);
4 }
5
6 main([1,n] args) {
7     [1,1] foo := "string";
8     return print(1,to_string( #foo)+"\n") -> 0;
9 }
```

test-access-hashtag-single-dim.xtnd - Expected Output

```
1 string
```

test-access-relative-range.xtnd

```
1 extern "stdlib.o" {
2     print(a,b);
3     to_string(a);
4 }
5
6 main([1,n] args) {
7     [4,4] foo := "string";
8     return print(1,to_string( foo[, [1]])+"\n") -> 0;
9 }
```

test-access-relative-range.xtnd - Expected Output

```
1 string
```

test-acos.xtnd

```
1 extern "stdlib.o" {
2     extend_acos(a);
3     printd(a,b);
4 }
5
6 [1,1] main(args) {
7     return printd(1, extend_acos(0.0)) -> 0;
8 }
```

test-acos.xtnd - Expected Output

```
1 1.570796
```

test-addition.xtnd

```
1 extern "stdlib.o" {
2     print(a,b);
3     to_string(a);
4 }
5
6 main([1,1] args){
7     return print(1,to_string( 5 + 7)+"\n") -> 0;
8 }
```

test-addition.xtnd - Expected Output

```
1 12
```

test-addition-empty.xtnd

```
1 extern "stdlib.o" {
2     print(a,b);
3     to_string(a);
4 }
5
6 main([1,1] args){
7     return print(1,to_string( empty + 5)+"\n") -> 0;
8 }
```

test-addition-empty.xtnd - Expected Output

```
1 empty
```

test-asin.xtnd

```
1 extern "stdlib.o" {
2     extend_asin(a);
3     printd(a,b);
4 }
5
6 [1,1] main([1,n] args) {
7     return printd(1, extend_asin(0.5)) -> 0;
8 }
```

test-asin.xtnd - Expected Output

```
1 0.523599
```

test-atan.xtnd

```
1 extern "stdlib.o" {
2     extend_atan(a);
3     printd(a,b);
4 }
5
6 [1,1] main([1,n] args) {
7     return printd(1, extend_atan(45.0)) -> 0;
8 }
```

test-atan.xtnd - Expected Output

```
1 1.548578
```

test-basic-func.xtnd

```
1 extern "stdlib.o" {
2   print(a,b);
3   to_string(a);
4 }
5
6 main([1,n] args) {
7   foo := 2;
8   bar := 3;
9   foobar := foo + bar;
10  return print(1,to_string( 0)+"\n") -> 0;
11 }
```

test-basic-func.xtnd - Expected Output

```
1 0
```

test-bitwise-and.xtnd

```
1 extern "stdlib.o" {
2   print(a,b);
3   to_string(a);
4 }
5
6 main([1,1] args){
7   return print(1,to_string( 23 & 12)+"\n") -> 0;
8 }
```

test-bitwise-and.xtnd - Expected Output

```
1 4
```

test-bitwise-and-empty.xtnd

```
1 extern "stdlib.o" {
2   print(a,b);
3   to_string(a);
4 }
5
6 main([1,1] args){
7   return print(1,to_string( empty & 4)+"\n") -> 0;
8 }
```

test-bitwise-and-empty.xtnd - Expected Output

```
1 empty
```

test-bitwise-left.xtnd

```
1 extern "stdlib.o" {
2   print(a,b);
3   to_string(a);
4 }
5
6 main([1,1] args){
7   return print(1,to_string( 14 << 2)+"\n") -> 0;
8 }
```

test-bitwise-left.xtnd - Expected Output

```
1 56
```

test-bitwise-left-empty.xtnd

```
1 extern "stdlib.o" {  
2     print(a,b);  
3     to_string(a);  
4 }  
5  
6 main([1,1] args){  
7     return print(1,to_string( empty >> 1)+"\n") -> 0;  
8 }
```

test-bitwise-left-empty.xtnd - Expected Output

```
1 empty
```

test-bitwise-or.xtnd

```
1 extern "stdlib.o" {  
2     print(a,b);  
3     to_string(a);  
4 }  
5  
6 main([1,1] args){  
7     return print(1,to_string( 14 | 12)+"\n") -> 0;  
8 }
```

test-bitwise-or.xtnd - Expected Output

```
1 14
```

test-bitwise-or-empty.xtnd

```
1 extern "stdlib.o" {  
2     print(a,b);  
3     to_string(a);  
4 }  
5  
6 main([1,1] args){  
7     return print(1,to_string( empty | 2)+"\n") -> 0;  
8 }
```

test-bitwise-or-empty.xtnd - Expected Output

```
1 empty
```

test-bitwise-right.xtnd

```
1 extern "stdlib.o" {  
2     print(a,b);  
3     to_string(a);  
4 }  
5  
6 main([1,1] args){  
7     return print(1,to_string( 12 >> 2)+"\n") -> 0;  
8 }
```

test-bitwise-right.xtnd - Expected Output

```
1 3
```

test-bitwise-right-empty.xtnd

```
1 extern "stdlib.o" {
2     print(a,b);
3     to_string(a);
4 }
5
6 main([1,1] args){
7     return print(1,to_string( empty >> 2)+"\n") -> 0;
8 }
```

test-bitwise-right-empty.xtnd - Expected Output

```
1 empty
```

test-bitwise-xor.xtnd

```
1 extern "stdlib.o" {
2     print(a,b);
3     to_string(a);
4 }
5
6 main([1,1] args){
7     return print(1,to_string( 14 ^ 12)+"\n") -> 0;
8 }
```

test-bitwise-xor.xtnd - Expected Output

```
1 2
```

test-bitwise-xor-empty.xtnd

```
1 extern "stdlib.o" {
2     print(a,b);
3     to_string(a);
4 }
5
6 main([1,1] args){
7     return print(1,to_string( empty ^ 2)+"\n") -> 0;
8 }
```

test-bitwise-xor-empty.xtnd - Expected Output

```
1 empty
```

test-boolean-equals.xtnd

```
1 extern "stdlib.o" {
2     print(a,b);
3     to_string(a);
4 }
5
6 main([1,1] args){
7     return print(1,to_string( 5 == 6)+"\n") -> 0;
8 }
```

test-boolean-equals.xtnd - Expected Output

```
1 0
```

test-boolean-equals-both-empty.xtnd

```
1 extern "stdlib.o" {
2     print(a,b);
3     to_string(a);
4 }
5
6 main([1,1] args){
7     return print(1,to_string( empty == empty)+"\n") -> 0;
8 }
```

test-boolean-equals-both-empty.xtnd - Expected Output

```
1 1
```

test-boolean-equals-harder.xtnd

```
1 extern "stdlib.o" {
2     printv(a,b);
3     printd(a,b);
4 }
5
6 main([1,1] args){
7     return
8         printv(1, "True cases for ==\n") ->
9         printd(1, (5 == 5)) ->
10        printd(1, (5 == 5.0)) ->
11        printd(1, (0.5 == 5e-1)) ->
12        printd(1, (50 == 5e1)) ->
13        printd(1, 2 + 2 == 4) ->
14        printd(1, "foo" == "foo") ->
15        printd(1, "" == "") ->
16        printd(1, empty == empty) ->
17        printd(1, empty == !empty) ->
18        printd(1, !"foo" == !"bar") ->
19        printd(1, (2 ? 3 : 4) == ("foo" ? 3 : "not 4") ) ->
20
21        printv(1, "\nFalse cases for ==\n") ->
22        printd(1, (5 == 6)) ->
23        printd(1, (5 == 5.01)) ->
24        printd(1, (0.5 == 5e-2)) ->
25        printd(1, (50 == 5e2)) ->
26        printd(1, 2 + 2 == 5) ->
27        printd(1, "foo" == "bar") ->
28        printd(1, "" == "foo") ->
29        printd(1, "" == empty) ->
30        printd(1, 2 == empty) ->
31        printd(1, empty == 2) ->
32        printd(1, (2 ? 3 : 4) == ("foo" ? "not 3" : 4) ) ->
33
34        printv(1, "\nTrue cases for !=\n") ->
35        printd(1, (5 != 6)) ->
36        printd(1, (5 != 5.01)) ->
37        printd(1, (0.5 != 5e-2)) ->
```

```

38     printf(1, (50 != 5e2)) ->
39     printf(1, 2 + 2 != 5) ->
40     printf(1, "foo" != "bar") ->
41     printf(1, "" != "foo") ->
42     printf(1, "" != empty) ->
43     printf(1, 2 != empty) ->
44     printf(1, empty != 2) ->
45     printf(1, (2 ? 3 : 4) != ("foo" ? "not 3" : 4) ) ->
46
47     printf(1, "\nFalse cases for !=\n") ->
48     printf(1, (5 != 5)) ->
49     printf(1, (5 != 5.0)) ->
50     printf(1, (0.5 != 5e-1)) ->
51     printf(1, (50 != 5e1)) ->
52     printf(1, 2 + 2 != 4) ->
53     printf(1, "foo" != "foo") ->
54     printf(1, "" != "") ->
55     printf(1, empty != empty) ->
56     printf(1, empty != !empty) ->
57     printf(1, !"foo" != !"bar") ->
58     printf(1, (2 ? 3 : 4) != ("foo" ? 3 : "not 4") ) ->
59
60     0;
61 }

```

test-boolean-equals-harder.xtnd - Expected Output

```

1 True cases for ==
2 1.000000
3 1.000000
4 1.000000
5 1.000000
6 1.000000
7 1.000000
8 1.000000
9 1.000000
10 1.000000
11 1.000000
12 1.000000
13
14 False cases for ==
15 0.000000
16 0.000000
17 0.000000
18 0.000000
19 0.000000
20 0.000000
21 0.000000
22 0.000000
23 0.000000
24 0.000000
25 0.000000
26
27 True cases for !=
28 1.000000
29 1.000000
30 1.000000

```

```

31 1.000000
32 1.000000
33 1.000000
34 1.000000
35 1.000000
36 1.000000
37 1.000000
38 1.000000
39
40 False cases for !=
41 0.000000
42 0.000000
43 0.000000
44 0.000000
45 0.000000
46 0.000000
47 0.000000
48 0.000000
49 0.000000
50 0.000000
51 0.000000

```

test-boolean-equals-one-empty.xtnd

```

1 extern "stdlib.o" {
2     print(a,b);
3     to_string(a);
4 }
5
6 main([1,1] args){
7     return print(1,to_string( empty == 5)+"\n") -> 0;
8 }

```

test-boolean-equals-one-empty.xtnd - Expected Output

```

1 0

```

test-boolean-logical-not-equals.xtnd

```

1 extern "stdlib.o" {
2     print(a,b);
3     to_string(a);
4 }
5
6 main([1,1] args){
7     return print(1,to_string( 6 != 7)+"\n") -> 0;
8 }

```

test-boolean-logical-not-equals.xtnd - Expected Output

```

1 1

```

test-boolean-logical-not-equals-both-empty.xtnd

```

1 extern "stdlib.o" {
2     print(a,b);
3     to_string(a);
4 }

```



```

5
6 main([1,1] args){
7     return print(1,to_string( empty != empty)+"\n") -> 0;
8 }

```

test-boolean-logical-not-equals-both-empty.xtnd - Expected Output

```

1 0

```

test-boolean-logical-not-equals-one-empty.xtnd

```

1 extern "stdlib.o" {
2     print(a,b);
3     to_string(a);
4 }
5
6 main([1,1] args){
7     return print(1,to_string( empty != 5)+"\n") -> 0;
8 }

```

test-boolean-logical-not-equals-one-empty.xtnd - Expected Output

```

1 1

```

test-calling-func-from-import.xtnd

```

1 extern "stdlib.o" {
2     print(a,b);
3     to_string(a);
4 }
5
6 import "../samples/gcd_func.xtnd";
7
8 main([1,n] args){
9     return print(1,to_string( gcd(70, 55))+"\n") -> 0;
10 }

```

test-calling-func-from-import.xtnd - Expected Output

```

1 5

```

test-ceil.xtnd

```

1 extern "stdlib.o" {
2     extend_ceil(a);
3     printf(a,b);
4 }
5
6 [1,1] main([1,n] args) {
7     return printf(1, extend_ceil(10.45)) -> 0;
8 }

```

test-ceil.xtnd - Expected Output

```

1 11.000000

```

test-cos.xtnd

```

1 extern "stdlib.o" {
2     extend_cos(a);
3     printf(a,b);
4 }
5
6 [1,1] main([1,n] args) {
7     return printf(1, extend_cos(45.0)) -> 0;
8 }

```

test-cos.xtnd - Expected Output

```

1 0.525322

```

test-cosh.xtnd

```

1 extern "stdlib.o" {
2     extend_cosh(a);
3     printf(a,b);
4 }
5
6 [1,1] main([1,n] args) {
7     return printf(1, extend_cosh(45.0)) -> 0;
8 }

```

test-cosh.xtnd - Expected Output

```

1 17467135528742547456.000000

```

test-division.xtnd

```

1 extern "stdlib.o" {
2     print(a,b);
3     to_string(a);
4 }
5
6 main([1,1] args){
7     /* Should evaluate to 4 */
8     return print(1,to_string( 20 / 5)+"\n") -> 0;
9 }

```

test-division.xtnd - Expected Output

```

1 4

```

test-division-empty.xtnd

```

1 extern "stdlib.o" {
2     print(a,b);
3     to_string(a);
4 }
5
6 main([1,n] args){
7     /* Should return empty */
8     return print(1,to_string( empty / 5)+"\n") -> 0;
9 }

```

test-division-empty.xtnd - Expected Output

```

1 empty

```

test-exp.xtnd

```
1 extern "stdlib.o" {
2     extend_exp(a);
3     printf(a,b);
4 }
5
6 [1,1] main([1,n] args) {
7     return printf(1, extend_exp(2.0)) -> 0;
8 }
```

test-exp.xtnd - Expected Output

```
1 7.389056
```

test-fabs.xtnd

```
1 extern "stdlib.o" {
2     extend_fabs(a);
3     printf(a,b);
4 }
5
6 [1,1] main([1,n] args) {
7     return printf(1, extend_fabs(-45.0)) -> 0;
8 }
```

test-fabs.xtnd - Expected Output

```
1 45.000000
```

test-file-close.xtnd

```
1 extern "stdlib.o" {
2     extend_open(a,b);
3     extend_close(a);
4     print(a,b);
5 }
6
7 [1,1] main(args){
8     return extend_close(extend_open("testcases/assets/test_file.txt", "r")) -> print(1, "
    Made it this far\n") -> 0;
9 }
```

test-file-close.xtnd - Expected Output

```
1 Made it this far
```

test-file-read.xtnd

```
1 extern "stdlib.o" {
2     extend_open(a,b);
3     extend_read(a,b);
4     extend_close(a);
5     print(a,b);
6 }
7
8 [1,1] main(args){
9     return print(1, extend_read(extend_open("testcases/assets/test_file.txt", "r"),5))
    -> 0;
10 }
```

test-file-read.xtnd - Expected Output

```
1 This
```

test-file-write.xtnd

```
1 extern "stdlib.o" {
2     extend_open(a,b);
3     extend_write(a,b);
4     extend_close(a);
5     print(a,b);
6 }
7
8 [1,1] main(args){
9     return extend_close(extend_write(extend_open("testcases/assets/test_file_write.out",
10         "w"), "Hello"))
11         -> print(1,"Made it this far\n")
12         -> 0;
13 }
```

test-file-write.xtnd - Expected Output

```
1 Made it this far
```

test-floor.xtnd

```
1 extern "stdlib.o" {
2     extend_floor(a);
3     printf(a,b);
4 }
5
6 [1,1] main([1,n] args) {
7     return printf(1, extend_floor(10.45)) -> 0;
8 }
```

test-floor.xtnd - Expected Output

```
1 10.000000
```

test-func-params.xtnd

```
1 extern "stdlib.o" {
2     print(a,b);
3     to_string(a);
4 }
5
6 main([1,n] args) {
7     return print(1,to_string( foo("string"))+"\n") -> 0;
8 }
9 [1,1] foo([1,1] arg) {
10     return arg;
11 }
```

test-func-params.xtnd - Expected Output

```
1 string
```

test-func-params-omit-dim.xtnd

```

1 extern "stdlib.o" {
2     print(a,b);
3     to_string(a);
4 }
5
6 main([1,n] args) {
7     return print(1,to_string( foo("string"))+"\n") -> 0;
8 }
9 foo([1,1] arg) {
10     return arg;
11 }

```

test-func-params-omit-dim.xtnd - Expected Output

```

1 string

```

test-global-hello.xtnd

```

1 extern "stdlib.o" {
2     print(a,b);
3     printv(a,b);
4 }
5
6 bar() {
7     foo := 5;
8     return 2;
9 }
10
11 global foo := printv(1,"Hello Globals!\n") -> 0;
12
13 [1,1] main(args) {
14     return foo;
15 }

```

test-global-hello.xtnd - Expected Output

```

1 Hello Globals!

```

test-global-masking.xtnd

```

1 extern "stdlib.o" {
2     print(a,b);
3     printv(a,b);
4 }
5
6 bar() {
7     foo := 5;
8     return 2;
9 }
10
11 global foo := printv(1,"Hello Globals!\n") -> 0;
12
13 [1,1] main(args) {
14     foo := printv(1,"Hello Locals!\n") -> 0;
15     return foo;
16 }

```

test-global-masking.xtnd - Expected Output

```
1 Hello Locals!
```

test-globals-between-imports.xtnd

```
1 import "../testcases/assets/string.xtnd";
2 global foo;
3 global [2, 5] bar;
4 import "../testcases/assets/string.xtnd";
```

test-globals-between-imports.xtnd - Expected Output

```
1 Hello
```

test-greater-than.xtnd

```
1 extern "stdlib.o" {
2     print(a,b);
3     to_string(a);
4 }
5
6 main([1,1] args){
7     return print(1,to_string( 6 > 5)+"\n") -> 0;
8 }
```

test-greater-than.xtnd - Expected Output

```
1 1
```

test-greater-than-empty.xtnd

```
1 extern "stdlib.o" {
2     print(a,b);
3     to_string(a);
4 }
5
6 main([1,1] args){
7     return print(1,to_string( empty > 5)+"\n") -> 0;
8 }
```

test-greater-than-empty.xtnd - Expected Output

```
1 empty
```

test-greater-than-or-equal.xtnd

```
1 extern "stdlib.o" {
2     print(a,b);
3     to_string(a);
4 }
5
6 main([1,1] args){
7     return print(1,to_string( 7 >= 7)+"\n") -> 0;
8 }
```

test-greater-than-or-equal.xtnd - Expected Output

```
1 1
```

test-greater-than-or-equal-empty.xtnd

```

1 extern "stdlib.o" {
2     print(a,b);
3     to_string(a);
4 }
5
6 main([1,1] args){
7     return print(1,to_string( empty >= 7)+"\n") -> 0;
8 }

```

test-greater-than-or-equal-empty.xtnd - Expected Output

```

1 empty

```

test-less-than.xtnd

```

1 extern "stdlib.o" {
2     print(a,b);
3     to_string(a);
4 }
5
6 main([1,1] args){
7     return print(1,to_string( 6 < 7)+"\n") -> 0;
8 }

```

test-less-than.xtnd - Expected Output

```

1 1

```

test-less-than-empty.xtnd

```

1 extern "stdlib.o" {
2     print(a,b);
3     to_string(a);
4 }
5
6 main([1,1] args){
7     return print(1,to_string( empty > 5)+"\n") -> 0;
8 }

```

test-less-than-empty.xtnd - Expected Output

```

1 empty

```

test-less-than-or-equal.xtnd

```

1 extern "stdlib.o" {
2     print(a,b);
3     to_string(a);
4 }
5
6 main([1,1] args){
7     return print(1,to_string( 7 <= 5)+"\n") -> 0;
8 }

```

test-less-than-or-equal.xtnd - Expected Output

```

1 0

```

test-less-than-or-equal-empty.xtnd

```

1 extern "stdlib.o" {
2     print(a,b);
3     to_string(a);
4 }
5
6 main([1,1] args){
7     return print(1,to_string( empty <= 8)+"\n") -> 0;
8 }

```

test-less-than-or-equal-empty.xtnd - Expected Output

```

1 empty

```

test-log.xtnd

```

1 extern "stdlib.o" {
2     extend_log(a);
3     printf(a,b);
4 }
5
6 [1,1] main([1,n] args) {
7     return printf(1, extend_log(10.0)) -> 0;
8 }

```

test-log.xtnd - Expected Output

```

1 2.302585

```

test-log10.xtnd

```

1 extern "stdlib.o" {
2     extend_log10(a);
3     printf(a,b);
4 }
5
6 [1,1] main([1,n] args) {
7     return printf(1, extend_log10(100.0)) -> 0;
8 }

```

test-log10.xtnd - Expected Output

```

1 2.000000

```

test-logical-and.xtnd

```

1 extern "stdlib.o" {
2     print(a,b);
3     to_string(a);
4 }
5
6 main([1,1] args){
7     return print(1,to_string( 1 && 6)+"\n") -> 0;
8 }

```

test-logical-and.xtnd - Expected Output

```

1 1

```

test-logical-and-empty.xtnd


```

1 extern "stdlib.o" {
2     print(a,b);
3     to_string(a);
4 }
5
6 main([1,1] args){
7     return print(1,to_string( empty && 1)+"\n") -> 0;
8 }

```

test-logical-and-empty.xtnd - Expected Output

```

1 empty

```

test-logical-not.xtnd

```

1 extern "stdlib.o" {
2     print(a,b);
3     to_string(a);
4 }
5
6 main([1,1] args){
7     return print(1,to_string( !5)+"\n") -> 0;
8 }

```

test-logical-not.xtnd - Expected Output

```

1 0

```

test-logical-not-empty.xtnd

```

1 extern "stdlib.o" {
2     print(a,b);
3     to_string(a);
4 }
5
6 main([1,1] args){
7     return print(1,to_string( !empty)+"\n") -> 0;
8 }

```

test-logical-not-empty.xtnd - Expected Output

```

1 empty

```

test-logical-or.xtnd

```

1 extern "stdlib.o" {
2     print(a,b);
3     to_string(a);
4 }
5
6 main([1,1] args){
7     return print(1,to_string( 5 || 6)+"\n") -> 0;
8 }

```

test-logical-or.xtnd - Expected Output

```

1 1

```

test-logical-or-empty.xtnd

```

1 extern "stdlib.o" {
2     print(a,b);
3     to_string(a);
4 }
5
6 main([1,1] args){
7     return print(1,to_string( empty || 4)+"\n") -> 0;
8 }

```

test-logical-or-empty.xtnd - Expected Output

```

1 empty

```

test-modulo.xtnd

```

1 extern "stdlib.o" {
2     print(a,b);
3     to_string(a);
4 }
5
6 main([1,n] args){
7     /* Should return 1 */
8     return print(1,to_string( 5 % 4)+"\n") -> 0;
9 }

```

test-modulo.xtnd - Expected Output

```

1 1

```

test-modulo-empty.xtnd

```

1 extern "stdlib.o" {
2     print(a,b);
3     to_string(a);
4 }
5
6 main([1,n] args){
7     /* Should return empty */
8     return print(1,to_string( empty % 5)+"\n") -> 0;
9 }

```

test-modulo-empty.xtnd - Expected Output

```

1 empty

```

test-multiple-imports.xtnd

```

1 import "../testcases/assets/string.xtnd";
2 import "../testcases/assets/string.xtnd";

```

test-multiple-imports.xtnd - Expected Output

```

1 Hello

```

test-multiplication.xtnd

```

1 extern "stdlib.o" {
2     print(a,b);
3     to_string(a);
4 }
5
6 main([1,n] args){
7     /* Should evaluate to 35 */
8     return print(1,to_string( 7 * 5)+"\n") -> 0;
9 }

```

test-multiplication.xtnd - Expected Output

```

1 35

```

test-multiplication-empty.xtnd

```

1 extern "stdlib.o" {
2     print(a,b);
3     to_string(a);
4 }
5
6 main([1,n] args){
7     /* Should evaluate to empty */
8     return print(1,to_string( empty * 5)+"\n") -> 0;
9 }

```

test-multiplication-empty.xtnd - Expected Output

```

1 empty

```

test-parse-error.xtnd

```

1 [1,1] main(args){
2     foo := 5$5;
3     return foo;
4 }

```

test-parse-error.xtnd - Expected Output

```

1 Syntax error in "./testcases/inputs_regression/test_parse_error.xtnd": Invalid
   character: $
2 Line 2 at character 11

```

test-parse-error-after-multiline-comment.xtnd

```

1 [1,1] main(args){
2     /* This is a comment spanning multiple lines.
3
4
5
6
7
8
9
10
11
12
13

```



```
1 Syntax error in "./testcases/inputs_regression/test_parse_error_newlines.xtnd":  
   Invalid character: $  
2 Line 3 at character 52
```

test-parse-error-string.xtnd

```
1 [1,1] main(args){  
2   foo := "Hello";$$$;  
3   return foo;  
4 }
```

test-parse-error-string.xtnd - Expected Output

```
1 Syntax error in "./testcases/inputs_regression/test_parse_error_string.xtnd": Invalid  
   character: $  
2 Line 2 at character 18
```

test-power.xtnd

```
1 extern "stdlib.o" {  
2   print(a,b);  
3   to_string(a);  
4 }  
5  
6 main([1,n] args){  
7   /* Should return 216 */  
8   return print(1,to_string( 6**3)+"\n") -> 0;  
9 }
```

test-power.xtnd - Expected Output

```
1 216
```

test-power-empty.xtnd

```
1 extern "stdlib.o" {  
2   print(a,b);  
3   to_string(a);  
4 }  
5  
6 main([1,n] args){  
7   /* Should return empty */  
8   return print(1,to_string( empty**5)+"\n") -> 0;  
9 }
```

test-power-empty.xtnd - Expected Output

```
1 empty
```

test-print-empty.xtnd

```
1 extern "stdlib.o" {  
2   print(a,b);  
3   to_string(a);  
4 }  
5  
6 main([1,n] args) {  
7   foo := empty;  
8   return print(1,to_string( foo)+"\n") -> 0;  
9 }
```

test-print-empty.xtnd - Expected Output

```
1 empty
```

test-print-nums.xtnd

```
1 extern "stdlib.o" {
2   print(a,b);
3   to_string(a);
4 }
5
6 main([1,n] args) {
7   foo := 1;
8   return print(1,to_string( foo)+"\n") -> 0;
9 }
```

test-print-nums.xtnd - Expected Output

```
1 1
```

test-print-str.xtnd

```
1 extern "stdlib.o" {
2   print(a,b);
3   to_string(a);
4 }
5
6 main([1,n] args) {
7   foo := "string";
8   return print(1,to_string( foo)+"\n") -> 0;
9 }
```

test-print-str.xtnd - Expected Output

```
1 string
```

test-ref-between-globals.xtnd

```
1 extern "stdlib.o" {
2   print(a,b);
3   to_string(a);
4 }
5
6 global [2,2] foo;
7 global [2,2] bar;
8 main([1,n] args) {
9   foo := 1;
10  bar := foo;
11  return print(1,to_string( bar)+"\n") -> 0;
12 }
```

test-ref-between-globals.xtnd - Expected Output

```
1 1
```

test-short-circuiting-and.xtnd

```

1 extern "stdlib.o" {
2     print(a,b);
3 }
4
5 main([1,1] args){
6     return 0 && print(1,"FAIL\n") -> print(1,"PASS\n") -> 0;
7 }

```

test-short-circuiting-and.xtnd - Expected Output

```

1 PASS

```

test-short-circuiting-and2.xtnd

```

1 extern "stdlib.o" {
2     print(a,b);
3 }
4
5 main([1,1] args){
6     return 1 && print(1,"PASS1\n") -> print(1,"PASS2\n") -> 0;
7 }

```

test-short-circuiting-and2.xtnd - Expected Output

```

1 PASS1
2 PASS2

```

test-short-circuiting-or.xtnd

```

1 extern "stdlib.o" {
2     print(a,b);
3 }
4
5 main([1,1] args){
6     return 0 || print(1,"PASS1\n") -> print(1,"PASS2\n") -> 0;
7 }

```

test-short-circuiting-or.xtnd - Expected Output

```

1 PASS1
2 PASS2

```

test-short-circuiting-or2.xtnd

```

1 extern "stdlib.o" {
2     print(a,b);
3 }
4
5 main([1,1] args){
6     return 1 || print(1,"FAIL\n") -> print(1,"PASS\n") -> 0;
7 }

```

test-short-circuiting-or2.xtnd - Expected Output

```

1 PASS

```

test-sin.xtnd

```

1 extern "stdlib.o" {
2     extend_sin(a);
3     printf(a,b);
4 }
5
6 [1,1] main([1,n] args) {
7     return printf(1, extend_sin(45.0)) -> 0;
8 }

```

test-sin.xtnd - Expected Output

```

1 0.850904

```

test-sin-through-function.xtnd

```

1 extern "stdlib.o" {
2     extend_sin(a);
3     printf(a,b);
4 }
5
6 internal_sin(x,y,z) {
7     return extend_sin(z);
8 }
9
10 [1,1] main([1,n] args) {
11     return printf(1, internal_sin(1,2,45.0)) -> 0;
12 }

```

test-sin-through-function.xtnd - Expected Output

```

1 0.850904

```

test-sin-through-function-and-global.xtnd

```

1 extern "stdlib.o" {
2     extend_sin(a);
3     printf(a,b);
4 }
5
6 global theta := 45.0;
7
8 internal_sin(x,y,z) {
9     return extend_sin(z);
10 }
11
12 [1,1] main([1,n] args) {
13     return printf(1, internal_sin(1,2,theta)) -> 0;
14 }

```

test-sin-through-function-and-global.xtnd - Expected Output

```

1 0.850904

```

test-single-import.xtnd

```

1 extern "stdlib.o" {
2     print(a,b);
3     to_string(a);

```



```

4 }
5
6 import "../samples/gcd_func.xtnd";
7
8 main([1,n] args) {
9     return print(1, to_string(gcd(70, 55)) + "\n") -> 0;
10 }

```

test-single-import.xtnd - Expected Output

```
1 5
```

test-sinh.xtnd

```

1 extern "stdlib.o" {
2     extend_sinh(a);
3     printf(a,b);
4 }
5
6 [1,1] main([1,n] args) {
7     return printf(1, extend_sinh(45.0)) -> 0;
8 }

```

test-sinh.xtnd - Expected Output

```
1 17467135528742547456.000000
```

test-sqrt.xtnd

```

1 extern "stdlib.o" {
2     extend_sqrt(a);
3     printf(a,b);
4 }
5
6 [1,1] main([1,n] args) {
7     return printf(1, extend_sqrt(9.0)) -> 0;
8 }

```

test-sqrt.xtnd - Expected Output

```
1 3.000000
```

test-string-concatenation.xtnd

```

1 extern "stdlib.o" {
2     printf(a,b);
3 }
4
5 [1,1] main(args) {
6     foo :=
7         printf(1,"Hello " + "World\n") ->
8         printf(1,"Hello " + "World" + "\n") ->
9         printf(1,("Hello " + "World") + (" " + "\n")) ->
10        0;
11     return foo;
12 }

```

test-string-concatenation.xtnd - Expected Output

```
1 Hello World
2 Hello World
3 Hello World
```

test-subtraction.xtnd

```
1 extern "stdlib.o" {
2   print(a,b);
3   to_string(a);
4 }
5
6 main([1,1] args){
7   return print(1,to_string( 7 - 5)+"\n") -> 0;
8 }
```

test-subtraction.xtnd - Expected Output

```
1 2
```

test-subtraction-empty.xtnd

```
1 extern "stdlib.o" {
2   print(a,b);
3   to_string(a);
4 }
5
6 main([1,1] args){
7   return print(1,to_string( empty - 2)+"\n") -> 0;
8 }
```

test-subtraction-empty.xtnd - Expected Output

```
1 empty
```

test-switch-v1.xtnd

```
1 extern "stdlib.o" {
2   print(a,b);
3   to_string(a);
4 }
5
6 main([1,1] args){
7   x := switch(1) {
8     case 1: 100;
9     case 2: 200;
10    default: 300;
11  };
12   return print(1,to_string(x)+"\n") -> 0;
13 }
```

test-switch-v1.xtnd - Expected Output

```
1 100
```

test-switch-v10.xtnd

```

1 extern "stdlib.o" {
2     printf(a,b);
3     to_string(a);
4 }
5
6 main([1,1] args){
7     x := switch {
8         case 0: 100;
9         case "also true": 200;
10        default: 99;
11    };
12    return printf(1,x) -> 0;
13 }

```

test-switch-v10.xtnd - Expected Output

```

1 200.000000

```

test-switch-v11.xtnd

```

1 extern "stdlib.o" {
2     printf(a,b);
3     to_string(a);
4 }
5
6 main([1,1] args){
7     x := switch {
8         case 0: 100;
9         default: 99;
10    };
11    return printf(1,x) -> 0;
12 }

```

test-switch-v11.xtnd - Expected Output

```

1 99.000000

```

test-switch-v2.xtnd

```

1 extern "stdlib.o" {
2     print(a,b);
3     to_string(a);
4 }
5
6 main([1,1] args){
7     x := switch(2) {
8         case 1: 100;
9         case 2: 200;
10        default: 300;
11    };
12    return print(1,to_string(x)+"\n") -> 0;
13 }

```

test-switch-v2.xtnd - Expected Output

```

1 200

```

test-switch-v3.xtnd

```

1 extern "stdlib.o" {
2     print(a,b);
3     to_string(a);
4 }
5
6 main([1,1] args){
7     x := switch(3) {
8         case 1: 100;
9         case 2: 200;
10        default: 300;
11    };
12    return print(1,to_string(x)+"\n") -> 0;
13 }

```

test-switch-v3.xtnd - Expected Output

```

1 300

```

test-switch-v4.xtnd

```

1 extern "stdlib.o" {
2     print(a,b);
3     to_string(a);
4 }
5
6 main([1,1] args){
7     x := switch(2) {
8         case 1, 2: 100;
9         default: 300;
10    };
11    return print(1,to_string(x)+"\n") -> 0;
12 }

```

test-switch-v4.xtnd - Expected Output

```

1 100

```

test-switch-v5.xtnd

```

1 extern "stdlib.o" {
2     print(a,b);
3     to_string(a);
4 }
5
6 main([1,1] args){
7     x := switch(3) {
8         case 1, 2: 100;
9         default: 300;
10    };
11    return print(1,to_string(x)+"\n") -> 0;
12 }

```

test-switch-v5.xtnd - Expected Output

```

1 300

```

test-switch-v6.xtnd

```

1 extern "stdlib.o" {
2     print(a,b);
3     to_string(a);
4 }
5
6 main([1,1] args){
7     x := switch(3) {
8         case 1, 2: 100;
9         case 0, 3: 200;
10        default: 300;
11    };
12    return print(1,to_string(x)+"\n") -> 0;
13 }

```

test-switch-v6.xtnd - Expected Output

```

1 200

```

test-switch-v7.xtnd

```

1 extern "stdlib.o" {
2     print(a,b);
3     to_string(a);
4 }
5
6 main([1,1] args){
7     x := switch(4) {
8         case 1, 2: 100;
9         case 0, 3: 200;
10    };
11    return print(1,to_string(x)+"\n") -> 0;
12 }

```

test-switch-v7.xtnd - Expected Output

```

1 empty

```

test-switch-v8.xtnd

```

1 extern "stdlib.o" {
2     print(a,b);
3     to_string(a);
4 }
5
6 main([1,1] args){
7     x := switch() {
8         case 1 > 2: 100;
9         case 3 > 0: 200;
10    };
11    return print(1,to_string(x)+"\n") -> 0;
12 }

```

test-switch-v8.xtnd - Expected Output

```

1 200

```

test-switch-v9.xtnd

```

1 extern "stdlib.o" {
2     printf(a,b);
3     to_string(a);
4 }
5
6 main([1,1] args){
7     x := switch {
8         case "true": 100;
9         case "also true": 200;
10    };
11    return printf(1,x) -> 0;
12 }

```

test-switch-v9.xtnd - Expected Output

```

1 100.000000

```

test-tan.xtnd

```

1 extern "stdlib.o" {
2     extend_tan(a);
3     printf(a,b);
4 }
5
6 [1,1] main([1,n] args) {
7     return printf(1, extend_tan(45.0)) -> 0;
8 }

```

test-tan.xtnd - Expected Output

```

1 1.619775

```

test-tanh.xtnd

```

1 extern "stdlib.o" {
2     extend_tanh(a);
3     printf(a,b);
4 }
5
6 [1,1] main([1,n] args) {
7     return printf(1, extend_tanh(45.0)) -> 0;
8 }

```

test-tanh.xtnd - Expected Output

```

1 1.000000

```

test-ternary-conditional.xtnd

```

1 extern "stdlib.o" {
2     print(a,b);
3     to_string(a);
4 }
5
6 main([1,1] args){
7     return print(1,to_string(5 ? 2 : 3) + "\n") -> 0;
8 }

```

test-ternary-conditional.xtnd - Expected Output

```
1 2
```

test-ternary-conditional-empty.xtnd

```
1 extern "stdlib.o" {
2     print(a,b);
3     to_string(a);
4 }
5
6 main([1,1] args){
7     return print(1,to_string( empty ? 5 : 6)+"\n") -> 0;
8 }
```

test-ternary-conditional-empty.xtnd - Expected Output

```
1 empty
```

test-unary-negation.xtnd

```
1 extern "stdlib.o" {
2     print(a,b);
3     to_string(a);
4 }
5
6 main([1,n] args){
7     /* Should return -33 */
8     return print(1,to_string( -33)+"\n") -> 0;
9 }
```

test-unary-negation.xtnd - Expected Output

```
1 -33
```

test-unary-negation-empty.xtnd

```
1 extern "stdlib.o" {
2     print(a,b);
3     to_string(a);
4 }
5
6 main([1,n] args){
7     return print(1,to_string( -empty)+"\n") -> 0;
8 }
```

test-unary-negation-empty.xtnd - Expected Output

```
1 empty
```

9. Git Logs

```
1 *   commit 8297f339f2e0d82ff2aa0b48840103441af0e7a5
2 |\ Merge: d87b73c 37470e9
3 || Author: Jared Samet <jared.samet@aya.yale.edu>
4 || Date:   Wed Dec 14 12:01:47 2016 -0500
5 ||
6 ||       Merge pull request #85 from ExtendLang/put-lt-back
7 ||
8 ||       Put back LT, comment out sys/time.h
9 ||
10 | *   commit 37470e97bb50de7678bb6ea084f11bb4169c8367
11 | / Author: oracleofnj <jared.samet@aya.yale.edu>
12 | Date:   Wed Dec 14 11:14:06 2016 -0500
13 |
14 |       Put back LT, comment out sys/time.h
15 |
16 *   commit d87b73c920b3943e1eaf83b0b4437e115ffca0a8
17 |\ Merge: 61bc9b6 d126e3c
18 || Author: Jared Samet <jared.samet@aya.yale.edu>
19 || Date:   Wed Dec 14 10:51:58 2016 -0500
20 ||
21 ||       Merge pull request #82 from ExtendLang/hard-to-repro-bug
22 ||
23 ||       Half the time it works
24 ||
25 | *   commit d126e3ca94535cdbdeb917737b0464f1b83f1483
26 | | Author: oracleofnj <jared.samet@aya.yale.edu>
27 | | Date:   Wed Dec 14 00:51:00 2016 -0500
28 | |
29 | |       Try with time.h instead of sys/time.h
30 | |
31 | *   commit a5356125e9e9c01e933ba553cd4ba6d168c48a25
32 | | Author: oracleofnj <jared.samet@aya.yale.edu>
33 | | Date:   Wed Dec 14 00:48:35 2016 -0500
34 | |
35 | |       Remove lrints
36 | |
37 | *   commit e8448532121cc49d4b8cb3b1b436d3268f643753
38 | | Author: oracleofnj <jared.samet@aya.yale.edu>
39 | | Date:   Wed Dec 14 00:34:37 2016 -0500
40 | |
41 | |       Initialize all variables and remove pointer math; bug appears fixed
42 | |
43 | *   commit 4cla421a814d132a57e1cda50e4b1c9c3709dd6d
44 | | Author: oracleofnj <jared.samet@aya.yale.edu>
```



```

45 | | Date: Tue Dec 13 22:55:07 2016 -0500
46 | |
47 | | Some formula is weird
48 | |
49 | * commit 5dbd4090286655786d4fdfe259f055e696eaa7b4
50 | |\ Merge: 879eaf3 37f5ce2
51 | | | Author: oracleofnj <jared.samet@aya.yale.edu>
52 | | | Date: Tue Dec 13 22:43:19 2016 -0500
53 | | |
54 | | | Merge branch 'hard-to-repro-bug' of https://github.com/ExtendLang/Extend
    | | | into hard-to-repro-bug
55 | | |
56 | | * commit 37f5ce24falee119ae55c4aa16799960e6aled3a
57 | | |\ Merge: e20f7e4 alcf5a
58 | | | Author: Jared Samet <jared.samet@aya.yale.edu>
59 | | | Date: Tue Dec 13 22:42:40 2016 -0500
60 | | |
61 | | | Merge pull request #83 from ExtendLang/rounding-for-read
62 | | |
63 | | | Added rounding at several places
64 | | |
65 | | | * commit alcf5a6513b5e67e2c99707834108b78b82ede3
66 | | |/ Author: Nigel Schuster <nigel.schusters@gmail.com>
67 | | | Date: Tue Dec 13 22:34:21 2016 -0500
68 | | |
69 | | | Added rounding at several places
70 | | |
71 | * | commit 879eaf3795a5b000e4bfa45cb7cd100fef32fe16
72 | |/ Author: oracleofnj <jared.samet@aya.yale.edu>
73 | | Date: Tue Dec 13 22:43:17 2016 -0500
74 | |
75 | | Testing
76 | |
77 | * commit e20f7e4b4a34eb102fd10408acb0c05db53a0ef7
78 | / Author: oracleofnj <jared.samet@aya.yale.edu>
79 | Date: Tue Dec 13 21:36:13 2016 -0500
80 |
81 | Half the time it works
82 |
83 | * commit 61bc9b646c9a47d9895c436fe070671cfd2dcccdf
84 | \ Merge: ae5b8a8 4a810df
85 | | Author: Nigel Schuster <Neitsch@users.noreply.github.com>
86 | | Date: Tue Dec 13 20:33:27 2016 -0500
87 | |
88 | | Merge pull request #81 from ExtendLang/fix-em-all
89 | |
90 | | Corrected testcase outputs
91 | |
92 | * commit 4a810dff997bbb03a5093ff83a4a5ac0ce51decbb
93 | / Author: Nigel Schuster <nigel.schusters@gmail.com>
94 | Date: Tue Dec 13 19:34:29 2016 -0500
95 |
96 | Corrected testcase outputs
97 |
98 | * commit ae5b8a8e130alb0e20f70d446097115be22018aa
99 | \ Merge: 8e6e9ba 70b2704

```

```

100 | | Author: Jared Samet <jared.samet@aya.yale.edu>
101 | | Date: Tue Dec 13 19:08:43 2016 -0500
102 | |
103 | | Merge pull request #80 from ExtendLang/select
104 | |
105 | | Select
106 | |
107 | * commit 70b270444f13c33a41366d8bfa0be971f11b0b30
108 | | Author: oracleofnj <jared.samet@aya.yale.edu>
109 | | Date: Tue Dec 13 19:02:32 2016 -0500
110 | |
111 | | No C99
112 | |
113 | * commit 15fd76223412a2a9c25f2854300f80248e93f5f4
114 | \ Merge: 7a93885 8e6e9ba
115 | / Author: oracleofnj <jared.samet@aya.yale.edu>
116 | / Date: Tue Dec 13 18:42:21 2016 -0500
117 | |
118 | | Merge branch 'master' into select
119 | |
120 | * commit 8e6e9ba23e7d8f401efc0d9e6fc52ca15eac78bc
121 | \ Merge: 8146d04 a483282
122 | | Author: Jared Samet <jared.samet@aya.yale.edu>
123 | | Date: Tue Dec 13 18:42:05 2016 -0500
124 | |
125 | | Merge pull request #78 from ExtendLang/unop-unary-minus
126 | |
127 | | Unop TypeOf
128 | |
129 | | * commit 7a93885edd8c186e3487dc57efdd00ad156f10dd
130 | | Author: oracleofnj <jared.samet@aya.yale.edu>
131 | | Date: Tue Dec 13 18:41:49 2016 -0500
132 | |
133 | | Calculate all formula indices
134 | |
135 | | * commit 07e63dc3ffeceecfc6e598f5faa1435c4c4a4bd7
136 | | Author: oracleofnj <jared.samet@aya.yale.edu>
137 | | Date: Tue Dec 13 18:19:58 2016 -0500
138 | |
139 | | Properly build instantiate var
140 | |
141 | | * commit 1a291292bfb2a95aa54d796c741ba622262c3137
142 | | Author: oracleofnj <jared.samet@aya.yale.edu>
143 | | Date: Tue Dec 13 17:24:16 2016 -0500
144 | |
145 | | Replace bools with chars for compatibility between C and LLVM
146 | |
147 | | * commit 12e78a30f51a2d6034bbfdf313b624383fa94885
148 | | Author: oracleofnj <jared.samet@aya.yale.edu>
149 | | Date: Tue Dec 13 17:17:54 2016 -0500
150 | |
151 | | Added debug output
152 | |
153 | * commit a4832821f5e532bbf2fcbc38270c9821bcf44ec0
154 | \ Merge: f8c9b43 8146d04
155 | / Author: oracleofnj <jared.samet@aya.yale.edu>

```

```

156 | / | | Date: Tue Dec 13 16:13:30 2016 -0500
157 | | |
158 | | | Merge branch 'master' into unop-unary-minus
159 | | |
160 * | | commit 8146d046adbf93a34157048b3c4b1777207334fc
161 | \ | \ Merge: dcd5766 94afc93
162 | | | Author: Jared Samet <jared.samet@aya.yale.edu>
163 | | | Date: Tue Dec 13 16:12:17 2016 -0500
164 | | |
165 | | | Merge pull request #75 from ExtendLang/fix-more-tc
166 | | |
167 | | | Fix more tc
168 | | |
169 | * | commit 94afc93afdbe4eb32322f9e8951ef250f1870263
170 | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
171 | | | Date: Tue Dec 13 16:02:06 2016 -0500
172 | | |
173 | | | Corrected expected TC
174 | | |
175 | * | commit f6f82767ec790d01dd5b6e7e05509fceaec322ec
176 | / / | Author: Nigel Schuster <nigel.schusters@googlemail.com>
177 | | | Date: Tue Dec 13 16:00:59 2016 -0500
178 | | |
179 | | | Fixed string.xtnd file
180 | | |
181 * | | commit dcd5766e9b71ae3445135a1c731dd35cdafd24d3
182 | \ | \ Merge: 23328f1 d9abfc0
183 | | | Author: Nigel Schuster <Neitsch@users.noreply.github.com>
184 | | | Date: Tue Dec 13 15:44:38 2016 -0500
185 | | |
186 | | | Merge pull request #74 from ExtendLang/fix-tc
187 | | |
188 | | | Fixed ternary testcase
189 | | |
190 | * | commit d9abfc09658cf505f5f9b9239953cb7b1347bef7
191 | | \ | Merge: 324779a 23328f1
192 | | / | Author: Nigel Schuster <Neitsch@users.noreply.github.com>
193 | / | | Date: Tue Dec 13 15:38:38 2016 -0500
194 | | |
195 | | | Merge branch 'master' into fix-tc
196 | | |
197 | * | commit 324779a30d19b74050584d4ef2bd536d3f1b4943
198 | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
199 | | | Date: Tue Dec 13 15:32:26 2016 -0500
200 | | |
201 | | | Corrected expected value
202 | | |
203 | * | commit fafe2e602f3a091d77ddbb90887680bcefc9bd95
204 | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
205 | | | Date: Tue Dec 13 15:29:00 2016 -0500
206 | | |
207 | | | Fixed string tc
208 | | |
209 | * | commit 022f05c5b303b6c567d38393517803276c90b157
210 | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
211 | | | Date: Tue Dec 13 15:23:59 2016 -0500

```

```

212 | | |
213 | | | Fixed testcase
214 | | |
215 | | * commit f8c9b435857d699209cfe033171f124ce78e3dae
216 | | | Author: oracleofnj <jared.samet@aya.yale.edu>
217 | | | Date: Tue Dec 13 16:13:09 2016 -0500
218 | | |
219 | | | Make TypeOf work
220 | | |
221 | | * commit bfe1c07e3b3ec6d5237043eb50ce73f2fac80fb9
222 | | \ Merge: 50ed49c 23328f1
223 | | _/ Author: oracleofnj <jared.samet@aya.yale.edu>
224 | / | | Date: Tue Dec 13 15:39:45 2016 -0500
225 | | |
226 | | | Merge branch 'master' into unop-unary-minus
227 | | |
228 * | | commit 23328f10c3bf28e27a7e6a081fa91eea27611b5b
229 | \ \ \ Merge: 6ad8512 b12fe37
230 | | | | Author: Nigel Schuster <Neitsch@users.noreply.github.com>
231 | | | | Date: Tue Dec 13 15:37:18 2016 -0500
232 | | | |
233 | | | | Merge pull request #73 from ExtendLang/and-or-xor
234 | | | |
235 | | | | And or xor
236 | | | |
237 | * | | commit b12fe378c00d0782b36cc270a1388b2bb9094a52
238 | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
239 | | | | Date: Tue Dec 13 15:18:57 2016 -0500
240 | | | |
241 | | | | Implemented and, or and xor
242 | | | |
243 | * | | commit 90cbaa089e26f712bdb0321c66604d5ab83d4a75
244 | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
245 | | | | Date: Tue Dec 13 15:16:31 2016 -0500
246 | | | |
247 | | | | Added left and right shift
248 | | | |
249 | * | | commit 571ee7e663823d3cdcc388c11c3ef4903cc07625
250 | | \ \ \ Merge: aeab40d e377567
251 | | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
252 | | | | | Date: Tue Dec 13 14:56:05 2016 -0500
253 | | | | |
254 | | | | | Merge branch 'power' of https://github.com/ExtendLang/Extend into power
255 | | | | |
256 | | * | | commit e377567e16ab447e156073e63e80ddd17545de28
257 | | | \ \ \ Merge: 71f395d 6ad8512
258 | | | _/ / / Author: Nigel Schuster <Neitsch@users.noreply.github.com>
259 | / | | / / Date: Tue Dec 13 14:53:28 2016 -0500
260 | | | / /
261 | | | | Merge branch 'master' into power
262 | | | |
263 * | | | commit 6ad85121a7782e13817dfd58a7b7308cccf9bf0d
264 | \ \ \ \ \ Merge: 668a0eb 6a04209
265 | | | | | Author: Jared Samet <jared.samet@aya.yale.edu>
266 | | | | | Date: Tue Dec 13 14:53:11 2016 -0500
267 | | | | |

```

```

268 | | | | | Merge pull request #69 from ExtendLang/unop-unary-minus
269 | | | | |
270 | | | | | Unop unary minus
271 | | | | |
272 | | * | | commit aeab40daf88e72d6bd61d80553c17c0c2db73930
273 | | | / / Author: Nigel Schuster <nigel.schusters@googlemail.com>
274 | | | | Date: Tue Dec 13 14:55:57 2016 -0500
275 | | | |
276 | | | | Removed unnecessary level of indirection
277 | | | |
278 | | * | | commit 71f395dcefc8f064d9cd0700da3345fc9658e083
279 | | / / Author: Nigel Schuster <nigel.schusters@googlemail.com>
280 | / | | Date: Tue Dec 13 14:46:27 2016 -0500
281 | | |
282 | | | Power to the people of Extend
283 | | |
284 | | * | | commit 50ed49cab80e64801ddef4b2242de0aabf886912
285 | | / Author: oracleofnj <jared.samet@aya.yale.edu>
286 | | Date: Tue Dec 13 15:38:04 2016 -0500
287 | |
288 | | Merging in main
289 | |
290 | * | | commit 6a04209cc7ebdb2a96fc67b66fed983b10c44c8
291 | | \ Merge: edb0ecc 668a0eb
292 | | / Author: oracleofnj <jared.samet@aya.yale.edu>
293 | / | Date: Tue Dec 13 14:45:46 2016 -0500
294 | |
295 | | Fix merge conflict
296 | |
297 | * | | commit 668a0eb548dbab171c854c1e08c24628cc5f268c
298 | \ \ Merge: f873242 866b68f
299 | | | Author: Jared Samet <jared.samet@aya.yale.edu>
300 | | | Date: Tue Dec 13 14:37:19 2016 -0500
301 | | |
302 | | | Merge pull request #68 from ExtendLang/mod-div
303 | | |
304 | | | Modulo and division
305 | | |
306 | * | | commit 866b68f65b6e94cb063a1fb88918f62ebba68bb6
307 | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
308 | | | Date: Tue Dec 13 14:32:18 2016 -0500
309 | | |
310 | | | Added modulo and division operation
311 | | |
312 | * | | commit 84dfc3315078447c8df4b64a2f00cda7bef47cf1
313 | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
314 | | | Date: Tue Dec 13 14:26:25 2016 -0500
315 | | |
316 | | | Crunched some code
317 | | |
318 | * | | commit f4d5a8114d50acf5683d58629eca82a776c32480
319 | | \ Merge: fc94112 f873242
320 | | / Author: Nigel Schuster <nigel.schusters@googlemail.com>
321 | / | | Date: Tue Dec 13 14:22:12 2016 -0500
322 | | |
323 | | | Merge branch 'master' into simplification

```

```

324 | | |
325 | * | commit fc941122a7b42a66c6db83c23e9f8abfce9c5a25
326 | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
327 | | | Date: Tue Dec 13 14:20:35 2016 -0500
328 | | |
329 | | | Added multiplication
330 | | |
331 | | * commit edb0eccb71dc27ceacae7362802b9d3562dd1f9b
332 | | | Author: oracleofnj <jared.samet@aya.yale.edu>
333 | | | Date: Tue Dec 13 14:43:32 2016 -0500
334 | | |
335 | | | Add unary minus
336 | | |
337 | | * commit 46d5aa6b0827bf684873459b2ea347f43294c488
338 | | | \ Merge: 76210eb f873242
339 | | | / Author: oracleofnj <jared.samet@aya.yale.edu>
340 | / | | Date: Tue Dec 13 14:26:35 2016 -0500
341 | | |
342 | | | Merge branch 'master' into unop-typeof
343 | | |
344 * | | | commit f873242d23ab48c0e4230881d4524d53bdca5b52
345 | \ \ \ Merge: 4afd78e 6c26c2c
346 | | | | Author: Nigel Schuster <Neitsch@users.noreply.github.com>
347 | | | | Date: Tue Dec 13 14:21:26 2016 -0500
348 | | | |
349 | | | | Merge pull request #65 from ExtendLang/subtraction
350 | | | |
351 | | | | Addition of Subtraction
352 | | | |
353 | * | | | commit 6c26c2cae72e3780299f447d1258e27dc799e1a3
354 | | \ \ \ Merge: bd90241 4afd78e
355 | | / / / Author: Nigel Schuster <Neitsch@users.noreply.github.com>
356 | / | | | Date: Tue Dec 13 14:19:07 2016 -0500
357 | | | |
358 | | | | Merge branch 'master' into subtraction
359 | | | |
360 | * | | | commit bd90241ce837eb554b67204b86d9db76c9888a0f
361 | | \ \ \ Merge: 4042259 c7d4162
362 | | | | | Author: Nigel Schuster <Neitsch@users.noreply.github.com>
363 | | | | | Date: Tue Dec 13 14:14:17 2016 -0500
364 | | | | |
365 | | | | | Merge branch 'master' into subtraction
366 | | | | |
367 | * | | | | commit 4042259e9960bc7d4e9fd50a45c8f94231770339
368 | | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
369 | | | | | Date: Tue Dec 13 14:13:09 2016 -0500
370 | | | | |
371 | | | | | Added subtraction
372 | | | | |
373 | * | | | | commit 82a3db27594ad2acc9f8a275a26b8fd01937adde
374 | | \ \ \ \ Merge: 1elf973 cc40008
375 | | | | / / Author: Nigel Schuster <nigel.schusters@googlemail.com>
376 | | | | / | Date: Tue Dec 13 14:11:31 2016 -0500
377 | | | | |
378 | | | | | Merge branch 'master' into subtraction
379 | | | | |

```

```

380 | * | | | | commit 1e1f9734f1e8b482c8cd7074014aa4e5c506e19c
381 | | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
382 | | | | | Date: Tue Dec 13 13:44:36 2016 -0500
383 | | | | |
384 | | | | | Subtraction
385 | | | | |
386 | | | | | * commit 76210ebd86f466b3e2c3e837b5aa9d1c4776d56b
387 | | _ | _ | / Author: oracleofnj <jared.samet@aya.yale.edu>
388 | / | | | | Date: Tue Dec 13 14:26:18 2016 -0500
389 | | | | |
390 | | | | | Start on it
391 | | | | |
392 * | | | | | commit 4afd78e4535d88d3c131af4c4dfd5a1812ec17e9
393 | \ \ \ \ \ Merge: c7d4162 d4d4388
394 | | _ | _ | / Author: Nigel Schuster <Neitsch@users.noreply.github.com>
395 | / | | | | Date: Tue Dec 13 14:18:55 2016 -0500
396 | | | | |
397 | | | | | Merge pull request #64 from ExtendLang/refactor-boolean-binops
398 | | | | |
399 | | | | | Change Lt, Lte in grammar; implement GTE
400 | | | | |
401 | * | | | | commit d4d4388775ad5c1ce66fa729f5161d324c0da91c
402 | | \ \ \ \ Merge: 663f399 c7d4162
403 | | / / / Author: Nigel Schuster <Neitsch@users.noreply.github.com>
404 | / | | | | Date: Tue Dec 13 14:15:58 2016 -0500
405 | | | | |
406 | | | | | Merge branch 'master' into refactor-boolean-binops
407 | | | | |
408 * | | | | | commit c7d41623a82f3819c7a2104a8dabdd1ca17c2a5d6
409 | | \ \ \ \ Merge: cc40008 97821c8
410 | | _ | _ | / Author: Jared Samet <jared.samet@aya.yale.edu>
411 | / | | | | Date: Tue Dec 13 14:02:13 2016 -0500
412 | | | | |
413 | | | | | Merge pull request #63 from ExtendLang/more-binops
414 | | | | |
415 | | | | | More binops
416 | | | | |
417 | | * | | | commit 663f399e023f753acalc87d758cf4cbdd167e466
418 | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
419 | | | | | Date: Tue Dec 13 14:12:57 2016 -0500
420 | | | | |
421 | | | | | Remove wildcard from BinOp pattern match
422 | | | | |
423 | | * | | | commit 1bf6bedffc1776bc66e7dbc413bd74d293778ead
424 | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
425 | | | | | Date: Tue Dec 13 14:09:47 2016 -0500
426 | | | | |
427 | | | | | Add TransformedAway exception for LogAnd and LogOr
428 | | | | |
429 | | * | | | commit 952778e9205d92208cfb48b6b8c92c2c697cfb40
430 | | / / | | Author: oracleofnj <jared.samet@aya.yale.edu>
431 | | | | | Date: Tue Dec 13 14:01:54 2016 -0500
432 | | | | |
433 | | | | | Change Lt, Lte in grammar; implement GTE
434 | | | | |
435 | * | | | | commit 97821c8291bc07c3607f98179832fb05a89f6e8d

```

```

436 | | | Author: oracleofnj <jared.samet@aya.yale.edu>
437 | | | Date: Tue Dec 13 13:47:52 2016 -0500
438 | | |
439 | | | GT
440 | | |
441 | * | commit e0a883a96e63eea084f350d58e12a8d56be8e157
442 | / / Author: oracleofnj <jared.samet@aya.yale.edu>
443 | | | Date: Tue Dec 13 13:37:57 2016 -0500
444 | | |
445 | | | Remove NotEq from AST since != is parsed to UnOp(LogNot,BinOp(Eq,...))
446 | | |
447 * | | commit cc40008a9c11d7b084feb72657cf2a69fb5dea62
448 | \ \ Merge: a656f57 7123ebc
449 | | | Author: Nigel Schuster <Neitsch@users.noreply.github.com>
450 | | | Date: Tue Dec 13 12:49:33 2016 -0500
451 | | |
452 | | | Merge pull request #60 from ExtendLang/addition2
453 | | |
454 | | | String concatenation
455 | | |
456 | * | commit 7123ebcb45ebe8a8dc4a924a17760a8eaf016473
457 | \ \ Merge: eb134b3 a656f57
458 | / / Author: Nigel Schuster <Neitsch@users.noreply.github.com>
459 | / | Date: Tue Dec 13 12:41:09 2016 -0500
460 | | |
461 | | | Merge branch 'master' into addition2
462 | | |
463 * | | commit a656f57efd95a86d97de787aa6694f65f39e265a
464 | \ \ Merge: 81533f4 a64cc15
465 | |_/ Author: Jared Samet <jared.samet@aya.yale.edu>
466 | / | Date: Tue Dec 13 12:38:12 2016 -0500
467 | | |
468 | | | Merge pull request #61 from ExtendLang/debug-unop
469 | | |
470 | | | Add Debug expr
471 | | |
472 | * | commit a64cc15c6f95beaf1f4fe85112c4869a2b9b12fb
473 | / / Author: oracleofnj <jared.samet@aya.yale.edu>
474 | | | Date: Tue Dec 13 12:14:45 2016 -0500
475 | | |
476 | | | Add Debug expr
477 | | |
478 | * | commit eb134b35a63c3f58eb744c921a9235b11399fba3
479 | | | Author: Nigel Schuster <nigel.schusters@gmail.com>
480 | | | Date: Tue Dec 13 12:29:53 2016 -0500
481 | | |
482 | | | Moved testcases
483 | | |
484 | * | commit 044c6bdda6cba9100614b3c20ed355bdecfbf2095
485 | | | Author: Nigel Schuster <nigel.schusters@gmail.com>
486 | | | Date: Tue Dec 13 12:29:07 2016 -0500
487 | | |
488 | | | Fixed off by one error
489 | | |
490 | * | commit 59858a0952c52893b5b5245b158b426bbb5c4a4d
491 | | | Author: oracleofnj <jared.samet@aya.yale.edu>

```



```

492 | | Date: Tue Dec 13 11:33:12 2016 -0500
493 | |
494 | | Whoops no space
495 | |
496 | * | commit 0426f3411166f44f628bcb94bdde53ec4a6bb5ba
497 | | Author: oracleofnj <jared.samet@aya.yale.edu>
498 | | Date: Tue Dec 13 11:30:26 2016 -0500
499 | |
500 | | Add test case
501 | |
502 | * | commit 49ffa8688e78b8add3b67dbb36fe505f759eac17
503 | | \ Merge: 3cdaa5a 81533f4
504 | | / Author: Nigel Schuster <Neitsch@users.noreply.github.com>
505 | / | Date: Tue Dec 13 11:19:14 2016 -0500
506 | |
507 | | Merge branch 'master' into addition2
508 | |
509 | * | commit 81533f4fc65b6303d3bc21bf755814053e9212ec
510 | \ \ Merge: 200b8b6 64d1760
511 | | | Author: Jared Samet <jared.samet@aya.yale.edu>
512 | | | Date: Tue Dec 13 11:13:44 2016 -0500
513 | | |
514 | | | Merge pull request #59 from ExtendLang/equal-rights
515 | | |
516 | | | Equal rights
517 | | |
518 | * | commit 64d1760ba8810a2d57de49736980530e5d803441
519 | | | Author: oracleofnj <jared.samet@aya.yale.edu>
520 | | | Date: Tue Dec 13 11:04:55 2016 -0500
521 | | |
522 | | | Wake up please, GitHub
523 | | |
524 | * | commit 840aeaf2a7afdlb88756ad9bb391d7763921c76e
525 | | | Author: oracleofnj <jared.samet@aya.yale.edu>
526 | | | Date: Tue Dec 13 10:48:03 2016 -0500
527 | | |
528 | | | Remove usage demonstration
529 | | |
530 | * | commit 61ff4395ca4fef30147df4e8a7733f3eb07d3ed3
531 | | | Author: oracleofnj <jared.samet@aya.yale.edu>
532 | | | Date: Tue Dec 13 03:26:35 2016 -0500
533 | | |
534 | | | Add string equality and test cases
535 | | |
536 | * | commit f3112e98f1a16888694bae635e3c2d8707c83469
537 | | | Author: oracleofnj <jared.samet@aya.yale.edu>
538 | | | Date: Tue Dec 13 01:57:10 2016 -0500
539 | | |
540 | | | Reduce cut & paste
541 | | |
542 | * | commit 08ce677a9c4db365fefdf59a91c89b8c8bae5d73
543 | | | Author: oracleofnj <jared.samet@aya.yale.edu>
544 | | | Date: Tue Dec 13 01:35:46 2016 -0500
545 | | |
546 | | | Remove obsolete testing file
547 | | |

```

```

548 | * | commit ae8a07e0ccd6e9959d83a800af7e68ef7267e39c
549 | \| \ Merge: 862b38c 6090713
550 | | | Author: oracleofnj <jared.samet@aya.yale.edu>
551 | | | Date: Tue Dec 13 01:23:26 2016 -0500
552 | | |
553 | | | Merge branch 'print_value_p' into equal-rights
554 | | |
555 | | * | commit 6090713b9c7b49a8a58308cb4caaaef6a1817aa9
556 | | | Author: oracleofnj <jared.samet@aya.yale.edu>
557 | | | Date: Tue Dec 13 01:22:47 2016 -0500
558 | | |
559 | | | Use correct printf specifier
560 | | |
561 | * | | commit 862b38c5f0190732e311a967863ff8862e92490c
562 | \| \ \ Merge: 50281b1 5e913ad
563 | | | / / Author: oracleofnj <jared.samet@aya.yale.edu>
564 | | | Date: Tue Dec 13 01:19:14 2016 -0500
565 | | |
566 | | | Merge branch 'print_value_p' into equal-rights
567 | | |
568 | | * | commit 5e913ad2d9e42cc06417b1de641cb812d712e52a
569 | | / / Author: oracleofnj <jared.samet@aya.yale.edu>
570 | / | | Date: Tue Dec 13 01:16:07 2016 -0500
571 | | |
572 | | | Add debug_print; remove print statement that was causing us to falsely
    | | | pass test cases from to_string; show usage in UnOp(Neg)
573 | | |
574 | * | commit 50281b1d3047296e7f3c5ed362bd4126a37dd0a2
575 | | | Author: oracleofnj <jared.samet@aya.yale.edu>
576 | | | Date: Tue Dec 13 00:47:28 2016 -0500
577 | | |
578 | | | Numeric equality
579 | | |
580 | * | commit 0f76aa4eac383b95c8c9b05c7a1073b18c4e53b4
581 | / / Author: oracleofnj <jared.samet@aya.yale.edu>
582 | | Date: Mon Dec 12 22:30:15 2016 -0500
583 | |
584 | | Remove print flags
585 | |
586 | * | commit 200b8b6bf132a7d80f6545ab1a8063b95f8b8477
587 | \| \ Merge: 8834635 7e7276b
588 | | | Author: Nigel Schuster <Neitsch@users.noreply.github.com>
589 | | | Date: Mon Dec 12 22:16:15 2016 -0500
590 | | |
591 | | | Merge pull request #57 from ExtendLang/addition2
592 | | |
593 | | | Addition
594 | | |
595 | | * | commit 3cdaa5a45e4644004a3cb274fd43c0c56ecd38a4
596 | | | Author: Nigel Schuster <nigel.schusters@gmail.com>
597 | | | Date: Tue Dec 13 11:12:41 2016 -0500
598 | | |
599 | | | String addition
600 | | |
601 | | * | commit da7c5433dc305f65f46f6bacec96c351aeaef60c
602 | | / / Author: Nigel Schuster <nigel.schusters@gmail.com>

```

```

603 | | Date: Mon Dec 12 12:43:31 2016 -0500
604 | |
605 | | Setting flag for addition
606 | |
607 | * commit 7e7276ba7b8c816cc7c3c3d8af46948e96b14a16
608 | | \ Merge: 8aa125f 8834635
609 | | / Author: Nigel Schuster <nigel.schusters@googlemail.com>
610 | / | Date: Mon Dec 12 12:37:35 2016 -0500
611 | |
612 | | Merge branch 'master' into addition2
613 | |
614 * | commit 88346350bff1466a192f24dbb77f572a329c7204
615 | \ \ Merge: 6ed303e 53ae9e0
616 | | | Author: Nigel Schuster <Neitsch@users.noreply.github.com>
617 | | | Date: Mon Dec 12 10:18:51 2016 -0500
618 | | |
619 | | | Merge pull request #55 from ExtendLang/runtime
620 | | |
621 | | | Extracted runtime into seperate file
622 | | |
623 | * | commit 53ae9e05b3affda0148b3112b8b6362d15cabb0
624 | | \ Merge: d1e196d 6ed303e
625 | | / Author: Nigel Schuster <Neitsch@users.noreply.github.com>
626 | / | Date: Mon Dec 12 10:06:24 2016 -0500
627 | | |
628 | | | Merge branch 'master' into runtime
629 | | |
630 * | | commit 6ed303e17033f6525a77d6640b2da8c6bf806f7b
631 | \ \ \ Merge: ecc620e ae49ce6
632 | | | | Author: Jared Samet <jared.samet@aya.yale.edu>
633 | | | | Date: Mon Dec 12 09:43:57 2016 -0500
634 | | | |
635 | | | | Merge pull request #56 from ExtendLang/truthy-fix
636 | | | |
637 | | | | Truthy fix
638 | | | |
639 | * | | commit ae49ce6bba03f66e939fb33bdb7b6a9af58f1696
640 | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
641 | | | | Date: Mon Dec 12 01:15:29 2016 -0500
642 | | | |
643 | | | | Remove extra file
644 | | | |
645 | * | | commit 7fe6a2234a976d5d85333147b0016d4099f71f5d
646 | / / / Author: oracleofnj <jared.samet@aya.yale.edu>
647 | | | | Date: Mon Dec 12 01:11:53 2016 -0500
648 | | | |
649 | | | | Falsey fix
650 | | | |
651 * | | | commit ecc620eba7af7a55e6da7a222e6ef6a2a08c461d
652 | \ \ \ Merge: a875b41 4c8caa5
653 | | | | Author: Nigel Schuster <Neitsch@users.noreply.github.com>
654 | | | | Date: Mon Dec 12 00:17:06 2016 -0500
655 | | | |
656 | | | | Merge pull request #54 from ExtendLang/final-draft-for-real
657 | | | |
658 | | | | Final draft for real

```

```

659 | | | |
660 | * | | | commit 4c8caa516cd2dbf01475669aalca048ee1c90130
661 | | \ \ \ Merge: 04d3b57 a875b41
662 | | / / / Author: Jared Samet <jared.samet@aya.yale.edu>
663 | / | | | Date: Mon Dec 12 00:09:16 2016 -0500
664 | | | |
665 | | | | Merge branch 'master' into final-draft-for-real
666 | | | |
667 | * | | | commit 04d3b57dd8f38dafb9edb7999ed203e2c9921f26
668 | | \ \ \ Merge: 718ecd3 39025b0
669 | | | | | Author: Nigel Schuster <Neitsch@users.noreply.github.com>
670 | | | | | Date: Mon Dec 12 00:00:29 2016 -0500
671 | | | | |
672 | | | | | Merge pull request #39 from ExtendLang/more-lrm-ed
673 | | | | |
674 | | | | | More lrm edits
675 | | | | |
676 | | * | | | commit 39025b056938eb96d3643228d86ff1fc93a9ce24
677 | | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
678 | | | | | Date: Sun Dec 11 23:59:18 2016 -0500
679 | | | | |
680 | | | | | Fixed examples, made small corrections
681 | | | | |
682 | | * | | | commit a0ed757bb534a010db452bef1409ec536cb17637
683 | | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
684 | | | | | Date: Sat Dec 10 23:31:38 2016 -0500
685 | | | | |
686 | | | | | Edited explanation for row() and column()
687 | | | | |
688 | | * | | | commit 7c50ef28d766680fd30067187d01d27fda12e7fa
689 | | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
690 | | | | | Date: Sat Dec 10 23:27:07 2016 -0500
691 | | | | |
692 | | | | | Added info for strings
693 | | | | |
694 | | * | | | commit 738e41b433a38008360696f2ece253a91fafdbe6
695 | | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
696 | | | | | Date: Sat Dec 10 23:24:20 2016 -0500
697 | | | | |
698 | | | | | Added boolean example
699 | | | | |
700 | | * | | | commit 5377fdff3f9824062de5a599b59ea07f5e2d2d4b
701 | | / / / Author: Nigel Schuster <nigel.schusters@googlemail.com>
702 | | | | | Date: Sat Dec 10 23:19:26 2016 -0500
703 | | | | |
704 | | | | | Added arithmetic example
705 | | | | |
706 | * | | | | commit 718ecd3b1c641c7b5816718c142423fa900d07ca
707 | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
708 | | | | | Date: Sat Dec 10 03:09:18 2016 -0500
709 | | | | |
710 | | | | | Some changes to LRM; add if(a,b,c)
711 | | | | |
712 | | * | | | commit d1e196da9518ddc18fd6df57a52c91aef3e71fe8
713 | | / / | Author: Nigel Schuster <nigel.schusters@googlemail.com>
714 | / | | | Date: Mon Dec 12 00:23:13 2016 -0500

```

```

715 | | |
716 | | |      Extracted runtime into seperate file
717 | | |
718 * | |      commit a875b4172c1d79172be671c297c09718b244dbdb
719 |\ \ \      Merge: b95d14f 616dd34
720 | | |      Author: Jared Samet <jared.samet@aya.yale.edu>
721 | | |      Date:   Sun Dec 11 23:51:30 2016 -0500
722 | | |
723 | | |      Merge pull request #53 from ExtendLang/truthy
724 | | |
725 | | |      Truthy
726 | | |
727 | * |      commit 616dd34bebf74dfb7b74191141880b7cb68a7019
728 | \ \ \      Merge: 0fa8255 b95d14f
729 | / / /      Author: oracleofnj <jared.samet@aya.yale.edu>
730 | / | |      Date:   Sun Dec 11 23:15:54 2016 -0500
731 | | |
732 | | |      Merge branch 'master' into truthy
733 | | |
734 * | |      commit b95d14f5dd690433f23be5a71e0affebce6b1829
735 |\ \ \ \      Merge: 2a905c7 6dea96f
736 | | |      Author: Jared Samet <jared.samet@aya.yale.edu>
737 | | |      Date:   Sun Dec 11 21:02:28 2016 -0500
738 | | |
739 | | |      Merge pull request #50 from ExtendLang/builder-hotfix
740 | | |
741 | | |      So many builders
742 | | |
743 | | * |      commit 0fa8255b29077c769514973ad17fa7eb7fec9351
744 | | |      Author: oracleofnj <jared.samet@aya.yale.edu>
745 | | |      Date:   Sun Dec 11 23:14:42 2016 -0500
746 | | |
747 | | |      Apparently still needs some work
748 | | |
749 | | * |      commit 78584d7808785b7a8e7253788bdfc0240e96cfd9
750 | | |      Author: oracleofnj <jared.samet@aya.yale.edu>
751 | | |      Date:   Sun Dec 11 23:09:07 2016 -0500
752 | | |
753 | | |      Thanks a lot Travis
754 | | |
755 | | * |      commit b5673d22f67f84f159cd1c28288a195be81aada3
756 | | |      Author: oracleofnj <jared.samet@aya.yale.edu>
757 | | |      Date:   Sun Dec 11 22:51:52 2016 -0500
758 | | |
759 | | |      TERRRRRRRRR NARRRRRRR EEEEEEEEEEEEEEEEEEEEE
760 | | |
761 | | * |      commit b81bclbab064a134c831685c9d76c47a4b2cd69c
762 | | / / /      Author: oracleofnj <jared.samet@aya.yale.edu>
763 | | |      Date:   Sun Dec 11 22:04:25 2016 -0500
764 | | |
765 | | |      Maybe Truthy
766 | | |
767 | * |      commit 6dea96f0c2d42f131990b37e515f2e6df198d2e0
768 | / / /      Author: oracleofnj <jared.samet@aya.yale.edu>
769 | | |      Date:   Sun Dec 11 20:40:47 2016 -0500
770 | | |

```

```

771 | | | So many builders
772 | | |
773 * | | commit 2a905c7b5b89a9a2d490bda9782e1e0e54fb8cb3
774 |\ \ \ Merge: 18fc1be 2bc6c85
775 | | | Author: Jared Samet <jared.samet@aya.yale.edu>
776 | | | Date: Sun Dec 11 19:15:47 2016 -0500
777 | | |
778 | | | Merge pull request #47 from ExtendLang/function-parameter
779 | | |
780 | | | Function parameter
781 | | |
782 | * | | commit 2bc6c85a240f72c3bc040bb2d54e4bb375b1720b
783 | | | Author: oracleofnj <jared.samet@aya.yale.edu>
784 | | | Date: Sun Dec 11 19:11:33 2016 -0500
785 | | |
786 | | | Add combined test case
787 | | |
788 | * | | commit 860a11b3e4368c6f21337ccd055df4c00f502b0f
789 | |\ \ \ Merge: 8c3499e 18fc1be
790 | / / / Author: oracleofnj <jared.samet@aya.yale.edu>
791 | / | | Date: Sun Dec 11 19:04:35 2016 -0500
792 | | |
793 | | | Merge branch 'master' into function-parameter
794 | | |
795 | * | | commit 8c3499e3dd670ff1e16911b823cff1933452c71f
796 | | | Author: oracleofnj <jared.samet@aya.yale.edu>
797 | | | Date: Sun Dec 11 19:03:39 2016 -0500
798 | | |
799 | | | Remove extraneous printlines
800 | | |
801 | * | | commit 99418c0215673e964074547dced2e0f69a432b06
802 | | | Author: oracleofnj <jared.samet@aya.yale.edu>
803 | | | Date: Sun Dec 11 19:02:31 2016 -0500
804 | | |
805 | | | Make function parameters work
806 | | |
807 | * | | commit 387559b44107284c1056db47378bf9b79458cb93
808 | | | Author: oracleofnj <jared.samet@aya.yale.edu>
809 | | | Date: Sun Dec 11 18:39:00 2016 -0500
810 | | |
811 | | | First attempt
812 | | |
813 | | | * commit 8aa125f1d5292eb3f7aa41be186d761248134535
814 | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
815 | | | Date: Sun Dec 11 20:15:52 2016 -0500
816 | | |
817 | | | Made som rpgroess
818 | | |
819 | | | * commit 6c00a72a2912c7f38af73fc49070f63e05243ca6
820 | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
821 | / | | Date: Sun Dec 11 18:45:46 2016 -0500
822 | | |
823 | | | Some progress
824 | | |
825 | * | | commit 18fc1beab2196a2a45d6ce4cdf2b528f2545c03e
826 |\ \ \ Merge: f1dd8a5 f7e9be8

```

```

827 | / / / Author: Nigel Schuster <Neitsch@users.noreply.github.com>
828 | / | | Date: Sun Dec 11 18:08:11 2016 -0500
829 | | |
830 | | | Merge pull request #45 from ExtendLang/empty
831 | | |
832 | | | Implemented empty, small flag setting fix
833 | | |
834 | * | | commit f7e9be8c20044c73ea00ce076bd075a24e809dfa
835 | | \ \ Merge: 18db166 fldd8a5
836 | / / / Author: Nigel Schuster <Neitsch@users.noreply.github.com>
837 | / | | Date: Sun Dec 11 16:30:05 2016 -0500
838 | | |
839 | | | Merge branch 'master' into empty
840 | | |
841 | * | | commit fldd8a5ca9976def254e61530ed91c650c09070b
842 | | \ \ Merge: 3c4681d 50366f4
843 | | | | Author: Jared Samet <jared.samet@aya.yale.edu>
844 | | | | Date: Sun Dec 11 16:18:44 2016 -0500
845 | | | |
846 | | | | Merge pull request #46 from ExtendLang/actually-make-global-scope
847 | | | |
848 | | | | Actually make global scope
849 | | | |
850 | * | | commit 50366f43d98220a48d90a6faa884c70e68778627
851 | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
852 | | | | Date: Sun Dec 11 15:38:05 2016 -0500
853 | | | |
854 | | | | Make sure locals are properly masking globals
855 | | | |
856 | * | | commit 046c7cc0ca4ad773ac83b7677dd7f14e3d8c9337
857 | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
858 | | | | Date: Sun Dec 11 15:30:53 2016 -0500
859 | | | |
860 | | | | Make globals work, fix bug
861 | | | |
862 | * | | commit a844a4635358aeba7edd4f6faa83871036c1fee4
863 | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
864 | | | | Date: Sun Dec 11 15:14:09 2016 -0500
865 | | | |
866 | | | | So close
867 | | | |
868 | * | | commit 67849f0177f0c67cd19a61121b9d971bab6a0474
869 | / / / Author: oracleofnj <jared.samet@aya.yale.edu>
870 | | | | Date: Sun Dec 11 15:01:52 2016 -0500
871 | | | |
872 | | | | Make the global scope object
873 | | | |
874 | * | | commit 18db1667adb6a80cc1d0bbb4b6b657ec6db0200e
875 | | \ \ Merge: 393d02c 3c4681d
876 | / / / Author: Nigel Schuster <Neitsch@users.noreply.github.com>
877 | / | | Date: Sun Dec 11 15:05:42 2016 -0500
878 | | | |
879 | | | | Merge branch 'master' into empty
880 | | | |
881 | * | | commit 3c4681d6a7aaf164d403b3148822d8992783754d
882 | | \ \ Merge: abcffd0 7be1001

```

```

883 | | | | Author: Jared Samet <jared.samet@aya.yale.edu>
884 | | | | Date: Sun Dec 11 13:31:12 2016 -0500
885 | | | |
886 | | | | Merge pull request #44 from ExtendLang/float-display-hotfix
887 | | | |
888 | | | | Floating point math hotfix
889 | | | |
890 | * | | | commit 7be1001d43b1082b4cc2fa6801f619f99f0679c8
891 | | \ \ \ Merge: 556da44 abcffd0
892 | | / / / Author: Jared Samet <jared.samet@aya.yale.edu>
893 | / | | | Date: Sun Dec 11 13:26:55 2016 -0500
894 | | | |
895 | | | | Merge branch 'master' into float-display-hotfix
896 | | | |
897 | * | | | commit abcffd04216b35160b67c1f7ca29de8020a57774
898 | | \ \ \ Merge: d65aad4 0ad195e
899 | | | | | Author: Jared Samet <jared.samet@aya.yale.edu>
900 | | | | | Date: Sun Dec 11 13:19:05 2016 -0500
901 | | | | |
902 | | | | | Merge pull request #42 from ExtendLang/encapsulate-build-scope
903 | | | | |
904 | | | | | Encapsulate a little more of building the scope
905 | | | | |
906 | * | | | | commit 0ad195e79e1bce957f7d265e7365de0ffc718988
907 | | \ \ \ \ Merge: 9caf464 d65aad4
908 | | / / / / Author: oracleofnj <jared.samet@aya.yale.edu>
909 | / | | | | Date: Sun Dec 11 12:42:42 2016 -0500
910 | | | | |
911 | | | | | Merge branch 'master' into encapsulate-build-scope
912 | | | | |
913 | * | | | | commit 9caf4645cf9a747101bed8801917dda953b3df5e
914 | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
915 | | | | | Date: Sun Dec 11 12:41:40 2016 -0500
916 | | | | |
917 | | | | | Encapsulate a little more of building the scope
918 | | | | |
919 | | * | | | commit 556da44ef36ec5b4b93aade0ef2bb2372f0cd0a4
920 | | / / / / Author: oracleofnj <jared.samet@aya.yale.edu>
921 | / | | | | Date: Sun Dec 11 13:18:15 2016 -0500
922 | | | | |
923 | | | | | Floating point math hotfix
924 | | | | |
925 | | * | | | commit 393d02c61efdaa9e246509fef8b3b61303bf0b96
926 | | / / / Author: Nigel Schuster <nigel.schusters@googlemail.com>
927 | / | | | | Date: Sun Dec 11 14:25:02 2016 -0500
928 | | | | |
929 | | | | | Implemented empty, small flag setting fix
930 | | | | |
931 | * | | | | commit d65aad46b60dde24fdc2740ec1e26676fa03a2e0
932 | | \ \ \ Merge: 9cee2fc 0f5a6ba
933 | | / / / Author: Jared Samet <jared.samet@aya.yale.edu>
934 | | | | | Date: Sun Dec 11 12:09:28 2016 -0500
935 | | | | |
936 | | | | | Merge pull request #40 from ExtendLang/make-global-scope
937 | | | | |
938 | | | | | Make global scope

```



```

939 | | |
940 | * | commit 0f5a6ba3e6c1e3e29a4ed02f70cf418ce059bd4b
941 | | \ Merge: 56b58d9 9cee2fc
942 | | / Author: oracleofnj <jared.samet@aya.yale.edu>
943 | / | Date: Sun Dec 11 12:04:05 2016 -0500
944 | | |
945 | | | Merge branch 'master' into make-global-scope
946 | | |
947 * | | commit 9cee2fc6bb72ad0cf074bb6b16faa4759778f5d1
948 | | | Author: kevinyel <kevinyel@users.noreply.github.com>
949 | | | Date: Sun Dec 11 10:07:36 2016 -0500
950 | | |
951 | | | Testcases (#38)
952 | | |
953 | | | * Updated testcases with to_string method
954 | | |
955 | | | * Merged with master
956 | | |
957 | * | commit 56b58d9b121d5b8e26a25bdb45572b9de5ed89bb
958 | | | Author: oracleofnj <jared.samet@aya.yale.edu>
959 | | | Date: Sun Dec 11 12:01:28 2016 -0500
960 | | |
961 | | | Encapsulate build_var_defns
962 | | |
963 | * | commit f25e5b377975ab5473d0592720ded03b7bce259f
964 | | | Author: oracleofnj <jared.samet@aya.yale.edu>
965 | | | Date: Sun Dec 11 11:43:19 2016 -0500
966 | | |
967 | | | Only construct var_defns once
968 | | |
969 | * | commit f3f4bef9eb2db0313d80dfbe6bdbf83f3a42076d
970 | | | Author: oracleofnj <jared.samet@aya.yale.edu>
971 | | | Date: Sun Dec 11 00:45:44 2016 -0500
972 | | |
973 | | | Make global variable to hold vardefns
974 | | |
975 | * | commit a8f4ad93aa0dd3365b6661a16035f40a8a1c5587
976 | | | Author: oracleofnj <jared.samet@aya.yale.edu>
977 | | | Date: Sat Dec 10 21:28:18 2016 -0500
978 | | |
979 | | | Isolate the part of building a scope for reuse with global variables
980 | | |
981 | * | commit 58f7a4d3c4aece7e9f12809f50479a2b8267f01d
982 | / / Author: Nigel Schuster <nigel.schusters@googlemail.com>
983 | | | Date: Sat Dec 10 18:05:01 2016 -0500
984 | | |
985 | | | Performing copy before returning, so that memory can be freed with alloca
986 | | |
987 * | | commit c0e56aa3b9853d9e44a27109f8c2666d752afe7b
988 | \ \ Merge: ef0e5e7 a4b35df
989 | | | Author: Nigel Schuster <Neitsch@users.noreply.github.com>
990 | | | Date: Sat Dec 10 17:07:00 2016 -0500
991 | | |
992 | | | Merge pull request #37 from ExtendLang/dereference
993 | | |
994 | | | Dereferencing 1x1 subrange

```

```

995 | | |
996 | * | commit a4b35df069c5c5243c499cd7aaeda27973678cb6
997 | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
998 | | | Date: Sat Dec 10 16:42:17 2016 -0500
999 | | |
1000 | | | Removed obsolete methods
1001 | | |
1002 | * | commit cf08a8cc461f1aa54f7f408c78bbf0e20269ccca
1003 | | \ Merge: ce833d4 ef0e5e7
1004 | | / Author: Nigel Schuster <Neitsch@users.noreply.github.com>
1005 | / | Date: Sat Dec 10 16:36:20 2016 -0500
1006 | | |
1007 | | | Merge branch 'master' into dereference
1008 | | |
1009 | * | commit ef0e5e7bc6c971c65c8f48460b27479aebb56075
1010 | | \ Merge: 0177dc2 38ba6e6
1011 | | | Author: Nigel Schuster <Neitsch@users.noreply.github.com>
1012 | | | Date: Sat Dec 10 16:36:03 2016 -0500
1013 | | |
1014 | | | Merge pull request #36 from ExtendLang/comp-warn
1015 | | |
1016 | | | Suppressing compiler warnings
1017 | | |
1018 | * | commit 38ba6e64fbc18f17f1611c695655b96670cda5b2
1019 | | / Author: Nigel Schuster <nigel.schusters@googlemail.com>
1020 | / | Date: Sat Dec 10 13:18:39 2016 -0500
1021 | | |
1022 | | | Suppressing compiler warnings
1023 | | |
1024 | * | commit 0177dc290c5e56e428fa135e0b0b9a4495c1e05f
1025 | | \ Merge: 127f99d e259556
1026 | | | Author: Nigel Schuster <Neitsch@users.noreply.github.com>
1027 | | | Date: Sat Dec 10 16:35:50 2016 -0500
1028 | | |
1029 | | | Merge pull request #35 from ExtendLang/linker
1030 | | |
1031 | | | Linker
1032 | | |
1033 | * | commit e2595560863fe9fff107818f713b5e8508d0c768
1034 | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
1035 | | | Date: Sat Dec 10 13:53:12 2016 -0500
1036 | | |
1037 | | | Removed nodefaultlibs directive
1038 | | |
1039 | * | commit 09c39619d89f65a19d537336c3b3680553f732a0
1040 | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
1041 | | | Date: Sat Dec 10 13:50:19 2016 -0500
1042 | | |
1043 | | | Modified linker to work for travis
1044 | | |
1045 | * | commit 36d662a599754648d00c4d773fb16eala59d335f
1046 | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
1047 | | | Date: Sat Dec 10 13:37:27 2016 -0500
1048 | | |
1049 | | | Attempt to link math
1050 | | |

```

```

1051 | * | | commit 2d4564a4f9aae99f67f72baf514b7095b452ee08
1052 | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
1053 | | | | Date: Sat Dec 10 13:22:14 2016 -0500
1054 | | | |
1055 | | | | Linking math library
1056 | | | |
1057 | * | | commit 9deac9bbd385607cbd5d80a97e929bc52f7f9633
1058 | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
1059 | | | | Date: Sat Dec 10 13:06:17 2016 -0500
1060 | | | |
1061 | | | | Modified compile script. Removed debug output
1062 | | | |
1063 | * | | commit d35607b7b7830c502d4af4d1d9c65c70bfa37d96
1064 | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
1065 | | | | Date: Sat Dec 10 13:04:30 2016 -0500
1066 | | | |
1067 | | | | Simplifier testscript
1068 | | | |
1069 | * | | commit d37dac2f40ea9f0013dd69b6ce40197fe4ec82ab
1070 | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
1071 | | | | Date: Sat Dec 10 12:36:45 2016 -0500
1072 | | | |
1073 | | | | Fixed duplicate import issue
1074 | | | |
1075 | * | | commit 31c26bccd855332e33505e8f887727069a450a00
1076 | | / / Author: Nigel Schuster <nigel.schusters@googlemail.com>
1077 | | | Date: Sat Dec 10 12:30:29 2016 -0500
1078 | | |
1079 | | | Added cmd args to link file
1080 | | |
1081 | * | | commit 127f99dc5c44b0cc673b7c254847c54b8a23610b
1082 | \ \ \ Merge: b2e881d a350720
1083 | | | | Author: Nigel Schuster <Neitsch@users.noreply.github.com>
1084 | | | | Date: Sat Dec 10 16:35:41 2016 -0500
1085 | | | |
1086 | | | | Merge pull request #34 from ExtendLang/rel-import
1087 | | | |
1088 | | | | Import path switch
1089 | | | |
1090 | * | | commit a350720f73027f0b5d99b73492d4326081324d84
1091 | | / / Author: Nigel Schuster <nigel.schusters@googlemail.com>
1092 | | | Date: Sat Dec 10 11:40:50 2016 -0500
1093 | | |
1094 | | | | Switched import style from root directory to relative path
1095 | | | |
1096 | * | | commit b2e881df8c2d033bbe299575b6d9092ba7ab51ea
1097 | \ \ \ Merge: 6a8f836 90e39b0
1098 | | / / Author: Nigel Schuster <Neitsch@users.noreply.github.com>
1099 | / | | Date: Sat Dec 10 16:35:31 2016 -0500
1100 | | |
1101 | | | | Merge pull request #33 from ExtendLang/ts-fix
1102 | | | |
1103 | | | | Testscript fix
1104 | | | |
1105 | * | | commit 90e39b0ba32729808fa7f835ce8adf9784ec5fb4
1106 | / / Author: Nigel Schuster <nigel.schusters@googlemail.com>

```

```

1107 | | Date: Sat Dec 10 11:24:19 2016 -0500
1108 | |
1109 | | Fixed issue in testscript that might report false results when it fails
      | | early
1110 | |
1111 | * commit ce833d4433f8b0e2ea4782478a3b6d61e85ec0d5
1112 | / Author: Nigel Schuster <nigel.schusters@googlemail.com>
1113 | Date: Sat Dec 10 16:14:34 2016 -0500
1114 |
1115 | Dereferencing lxl subrange
1116 |
1117 * commit 6a8f8363850840afc597a0b97853ae66a8709683
1118 | \ Merge: eac9e77 fc886a9
1119 | | Author: Jared Samet <jared.samet@aya.yale.edu>
1120 | | Date: Fri Dec 9 18:29:22 2016 -0500
1121 | |
1122 | | Merge pull request #24 from ExtendLang/final-draft-lrm
1123 | |
1124 | | Almost the final draft!
1125 | |
1126 | * commit fc886a96416ec9513edbelafac1c94d39b41ae02
1127 | | \ Merge: eac9e77 cda63cb
1128 | / / Author: oracleofnj <jared.samet@aya.yale.edu>
1129 | | Date: Fri Dec 9 18:23:52 2016 -0500
1130 | |
1131 | | Merge branch 'final-draft-lrm'
1132 | |
1133 | * commit cda63cb3e7aeb7efa06b4af110a9064709b384ac
1134 | | \ Merge: fb18949 eac9e77
1135 | | / Author: oracleofnj <jared.samet@aya.yale.edu>
1136 | / | Date: Fri Dec 9 18:23:24 2016 -0500
1137 | |
1138 | | Fix merge conflict
1139 | |
1140 * | commit eac9e77c743d4730ef72b6fa7f4a72304808e35d
1141 | \ \ Merge: 90fc58e fe825f4
1142 | | | Author: Nigel Schuster <Neitsch@users.noreply.github.com>
1143 | | | Date: Fri Dec 9 18:04:08 2016 -0500
1144 | | |
1145 | | | Merge pull request #29 from ExtendLang/refactor
1146 | | |
1147 | | | Made new execution strategy work
1148 | | |
1149 | * | commit fe825f42d95f8d620d082764118657e3791b33ff
1150 | | | Author: oracleofnj <jared.samet@aya.yale.edu>
1151 | | | Date: Fri Dec 9 17:55:39 2016 -0500
1152 | | |
1153 | | | Compact last bit
1154 | | |
1155 | * | commit b02dbbe930e7ef9bf5a4e454a2e367aa0b50fcel
1156 | | | Author: oracleofnj <jared.samet@aya.yale.edu>
1157 | | | Date: Fri Dec 9 17:49:00 2016 -0500
1158 | | |
1159 | | | Give formula functions names
1160 | | |
1161 | * | commit edd7aa40ee048a31d84a1f14c73a96d0722c242b

```

```

1162 | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
1163 | | | Date:   Fri Dec 9 17:40:57 2016 -0500
1164 | | |
1165 | | |     Removed artifcats
1166 | | |
1167 | * | commit 9b49e202be310550b9391dc6de3ad1aee961f53a
1168 | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
1169 | | | Date:   Fri Dec 9 17:37:59 2016 -0500
1170 | | |
1171 | | |     Fixed I/O testcases
1172 | | |
1173 | * | commit a4ad4b19eae1f9bfaeae27c3a6810b26da856fee
1174 | | \ Merge: b07398b ed01567
1175 | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
1176 | | | Date:   Fri Dec 9 17:18:13 2016 -0500
1177 | | |
1178 | | |     Merge
1179 | | |
1180 | | * | commit ed01567417fae9a616e32e1d08614fa08424a6bd
1181 | | | Author: oracleofnj <jared.samet@aya.yale.edu>
1182 | | | Date:   Fri Dec 9 17:17:06 2016 -0500
1183 | | |
1184 | | |     Make sizeof not break tests
1185 | | |
1186 | * | commit b07398bd13daed37fc65f0317100f816940257c9
1187 | | / Author: Nigel Schuster <nigel.schusters@googlemail.com>
1188 | | | Date:   Fri Dec 9 17:17:19 2016 -0500
1189 | | |
1190 | | |     Added macro for function definition
1191 | | |
1192 | * | commit a0a7054e9ce6d82ee760a64a0e3cac3d1ae89c85
1193 | | | Author: oracleofnj <jared.samet@aya.yale.edu>
1194 | | | Date:   Fri Dec 9 17:01:20 2016 -0500
1195 | | |
1196 | | |     Use symbol table
1197 | | |
1198 | * | commit 56fd61b31aa673bc532a40300d61349e0205bb53
1199 | | \ Merge: 38aedba dfb702e
1200 | | | Author: oracleofnj <jared.samet@aya.yale.edu>
1201 | | | Date:   Fri Dec 9 16:11:10 2016 -0500
1202 | | |
1203 | | |     Merge branch 'refactor' of https://github.com/ExtendLang/Extend into
1204 | | | refactor
1205 | | * | commit dfb702e023849665c7fc46b0488c85cc8c46c26f
1206 | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
1207 | | | Date:   Fri Dec 9 16:01:08 2016 -0500
1208 | | |
1209 | | |     Converted more to value_p from subrange_p
1210 | | |
1211 | | * | commit e963186b1712b1531c8982c35e2ead98e2da3907
1212 | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
1213 | | | Date:   Fri Dec 9 15:42:35 2016 -0500
1214 | | |
1215 | | |     Made example TC work
1216 | | |

```

```

1217 | * | | commit 38aedba9bfcd2f48a1868fe8f256d9e512dcc0cc
1218 | | / / Author: oracleofnj <jared.samet@aya.yale.edu>
1219 | | | Date: Fri Dec 9 16:10:35 2016 -0500
1220 | | |
1221 | | | Create symbol table
1222 | | |
1223 | * | commit eb76234747c9b70a41891853ddfd4730f1950a2c
1224 | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
1225 | | | Date: Fri Dec 9 11:14:58 2016 -0500
1226 | | |
1227 | | | Made Hello World work again
1228 | | |
1229 | * | commit 08aeb70eac43c1fd2e1b8e9f6ead7bellfbc9c12
1230 | | | Author: oracleofnj <jared.samet@aya.yale.edu>
1231 | | | Date: Fri Dec 9 02:13:09 2016 -0500
1232 | | |
1233 | | | Done for the night
1234 | | |
1235 | * | commit cb39114298fe29c6652e422d86d663482b2d2961
1236 | | | Author: oracleofnj <jared.samet@aya.yale.edu>
1237 | | | Date: Fri Dec 9 01:35:36 2016 -0500
1238 | | |
1239 | | | More refactoring
1240 | | |
1241 | * | commit 7974bbdf6d9be11657ebf61e534f331efd920ff2
1242 | | | Author: oracleofnj <jared.samet@aya.yale.edu>
1243 | | | Date: Thu Dec 8 23:53:31 2016 -0500
1244 | | |
1245 | | | Banish the term extern
1246 | | |
1247 | * | commit 49af9724790d0c38ccfcf336c07014f2165948ed
1248 | | | Author: oracleofnj <jared.samet@aya.yale.edu>
1249 | | | Date: Thu Dec 8 23:45:30 2016 -0500
1250 | | |
1251 | | | Add a couple comments
1252 | | |
1253 | * | commit 0fbf461200018cf1345a9b3ee2b5efd37189c2ce
1254 | | | Author: oracleofnj <jared.samet@aya.yale.edu>
1255 | | | Date: Thu Dec 8 21:52:24 2016 -0500
1256 | | |
1257 | | | Get my bearings
1258 | | |
1259 | * | commit 5ecb59941a875361553eb25544f92aaaf5231ca0
1260 | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
1261 | | | Date: Thu Dec 8 19:47:51 2016 -0500
1262 | | |
1263 | | | Added some documentation
1264 | | |
1265 | * | commit 65066fc265e8c4e89272ce050c4424a4873b0674
1266 | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
1267 | | | Date: Thu Dec 8 12:18:57 2016 -0500
1268 | | |
1269 | | | Added name display for variable
1270 | | |
1271 | * | commit f985cc8022350d02fd0c3091aa8e10b85f334c38
1272 | | \ Merge: 4b58ce9 a65c24e

```

```

1273 | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
1274 | | | | Date:   Wed Dec 7 12:14:59 2016 -0500
1275 | | | |
1276 | | | |      Merge branch 'finish-transformations' into get-val-rev
1277 | | | |
1278 | | * | | commit a65c24ed8b30ed639f67913bdd1ffef1b31b0437
1279 | | | \ \ Merge: 29d02d9 90fc58e
1280 | | _|/ / Author: oracleofnj <jared.samet@aya.yale.edu>
1281 | /| | | Date:   Tue Dec 6 16:14:10 2016 -0500
1282 | | | |
1283 | | | |      Merge branch 'master' into finish-transformations
1284 | | | |
1285 | | * | | commit 29d02d923ffa6af98aacbd2375d6e533f6035ccf
1286 | | | \ \ Merge: 0e8398f 52e7a8a
1287 | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
1288 | | | | | Date:   Mon Dec 5 13:34:27 2016 -0500
1289 | | | | |
1290 | | | | |      Fix merge conflict - keep expr_loc
1291 | | | | |
1292 | | * | | commit 0e8398f46809eb8f11d393a663f0096966d2573e
1293 | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
1294 | | | | | Date:   Sun Dec 4 13:23:27 2016 -0500
1295 | | | | |
1296 | | | | |      Transform all LHS expressions including integers to IDs; check for
1297 | | | | | strings or range literals and disallow
1298 | | * | | commit 36f5848cd9bd8421e6a61e5b79553c5b0aeeac8e
1299 | | | \ \ \ Merge: 330bec3 2ae2b83
1300 | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
1301 | | | | | Date:   Sat Dec 3 14:07:39 2016 -0500
1302 | | | | |
1303 | | | | |      Merge branch 'master' into finish-transformations
1304 | | | | |
1305 | | * | | | commit 330bec3fea9d441236961977886b8c8e7d09df75
1306 | | | \ \ \ \ Merge: 9702e5b 8a60995
1307 | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
1308 | | | | | Date:   Fri Dec 2 13:49:34 2016 -0500
1309 | | | | |
1310 | | | | |      Merge branch 'master' into finish-transformations
1311 | | | | |
1312 | | * | | | | commit 9702e5b56018c6187cb0e9421889369f9c867bd6
1313 | | | \ \ \ \ \ Merge: 047cfec 5bdd52c
1314 | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
1315 | | | | | Date:   Thu Dec 1 21:14:35 2016 -0500
1316 | | | | |
1317 | | | | |      Merge branch 'master' into finish-transformations
1318 | | | | |
1319 | | * | | | | commit 047cfec9cf0a8e645b8b61d52238a107a479db38
1320 | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
1321 | | | | | Date:   Thu Dec 1 18:42:04 2016 -0500
1322 | | | | |
1323 | | | | |      Add short circuiting test cases
1324 | | | | |
1325 | | * | | | | commit 6acd7f624a1db65cba846ea8745a6457f638ce7c
1326 | | | \ \ \ \ \ Merge: 5762112 72360f4
1327 | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>

```

```

1328 | | | | | | | | Date: Thu Dec 1 18:31:33 2016 -0500
1329 | | | | | | | |
1330 | | | | | | | | Merge remote-tracking branch 'origin/fail-silent' into finish-
      transformations
1331 | | | | | | | |
1332 | | * | | | | | | commit 57621128490ab146c2c82339fe3edfde4fdb0029
1333 | | | | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
1334 | | | | | | | | Date: Thu Dec 1 16:04:06 2016 -0500
1335 | | | | | | | |
1336 | | | | | | | | Get rid of wildcard pattern match in interpreter
1337 | | | | | | | |
1338 | | * | | | | | | commit a90a34323a7fe253998bbffb3eee8e8212017587
1339 | | | \ \ \ \ \ \ \ Merge: 85bc21d 7abb394
1340 | | | | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
1341 | | | | | | | | Date: Thu Dec 1 15:59:40 2016 -0500
1342 | | | | | | | |
1343 | | | | | | | | Merge branch 'master' into finish-transformations
1344 | | | | | | | |
1345 | | * | | | | | | commit 85bc21d8c88af6bdce2a7ca4eec8939dad0783a2
1346 | | | | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
1347 | | | | | | | | Date: Thu Dec 1 15:59:05 2016 -0500
1348 | | | | | | | |
1349 | | | | | | | | Remove unnecessary file
1350 | | | | | | | |
1351 | | * | | | | | | commit 81fe5654c032c936924bc103c97d1ae4cd23409c
1352 | | | | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
1353 | | | | | | | | Date: Thu Dec 1 15:58:40 2016 -0500
1354 | | | | | | | |
1355 | | | | | | | | Finish range literals
1356 | | | | | | | |
1357 | | * | | | | | | commit da41520e92a27c3ceeaf63b6fb32bda57d16baf0
1358 | | | | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
1359 | | | | | | | | Date: Thu Dec 1 14:54:21 2016 -0500
1360 | | | | | | | |
1361 | | | | | | | | So close
1362 | | | | | | | |
1363 | | * | | | | | | commit 2cabadcf4a488b40a33944794bebe5a3afec6344
1364 | | | \ \ \ \ \ \ \ Merge: cf36f70 13cd317
1365 | | | | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
1366 | | | | | | | | Date: Thu Dec 1 11:58:03 2016 -0500
1367 | | | | | | | |
1368 | | | | | | | | Merge branch 'master' into finish-transformations
1369 | | | | | | | |
1370 | | * | | | | | | commit cf36f702053bdb168c51d2ae78d62a013f73cc4d
1371 | | | | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
1372 | | | | | | | | Date: Thu Dec 1 09:47:38 2016 -0500
1373 | | | | | | | |
1374 | | | | | | | | Sample digits function
1375 | | | | | | | |
1376 | | * | | | | | | commit febcbff889be83fe6edc63cbffdaefc0d7b02d519
1377 | | | | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
1378 | | | | | | | | Date: Wed Nov 30 15:54:45 2016 -0500
1379 | | | | | | | |
1380 | | | | | | | | Add oddball formula test case and try out theory for range
      literal
1381 | | | | | | | |

```



```

1490 | | | | | | | | | |
1491 | * | | | | | | | | | commit 1ce7f83e9cc542daba3b0395f2bebbc8c779e473
1492 | | | | | | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
1493 | | | | | | | | | | Date: Mon Dec 5 14:18:41 2016 -0500
1494 | | | | | | | | | |
1495 | | | | | | | | | | Create patch file
1496 | | | | | | | | | |
1497 | * | | | | | | | | | commit f1b1lee94f4792ff5a6ddf2e837fffe1a3396dbe
1498 | | | | | | | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
1499 | | | | | | | | | | Date: Mon Dec 5 12:46:35 2016 -0500
1500 | | | | | | | | | |
1501 | | | | | | | | | | Skeleton for get_val
1502 | | | | | | | | | |
1503 | | | | | | | | | * commit fb1894999d655ab960f63a01c0c820373b6d8c1a
1504 | | | | | | | | | \ Merge: 4aab3dc 90fc58e
1505 | | _|_|_|_|_|_|_|_|_|/ Author: oracleofnj <jared.samet@aya.yale.edu>
1506 | /| | | | | | | | | Date: Wed Dec 7 23:44:17 2016 -0500
1507 | | | | | | | | | |
1508 | | | | | | | | | | Merge branch 'master' into final-draft-lrm
1509 | | | | | | | | | |
1510 | * | | | | | | | | | commit 90fc58e52cb1bdec2ea03f5b81c0bbda9c0f954b
1511 | \ \ \ \ \ \ \ \ \ \ Merge: 910bd01 04c2c65
1512 | | _|/ / / / / / / / / Author: Jared Samet <jared.samet@aya.yale.edu>
1513 | /| | | | | | | | | Date: Mon Dec 5 22:14:41 2016 -0500
1514 | | | | | | | | | |
1515 | | | | | | | | | | Merge pull request #23 from ExtendLang/read-empty
1516 | | | | | | | | | |
1517 | | | | | | | | | | Read empty
1518 | | | | | | | | | |
1519 | * | | | | | | | | | commit 04c2c655bfe133f98aa03b93739a0545a8dc8e37
1520 | | | | | | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
1521 | | | | | | | | | | Date: Mon Dec 5 15:53:35 2016 -0500
1522 | | | | | | | | | |
1523 | | | | | | | | | | Add slurp by passing 0 max bytes
1524 | | | | | | | | | |
1525 | * | | | | | | | | | commit d8cf3168a0145845d437816b173f36d8e6bbbded
1526 | / / / / / / / / / / Author: oracleofnj <jared.samet@aya.yale.edu>
1527 | | | | | | | | | | Date: Mon Dec 5 14:46:46 2016 -0500
1528 | | | | | | | | | |
1529 | | | | | | | | | | Start handling empty
1530 | | | | | | | | | |
1531 | | | | | | | | | * commit 4aab3dcdf8fdef825fc6f517c906c714925fff95
1532 | | | | | | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
1533 | | | | | | | | | | Date: Wed Dec 7 23:43:25 2016 -0500
1534 | | | | | | | | | |
1535 | | | | | | | | | | Update PDF
1536 | | | | | | | | | |
1537 | | | | | | | | | * commit ed44d27981de0127f73a6cc6f3f32cbf6bcac8ce
1538 | | | | | | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
1539 | | | | | | | | | | Date: Wed Dec 7 23:43:01 2016 -0500
1540 | | | | | | | | | |
1541 | | | | | | | | | | Fix failing test cases
1542 | | | | | | | | | |
1543 | | | | | | | | | * commit 9354fa7a2111e943f000edb1409a05f781b14276
1544 | | | | | | | | | \ Merge: 0722412 78649f4
1545 | | | | | | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>

```

```

1546 | | | | | | | | | | Date: Wed Dec 7 23:06:36 2016 -0500
1547 | | | | | | | | | |
1548 | | | | | | | | | | Final draft candidate
1549 | | | | | | | | | |
1550 | | | | | | | | | | * commit 78649f4df74af3ca528e3fe72fafee42fcd9e925
1551 | | | | | | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
1552 | | | | | | | | | | Date: Wed Dec 7 18:09:46 2016 -0500
1553 | | | | | | | | | |
1554 | | | | | | | | | | Almost done
1555 | | | | | | | | | |
1556 | | | | | | | | | | * commit 05ded19fed74d99c33fe3e5333ce190413f11c7c
1557 | | | | | | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
1558 | | | | | | | | | | Date: Wed Dec 7 15:47:52 2016 -0500
1559 | | | | | | | | | |
1560 | | | | | | | | | | More work
1561 | | | | | | | | | |
1562 | | | | | | | | | | * | commit 07224121c0ac2f0d2cced8d57d8df81620a3c881
1563 | | | | | | | | | | | / Author: oracleofnj <jared.samet@aya.yale.edu>
1564 | | | | | | | | | | Date: Wed Dec 7 11:32:11 2016 -0500
1565 | | | | | | | | | |
1566 | | | | | | | | | | Working
1567 | | | | | | | | | |
1568 | | | | | | | | | | * commit cbb05776c7518c07430d1fb482fd4ee30c70a3e6
1569 | | | | | | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
1570 | | | | | | | | | | Date: Wed Dec 7 02:35:06 2016 -0500
1571 | | | | | | | | | |
1572 | | | | | | | | | | Still WIP
1573 | | | | | | | | | |
1574 | | | | | | | | | | * commit e3c94366de1322d5409194ca08043d6d0cc355f5
1575 | | | | | | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
1576 | | | | | | | | | | Date: Wed Dec 7 00:44:22 2016 -0500
1577 | | | | | | | | | |
1578 | | | | | | | | | | WIP
1579 | | | | | | | | | |
1580 | | | | | | | | | | * commit 18bb1821e80b81c4ec98f834e28f48c838216de8
1581 | | | | | | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
1582 | | | | | | | | | | Date: Wed Dec 7 00:35:06 2016 -0500
1583 | | | | | | | | | |
1584 | | | | | | | | | | Still work in progress
1585 | | | | | | | | | |
1586 | | | | | | | | | | * commit af69b92ab6338b061bf9d8b767b3e2e8blab50a4
1587 | | | | | | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
1588 | | | | | | | | | | Date: Tue Dec 6 17:23:45 2016 -0500
1589 | | | | | | | | | |
1590 | | | | | | | | | | More updates
1591 | | | | | | | | | |
1592 | | | | | | | | | | * commit 85a4ccb27d496fbe4c377941e024d0088fae8d1b
1593 | | | | | | | | | | | / Author: oracleofnj <jared.samet@aya.yale.edu>
1594 | / | | | | | | | | | Date: Tue Dec 6 16:12:31 2016 -0500
1595 | | | | | | | | | |
1596 | | | | | | | | | | LRM update part 1
1597 | | | | | | | | | |
1598 * | | | | | | | | | | commit 910bd01aacd801d145b0dde2f55a92827c691ae4
1599 | \ \ \ \ \ \ \ \ \ \ Merge: 52e7a8a 88480fb
1600 | | | | | | | | | | | / Author: Jared Samet <jared.samet@aya.yale.edu>
1601 | / | | | | | | | | | Date: Mon Dec 5 14:27:07 2016 -0500

```

```

1602 | | | | | | | | |
1603 | | | | | | | | | Merge pull request #21 from ExtendLang/fileio
1604 | | | | | | | | |
1605 | | | | | | | | | File IO Stdlib functions
1606 | | | | | | | | |
1607 | * | | | | | | | | commit 88480fbc58a8931e54098dc99ba74f96c9d780bb
1608 | | \ \ \ \ \ \ \ \ Merge: bfa906b 52e7a8a
1609 | | / / / / / / / / Author: Nigel Schuster <Neitsch@users.noreply.github.com>
1610 | / | | | | | | | | Date: Mon Dec 5 13:36:28 2016 -0500
1611 | | | | | | | | |
1612 | | | | | | | | | Merge branch 'master' into fileio
1613 | | | | | | | | |
1614 | * | | | | | | | | commit 52e7a8a8aa3c5b38ca324a24fa4f0ce15059ab2f
1615 | | \ \ \ \ \ \ \ \ Merge: a5b9066 e4e5e26
1616 | | _ | / / / / / / / Author: Jared Samet <jared.samet@aya.yale.edu>
1617 | / | | | | | | | | Date: Mon Dec 5 13:32:54 2016 -0500
1618 | | | | | | | | |
1619 | | | | | | | | | Merge pull request #22 from ExtendLang/rm-micro
1620 | | | | | | | | |
1621 | | | | | | | | | Removed microc
1622 | | | | | | | | |
1623 | * | | | | | | | | commit e4e5e264cf4369c696f0259c1825552aaee13bc2
1624 | / / / / / / / / Author: Nigel Schuster <nigel.schusters@googlemail.com>
1625 | | | | | | | | | Date: Mon Dec 5 09:25:17 2016 -0500
1626 | | | | | | | | |
1627 | | | | | | | | | Removed microc reference implementation
1628 | | | | | | | | |
1629 | * | | | | | | | | commit bfa906b7661ba27c0f28bed535033a0f4b5eff06
1630 | | | | | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
1631 | | | | | | | | | Date: Mon Dec 5 13:28:03 2016 -0500
1632 | | | | | | | | |
1633 | | | | | | | | | Fix off-by-one bug
1634 | | | | | | | | |
1635 | * | | | | | | | | commit eb8dd71e3eb2f92d1a82dd19d03b1e090176bbef
1636 | | | | | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
1637 | | | | | | | | | Date: Mon Dec 5 13:20:03 2016 -0500
1638 | | | | | | | | |
1639 | | | | | | | | | Address issues
1640 | | | | | | | | |
1641 | * | | | | | | | | commit 270da2b340e5f3f2e8240495ff32c4e31e58456b
1642 | | \ \ \ \ \ \ \ \ Merge: b928e98 a5b9066
1643 | | / / / / / / / / Author: Ishaan Kolluri <ishaankolluri@users.noreply.github.com>
1644 | / | | | | | | | | Date: Mon Dec 5 02:40:59 2016 -0500
1645 | | | | | | | | |
1646 | | | | | | | | | Merge branch 'master' into fileio
1647 | | | | | | | | |
1648 | * | | | | | | | | commit a5b9066a498c16d2c10664d59e42d71f731959ea
1649 | | \ \ \ \ \ \ \ \ Merge: 2ae2b83 35e9471
1650 | | _ | _ | _ | _ | _ / Author: Jared Samet <jared.samet@aya.yale.edu>
1651 | / | | | | | | | | Date: Sun Dec 4 14:00:30 2016 -0500
1652 | | | | | | | | |
1653 | | | | | | | | | Merge pull request #20 from ExtendLang/lhs-all-ids
1654 | | | | | | | | |
1655 | | | | | | | | | Lhs all ids
1656 | | | | | | | | |
1657 | * | | | | | | | | commit 35e947123ee5d51957a79780b78e2cfc816ae0e3

```

```

1658 | | | | | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
1659 | | | | | | | | | Date: Sun Dec 4 13:38:44 2016 -0500
1660 | | | | | | | | |
1661 | | | | | | | | | Put back Id(s) as it was
1662 | | | | | | | | |
1663 | * | | | | | | | | commit 641d4546969477380744cdec08904ca7fa061f73
1664 | / / / / / / / / | Author: oracleofnj <jared.samet@aya.yale.edu>
1665 | | | | | | | | | Date: Sun Dec 4 13:36:36 2016 -0500
1666 | | | | | | | | |
1667 | | | | | | | | | Always transform to ID on LHS, even for LitInts
1668 | | | | | | | | |
1669 | * | | | | | | | | commit b928e9826f816e841ee2ba0a7a59d33a0e1daf0f
1670 | | | | | | | | | Author: Ishaan <ishaankolluri@gmail.com>
1671 | | | | | | | | | Date: Mon Dec 5 02:40:10 2016 -0500
1672 | | | | | | | | |
1673 | | | | | | | | | Remove bloat
1674 | | | | | | | | |
1675 | * | | | | | | | | commit 894b511acf6e592bbe552b9cf639bc764e2a2342
1676 | | | | | | | | | Author: Ishaan <ishaankolluri@gmail.com>
1677 | | | | | | | | | Date: Mon Dec 5 02:32:49 2016 -0500
1678 | | | | | | | | |
1679 | | | | | | | | | Added testcase
1680 | | | | | | | | |
1681 | * | | | | | | | | commit 62b8e83bf20e352b8c6ce5523b519e12371b88f5
1682 | | | | | | | | | Author: Ishaan <ishaankolluri@gmail.com>
1683 | | | | | | | | | Date: Mon Dec 5 02:30:16 2016 -0500
1684 | | | | | | | | |
1685 | | | | | | | | | Added fwrite implementation
1686 | | | | | | | | |
1687 | * | | | | | | | | commit 77a23ae8b083fba49a27adb2a9fdb479258c2b1b
1688 | | | | | | | | | Author: Ishaan <ishaankolluri@gmail.com>
1689 | | | | | | | | | Date: Mon Dec 5 01:39:30 2016 -0500
1690 | | | | | | | | |
1691 | | | | | | | | | Added read
1692 | | | | | | | | |
1693 | * | | | | | | | | commit 46e9b5893df2c7a7ff36ab9642880517c7ccec3c
1694 | | | | | | | | | Author: Ishaan <ishaankolluri@gmail.com>
1695 | | | | | | | | | Date: Mon Dec 5 00:07:16 2016 -0500
1696 | | | | | | | | |
1697 | | | | | | | | | Make refactoring changes and new helpers
1698 | | | | | | | | |
1699 | * | | | | | | | | commit f47f2ba753dad63a6232136b2f8be7ceb61c1b44
1700 | | | | | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
1701 | | | | | | | | | Date: Sun Dec 4 10:30:44 2016 -0500
1702 | | | | | | | | |
1703 | | | | | | | | | Add error handling to close() and add a couple test cases
1704 | | | | | | | | |
1705 | * | | | | | | | | commit e95a95a32ec5aca30c1f49fb2aeb48c3b160b935
1706 | | | | | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
1707 | | | | | | | | | Date: Sun Dec 4 10:07:01 2016 -0500
1708 | | | | | | | | |
1709 | | | | | | | | | Add assertSingleNumber and get_number to eliminate more copy &
    | | | | | | | | | paste
1710 | | | | | | | | |
1711 | * | | | | | | | | commit 543e7209e425928904d9ccbd8299cd518865ffb2
1712 | | | | | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>

```

```

1713 | | | | | | | Date: Sun Dec 4 09:47:03 2016 -0500
1714 | | | | | | |
1715 | | | | | | | Add new_number() to eliminate some copy and paste
1716 | | | | | | |
1717 | * | | | | | | commit d7f10c926a7701189e4df2acb12bcac88b86d875
1718 | | | | | | | Author: Ishaan <ishaankolluri@gmail.com>
1719 | | | | | | | Date: Sun Dec 4 02:31:03 2016 -0500
1720 | | | | | | |
1721 | | | | | | | Tentative drafts of fileio functions
1722 | | | | | | |
1723 | * | | | | | | commit 7d81e43b768ea42ee806a6ad23869f7efacbf8a8
1724 | | | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
1725 | | | | | | | Date: Sun Dec 4 00:15:20 2016 -0500
1726 | | | | | | |
1727 | | | | | | | add diagnostic prinfo
1728 | | | | | | |
1729 | * | | | | | | commit 868d9a4491fe90a050b9875f701ffe6d4d84cef6
1730 | | | | | | | Author: Ishaan <ishaankolluri@gmail.com>
1731 | | | | | | | Date: Sat Dec 3 23:46:01 2016 -0500
1732 | | | | | | |
1733 | | | | | | | Cleanup
1734 | | | | | | |
1735 | * | | | | | | commit aale0148bf45498311b2cb9eb0fe1ebdf38cc07a
1736 | | | | | | | Author: Ishaan <ishaankolluri@gmail.com>
1737 | | | | | | | Date: Sat Dec 3 23:42:46 2016 -0500
1738 | | | | | | |
1739 | | | | | | | Add file pointer array
1740 | | | | | | |
1741 | * | | | | | | commit 88d05de75293d0073ef4823ffa7f92297b6ccd85
1742 | / / / / / / / Author: Ishaan <ishaankolluri@gmail.com>
1743 | | | | | | | Date: Sat Dec 3 18:38:34 2016 -0500
1744 | | | | | | |
1745 | | | | | | | Working on fopen
1746 | | | | | | |
1747 | * | | | | | | commit 2ae2b839a88fad0ed90c1b4b6666e9c37c54a911
1748 | \ \ \ \ \ \ \ Merge: 8a60995 7c78a23
1749 | | _ | _ | _ | _ / Author: Jared Samet <jared.samet@aya.yale.edu>
1750 | / | | | | | | Date: Sat Dec 3 14:06:40 2016 -0500
1751 | | | | | | |
1752 | | | | | | | Merge pull request #15 from ExtendLang/stdlib-fun
1753 | | | | | | |
1754 | | | | | | | Stdlib fun
1755 | | | | | | |
1756 | * | | | | | | commit 7c78a2359f174dd9d3d68a02a23c587186b3e20d
1757 | | | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
1758 | | | | | | | Date: Sat Dec 3 14:02:51 2016 -0500
1759 | | | | | | |
1760 | | | | | | | Move test_fabs out of regression test suite
1761 | | | | | | |
1762 | * | | | | | | commit 0a8055b8851a60af3e6f23d19ebd24f126f1bdf9
1763 | | | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
1764 | | | | | | | Date: Sat Dec 3 13:48:19 2016 -0500
1765 | | | | | | |
1766 | | | | | | | make test | grep REGRESSION
1767 | | | | | | |
1768 | * | | | | | | commit a24742bca26352fb2f9a255ca1abf4036b975a9d

```

```

1769 | |\ \ \ \ \ \ Merge: 5243c5a 8a60995
1770 | / / / / / / / Author: Kevin <kevinye1113@gmail.com>
1771 | / | | | | | | Date: Fri Dec 2 22:50:43 2016 -0500
1772 | | | | | | |
1773 | | | | | | | Merged stdlib with master
1774 | | | | | | |
1775 * | | | | | | | commit 8a60995f5d9e25c950066478d4b6f451f0c17a16
1776 |\ \ \ \ \ \ \ Merge: 96a3028 f0d33e2
1777 | | | | | | | Author: Jared Samet <jared.samet@aya.yale.edu>
1778 | | | | | | | Date: Thu Dec 1 23:38:54 2016 -0500
1779 | | | | | | |
1780 | | | | | | | Merge pull request #18 from ExtendLang/parser-error
1781 | | | | | | |
1782 | | | | | | | Parser error
1783 | | | | | | |
1784 | * | | | | | | commit f0d33e292b39aab8e6d615cb28d6a011697a4115
1785 | | | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
1786 | | | | | | | Date: Thu Dec 1 23:18:39 2016 -0500
1787 | | | | | | |
1788 | | | | | | | Move error handling
1789 | | | | | | |
1790 | * | | | | | | commit 3b24c3a9570d8c03150bf405fcb3c15a407083fd
1791 | | | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
1792 | | | | | | | Date: Thu Dec 1 23:16:53 2016 -0500
1793 | | | | | | |
1794 | | | | | | | Adjust test script
1795 | | | | | | |
1796 | * | | | | | | commit 60a732fe464f53cb119d5a5d0dca2e10202baf27
1797 | |\ \ \ \ \ \ \ Merge: 5dec6a2 96a3028
1798 | / / / / / / / Author: oracleofnj <jared.samet@aya.yale.edu>
1799 | / | | | | | | Date: Thu Dec 1 22:55:28 2016 -0500
1800 | | | | | | |
1801 | | | | | | | Merge branch 'master' into parser-error
1802 | | | | | | |
1803 * | | | | | | | commit 96a30280911892931866a653d3271e1c8a8eb085
1804 |\ \ \ \ \ \ \ Merge: 5bdd52c 7912d5a
1805 | | _ | _ | _ | _ / Author: Nigel Schuster <Neitsch@users.noreply.github.com>
1806 | / | | | | | | Date: Thu Dec 1 22:19:21 2016 -0500
1807 | | | | | | |
1808 | | | | | | | Merge pull request #16 from ExtendLang/fail-silent
1809 | | | | | | |
1810 | | | | | | | Reduce testscript output
1811 | | | | | | |
1812 | * | | | | | | commit 7912d5ad15e53af8b7fd5d4fa50086489683abe5
1813 | |\ \ \ \ \ \ \ Merge: 72360f4 5bdd52c
1814 | / / / / / / / Author: Nigel Schuster <Neitsch@users.noreply.github.com>
1815 | / | | | | | / Date: Thu Dec 1 21:26:03 2016 -0500
1816 | | | _ | _ | _ | /
1817 | | / | | | | | Merge branch 'master' into fail-silent
1818 | | | | | | |
1819 | * | | | | | | commit 72360f4eb9c1dcf6ef01987c5a3764bb3ffda0d8
1820 | | | _ | _ | _ / Author: Nigel Schuster <nigel.schusters@googlemail.com>
1821 | | / | | | | | Date: Thu Dec 1 17:09:08 2016 -0500
1822 | | | | | | |
1823 | | | | | | | Minified error output for outputs that have not passed yet
1824 | | | | | | |

```



```

1825 | | * | | | commit 5dec6a2973176e9aa664adec96fae44e28b8a30b
1826 | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
1827 | | | | | Date: Thu Dec 1 22:55:05 2016 -0500
1828 | | | | |
1829 | | | | | Thank you Nigel!!!
1830 | | | | |
1831 | | * | | | commit 6c3696ce28a857d6cbdca227677d454b0dbd280b
1832 | | / / / / | Author: oracleofnj <jared.samet@aya.yale.edu>
1833 | / | | | | Date: Thu Dec 1 21:59:40 2016 -0500
1834 | | | | |
1835 | | | | | Figure out why test is failing
1836 | | | | |
1837 * | | | | | commit 5bdd52cc31c181789388acdafba2eb69709658f0
1838 | \ \ \ \ \ | Merge: 8c7b6ce 8893255
1839 | | | | | Author: Jared Samet <jared.samet@aya.yale.edu>
1840 | | | | | Date: Thu Dec 1 21:13:45 2016 -0500
1841 | | | | |
1842 | | | | | Merge pull request #17 from ExtendLang/lexbuf-pos
1843 | | | | |
1844 | | | | | Lexbuf pos
1845 | | | | |
1846 | * | | | | commit 889325556f4f0f43d4a6048fb15ea0b59e02b0fc
1847 | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
1848 | | | | | Date: Thu Dec 1 20:35:04 2016 -0500
1849 | | | | |
1850 | | | | | Add a couple test cases
1851 | | | | |
1852 | * | | | | commit 2868653773273d33ce8d4a5ddb12c02086d7e037
1853 | / / / / / | Author: oracleofnj <jared.samet@aya.yale.edu>
1854 | | | | | Date: Thu Dec 1 20:23:01 2016 -0500
1855 | | | | |
1856 | | | | | Use lexbuf.lex_curr_p to calculate position
1857 | | | | |
1858 * | | | | | commit 8c7b6cea2bb4fd797d7dda9e2cdcbb819890f23c
1859 | \ \ \ \ \ | Merge: 7abb394 2885ac7
1860 | / / / / / | Author: Ishaan Kolluri <ishaankolluri@users.noreply.github.com>
1861 | / | | | | Date: Thu Dec 1 18:59:49 2016 -0500
1862 | | | | |
1863 | | | | | Merge pull request #11 from ExtendLang/parse_error
1864 | | | | |
1865 | | | | | Add line numbers to Syntax Errors
1866 | | | | |
1867 | * | | | | commit 2885ac7e0d21ca4132922e5e5215155e6bf4c137
1868 | | | | | Author: Ishaan <ishaankolluri@gmail.com>
1869 | | | | | Date: Thu Dec 1 18:56:15 2016 -0500
1870 | | | | |
1871 | | | | | Added test case for string
1872 | | | | |
1873 | * | | | | commit e9fb1c2dca7114154c79f31f534da8e5cc1d0f7f
1874 | | | | | Author: Ishaan <ishaankolluri@gmail.com>
1875 | | | | | Date: Thu Dec 1 15:03:29 2016 -0500
1876 | | | | |
1877 | | | | | Added increment to string buffer and tests
1878 | | | | |
1879 | * | | | | commit eb7c1e843628a9714a50a847fca26ce8e4be75c4
1880 | | | | | Author: Ishaan <ishaankolluri@gmail.com>

```

```

1881 | | | | | Date: Thu Dec 1 03:50:53 2016 -0500
1882 | | | | |
1883 | | | | | Add partial character indexing
1884 | | | | |
1885 | * | | | | commit df09aeaa3fb16283f652ab549ab2e246c4d79464
1886 | | | | | Author: Ishaan <ishaankolluri@gmail.com>
1887 | | | | | Date: Thu Dec 1 00:41:07 2016 -0500
1888 | | | | |
1889 | | | | | Add expected parse testcase intermediate
1890 | | | | |
1891 | * | | | | commit 712a71006cf73e566676393cb09cca2cd84e311e
1892 | | | | | Author: Ishaan <ishaankolluri@gmail.com>
1893 | | | | | Date: Wed Nov 30 16:42:32 2016 -0500
1894 | | | | |
1895 | | | | | Added tentative scanner-level line number
1896 | | | | |
1897 | * | | | | commit bf4ee6c3bc017d4d2a7140526717edfc6c70a896
1898 | / / / / Author: Ishaan <ishaankolluri@gmail.com>
1899 | | | | | Date: Wed Nov 30 15:53:43 2016 -0500
1900 | | | | |
1901 | | | | | Added SyntaxError Exception at scan level
1902 | | | | |
1903 | * | | | | commit 7abb39400db2ebc10312943f011b363532dbd872
1904 | \ \ \ \ Merge: 13cd317 e0b7fdb
1905 | | _ | / Author: Nigel Schuster <Neitsch@users.noreply.github.com>
1906 | / | | | Date: Thu Dec 1 14:07:58 2016 -0500
1907 | | | | |
1908 | | | | | Merge pull request #14 from ExtendLang/sinner
1909 | | | | |
1910 | | | | | Sinner
1911 | | | | |
1912 | * | | | | commit e0b7fdbea24db1a5d6606dfc19848d8755cb931c
1913 | | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
1914 | | | | | Date: Thu Dec 1 14:05:38 2016 -0500
1915 | | | | |
1916 | | | | | Rename empty to new_val
1917 | | | | |
1918 | * | | | | commit 6ea8cff367f1465f7903fc249fdd65a94e5fed27
1919 | | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
1920 | | | | | Date: Thu Dec 1 10:10:26 2016 -0500
1921 | | | | |
1922 | | | | | Using define instead of magic numbers
1923 | | | | |
1924 | * | | | | commit cd7d261cc0f4faf9f3fb3178228fce7e81ac551c
1925 | | \ \ \ Merge: 3986f38 13cd317
1926 | | / / / Author: Nigel Schuster <nigel.schusters@googlemail.com>
1927 | / | | | Date: Thu Dec 1 10:07:10 2016 -0500
1928 | | | | |
1929 | | | | | Merge branch 'master' into sinner
1930 | | | | |
1931 | * | | | | commit 13cd31794ee3313e0b87a106100fe4e752969a45
1932 | \ \ \ \ Merge: effc20b 4eed07
1933 | | | | | Author: Nigel Schuster <Neitsch@users.noreply.github.com>
1934 | | | | | Date: Thu Dec 1 10:06:25 2016 -0500
1935 | | | | |
1936 | | | | | Merge pull request #13 from ExtendLang/value_p

```

```

1937 | | | | |
1938 | | | | | Moved all function signatures to value_p return value
1939 | | | | |
1940 | * | | | commit 4eed0719d5cee5cae27e2debfla26572a2a96ad
1941 | | | | | Author: Ishaan <ishaankolluri@gmail.com>
1942 | | | | | Date: Thu Dec 1 01:02:56 2016 -0500
1943 | | | | |
1944 | | | | | Change print return type to empty
1945 | | | | |
1946 | | | * | commit 5243c5a654a69cc7b5239e0236c3318b132fec8e
1947 | | | | | Author: Kevin <kevinyell113@gmail.com>
1948 | | | | | Date: Fri Dec 2 18:16:36 2016 -0500
1949 | | | | |
1950 | | | | | Removed magic numbers and add fabs test
1951 | | | | |
1952 | | | * | commit fa42f2718eada7ac73302690aac7a4982e20da4f
1953 | | | | | Author: Kevin <kevinyell113@gmail.com>
1954 | | | | | Date: Thu Dec 1 00:41:47 2016 -0500
1955 | | | | |
1956 | | | | | Fixed acos function
1957 | | | | |
1958 | | | * | commit 53d34adc07f06d185d57de7d3ed3901706eeb4f3
1959 | | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
1960 | | | | | Date: Thu Dec 1 00:29:32 2016 -0500
1961 | | | | |
1962 | | | | | Moved double values type to numeric
1963 | | | | |
1964 | | | * | commit f769c61b5da538e4934721de651d2772359fa2c5
1965 | | | \ \ Merge: 7f0bc86 3986f38
1966 | | | / / Author: Nigel Schuster <nigel.schusters@googlemail.com>
1967 | | / | | Date: Thu Dec 1 00:18:07 2016 -0500
1968 | | | | |
1969 | | | | | Merge branch 'sinner' into stdlib-fun
1970 | | | | |
1971 | | * | | commit 3986f38a3fe7b5cbe849af80e5c00fc7881dc13e
1972 | | \ \ \ Merge: 4604545 5bd87f9
1973 | | / / / Author: Nigel Schuster <nigel.schusters@googlemail.com>
1974 | | / | | Date: Thu Dec 1 00:17:21 2016 -0500
1975 | | | | |
1976 | | | | | Merge branch 'value_p' into sinner
1977 | | | | |
1978 | * | | | commit 5bd87f9233c2b4be9f60cecd0bbab3792df5d8b7
1979 | | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
1980 | | | | | Date: Wed Nov 30 23:37:58 2016 -0500
1981 | | | | |
1982 | | | | | Explicitly declaring to link math library
1983 | | | | |
1984 | | * | | commit 4604545c905e2a9a77524cbaeb54a93a9adfb249
1985 | | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
1986 | | | | | Date: Thu Dec 1 00:12:08 2016 -0500
1987 | | | | |
1988 | | | | | Consistently using floats
1989 | | | | |
1990 | | * | | commit 38b9824041ecc7c1b9985cf44436a31f5daa90f3
1991 | | \ \ \ Merge: e085977 3303575
1992 | | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>

```

```

1993 | | | | | Date: Wed Nov 30 23:46:14 2016 -0500
1994 | | | | |
1995 | | | | | Merge branch 'value_p' into sinner
1996 | | | | |
1997 | | | * | | commit 330357567faf2c6fc8b1c3e769d7aa3e047629de
1998 | | | / / / Author: Nigel Schuster <nigel.schusters@googlemail.com>
1999 | | / | | | Date: Wed Nov 30 23:37:58 2016 -0500
2000 | | | | |
2001 | | | | | Explicitly declaring to link math library
2002 | | | | |
2003 | * | | | | commit 31a74ecd91861165dfdcfed780a8d59ba99042e7
2004 | | \ \ \ \ \ Merge: 206ee5a effc20b
2005 | | / / / / / Author: Nigel Schuster <nigel.schusters@googlemail.com>
2006 | / | | | | Date: Wed Nov 30 23:35:34 2016 -0500
2007 | | | | |
2008 | | | | | Merge branch 'master' into value_p
2009 | | | | |
2010 * | | | | | commit effc20bd8f99f841b0ae5a0af19042599db89345
2011 | \ \ \ \ \ \ Merge: 3a8efbc 3b6d7b7
2012 | | _ | _ | _ / Author: Ishaan Kolluri <ishaankolluri@users.noreply.github.com>
2013 | / | | | | Date: Wed Nov 30 18:45:52 2016 -0500
2014 | | | | |
2015 | | | | | Merge pull request #12 from ExtendLang/easy-compile
2016 | | | | |
2017 | | | | | Added script to compile and link
2018 | | | | |
2019 | * | | | | commit 3b6d7b740336c5fef9a60be7f77351f04ab46de
2020 | / / / / / Author: Nigel Schuster <nigel.schusters@googlemail.com>
2021 | | | | | Date: Wed Nov 30 17:51:19 2016 -0500
2022 | | | | |
2023 | | | | | Added script to compile and link
2024 | | | | |
2025 | | | * | | commit 7f0bc8650ald6d89ac6ddd1d4862c245e7dfbc7e
2026 | | | | | Author: Kevin <kevinyell113@gmail.com>
2027 | | | | | Date: Wed Nov 30 23:04:34 2016 -0500
2028 | | | | |
2029 | | | | | Finished remainder of stdlib
2030 | | | | |
2031 | | | * | | commit cd160df84fff7fceb7340b5fad27fe9360c436cd
2032 | | | / | | Author: Kevin <kevinyell113@gmail.com>
2033 | | | | | Date: Wed Nov 30 22:50:18 2016 -0500
2034 | | | | |
2035 | | | | | Added more c functions to stdlib
2036 | | | | |
2037 | | * | | | | commit e085977d3edd3785dc682c42161601b6b975cc41
2038 | | / | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
2039 | | | | | Date: Wed Nov 30 19:59:57 2016 -0500
2040 | | | | |
2041 | | | | | Made sin function work
2042 | | | | |
2043 | * | | | | commit 206ee5ab0f46bc5411cc3834f8107134c6e6e80b
2044 | / | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
2045 | | | | | Date: Wed Nov 30 19:07:28 2016 -0500
2046 | | | | |
2047 | | | | | Moved all function signatures to value_p return value
2048 | | | | |

```

```

2049 *   commit 3a8efbc6575cb2d73b305c087b6f2294cf1c3511
2050 |\   Merge: 5b3edb0 7a2af49
2051 | |   Author: Nigel Schuster <Neitsch@users.noreply.github.com>
2052 | |   Date:   Mon Nov 28 23:05:28 2016 -0500
2053 | |
2054 | |       Merge pull request #9 from ExtendLang/func-calls
2055 | |
2056 | |       Function calls work now
2057 | |
2058 *   commit 7a2af4937e6abf0621cfabad4ee463dfa7b64e20
2059 | |   Author: Nigel Schuster <nigel.schusters@googlemail.com>
2060 | |   Date:   Mon Nov 28 20:33:53 2016 -0500
2061 | |
2062 | |       Removed another ocaml 4.3 dep
2063 | |
2064 *   commit 468e79f88bcecd2d60c822799c353bc00ee11099
2065 | |   Author: Nigel Schuster <nigel.schusters@googlemail.com>
2066 | |   Date:   Mon Nov 28 19:50:53 2016 -0500
2067 | |
2068 | |       Added ocaml 4.3 as dep for travis (hopefully this works)
2069 | |
2070 *   commit a40876103c19a444466a07019854cf508ccfe276
2071 | |   Author: Nigel Schuster <nigel.schusters@googlemail.com>
2072 | |   Date:   Mon Nov 28 19:35:49 2016 -0500
2073 | |
2074 | |       Fixed String.equal
2075 | |
2076 *   commit 90c3cafldc0ebf18d2e8709df39be6333266bc56
2077 | |   Author: Nigel Schuster <nigel.schusters@googlemail.com>
2078 | |   Date:   Sun Nov 27 22:52:14 2016 -0500
2079 | |
2080 | |       Fixed interpreter for now
2081 | |
2082 *   commit al8da787c64c95ed8cc31f8f188bb1ef9507453b
2083 | |   Author: Nigel Schuster <nigel.schusters@googlemail.com>
2084 | |   Date:   Sun Nov 27 22:42:27 2016 -0500
2085 | |
2086 | |       Added accidentally created file
2087 | |
2088 *   commit 5647312ef3939b7417783c3ff7620a90d3b002fe
2089 | |   Author: Nigel Schuster <nigel.schusters@googlemail.com>
2090 | |   Date:   Sun Nov 27 22:41:22 2016 -0500
2091 | |
2092 | |       Made extern function calls work
2093 | |
2094 *   commit 872aa8c4c287ddb35dd5d529df2122e1b38a0fb9
2095 |\   Merge: 26ef1cc 877336f
2096 | | |   Author: Nigel Schuster <nigel.schusters@googlemail.com>
2097 | | |   Date:   Sun Nov 27 13:52:44 2016 -0500
2098 | | |
2099 | | |       Merge branch 'func-calls' of https://github.com/ExtendLang/Extend into func-
2100 | | |       calls
2101 | | *   commit 877336f09f10848092540a3349bb1a42c9a6ebd9
2102 | | |\  Merge: 374273f 5b3edb0
2103 | |_|/  Author: Nigel Schuster <Neitsch@users.noreply.github.com>

```

```

2104 | / | | Date: Sun Nov 27 12:15:11 2016 -0500
2105 | | |
2106 | | | Merge branch 'master' into func-calls
2107 | | |
2108 * | | commit 5b3edb0d0b6e8764cadd9290eab3c9dad7fb782d
2109 | \ \ \ Merge: 442ae91 952aab8
2110 | | | Author: Nigel Schuster <Neitsch@users.noreply.github.com>
2111 | | | Date: Sun Nov 27 12:14:43 2016 -0500
2112 | | |
2113 | | | Merge pull request #8 from ExtendLang/stdlib-template
2114 | | |
2115 | | | Stdlib template
2116 | | |
2117 | | * | commit 26ef1cc506931d6771cf950fdae016f9f3874bbe
2118 | | | / Author: Nigel Schuster <nigel.schusters@googlemail.com>
2119 | | | Date: Sun Nov 27 13:51:06 2016 -0500
2120 | | |
2121 | | | Merging list of functions
2122 | | |
2123 | | * | commit 374273f2ceb3b69ee489c39c0f3767ab4748f246
2124 | | | / Author: Nigel Schuster <nigel.schusters@googlemail.com>
2125 | | | Date: Sun Nov 27 12:13:52 2016 -0500
2126 | | |
2127 | | | Function calls work now
2128 | | |
2129 | | * | commit 952aab83190eb5b8e3be609fe44fa0fc0cbe8f2b
2130 | | \ Merge: ac6268f 554fbb2
2131 | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
2132 | | | Date: Sun Nov 27 09:54:12 2016 -0500
2133 | | |
2134 | | | Merge extern
2135 | | |
2136 | | * | commit 554fbb28bf7e995117790bd17770af810b1caa37
2137 | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
2138 | | | | Date: Wed Nov 23 22:28:29 2016 -0500
2139 | | | |
2140 | | | | Better error message for WrongNumberArgs
2141 | | | |
2142 | | * | commit f09e40e405befe48f3b2310c6f4a0aecd48dbcf7
2143 | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
2144 | | | | Date: Wed Nov 23 12:47:39 2016 -0500
2145 | | | |
2146 | | | | Make sequence work
2147 | | | |
2148 | | * | commit 053980bc62d59316a73fe42efaf1706d8f8d813c
2149 | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
2150 | | | | Date: Tue Nov 22 16:02:27 2016 -0500
2151 | | | |
2152 | | | | Actually commit all the extern stuff
2153 | | | |
2154 | | * | commit ac6268fd89de976aa5d355e09215f5bab22d128f
2155 | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
2156 | | | | Date: Sat Nov 26 23:06:00 2016 -0500
2157 | | | |
2158 | | | | Boxing ints, added unop sizeof, actually returning subrange not dummy object
2159 | | | |

```

```

2160 | * | commit ca07be3d9e79a9398b1da5fbc30a22976fbe2246
2161 | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
2162 | | | Date: Sat Nov 26 21:27:19 2016 -0500
2163 | | |
2164 | | | Unboxing hello world to and from subrange
2165 | | |
2166 | * | commit aef6c1964b6e8ca2b0d05d3a5e1efb8a5b0f8159
2167 | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
2168 | | | Date: Sat Nov 26 16:55:48 2016 -0500
2169 | | |
2170 | | | Made Hello World somewhat workable
2171 | | |
2172 | * | commit cfb637efdc1747331c83e6ccfe9f417fb8fc3c1d
2173 | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
2174 | | | Date: Fri Nov 25 18:27:37 2016 -0500
2175 | | |
2176 | | | Fixed faulty setup on call
2177 | | |
2178 | * | commit ebf926a2c3cd85309b6dfac4febc166d381c8ca2
2179 | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
2180 | | | Date: Fri Nov 25 17:48:57 2016 -0500
2181 | | |
2182 | | | Added template in C
2183 | | |
2184 | * | commit 0e0fa23bclb0fb0bc039fa0d1cbcb5f617f0cc9c
2185 | | / Author: Nigel Schuster <nigel.schusters@googlemail.com>
2186 | | Date: Tue Nov 22 14:36:54 2016 -0500
2187 | | |
2188 | | | Added extern in Ast
2189 | | |
2190 | * | commit aac63be5941852ff37036e5a69d1347ba7744a82
2191 | | | Author: oracleofnj <jared.samet@aya.yale.edu>
2192 | | | Date: Mon Nov 21 23:52:25 2016 -0500
2193 | | |
2194 | | | Better duplicate definition checking
2195 | | |
2196 | * | commit 08e2d073c08f1b2067f9e8a130595114815091ae
2197 | | | Author: oracleofnj <jared.samet@aya.yale.edu>
2198 | | | Date: Mon Nov 21 23:29:28 2016 -0500
2199 | | |
2200 | | | Check assertions before evaluating fn return expression
2201 | | |
2202 | * | commit 69fa332091dfb61848a9c7b47092c31244e74153
2203 | | | Author: oracleofnj <jared.samet@aya.yale.edu>
2204 | | | Date: Mon Nov 21 18:01:23 2016 -0500
2205 | | |
2206 | | | Add size assertions
2207 | | |
2208 | * | commit 22541c427b7883b6fc89841b077ec8clc9d41c80
2209 | | | Author: oracleofnj <jared.samet@aya.yale.edu>
2210 | | | Date: Mon Nov 21 12:48:34 2016 -0500
2211 | | |
2212 | | | Fix bug in Call()
2213 | | |
2214 | * | commit 9ald24bb89721d76c3987f50d3095b16c182f38e
2215 | | | Author: oracleofnj <jared.samet@aya.yale.edu>

```

```

2216 | | Date:   Mon Nov 21 12:39:41 2016 -0500
2217 | |
2218 | |     Working on crazy bug
2219 | |
2220 | | * commit a485ceef7e987cdef7cd4321d01849bc73342e78
2221 | | Author: oracleofnj <jared.samet@aya.yale.edu>
2222 | | Date:   Sun Nov 20 22:13:46 2016 -0500
2223 | |
2224 | |     Add test case for foo([m, n] arg)
2225 | |
2226 | | * commit 10afe9a2fdb645fbb146c724d6281d34d4ba3f8a
2227 | | Author: oracleofnj <jared.samet@aya.yale.edu>
2228 | | Date:   Sun Nov 20 22:07:17 2016 -0500
2229 | |
2230 | |     Expand function signature
2231 | |
2232 | | * commit 325e9ba5b275d2d312953e02eb1abca561a31a74
2233 | | Author: oracleofnj <jared.samet@aya.yale.edu>
2234 | | Date:   Sun Nov 20 18:53:52 2016 -0500
2235 | |
2236 | |     Well, this is awkward
2237 | |
2238 | | * commit 0a76dc987d319e86able342c5d96bbc624c4bbdb
2239 | | Author: oracleofnj <jared.samet@aya.yale.edu>
2240 | | Date:   Sun Nov 20 18:41:12 2016 -0500
2241 | |
2242 | |     Add check of return value
2243 | |
2244 | | * commit 488e34ea0aeebc91443a355a94e9586351d57049
2245 | | Author: oracleofnj <jared.samet@aya.yale.edu>
2246 | | Date:   Sun Nov 20 18:31:39 2016 -0500
2247 | |
2248 | |     Add sample #1
2249 | |
2250 | | * commit 93eebc5c3442929d4d2f2a1f99c199538009d707
2251 | | Author: oracleofnj <jared.samet@aya.yale.edu>
2252 | | Date:   Sun Nov 20 18:27:23 2016 -0500
2253 | |
2254 | |     Add semantic checking to make sure functions and variables on RHS exist
2255 | |
2256 | | * commit 881f164a15251a781692045ce602dc5153adddb7
2257 | | Author: oracleofnj <jared.samet@aya.yale.edu>
2258 | | Date:   Sun Nov 20 17:22:40 2016 -0500
2259 | |
2260 | |     Check RHS slice to ensure end > start, otherwise evaluate to empty
2261 | |
2262 | | * commit 442ae9121c99adeb30215e524022e1dd046e0e66
2263 | | Merge: 367bc2b f7f701d
2264 | | Author: Nigel Schuster <Neitsch@users.noreply.github.com>
2265 | | Date:   Sun Nov 20 11:42:54 2016 -0500
2266 | |
2267 | |     Merge pull request #73 from Neitsch/interpreter-global
2268 | |
2269 | |     Added use of global variables to interpreter
2270 | |
2271 | | * commit f7f701d56f6e553742adcf396bffd7d0d2021cd2

```



```

2272 |/ Author: Nigel Schuster <nigel.schusters@googlemail.com>
2273 | Date: Sun Nov 20 11:30:06 2016 -0500
2274 |
2275 | Added use of global variables to interpreter, fixed specs for logical or and
      | and testcases with empty
2276 |
2277 * commit 367bc2b3b07ed593eba6475aa42d3d66bada93c9
2278 |\ Merge: e956238 bdca834
2279 || Author: Nigel Schuster <Neitsch@users.noreply.github.com>
2280 || Date: Sun Nov 20 00:33:17 2016 -0500
2281 ||
2282 || Merge pull request #72 from Neitsch/codegen-part-app-fix
2283 ||
2284 || Fixed partial function application warning
2285 ||
2286 | * commit bdca8349c823e69a9065ca2f7efd24d8ba3d3482
2287 | |\ Merge: 9b742d1 e956238
2288 | |/ Author: Nigel Schuster <Neitsch@users.noreply.github.com>
2289 |/| Date: Sun Nov 20 00:31:04 2016 -0500
2290 | |
2291 | | Merge branch 'master' into codegen-part-app-fix
2292 | |
2293 * | commit e956238e31b3811552c0c4fblab6ea10f00107e1
2294 |\ \ Merge: f87cb94 32f2989
2295 | | | Author: Nigel Schuster <Neitsch@users.noreply.github.com>
2296 | | | Date: Sun Nov 20 00:28:49 2016 -0500
2297 | | |
2298 | | | Merge pull request #71 from Neitsch/tc-fixes
2299 | | |
2300 | | | Tc fixes
2301 | | |
2302 | * | commit 32f2989ad2a169fe36509509b8bf8aeb61e7d8f4
2303 | |\ \ Merge: 05f317a f87cb94
2304 | | / Author: Nigel Schuster <Neitsch@users.noreply.github.com>
2305 |/| | Date: Sun Nov 20 00:20:51 2016 -0500
2306 | | |
2307 | | | Merge branch 'master' into tc-fixes
2308 | | |
2309 * | | commit f87cb946eff0389be01d20ba5e62ff9882b6b94f
2310 |\ \ \ Merge: 6d73717 842ee5a
2311 | | | | Author: Nigel Schuster <Neitsch@users.noreply.github.com>
2312 | | | | Date: Sun Nov 20 00:20:35 2016 -0500
2313 | | | |
2314 | | | | Merge pull request #69 from Neitsch/regression-tests
2315 | | | |
2316 | | | | Regression tests
2317 | | | |
2318 | * | | commit 842ee5aa0e12f721fc985689fa2bflbfff66a167b
2319 | |\ \ \ Merge: 214ab9d 6d73717
2320 | | / / Author: Nigel Schuster <Neitsch@users.noreply.github.com>
2321 |/| | | Date: Sun Nov 20 00:18:56 2016 -0500
2322 | | | |
2323 | | | | Merge branch 'master' into regression-tests
2324 | | | |
2325 * | | | commit 6d73717a9152d479eceebl778f90baaaca6a9083
2326 |\ \ \ \ Merge: 36f72a1 04e5c4a

```

```

2327 | | | | | Author: Jared Samet <jared.samet@aya.yale.edu>
2328 | | | | | Date: Sat Nov 19 23:55:35 2016 -0500
2329 | | | | |
2330 | | | | | Merge pull request #66 from Neitsch/fix-test-cases
2331 | | | | |
2332 | | | | | Fix test cases
2333 | | | | |
2334 | | * | | commit 214ab9d5df7fa62828bbb0d1284e761365885a23
2335 | | \ \ \ Merge: fb31505 5e39ba7
2336 | | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
2337 | | | | | Date: Sat Nov 19 22:10:33 2016 -0500
2338 | | | | |
2339 | | | | | Merge
2340 | | | | |
2341 | | | * | | commit 5e39ba7b6f490b9f810c37e8157056ab31151b4c
2342 | | | \ \ \ Merge: 25263fe 36f72a1
2343 | | _ _ / / / Author: Nigel Schuster <nigel.schusters@googlemail.com>
2344 | / | | | | Date: Sat Nov 19 21:55:03 2016 -0500
2345 | | | | |
2346 | | | | | Merge
2347 | | | | |
2348 | | | * | | commit 25263fe599515a7bc97a37f874d5995371f77b2c
2349 | | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
2350 | | | | | Date: Sat Nov 19 21:50:21 2016 -0500
2351 | | | | |
2352 | | | | | Removed travis from build, removed super verbose output
2353 | | | | |
2354 | | | * | | commit 0554ad9c6ceb85f2f30929b4b63f1476b9e52780
2355 | | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
2356 | | | | | Date: Sat Nov 19 21:42:28 2016 -0500
2357 | | | | |
2358 | | | | | Using precise lli version
2359 | | | | |
2360 | | | * | | commit 2825ada0022a5df1aeac7d1871904a16f676cb9c
2361 | | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
2362 | | | | | Date: Fri Nov 18 12:51:33 2016 -0500
2363 | | | | |
2364 | | | | | Added branch to build
2365 | | | | |
2366 | | | * | | commit aafabb2f0b48af7e1961740edccd63c676fb2595
2367 | | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
2368 | | | | | Date: Fri Nov 18 12:50:56 2016 -0500
2369 | | | | |
2370 | | | | | Verbose output for travis debug
2371 | | | | |
2372 | | * | | | commit fb315051683e9323f494ec9b4cd42a87103f7ea9
2373 | | / / / / Author: Nigel Schuster <nigel.schusters@googlemail.com>
2374 | / | | | | Date: Sat Nov 19 22:08:42 2016 -0500
2375 | | | | |
2376 | | | | | Passing testcases are in separate directory. Output of stats
2377 | | | | |
2378 | | | * | | commit 05f317a8560e8cc6f59af840b3e5d97581fec1be
2379 | | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
2380 | | | | | Date: Sat Nov 19 22:37:36 2016 -0500
2381 | | | | |
2382 | | | | | Fixed output on TCs

```

```

2383 | | | | |
2384 | | | * | commit aald974fec852f9d6c2c524eb4e0e585c000d2a8
2385 | | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
2386 | | | | | Date: Sat Nov 19 22:33:40 2016 -0500
2387 | | | | |
2388 | | | | | Fixed expected value for ternary
2389 | | | | |
2390 | | | * | commit ab7653af94b6eb3c441ab2aa4cb1a0238cb0103f
2391 | | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
2392 | | | | | Date: Sat Nov 19 22:32:27 2016 -0500
2393 | | | | |
2394 | | | | | Fixed import testcases
2395 | | | | |
2396 | | | * | commit 848066cd1ea2e7e580ec4710181b5e36273ea9a8
2397 | | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
2398 | | | | | Date: Sat Nov 19 22:24:55 2016 -0500
2399 | | | | |
2400 | | | | | Moved testcase asset to asset folder
2401 | | | | |
2402 | | | * | commit 53c920628b8283b501dd70df16b76c1b668b5a66
2403 | | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
2404 | | | | | Date: Sat Nov 19 22:21:48 2016 -0500
2405 | | | | |
2406 | | | | | Corrected use of global variable in test_globals
2407 | | | | |
2408 | | | * | commit 5fe74a8c9883277c3ef1df944623a54f6782030c
2409 | | | / / Author: Nigel Schuster <nigel.schusters@googlemail.com>
2410 | | / | | Date: Sat Nov 19 22:19:47 2016 -0500
2411 | | | | |
2412 | | | | | Fixed expected output for test_access_column_cells
2413 | | | | |
2414 | * | | | commit 04e5c4a7bc3ea9d79a2cbdf9e03011df96b5af6f
2415 | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
2416 | | | | | Date: Sat Nov 19 18:30:32 2016 -0500
2417 | | | | |
2418 | | | | | Add more operators to interpreter
2419 | | | | |
2420 | * | | | commit e4a190c32f50362cae49e27679e8860a1b95f49b
2421 | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
2422 | | | | | Date: Sat Nov 19 17:14:04 2016 -0500
2423 | | | | |
2424 | | | | | Add argument to main and remove _expected from filenames
2425 | | | | |
2426 | * | | | commit 7cd2b3ae0741a05e19d1f0dc39afb4f3d6be3bff
2427 | | \ \ \ Merge: d1fddfd 36f72a1
2428 | | / / / Author: oracleofnj <jared.samet@aya.yale.edu>
2429 | / | | | Date: Sat Nov 19 16:53:12 2016 -0500
2430 | | | | |
2431 | | | | | Merge branch 'master' into fix-test-cases
2432 | | | | |
2433 | * | | | commit d1fddfd7bf6ba8190ff38aaed5a22cd3b3f85ef4
2434 | | \ \ \ Merge: a9320f3 7b6b79e
2435 | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
2436 | | | | | Date: Sat Nov 19 16:52:48 2016 -0500
2437 | | | | |
2438 | | | | | Merge branch 'fix-test-cases' of https://github.com/Neitsch/plt into fix

```

```

      -test-cases
2439 | | | | |
2440 | | * | | | commit 7b6b79ed80d9356bfeala038757c746788c41fee
2441 | | | \ \ \ Merge: 24a3625 de262b4
2442 | | | | | Author: Jared Samet <jared.samet@aya.yale.edu>
2443 | | | | | Date: Sat Nov 19 14:31:39 2016 -0500
2444 | | | | |
2445 | | | | | Merge branch 'master' into fix-test-cases
2446 | | | | |
2447 | * | | | | commit a9320f3f249e2b2a342add2c750ee9beb4bec2de
2448 | | \ \ \ \ Merge: 24a3625 de262b4
2449 | | | / / / Author: oracleofnj <jared.samet@aya.yale.edu>
2450 | | / / / / Date: Sat Nov 19 14:29:51 2016 -0500
2451 | | | / / /
2452 | | | | | Merge branch 'master' into fix-test-cases
2453 | | | | |
2454 | * | | | | commit 24a3625ee0a2ebe90f49563eb355549fe6938905
2455 | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
2456 | | | | | Date: Sat Nov 19 14:27:48 2016 -0500
2457 | | | | |
2458 | | | | | Add switch tests
2459 | | | | |
2460 | * | | | | commit 75e3f71b55e8f57c4092924b1294a6c717bd223e
2461 | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
2462 | | | | | Date: Fri Nov 18 20:39:23 2016 -0500
2463 | | | | |
2464 | | | | | Fix parsing errors in test cases
2465 | | | | |
2466 | | | | * commit 9b742d16a14b81215a51612eb237875017320a97
2467 | | _|_| / Author: Nigel Schuster <nigel.schusters@googlemail.com>
2468 | /| | | | Date: Sun Nov 20 00:24:39 2016 -0500
2469 | | | | |
2470 | | | | | Fixed partial function application warning
2471 | | | | |
2472 | * | | | | commit 36f72a190820385610b63dd1c8e9e94a0fc564a9
2473 | | \ \ \ \ Merge: de262b4 c46c87b
2474 | | _|_| / Author: Jared Samet <jared.samet@aya.yale.edu>
2475 | /| | | | Date: Sat Nov 19 16:49:34 2016 -0500
2476 | | | | |
2477 | | | | | Merge pull request #67 from Neitsch/test_cases
2478 | | | | |
2479 | | | | | Test cases
2480 | | | | |
2481 | * | | | | commit c46c87bef9282aae0e4d754d74e77cd1c5399a13
2482 | | \ \ \ \ Merge: 642ce76 de262b4
2483 | | / / / / Author: Jared Samet <jared.samet@aya.yale.edu>
2484 | /| | | | Date: Sat Nov 19 16:47:26 2016 -0500
2485 | | | | |
2486 | | | | | Merge branch 'master' into test_cases
2487 | | | | |
2488 | * | | | | commit de262b4bbe84505ef0dc460afe5fa4fea8da6ba3
2489 | | \ \ \ \ Merge: 7146dce 4e38757
2490 | | _|_| / Author: Jared Samet <jared.samet@aya.yale.edu>
2491 | /| | | | Date: Sat Nov 19 14:24:39 2016 -0500
2492 | | | | |
2493 | | | | | Merge pull request #60 from Neitsch/box-args

```

```

2494 | | | |
2495 | | | |      Box args
2496 | | | |
2497 | * | | |      commit 4e3875716c30388817c66fad2df038141b89ca51
2498 | | \ \ \ \      Merge: faecfa1 7146dce
2499 | | / / / /      Author: Jared Samet <jared.samet@aya.yale.edu>
2500 | / | | |      Date:   Fri Nov 18 16:00:10 2016 -0500
2501 | | | |
2502 | | | |      Merge branch 'master' into box-args
2503 | | | |
2504 | * | | |      commit 7146dce9e2cc8dc459dc51e0dbfeb7dd426551a0
2505 | | \ \ \ \ \ \      Merge: 124d61e 09cb42f
2506 | | _ | _ | /      Author: Jared Samet <jared.samet@aya.yale.edu>
2507 | / | | |      Date:   Fri Nov 18 15:59:54 2016 -0500
2508 | | | |
2509 | | | |      Merge pull request #64 from Neitsch/reorg-test
2510 | | | |
2511 | | | |      Reorg test
2512 | | | |
2513 | * | | |      commit 09cb42f8e184a376bc7a00f69ed57d1d60a30737
2514 | | | |      Author: oracleofnj <jared.samet@aya.yale.edu>
2515 | | | |      Date:   Fri Nov 18 14:07:39 2016 -0500
2516 | | | |
2517 | | | |      Fix parse difference
2518 | | | |
2519 | * | | |      commit 39634bbf216c4eb75ed7653687739f3d22bf68a4
2520 | | | |      Author: oracleofnj <jared.samet@aya.yale.edu>
2521 | | | |      Date:   Fri Nov 18 14:01:21 2016 -0500
2522 | | | |
2523 | | | |      Remove unnecessary files
2524 | | | |
2525 | * | | |      commit d772725d15628dd48f6542cd70bd234c5d157304
2526 | | | |      Author: oracleofnj <jared.samet@aya.yale.edu>
2527 | | | |      Date:   Fri Nov 18 14:01:02 2016 -0500
2528 | | | |
2529 | | | |      Make inputs work with interpreter
2530 | | | |
2531 | | * | |      commit faecfa1905804786fa0fc08086ef32e37b811964
2532 | | | \ \ \      Merge: 6f63e89 41a81ce
2533 | | | | |      Author: oracleofnj <jared.samet@aya.yale.edu>
2534 | | | | |      Date:   Fri Nov 18 01:48:44 2016 -0500
2535 | | | | |
2536 | | | | |      Fix merge conflict in box_args
2537 | | | | |
2538 | | | * |      commit 41a81cead3637dc09da80d33ba04ee453d873b88
2539 | | | | |      Author: oracleofnj <jared.samet@aya.yale.edu>
2540 | | | | |      Date:   Fri Nov 18 01:40:11 2016 -0500
2541 | | | | |
2542 | | | | |      Move argument boxing into a function
2543 | | | | |
2544 | | | | *      commit 642ce765c175416879ca7d17f68f31966a027710
2545 | | | | |      Author: Kevin <kevinyell113@gmail.com>
2546 | | | | |      Date:   Sat Nov 19 16:39:50 2016 -0500
2547 | | | | |
2548 | | | | |      Fixed helloworld bug
2549 | | | | |

```

```

2550 | | | | * commit ac3d7fa3ff769a6b987e3837a05652be00b1057b
2551 | | | | | Author: Kevin <kevinyell113@gmail.com>
2552 | | | | | Date: Sat Nov 19 16:10:53 2016 -0500
2553 | | | | |
2554 | | | | | Added corresponding AST result for gcd function
2555 | | | | |
2556 | | | | * commit f483ac7bc551d831ef314f07ecf51eb3c571b184
2557 | | | | | Author: Kevin <kevinyell113@gmail.com>
2558 | | | | | Date: Fri Nov 18 14:10:32 2016 -0500
2559 | | | | |
2560 | | | | | Updated print statement for each test
2561 | | | | |
2562 | | | | * commit f4456f8815d6e4d181920a4800da3257d75edbf7
2563 | | | | | Merge: 00aafb7 124d61e
2564 | | | | | Author: kevinyel <kevinyel@users.noreply.github.com>
2565 | | | | | Date: Fri Nov 18 13:17:25 2016 -0500
2566 | | | | |
2567 | | | | | Merge branch 'master' into test_cases
2568 | | | | |
2569 * | | | | commit 124d61e803cd209a0b7edf977473240falb450f5
2570 | | | | | Merge: 6f63e89 82cf599
2571 | | | | | Author: Nigel Schuster <Neitsch@users.noreply.github.com>
2572 | | | | | Date: Fri Nov 18 12:44:50 2016 -0500
2573 | | | | |
2574 | | | | | Merge pull request #61 from Neitsch/reorg-test
2575 | | | | |
2576 | | | | | Modify test script to compare interpreter and compiler with expected
2577 | | | | |
2578 | * | | | commit 82cf59904f569175b2dcd4de51e4faf5ff925d7d
2579 | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
2580 | | | | | Date: Fri Nov 18 12:34:57 2016 -0500
2581 | | | | |
2582 | | | | | Modify test script to compare interpreter and compiler with expected
2583 | | | | |
2584 * | | | | commit 6f63e898f49513139ab852300ee219aad742020b
2585 | | | | | Merge: 012caaa 088dc45
2586 | | | | | Author: Jared Samet <jared.samet@aya.yale.edu>
2587 | | | | | Date: Fri Nov 18 00:48:07 2016 -0500
2588 | | | | |
2589 | | | | | Merge pull request #59 from Neitsch/hello-hello
2590 | | | | |
2591 | | | | | Hello world
2592 | | | | |
2593 | * | | | commit 088dc45e0b59702e41d2a3d4e13ef6eb11f580dc
2594 | | | | | Merge: f84757b 012caaa
2595 | | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
2596 | | | | | Date: Fri Nov 18 00:29:45 2016 -0500
2597 | | | | |
2598 | | | | | Merge
2599 | | | | |
2600 * | | | | commit 012caaf689c35d226b447301bb847b14d645cab
2601 | | | | | Merge: 5e63cee 9463afa
2602 | | | | | Author: Jared Samet <jared.samet@aya.yale.edu>
2603 | | | | | Date: Fri Nov 18 00:12:40 2016 -0500
2604 | | | | |
2605 | | | | | Merge pull request #58 from Neitsch/copy-argv

```

```

2606 | | | |
2607 | | | | Copy argv
2608 | | | |
2609 | * | | commit 9463afa08811674f462b4a52b8433c8a7c19a7e9
2610 | | \ \ \ Merge: 54858ab bb11d6d
2611 | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
2612 | | | | Date: Thu Nov 17 23:12:41 2016 -0500
2613 | | | |
2614 | | | | Merge branch 'copy-argv' of https://github.com/Neitsch/plt into copy-
      argv
2615 | | | |
2616 | * | | | commit 54858ab5bbcc0b1879f4b4d7387d9e7e839e0677
2617 | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
2618 | | | | Date: Thu Nov 17 23:11:29 2016 -0500
2619 | | | |
2620 | | | | Add => infix operator to cut down on all the build_struct_gep calls
2621 | | | |
2622 | | | * | commit f84757bec65346775db43303f5b02754aa155e29
2623 | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
2624 | | | | Date: Fri Nov 18 00:02:34 2016 -0500
2625 | | | |
2626 | | | | Removed unnecessary files
2627 | | | |
2628 | | | * | commit 18fbff1163bf5660839cc7f2d03ab00299fbec29
2629 | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
2630 | | | | Date: Fri Nov 18 00:01:49 2016 -0500
2631 | | | |
2632 | | | | Removed dummy arg reading, added printing to interpreter - helloworld TC
      passes
2633 | | | |
2634 | | | * | commit b866da35db69204510000fd87c33cb6aade0532d
2635 | | | / / Author: Nigel Schuster <nigel.schusters@googlemail.com>
2636 | | | | Date: Thu Nov 17 23:31:42 2016 -0500
2637 | | | |
2638 | | | | Made hello world work
2639 | | | |
2640 | | * | | commit bb11d6dfa4ba716dfd50347645619d8f79321a66
2641 | | | \ \ Merge: e123652 5e63cee
2642 | | / / / Author: Nigel Schuster <Neitsch@users.noreply.github.com>
2643 | | / / / Date: Thu Nov 17 23:10:24 2016 -0500
2644 | | / /
2645 | / | | Merge branch 'master' into copy-argv
2646 | | |
2647 | * | | | commit 5e63ceebeaf8664dfdedb23271f70769609eb4b2
2648 | \ \ \ \ Merge: cafe20e 4a4a806
2649 | | | | Author: Nigel Schuster <Neitsch@users.noreply.github.com>
2650 | | | | Date: Thu Nov 17 17:40:31 2016 -0500
2651 | | | |
2652 | | | | Merge pull request #54 from Neitsch/operation_tests
2653 | | | |
2654 | | | | Operation tests.
2655 | | | |
2656 | * | | | commit 4a4a8063c0da95694f00b7683f9985f40877cbda
2657 | | \ \ \ Merge: 4b28df2 cafe20e
2658 | | / / / Author: Nigel Schuster <Neitsch@users.noreply.github.com>
2659 | / | | | Date: Thu Nov 17 17:19:13 2016 -0500

```

```

2660 | | | |
2661 | | | | Merge branch 'master' into operation_tests
2662 | | | |
2663 * | | | | commit cafe20e6e0f4dfe020f2c34caafa3f8879a944ee
2664 |\ \ \ \ Merge: d43a87b b728e2e
2665 | | | | | Author: Nigel Schuster <Neitsch@users.noreply.github.com>
2666 | | | | | Date: Thu Nov 17 17:19:11 2016 -0500
2667 | | | | |
2668 | | | | | Merge pull request #52 from Neitsch/one-main-arg
2669 | | | | |
2670 | | | | | Call main() with first argument <empty> in interpreter
2671 | | | | |
2672 | * | | | | commit b728e2e80558871009f78e188adaea92ff9826af
2673 | |\ \ \ \ Merge: e490548 d43a87b
2674 | / / / / Author: Nigel Schuster <Neitsch@users.noreply.github.com>
2675 | / | | | | Date: Thu Nov 17 17:16:20 2016 -0500
2676 | | | | |
2677 | | | | | Merge branch 'master' into one-main-arg
2678 | | | | |
2679 | * | | | | commit e490548c499782a30071fb192311adea1711422f
2680 | |\ \ \ \ Merge: 79ee3de a4cf367
2681 | | | | | Author: Jared Samet <jared.samet@aya.yale.edu>
2682 | | | | | Date: Thu Nov 17 15:50:35 2016 -0500
2683 | | | | |
2684 | | | | | Merge branch 'master' into one-main-arg
2685 | | | | |
2686 | * | | | | commit 79ee3dea137cf7fd4ace406716195cf758cabd75
2687 | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
2688 | | | | | Date: Thu Nov 17 15:18:58 2016 -0500
2689 | | | | |
2690 | | | | | Call main() with first argument <empty> in interpreter
2691 | | | | |
2692 | | | * | | | commit 4b28df26b724eccc92829ca84bd1ccdbc84c6729
2693 | | | |\ \ \ Merge: 3255e1b d43a87b
2694 | | | | / / / Author: Nigel Schuster <Neitsch@users.noreply.github.com>
2695 | / | | | | Date: Thu Nov 17 17:17:44 2016 -0500
2696 | | | | |
2697 | | | | | Merge branch 'master' into operation_tests
2698 | | | | |
2699 * | | | | | commit d43a87b8ab2c8e52d42c16af44d90a48301752c7
2700 |\ \ \ \ \ Merge: a4cf367 b1238a0
2701 | | | / / / / Author: Nigel Schuster <Neitsch@users.noreply.github.com>
2702 | / | | | | Date: Thu Nov 17 17:15:28 2016 -0500
2703 | | | | |
2704 | | | | | Merge pull request #55 from Neitsch/shell-fix
2705 | | | | |
2706 | | | | | Using bourne shell style redirection:
2707 | | | | |
2708 | * | | | | commit b1238a0145da88327094dd2fa59a243cc72856c2
2709 | | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
2710 | | | | | Date: Thu Nov 17 17:08:56 2016 -0500
2711 | | | | |
2712 | | | | | Shell is not my strength
2713 | | | | |
2714 | * | | | | commit a6cc0eafe87931aa48498436441c6351ealbfe0c
2715 | | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>

```



```

2716 | | | | | Date: Thu Nov 17 17:05:09 2016 -0500
2717 | | | | |
2718 | | | | | Screw you bourne shell
2719 | | | | |
2720 | * | | | | commit 51fbe679918ad80440986f53208c9d393a98f2bb
2721 | / / / / / Author: Nigel Schuster <nigel.schusters@googlemail.com>
2722 | | | | | Date: Thu Nov 17 16:59:50 2016 -0500
2723 | | | | |
2724 | | | | | Using bourne shell style redirection:
2725 | | | | |
2726 | | * | | | commit 3255e1b771252f98fbbfffc2067744e7761ee4a2e
2727 | | | | | Author: Ishaan <ishaankolluri@gmail.com>
2728 | | | | | Date: Thu Nov 17 16:32:12 2016 -0500
2729 | | | | |
2730 | | | | | Modify test suite specs
2731 | | | | |
2732 | | * | | | commit f0ab4d8880196a8f693891ef5ab6c8510e9a305d
2733 | | | | | Author: Ishaan <ishaankolluri@gmail.com>
2734 | | | | | Date: Thu Nov 17 16:30:42 2016 -0500
2735 | | | | |
2736 | | | | | Moved expected output text files to directory
2737 | | | | |
2738 | | * | | | commit 06d330c8e0470daaeefa0c06ad7b89a31f2ea4e5
2739 | | / / / Author: Ishaan <ishaankolluri@gmail.com>
2740 | / | | | Date: Thu Nov 17 16:07:02 2016 -0500
2741 | | | | |
2742 | | | | | 75% through operator cases
2743 | | | | |
2744 | | * | | | commit e123652c39885430a3ae0d6773b186cd102d37ac
2745 | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
2746 | | | | | Date: Thu Nov 17 22:28:12 2016 -0500
2747 | | | | |
2748 | | | | | Add byte for zero
2749 | | | | |
2750 | | * | | | commit 26a03b7e32173ed2ef947084d16e25eaa82ff921
2751 | | / / / Author: oracleofnj <jared.samet@aya.yale.edu>
2752 | / | | | Date: Thu Nov 17 22:24:17 2016 -0500
2753 | | | | |
2754 | | | | | Add new_string function
2755 | | | | |
2756 | * | | | | commit a4cf367633dd1dba496a2ec83472d8f513f56aaf
2757 | \ \ \ \ Merge: 7af929a c4f7437
2758 | | / / / Author: Jared Samet <jared.samet@aya.yale.edu>
2759 | / | | | Date: Thu Nov 17 15:50:29 2016 -0500
2760 | | | | |
2761 | | | | | Merge pull request #51 from Neitsch/test-script
2762 | | | | |
2763 | | | | | Test script
2764 | | | | |
2765 | * | | | | commit c4f74379a669e99fa7941712f9d5f3406bd2a390
2766 | | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
2767 | | | | | Date: Thu Nov 17 14:39:38 2016 -0500
2768 | | | | |
2769 | | | | | Removed version specific lli
2770 | | | | |
2771 | * | | | | commit 7b2236b3671a30fe5c49dcd15de8e1d7ec40fd7d

```

```

2772 | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
2773 | | | Date: Thu Nov 17 14:35:55 2016 -0500
2774 | | |
2775 | | | Fixed if no flag is given
2776 | | |
2777 | * | commit e10f6566f1d0a786eb103f542bd5572a7dc049b1
2778 | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
2779 | | | Date: Thu Nov 17 14:24:20 2016 -0500
2780 | | |
2781 | | | Outputting diff only if -p flag is given
2782 | | |
2783 | * | commit 2d2959718512cd1c4ab4d8618595e0fbf5a1afb2
2784 | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
2785 | | | Date: Thu Nov 17 14:19:30 2016 -0500
2786 | | |
2787 | | | Added it as build target
2788 | | |
2789 | * | commit 7af929a4a10c0b447752ddb02b189ff74e48c7ed
2790 | \ \ \ Merge: 7feb392 6ea43f6
2791 | | / / Author: Jared Samet <jared.samet@aya.yale.edu>
2792 | | | Date: Thu Nov 17 14:12:19 2016 -0500
2793 | | |
2794 | | | Merge pull request #50 from Neitsch/test-script
2795 | | |
2796 | | | Test script
2797 | | |
2798 | * | commit 6ea43f63f5e0e05287b1b7c85d60b25a3fa471d9
2799 | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
2800 | | | Date: Thu Nov 17 13:54:55 2016 -0500
2801 | | |
2802 | | | Added more env variables to avoid copy paste
2803 | | |
2804 | * | commit 05f27a219ca18a846664c65754bf682a6bc3c58d
2805 | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
2806 | | | Date: Thu Nov 17 12:45:11 2016 -0500
2807 | | |
2808 | | | Made simple testscript
2809 | | |
2810 | * | commit aca43c139c3b3a1c925473669f664b41acc7c0df
2811 | / / Author: Nigel Schuster <nigel.schusters@googlemail.com>
2812 | | Date: Thu Nov 17 11:08:11 2016 -0500
2813 | | |
2814 | | | Removed accidentally added files
2815 | | |
2816 | * | commit 00aafb71be96f422170e12cd38bd29a624cae40f
2817 | | Author: Kevin <kevinyle1113@gmail.com>
2818 | | Date: Fri Nov 18 13:16:08 2016 -0500
2819 | | |
2820 | | | Renamed inputs folder
2821 | | |
2822 | * | commit 99db652ae157dce96aa01290d6a492e4467e63ed
2823 | | Author: Kevin <kevinyle1113@gmail.com>
2824 | | Date: Fri Nov 18 12:51:40 2016 -0500
2825 | | |
2826 | | | Renamed expected output extension and created input folder for test cases
2827 | | |

```

```

2828 | * commit b8028f9b923d3cdd5600088f2582c8b29bc9bc5b
2829 | | Author: Kevin <kevinye1113@gmail.com>
2830 | | Date: Thu Nov 17 20:27:37 2016 -0500
2831 | |
2832 | |     Removed files from test folder
2833 | |
2834 | * commit c85d9b71b9b5073e2f2e2b2ecaf5253c2ffc1203
2835 | | Author: Kevin <kevinye1113@gmail.com>
2836 | | Date: Thu Nov 17 20:25:21 2016 -0500
2837 | |
2838 | |     Move testcases to testcases directory
2839 | |
2840 | * commit f17c6b67700996588a2696886faba09bc5484880
2841 | | Author: Kevin Ye <kevinye1113@gmail.com>
2842 | | Date: Thu Nov 17 20:21:38 2016 -0500
2843 | |
2844 | |     Complete testcases for List/Range/Function/Expression with expected outputs
2845 | |
2846 | * commit 9228eacadf83a6c901ce34947cad67eceb1dabca
2847 | / Author: Kevin Ye <kevinye1113@gmail.com>
2848 | Date: Thu Nov 17 04:52:31 2016 -0500
2849 |
2850 |     Test cases for List of Tests and Range/Function/Expression Tests
2851 |
2852 * commit 7feb392e1193b63f71f99995db0981330668253a
2853 | \ Merge: 41ef578 6e42afa
2854 | | Author: Ishaan Kolluri <ishaankolluri@users.noreply.github.com>
2855 | | Date: Thu Nov 17 00:28:53 2016 -0500
2856 | |
2857 | |     Merge pull request #48 from Neitsch/testing_list
2858 | |
2859 | |     Added initial testing list
2860 | |
2861 | * commit 6e42afa596a96286149edacb4a44a0ff0c6f2a47
2862 | | \ Merge: e40734b 41ef578
2863 | | / Author: Ishaan Kolluri <ishaankolluri@users.noreply.github.com>
2864 | / Date: Thu Nov 17 00:27:13 2016 -0500
2865 | |
2866 | |     Merge branch 'master' into testing_list
2867 | |
2868 * | commit 41ef57814b0e03cb890a26a11021150a1cb6793f
2869 | \ \ Merge: 1570836 3cbf089
2870 | | | Author: Jared Samet <jared.samet@aya.yale.edu>
2871 | | | Date: Wed Nov 16 17:50:03 2016 -0500
2872 | | |
2873 | | |     Merge pull request #49 from Neitsch/consume-command-line-args
2874 | | |
2875 | | |     Consume command line args
2876 | | |
2877 | * | commit 3cbf089e698f117fe7c093c0ad965ae75f7f5060
2878 | | \ Merge: 2fa73be 1570836
2879 | | / Author: oracleofnj <jared.samet@aya.yale.edu>
2880 | / | Date: Wed Nov 16 17:45:58 2016 -0500
2881 | | |
2882 | | |     Fix merge conflict
2883 | | |

```

```

2884 * | | | commit 1570836bf3643d233e52d1ceb1d46e072e37222a
2885 |\ \ \ Merge: 4a1fcac a8fbced
2886 | | | | Author: Nigel Schuster <Neitsch@users.noreply.github.com>
2887 | | | | Date: Wed Nov 16 16:51:05 2016 -0500
2888 | | | |
2889 | | | | Merge pull request #45 from Neitsch/doc
2890 | | | |
2891 | | | | Added a little documentation
2892 | | | |
2893 | * | | | commit a8fbcedf43a46b3e398838e15b1f713ba5874ad3
2894 | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
2895 | | | | Date: Wed Nov 16 16:38:49 2016 -0500
2896 | | | |
2897 | | | | Fixed minor syntax error
2898 | | | |
2899 | * | | | commit c2f37c8a801eda2b33c09d24ab762e10c0336168
2900 | |\ \ \ Merge: 92fb7a3 4a1fcac
2901 | | / / / Author: Nigel Schuster <nigel.schusters@googlemail.com>
2902 | / | | | Date: Wed Nov 16 16:30:43 2016 -0500
2903 | | | |
2904 | | | | Merge
2905 | | | |
2906 | * | | | commit 92fb7a326d3811b20ca2f64938fc46e3879dc090
2907 | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
2908 | | | | Date: Tue Nov 15 23:57:37 2016 -0500
2909 | | | |
2910 | | | | Added a little documentation
2911 | | | |
2912 | | * | | commit 2fa73be6cebe75a58364148689a0b496940cccd9
2913 | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
2914 | | | | Date: Wed Nov 16 16:05:37 2016 -0500
2915 | | | |
2916 | | | | Set return code to length of argv[1]
2917 | | | |
2918 | | * | | commit cd0d156ceeab4830df0f3a30b3265f34d3c07c68
2919 | | / / Author: oracleofnj <jared.samet@aya.yale.edu>
2920 | / | | Date: Wed Nov 16 15:50:39 2016 -0500
2921 | | | |
2922 | | | | Start processing command line args
2923 | | | |
2924 | | * | | commit e40734b736193f27f346eda80f4748132316db43
2925 | | | | Author: Ishaan <ishaankolluri@gmail.com>
2926 | | | | Date: Wed Nov 16 23:25:01 2016 -0500
2927 | | | |
2928 | | | | Added more test scenarios
2929 | | | |
2930 | | * | | commit bc21af622d3a32bb6c9fe2e86f83129886b8e574
2931 | | / / Author: Ishaan <ishaankolluri@gmail.com>
2932 | / | | Date: Wed Nov 16 15:54:12 2016 -0500
2933 | | | |
2934 | | | | Added initial testing list
2935 | | | |
2936 | * | | | commit 4a1fcacf86ee892bc72582fd2ea3a9dc74d0e63c
2937 | \ \ \ Merge: 8944b9a flb481e
2938 | | | | Author: Nigel Schuster <Neitsch@users.noreply.github.com>
2939 | | | | Date: Wed Nov 16 13:55:46 2016 -0500

```

```

2940 | | |
2941 | | | Merge pull request #46 from Neitsch/number-type
2942 | | |
2943 | | | Added number type that defaults to int
2944 | | |
2945 | * | commit flb481efad2233c37a8722580c655266116dad55
2946 | / / Author: Nigel Schuster <nigel.schusters@googlemail.com>
2947 | | Date: Wed Nov 16 11:04:44 2016 -0500
2948 | |
2949 | | Added number type that defaults to int
2950 | |
2951 * | commit 8944b9a1fb299e43a8de33dfb342243f75f7718a
2952 | \ \ Merge: fal741a bcbde36
2953 | / / Author: Nigel Schuster <Neitsch@users.noreply.github.com>
2954 | / | Date: Wed Nov 16 00:19:33 2016 -0500
2955 | |
2956 | | Merge pull request #44 from Neitsch/fix-arg
2957 | |
2958 | | Using subranges instead of ranges everywhere
2959 | |
2960 | * commit bcbde3610ba04bc8b125f34a502ad20f1ae80bd8
2961 | | \ Merge: 57b2162 fal741a
2962 | | / Author: Nigel Schuster <Neitsch@users.noreply.github.com>
2963 | / | Date: Tue Nov 15 23:49:07 2016 -0500
2964 | |
2965 | | Merge branch 'master' into fix-arg
2966 | |
2967 * | commit fal741a24c0bd75650d06a79fee7ea1401382c0a
2968 | \ \ Merge: 660c049 9407677
2969 | | | Author: Jared Samet <jared.samet@aya.yale.edu>
2970 | | | Date: Tue Nov 15 23:03:23 2016 -0500
2971 | | |
2972 | | | Merge pull request #43 from Neitsch/more-llvm-gen-js
2973 | | |
2974 | | | More llvm gen from JS
2975 | | |
2976 | * | commit 9407677c52644d935175cf5abe25a31127bf1787
2977 | | | Author: oracleofnj <jared.samet@aya.yale.edu>
2978 | | | Date: Tue Nov 15 22:31:03 2016 -0500
2979 | | |
2980 | | | Add hash table for common functions and add dereference-the-range
2981 | | |
2982 | * | commit 46elfd5752c75220787e60a50a8c51064e50c7ce
2983 | / / Author: oracleofnj <jared.samet@aya.yale.edu>
2984 | | Date: Tue Nov 15 21:38:51 2016 -0500
2985 | |
2986 | | Eliminate some copy & paste
2987 | |
2988 | * commit 57b2162706267d8ae19c759d4ef4b1199f9946e3
2989 | / Author: Nigel Schuster <nigel.schusters@googlemail.com>
2990 | Date: Tue Nov 15 22:39:38 2016 -0500
2991 |
2992 | Using subranges instead of ranges everywhere
2993 |
2994 * | commit 660c049fb22baac3526cd251b0b366ee3d1b8305
2995 | \ Merge: 8ad5a19 25b23cd

```

```

2996 | | Author: Jared Samet <jared.samet@aya.yale.edu>
2997 | | Date: Tue Nov 15 20:54:33 2016 -0500
2998 | |
2999 | | Merge pull request #42 from Neitsch/llvm-gen
3000 | |
3001 | | Lllvm gen
3002 | |
3003 | * commit 25b23cd3df1e93a00a330b2e02166d41e4dfea
3004 | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
3005 | | Date: Tue Nov 15 17:23:54 2016 -0500
3006 | |
3007 | | Fixed column retrieval for 1x1
3008 | |
3009 | * commit 3f022031254b6da7a3278153288a09d052d16fa2
3010 | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
3011 | | Date: Tue Nov 15 17:17:02 2016 -0500
3012 | |
3013 | | Fixed tests
3014 | |
3015 | * commit 26b8fcf1a26394ee6c82f447330ef2510aa32dc8
3016 | | Merge: e347a87 aed28b3
3017 | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
3018 | | Date: Tue Nov 15 17:15:08 2016 -0500
3019 | |
3020 | | Merge
3021 | |
3022 | * commit aed28b3e2ad836558e1ecdb4afb276f5bd022114
3023 | | Author: oracleofnj <jared.samet@aya.yale.edu>
3024 | | Date: Tue Nov 15 17:08:07 2016 -0500
3025 | |
3026 | | Add is_subrange_1x1
3027 | |
3028 | * commit cf5cbf0613395995f0ee23f19ab4a1a879364385
3029 | | Merge: c71d469 4b34abd
3030 | | Author: oracleofnj <jared.samet@aya.yale.edu>
3031 | | Date: Tue Nov 15 14:51:40 2016 -0500
3032 | |
3033 | | Merge branch 'llvm-gen' of https://github.com/Neitsch/plt into llvm-gen
3034 | |
3035 | | * commit 4b34abd125b1651df31211d8779b7c549d515ae7
3036 | | Merge: a80a6d0 8ad5a19
3037 | | Author: Nigel Schuster <Neitsch@users.noreply.github.com>
3038 | | Date: Tue Nov 15 14:41:37 2016 -0500
3039 | |
3040 | | Merge branch 'master' into llvm-gen
3041 | |
3042 | * commit 8ad5a19ed1d3e25da7ee9995fa1079fab3d08cb5
3043 | | Merge: d6daff3 3f0362a
3044 | | Author: Jared Samet <jared.samet@aya.yale.edu>
3045 | | Date: Tue Nov 15 14:33:40 2016 -0500
3046 | |
3047 | | Merge pull request #40 from Neitsch/interpreter
3048 | |
3049 | | Interpreter
3050 | |
3051 | * commit 3f0362aff5a44f3c39f0d991c5cc11b85dc7c47e

```

```

3052 | |\ \ \ \ Merge: d5f4024 d6daff3
3053 | | / / / Author: Jared Samet <jared.samet@aya.yale.edu>
3054 | / | | | Date: Tue Nov 15 14:28:44 2016 -0500
3055 | | | | |
3056 | | | | | Merge branch 'master' into interpreter
3057 | | | | |
3058 | * | | | | commit d6daff3a9c877c8277f1bd86f080f0defa597ead
3059 | \ \ \ \ \ Merge: 443a818 6afe599
3060 | | | | | Author: Jared Samet <jared.samet@aya.yale.edu>
3061 | | | | | Date: Sun Nov 13 20:26:14 2016 -0500
3062 | | | | |
3063 | | | | | Merge pull request #41 from Neitsch/LRM_String_Update
3064 | | | | |
3065 | | | | | Added changes relating to strings.
3066 | | | | |
3067 | * | | | | commit 6afe5996ee757b714490685718f04027e04532ab
3068 | | | | | Author: Ishaan Kolluri <ishaankolluri@gmail.com>
3069 | | | | | Date: Sun Nov 13 18:44:58 2016 -0500
3070 | | | | |
3071 | | | | | Added changes relating to strings.
3072 | | | | |
3073 | | | | * | commit c71d469cf691b624960ee4657697d623f0055f55
3074 | | | | | / Author: oracleofnj <jared.samet@aya.yale.edu>
3075 | | | | | Date: Tue Nov 15 14:51:19 2016 -0500
3076 | | | | |
3077 | | | | | Replace String.equal with =
3078 | | | | |
3079 | | | | * | commit a80a6d0053c71eb473fbb5b3cdcaca004be4b48
3080 | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
3081 | | | | | Date: Tue Nov 15 14:41:07 2016 -0500
3082 | | | | |
3083 | | | | | Add compile option to main
3084 | | | | |
3085 | | | | * | commit e347a876c8713deff609324bae48d55ba8a69f04
3086 | | | | / Author: Nigel Schuster <nigel.schusters@gmail.com>
3087 | | | | | Date: Tue Nov 15 17:12:26 2016 -0500
3088 | | | | |
3089 | | | | | Using more generic flag for values
3090 | | | | |
3091 | | | | * | commit c0c95a24323efbbe0abab2632671275c6ba0854a
3092 | | | | \ Merge: aa61ac9 d5f4024
3093 | | | | / Author: Nigel Schuster <nigel.schusters@gmail.com>
3094 | | | | / Date: Tue Nov 15 14:16:13 2016 -0500
3095 | | | | |
3096 | | | | | Merge
3097 | | | | |
3098 | | * | | commit d5f4024aab33d3307ddea75005f9885dc03395dc
3099 | | | | | Author: Nigel Schuster <nigel.schusters@gmail.com>
3100 | | | | | Date: Tue Nov 15 13:44:44 2016 -0500
3101 | | | | |
3102 | | | | | Moved failing TCs
3103 | | | | |
3104 | | * | | commit 42fd9ef88b59f80351a1170a3c5fc3d393ad55b6
3105 | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
3106 | | | | | Date: Tue Nov 15 12:21:57 2016 -0500
3107 | | | | |

```

```

3108 | | | | Fix bug in import
3109 | | | |
3110 | | * | commit 9c567c954d5e4f576f1ffaec3c5f152e86761c65
3111 | | | | Author: Nigel Schuster <nigel.schusters@gmail.com>
3112 | | | | Date: Tue Nov 15 11:11:30 2016 -0500
3113 | | | |
3114 | | | | Working on imports, fixed most testcases
3115 | | | |
3116 | | * | commit cf1ebf94b4dfacc0163e479d2ed6979146b47e2e
3117 | | | | Author: oraclofnj <jared.samet@aya.yale.edu>
3118 | | | | Date: Sun Nov 13 23:09:30 2016 -0500
3119 | | | |
3120 | | | | Rewrite main to take options; fix bug where import didn't know about first
      | | | | filename
3121 | | | |
3122 | | * | commit 0a5d484dba0b5e288e3b2bad4d6a4c044e1f8da9
3123 | | | | Author: oraclofnj <jared.samet@aya.yale.edu>
3124 | | | | Date: Sun Nov 13 18:45:29 2016 -0500
3125 | | | |
3126 | | | | Revert "Generating function header"
3127 | | | |
3128 | | | | This reverts commit f83a0bcc6ca8d58d79bdfb4b49471c6031e8d201.
3129 | | | |
3130 | | * | commit 137d7e2b95b7340894b1d826dc38ef91a958e731
3131 | | | \ Merge: 118bfc5 bf1d8bb
3132 | | | | Author: oraclofnj <jared.samet@aya.yale.edu>
3133 | | | | Date: Sun Nov 13 18:39:33 2016 -0500
3134 | | | |
3135 | | | | Merge branch 'interpreter' of https://github.com/Neitsch/plt into
      | | | | interpreter
3136 | | | |
3137 | | * | commit 118bfc54ba67ecba87442b26112557c59868220e
3138 | | | | Author: oraclofnj <jared.samet@aya.yale.edu>
3139 | | | | Date: Sun Nov 13 18:38:34 2016 -0500
3140 | | | |
3141 | | | | Allow single slice on RHS; make hashtag work
3142 | | | |
3143 | | * | commit 3addcc8b8c5029a7635a4b7edb5d718917618304
3144 | | | | Author: oraclofnj <jared.samet@aya.yale.edu>
3145 | | | | Date: Sun Nov 13 14:38:11 2016 -0500
3146 | | | |
3147 | | | | Make size(expr) an operator instead of built-in function
3148 | | | |
3149 | | * | commit 9a74e14abc798cab3268420451b7b9f3a9a2b7aa
3150 | | | | Author: oraclofnj <jared.samet@aya.yale.edu>
3151 | | | | Date: Sun Nov 13 14:22:44 2016 -0500
3152 | | | |
3153 | | | | Changing size() to be an operator
3154 | | | |
3155 | | | * commit aa61ac98d42571ab91a54ff06cb80be3deaa76b7
3156 | | | | Author: Nigel Schuster <nigel.schusters@gmail.com>
3157 | | | | Date: Tue Nov 15 09:31:42 2016 -0500
3158 | | | |
3159 | | | | Allocating scope object
3160 | | | |
3161 | | | * commit 574953833ac17b8d9c86faa967568ee8a27c7c22

```



```

3162 | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
3163 | | | | Date: Sun Nov 13 21:59:28 2016 -0500
3164 | | | |
3165 | | | | Added main function
3166 | | | |
3167 | | | | * commit e376270c9b5af196cac37f5cd2f64071b9768ace
3168 | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
3169 | | | | Date: Sun Nov 13 17:55:41 2016 -0500
3170 | | | |
3171 | | | | Added type arguments for functions
3172 | | | |
3173 | | | | * commit 5cfb5199f5effd2ce86663bfce7ba388a169bb6f
3174 | | | | / Author: Nigel Schuster <nigel.schusters@googlemail.com>
3175 | | | | Date: Sun Nov 13 17:26:23 2016 -0500
3176 | | | |
3177 | | | | Set more types up
3178 | | | |
3179 | | | | * commit bf1d8bb2b3965e4aeab26a1f4d1c91569c2eb6f2
3180 | | | | \ Merge: f83a0bc d6d2eaa
3181 | | | | / Author: Nigel Schuster <nigel.schusters@googlemail.com>
3182 | | | | / Date: Sun Nov 13 15:30:35 2016 -0500
3183 | | | |
3184 | | | | Merge branch 'interpreter' of https://github.com/Neitsch/plt into
      | | | | interpreter
3185 | | | |
3186 | | | | * commit d6d2eaa5440fbb39958b104a520197857514ea06
3187 | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
3188 | | | | Date: Sun Nov 13 00:08:41 2016 -0500
3189 | | | |
3190 | | | | Add closure to interpreter_variable
3191 | | | |
3192 | | | | * commit 64fba825d3c3d6eb216bfa6228cab91c8985d048
3193 | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
3194 | | | | Date: Sat Nov 12 22:38:39 2016 -0500
3195 | | | |
3196 | | | | Added bsearch to show logic bug
3197 | | | |
3198 | | | | * commit 66ffdb1cdd48dfdb93653a30e07ea47573fbb42c
3199 | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
3200 | | | | Date: Sat Nov 12 19:21:07 2016 -0500
3201 | | | |
3202 | | | | Add alpha version of function calls
3203 | | | |
3204 | | | | * commit 376b29aelf4b72fedb438e13b5fd1b3373c827bb
3205 | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
3206 | | | | Date: Sat Nov 12 17:17:23 2016 -0500
3207 | | | |
3208 | | | | Add string as value type
3209 | | | |
3210 | | | | * commit 08c61ee7724a15c9f95f1afbaabfe69aff63c2a4
3211 | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
3212 | | | | Date: Sat Nov 12 17:14:47 2016 -0500
3213 | | | |
3214 | | | | Clean up discrepancies
3215 | | | |
3216 | | | | * commit f83a0bcc6ca8d58d79bdfb4b49471c6031e8d201

```

```

3217 | | | / Author: Nigel Schuster <nigel.schusters@googlemail.com>
3218 | | | Date: Sun Nov 13 15:30:28 2016 -0500
3219 | | |
3220 | | | Generating function header
3221 | | |
3222 | | | * commit a18d5fc571d7ad30c4720b2a2ab5a58865726120
3223 | | | Author: oracleofnj <jared.samet@aya.yale.edu>
3224 | | | Date: Tue Nov 8 11:38:22 2016 -0500
3225 | | |
3226 | | | Fix bug with x[-1]
3227 | | |
3228 | | | * commit 962f81284eea2537d5b92451fb5494e52c1600a1
3229 | | | Author: oracleofnj <jared.samet@aya.yale.edu>
3230 | | | Date: Mon Nov 7 23:27:08 2016 -0500
3231 | | |
3232 | | | Refactor scope for interpreter; resolve variables on demand; make selections
    | | | work properly
3233 | | |
3234 | | | * commit 47bbef19ea8c13e555d5751a1ae4f369b8d0007d
3235 | | | Author: oracleofnj <jared.samet@aya.yale.edu>
3236 | | | Date: Sun Nov 6 22:05:55 2016 -0500
3237 | | |
3238 | | | Minor adjustments to interpreter to work with mapped AST
3239 | | |
3240 | | | * commit fddc6bc63b746a5aa19fb96fc1100a9da62ef94f
3241 | | | Author: oracleofnj <jared.samet@aya.yale.edu>
3242 | | | Date: Sun Nov 6 18:32:17 2016 -0500
3243 | | |
3244 | | | Eliminate extraneous nulls in JSON
3245 | | |
3246 | | | * commit ffddb17fb3d6995f44b24199a1fa6a45289ec8a3
3247 | | | Author: oracleofnj <jared.samet@aya.yale.edu>
3248 | | | Date: Sun Nov 6 18:15:40 2016 -0500
3249 | | |
3250 | | | Turn statement and function lists into StringMaps
3251 | | |
3252 | | | * Lists of statements are organized into StringMaps, mapping
3253 | | | * variable names to (dimensions, [formula assignments])
3254 | | | * Each dimension is either an integer or an ID (no "Some")
3255 | | | * For each formula assignment, row_start, row_end, col_start, col_end
3256 | | | * are now either integers or IDs
3257 | | | *
3258 | | | * Lists of functions are organized into StringMaps, mapping function
3259 | | | * names to (parameters, StringMap{variable name: def}, return val)
3260 | | | *
3261 | | | * Throws an exception if duplicate variable definitions or function
3262 | | | * definitions are encountered
3263 | | | *
3264 | | | * Throws an exception if an assignment to an unknown variable
3265 | | | * is encountered
3266 | | |
3267 | | | * commit 6810003e25ead7a1cb013849cb94600ab501dae8
3268 | | | Author: oracleofnj <jared.samet@aya.yale.edu>
3269 | | | Date: Sat Nov 5 19:47:57 2016 -0400
3270 | | |
3271 | | | Fix pattern matching warning

```

```

3272 | | |
3273 | | | * commit 7107a46504b717448dcd1f51a969fe5d0fc289e8
3274 | | | Author: oracleofnj <jared.samet@aya.yale.edu>
3275 | | | Date: Sat Nov 5 18:01:34 2016 -0400
3276 | | |
3277 | | | Add function to check range literals for legality at parse time
3278 | | |
3279 | | | * commit 80b13d19d1c44db5109f6bcaf2a484175a0340a6
3280 | | | Author: oracleofnj <jared.samet@aya.yale.edu>
3281 | | | Date: Sat Nov 5 15:13:10 2016 -0400
3282 | | |
3283 | | | Handle selections better
3284 | | |
3285 | | | * commit 6cbb009ff6e6d1cee972a5cf79520797c685a52f
3286 | | | Author: oracleofnj <jared.samet@aya.yale.edu>
3287 | | | Date: Fri Nov 4 15:48:58 2016 -0400
3288 | | |
3289 | | | Count to 1,000,000 using tail-recursive versions of List.map and cartesian
    | | | product
3290 | | |
3291 | | | * commit 9b2252d163cb36dee96cc654b3b33a2adb96825e
3292 | | | Author: oracleofnj <jared.samet@aya.yale.edu>
3293 | | | Date: Fri Nov 4 15:25:13 2016 -0400
3294 | | |
3295 | | | Show enter and exit
3296 | | |
3297 | | | * commit 3585e432acf0c6aef34585cad6ef491f1e0aa1cd
3298 | | | Author: oracleofnj <jared.samet@aya.yale.edu>
3299 | | | Date: Fri Nov 4 02:21:38 2016 -0400
3300 | | |
3301 | | | See how high it can count recursively
3302 | | |
3303 | | | * commit 38cf541bfd31e7fe9458c287c39c639ddda2792f
3304 | | | Author: oracleofnj <jared.samet@aya.yale.edu>
3305 | | | Date: Fri Nov 4 02:15:50 2016 -0400
3306 | | |
3307 | | | Get the easy parts of the interpreter working
3308 | | |
3309 | | | * commit 5d81d6ebc492cd13db7ea52a35f104512a5873c9
3310 | | | Author: oracleofnj <jared.samet@aya.yale.edu>
3311 | | | Date: Thu Nov 3 17:17:51 2016 -0400
3312 | | |
3313 | | | Start working on interpreter
3314 | | |
3315 | | | * commit 0078cee2ecbb431212849db233909580a2bb8c41
3316 | | | Author: oracleofnj <jared.samet@aya.yale.edu>
3317 | | | Date: Tue Nov 1 23:40:57 2016 -0400
3318 | | |
3319 | | | Got a non-tail-recursive version of topological sort working
3320 | | |
3321 | | | * commit 85df17519a57bf08f21d20c97c7c768e294673ac
3322 | | | Author: oracleofnj <jared.samet@aya.yale.edu>
3323 | | | Date: Tue Nov 1 15:39:10 2016 -0400
3324 | | |
3325 | | | Irrelevant highlighting thing
3326 | | |

```

```

3327 | | * commit 84c719aaaf45a702e2ad81ffb08b52505d780b3f
3328 | | | Author: oracleofnj <jared.samet@aya.yale.edu>
3329 | | | Date: Tue Nov 1 14:39:49 2016 -0400
3330 | | |
3331 | | | Rearrange nested functions
3332 | | |
3333 | | * commit 557dc4e32897ad2a47585d6a64cee6f649813000
3334 | | | Author: oracleofnj <jared.samet@aya.yale.edu>
3335 | | | Date: Tue Nov 1 13:50:52 2016 -0400
3336 | | |
3337 | | | Add circular import test case
3338 | | |
3339 | | * commit c47679884ae0408cce7352f2b7c12032596d1813
3340 | | | Author: oracleofnj <jared.samet@aya.yale.edu>
3341 | | | Date: Tue Nov 1 13:35:46 2016 -0400
3342 | | |
3343 | | | Fix syntax errors
3344 | | |
3345 | | * commit af5a31da8b2c2dbf99b1f7238aa75f5424a4917a
3346 | | | \ Merge: 02ca24f d451cc4
3347 | | | | Author: Jared Samet <jared.samet@aya.yale.edu>
3348 | | | | Date: Tue Nov 1 13:31:49 2016 -0400
3349 | | | |
3350 | | | | Merge pull request #37 from Neitsch/import-rec
3351 | | | |
3352 | | | | Recursively looking up dependencies
3353 | | | |
3354 | | | * commit d451cc42d244b6fbab30c782cb2c2a4005be539b
3355 | | | | \ Merge: 6a28c05 e673dca
3356 | | | | | Author: Jared Samet <jared.samet@aya.yale.edu>
3357 | | | | | Date: Tue Nov 1 13:31:33 2016 -0400
3358 | | | | |
3359 | | | | | Merge pull request #38 from Neitsch/import-load
3360 | | | | |
3361 | | | | | Loading data from all imports
3362 | | | | |
3363 | | | | * commit e673dca67cec24791211af2b9f80791cb14de39d
3364 | | | | / Author: Neitsch <ns3158@columbia.edu>
3365 | | | | | Date: Mon Oct 31 15:56:43 2016 -0400
3366 | | | | |
3367 | | | | | Loading data from all imports
3368 | | | | |
3369 | | | * commit 6a28c05cd7632d70aedb70fdcfdc8d264e3e3479
3370 | | | | Author: Neitsch <ns3158@columbia.edu>
3371 | | | | Date: Mon Oct 31 15:40:41 2016 -0400
3372 | | | |
3373 | | | | Recursively looking up dependencies
3374 | | | |
3375 | | | * commit 02ca24fac16c38f7b7f2dcbe2010ec04af49e007
3376 | | | \ \ Merge: 3f28289 6fa0e39
3377 | | | | | Author: Jared Samet <jared.samet@aya.yale.edu>
3378 | | | | | Date: Tue Nov 1 13:30:47 2016 -0400
3379 | | | | |
3380 | | | | | Merge pull request #39 from Neitsch/wild-exc
3381 | | | | |
3382 | | | | | Raising exceptions on certain values

```

```

3383 | | | | |
3384 | | | * | commit 6fa0e392019110045c14f1cdddc007e2bf5533ec
3385 | | | | / Author: Neitsch <ns3158@columbia.edu>
3386 | | | | Date: Mon Oct 31 16:43:17 2016 -0400
3387 | | | |
3388 | | | | Raising exceptions on certain values
3389 | | | |
3390 | | | * | commit 3f28289a73cf2fe837b2b44eaab7b105bb2b81fb
3391 | | | | \ Merge: 7d70af2 4eae3b
3392 | | | | / Author: Jared Samet <jared.samet@aya.yale.edu>
3393 | | | | / Date: Mon Oct 31 11:53:10 2016 -0400
3394 | | | |
3395 | | | | Merge pull request #36 from Neitsch/import-arrange
3396 | | | |
3397 | | | | Added unsorted function, globals and imports
3398 | | | |
3399 | | | * commit 4eae3b64c7b0404fea7876990e621d4f6ded907
3400 | | | | Author: Neitsch <ns3158@columbia.edu>
3401 | | | | Date: Mon Oct 31 11:01:00 2016 -0400
3402 | | | |
3403 | | | | Removed obsolete parts
3404 | | | |
3405 | | | * commit 7d7b1e51de024be3acc6f276df2027ea80c8b35f
3406 | | | / Author: Neitsch <ns3158@columbia.edu>
3407 | | | | Date: Mon Oct 31 10:59:12 2016 -0400
3408 | | | |
3409 | | | | Added unsorted function, globals and imports
3410 | | | |
3411 | | | * commit 7d70af2345c2cc0d09944e19f7517b0378635ce7
3412 | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
3413 | | | | Date: Sun Oct 30 15:23:04 2016 -0400
3414 | | | |
3415 | | | | Add some explanatory comments
3416 | | | |
3417 | | | * commit 40d6b169baf47be6839cc792f7fa17e79b83dab6
3418 | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
3419 | | | | Date: Sun Oct 30 15:03:32 2016 -0400
3420 | | | |
3421 | | | | More expansion samples
3422 | | | |
3423 | | | * commit af9b01c245de91965ee64957c7ce5112e8e059db
3424 | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
3425 | | | | Date: Sun Oct 30 14:48:44 2016 -0400
3426 | | | |
3427 | | | | Refactor expansion code
3428 | | | |
3429 | | | * commit 903bc3fe44b93382ae8211e1828a66526954058a
3430 | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
3431 | | | | Date: Sun Oct 30 00:19:10 2016 -0400
3432 | | | |
3433 | | | | Add test output
3434 | | | |
3435 | | | * commit 68b7b0340a9b6e21ca71140207c46a00e36d3396
3436 | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
3437 | | | | Date: Sun Oct 30 00:17:02 2016 -0400
3438 | | | |

```

```

3439 | | | Add test case
3440 | | |
3441 | | * commit a8bdf33619977487b629fc12bb99a66a5a5c41ed
3442 | | | Author: oracleofnj <jared.samet@aya.yale.edu>
3443 | | | Date: Sun Oct 30 00:04:05 2016 -0400
3444 | | |
3445 | | | Add LHS slice expansion
3446 | | |
3447 | | * commit 4ee6fdf258900ecbae3a8d1efb66e1019c188ce8
3448 | | | Author: oracleofnj <jared.samet@aya.yale.edu>
3449 | | | Date: Sat Oct 29 17:36:17 2016 -0400
3450 | | |
3451 | | | Add output
3452 | | |
3453 | | * commit 2b8bced61855a2361384912a47b3459c98828e2c
3454 | | / Author: oracleofnj <jared.samet@aya.yale.edu>
3455 | / | Date: Sat Oct 29 17:27:22 2016 -0400
3456 | | |
3457 | | | Expand dimension expressions
3458 | | |
3459 * | | commit 443a818dc4fd9ec42da4afd6e733dca35cc198e8
3460 | \ \ Merge: 022e8cd 9ba3c65
3461 | / | Author: Ishaan Kolluri <ishaankolluri@users.noreply.github.com>
3462 | | | Date: Wed Oct 26 16:31:51 2016 -0400
3463 | | |
3464 | | | Merge pull request #35 from ishaankolluri/master
3465 | | |
3466 | | | Add UNIs
3467 | | |
3468 | | * commit 9ba3c657b5a67bc5d2fc69be83b2e73809ab96ce
3469 | | | Author: Ishaan Kolluri <ishaankolluri@gmail.com>
3470 | | | Date: Wed Oct 26 16:31:00 2016 -0400
3471 | | |
3472 | | | Add UNIs
3473 | | |
3474 * | | commit 022e8cdff18e452b0d3c518836141f30f93a9356
3475 | \ \ Merge: 0bd9c4a 808aae5
3476 | / | Author: Ishaan Kolluri <ishaankolluri@users.noreply.github.com>
3477 | | | Date: Wed Oct 26 16:25:57 2016 -0400
3478 | | |
3479 | | | Merge pull request #34 from ishaankolluri/master
3480 | | |
3481 | | | One more edit
3482 | | |
3483 | | * commit 808aae5ee501ea7b86371ffe3b03ca489af02c07
3484 | / | Author: Ishaan Kolluri <ishaankolluri@gmail.com>
3485 | | | Date: Wed Oct 26 16:22:10 2016 -0400
3486 | | |
3487 | | | Added change to precedence operators
3488 | | |
3489 * | | commit 0bd9c4af5b3575b5c8f60973b8a2e9593ec207ee
3490 | \ \ Merge: e7020ec fb2b382
3491 | | | Author: Jared Samet <jared.samet@aya.yale.edu>
3492 | | | Date: Wed Oct 26 15:59:53 2016 -0400
3493 | | |
3494 | | | Merge pull request #33 from Neitsch/final-slicing-comments

```

```

3495 | |
3496 | |     Thats all for now folks
3497 | |
3498 | * commit fb2b382bd13dcdd469a3c2d7a6cce429f41ea632
3499 |/ Author: oracleofnj <jared.samet@aya.yale.edu>
3500 | Date:   Wed Oct 26 15:54:11 2016 -0400
3501 |
3502 |     Thats all for now folks
3503 |
3504 *   commit e7020ec688f7a86453b383714d8b5db98c699d2a
3505 |\ Merge: 4b7984a 4683f14
3506 | | Author: Jared Samet <jared.samet@aya.yale.edu>
3507 | | Date:   Wed Oct 26 15:00:11 2016 -0400
3508 | |
3509 | |     Merge pull request #32 from Neitsch/final-lrm-edits
3510 | |
3511 | |     Flesh out switch expressions, add precedence
3512 | |
3513 | * commit 4683f14715a99d5cdb53089a3bd247035847f3c5
3514 |/ Author: oracleofnj <jared.samet@aya.yale.edu>
3515 | Date:   Wed Oct 26 14:48:41 2016 -0400
3516 |
3517 |     Flesh out switch expressions, add precedence
3518 |
3519 *   commit 4b7984ac504b295e16f8bc77db8024742c6543ab
3520 |\ Merge: 0c42b9c 3d587c5
3521 | | Author: Jared Samet <jared.samet@aya.yale.edu>
3522 | | Date:   Wed Oct 26 11:15:03 2016 -0400
3523 | |
3524 | |     Merge pull request #31 from Neitsch/more-lrm-edits
3525 | |
3526 | |     Incorporate requested edits and a few more clarifications
3527 | |
3528 | * commit 3d587c5997c06337022a69c4a064ee1f33ca6bd2
3529 |/ Author: oracleofnj <jared.samet@aya.yale.edu>
3530 | Date:   Wed Oct 26 11:10:15 2016 -0400
3531 |
3532 |     Incorporate requested edits and a few more clarifications
3533 |
3534 *   commit 0c42b9cf523a95903a4464420864a25a638c8bfb
3535 |\ Merge: 63fb02b cd81040
3536 | | Author: Jared Samet <jared.samet@aya.yale.edu>
3537 | | Date:   Wed Oct 26 09:22:08 2016 -0400
3538 | |
3539 | |     Merge pull request #30 from ishaankolluri/LRM_update
3540 | |
3541 | |     Added changes to first half of LRM
3542 | |
3543 | * commit cd81040e2c5ba5da130c04bc67eaef13fa6522db
3544 |/ Author: ishaankolluri <ishaankolluri@gmail.com>
3545 | Date:   Wed Oct 26 03:30:20 2016 -0400
3546 |
3547 |     Added changes to first half of LRM
3548 |
3549 *   commit 63fb02be16a805fd845fb89632ca258ddbeedeb6
3550 |\ Merge: 5932551 0941e96

```

```

3551 | | Author: Jared Samet <jared.samet@aya.yale.edu>
3552 | | Date:   Wed Oct 26 02:13:17 2016 -0400
3553 | |
3554 | |     Merge pull request #29 from Neitsch/lrm-edits
3555 | |
3556 | |     Lrm edits
3557 | |
3558 | * commit 0941e967272126c6417f458450c611109792e6b8
3559 | | Author: oracleofnj <jared.samet@aya.yale.edu>
3560 | | Date:   Wed Oct 26 02:04:47 2016 -0400
3561 | |
3562 | |     Rebuild PDF
3563 | |
3564 | * commit cb0406921e69f750eca4de178dc2eb8d0733f90f
3565 | | Author: oracleofnj <jared.samet@aya.yale.edu>
3566 | | Date:   Wed Oct 26 02:04:01 2016 -0400
3567 | |
3568 | |     Add built in functions
3569 | |
3570 | * commit 4abf638392a694426f3ca02fb7fcd8ce5b5d3098
3571 | | Author: oracleofnj <jared.samet@aya.yale.edu>
3572 | | Date:   Wed Oct 26 01:56:38 2016 -0400
3573 | |
3574 | |     Add built in functions
3575 | |
3576 | * commit 7661925a96608a3ca00325d343f3feb4c6a0803a
3577 | / Author: oracleofnj <jared.samet@aya.yale.edu>
3578 | Date:   Wed Oct 26 00:04:22 2016 -0400
3579 |
3580 |     Initial comments
3581 |
3582 | * commit 59325515eae265feca08ec854902c0e14fcd0565
3583 | \ Merge: b978f00 cc66297
3584 | | Author: Nigel Schuster <Neitsch@users.noreply.github.com>
3585 | | Date:   Tue Oct 25 21:30:40 2016 -0400
3586 | |
3587 | |     Merge pull request #28 from Neitsch/func-doc-fix
3588 | |
3589 | |     Fixed mistakes in functions part of the doc
3590 | |
3591 | * commit cc66297fd6785a81de7d638a838cc2e4e6645c54
3592 | / Author: Nigel Schuster <nigel.schusters@googlemail.com>
3593 | Date:   Tue Oct 25 20:14:27 2016 -0400
3594 |
3595 |     Fixed mistakes in functions part of the doc
3596 |
3597 | * commit b978f0021545f18d4481658176c1118d7186e624
3598 | \ Merge: 2e1ea60 125a5bb
3599 | | Author: Ishaan Kolluri <ishaankolluri@users.noreply.github.com>
3600 | | Date:   Tue Oct 25 13:04:05 2016 -0400
3601 | |
3602 | |     Merge pull request #27 from ishaankolluri/master
3603 | |
3604 | |     Removed AUX file
3605 | |
3606 | * commit 125a5bb4578448db78b45022651c818e6e07c842

```



```

3607 |/ Author: Ishaan Kolluri <ishaankolluri@gmail.com>
3608 | Date: Tue Oct 25 12:49:38 2016 -0400
3609 |
3610 | Removed AUX file
3611 |
3612 * commit 2e1ea607d20d2b7c0ce2b283c77720b067e197a9
3613 |\ Merge: eb24036 84b03ee
3614 | | Author: Jared Samet <jared.samet@aya.yale.edu>
3615 | | Date: Tue Oct 25 11:30:35 2016 -0400
3616 | |
3617 | | Merge pull request #26 from Neitsch/better-regexp
3618 | |
3619 | | Better regexp
3620 | |
3621 * commit 84b03ee4733dale7a89376cfce43bb76f0884999
3622 | | Author: oracleofnj <jared.samet@aya.yale.edu>
3623 | | Date: Tue Oct 25 01:22:31 2016 -0400
3624 | |
3625 | | Fix let order
3626 | |
3627 * commit 91b40c52f9891693fe9cbb7d00418c9877fbae37
3628 |/ Author: oracleofnj <jared.samet@aya.yale.edu>
3629 | Date: Tue Oct 25 01:14:43 2016 -0400
3630 |
3631 | Improve regexp
3632 |
3633 * commit eb240362462693dca5918e90054fc421c65113fe
3634 |\ Merge: ec7cc9c 991c918
3635 | | Author: Jared Samet <jared.samet@aya.yale.edu>
3636 | | Date: Mon Oct 24 23:55:38 2016 -0400
3637 | |
3638 | | Merge pull request #23 from Neitsch/file-io
3639 | |
3640 | | File io
3641 | |
3642 * commit 991c918de9ba07ab68722fe8042a1edc96b555a5
3643 | | Author: oracleofnj <jared.samet@aya.yale.edu>
3644 | | Date: Mon Oct 24 23:20:12 2016 -0400
3645 | |
3646 | | Replace fopen, fclose etc. with open, close etc.
3647 | |
3648 * commit 338faa006d8d4c7858477028eb8074d69c24aee1
3649 | | Author: oracleofnj <jared.samet@aya.yale.edu>
3650 | | Date: Mon Oct 24 23:14:30 2016 -0400
3651 | |
3652 | | Fix file inclusion and rebuild PDF
3653 | |
3654 * commit b24edd3a58eed2763abdlfe1d4023138af05bbf7
3655 |\ Merge: 44alcc5 2f09a64
3656 | | Author: oracleofnj <jared.samet@aya.yale.edu>
3657 | | Date: Mon Oct 24 23:11:50 2016 -0400
3658 | |
3659 | | Merge in expressions section
3660 | |
3661 * commit 2f09a643221dec47af99592d0a53e15bcc7d67d7
3662 | | Author: Kevin <kevinyell113@gmail.com>

```

```

3663 | | | Date:   Mon Oct 24 15:52:10 2016 -0400
3664 | | |
3665 | | |     Added the Expression Section 4 to LRM
3666 | | |
3667 | | *   commit 8cd39aca8a29d19fe986ad4e7a3540fa4347221f
3668 | | | \ Merge: e5d2478 3609366
3669 | | | | Author: Kevin <kevinye1113@gmail.com>
3670 | | | | Date:   Mon Oct 24 11:05:33 2016 -0400
3671 | | | |
3672 | | | |     Added string literals to scanner
3673 | | | |
3674 | | * |   commit e5d247809c610f428568dce665213019bc368047
3675 | | | | Author: Kevin <kevinye1113@gmail.com>
3676 | | | | Date:   Mon Oct 24 11:00:39 2016 -0400
3677 | | | |
3678 | | | |     Added string literals to scanner
3679 | | | |
3680 | | * | |   commit 44a1cc53aee333cb2d9832f66c0750b8a7add7d8
3681 | | \ \ \ Merge: 1ea3c28 3609366
3682 | | | | / Author: oracleofnj <jared.samet@aya.yale.edu>
3683 | | | | / Date:   Mon Oct 24 23:06:07 2016 -0400
3684 | | | |
3685 | | | |     Merge scanner changes and add regexp to properly escape strings
3686 | | | |
3687 | | * |   commit 36093668180db647b45e208d1dfe0158e7281281
3688 | | | | Author: kevinyel <kevinyel@users.noreply.github.com>
3689 | | | | Date:   Thu Oct 20 21:14:00 2016 -0400
3690 | | | |
3691 | | | |     Update scanner.mll
3692 | | | |
3693 | | * | |   commit 0d57652f53d8a6f4dbbb84d58a6807e5ef59a6fe
3694 | | | | Author: Kevin <kevinye1113@gmail.com>
3695 | | | | Date:   Thu Oct 20 21:10:27 2016 -0400
3696 | | | |
3697 | | | |     Fixed bug in scanner
3698 | | | |
3699 | | * |   commit 1848813885a8c826151e62a27b4ef97b99004a96
3700 | | | | Author: Kevin <kevinye1113@gmail.com>
3701 | | | | Date:   Thu Oct 20 20:21:49 2016 -0400
3702 | | | |
3703 | | | |     Made scanner
3704 | | | |
3705 | | * | |   commit 1ea3c283f511068ed0886e67e5e65e974b646015
3706 | | \ \ \ Merge: a692466 ec7cc9c
3707 | | / / / Author: oracleofnj <jared.samet@aya.yale.edu>
3708 | / | | | Date:   Mon Oct 24 15:26:16 2016 -0400
3709 | | | |
3710 | | | |     Merge branch 'master' into file-io
3711 | | | |
3712 | * | | |   commit ec7cc9c42b88c8cab9bbfe748fb03109eb18bd9b
3713 | | _ | / Author: oracleofnj <jared.samet@aya.yale.edu>
3714 | / | | | Date:   Mon Oct 24 15:06:06 2016 -0400
3715 | | | |
3716 | | | |     Replace repetitive code with more idiomatic OCaml
3717 | | | |
3718 | | * | |   commit a69246646b48528da4e0f0ddf422cbc35522aeaf

```

```

3719 | | | Author: oracleofnj <jared.samet@aya.yale.edu>
3720 | | | Date:   Mon Oct 24 01:09:21 2016 -0400
3721 | | |
3722 | | |     Fix tests until strings ready
3723 | | |
3724 | * | commit 8553a50322c8741d29ed4f4b963a69f232f54750
3725 | | | Author: oracleofnj <jared.samet@aya.yale.edu>
3726 | | | Date:   Mon Oct 24 01:08:29 2016 -0400
3727 | | |
3728 | | |     Fix tests until string ready
3729 | | |
3730 | * | commit 0ed4ad70e5f0db7388b1228df374a1f214f450e6
3731 | | \ Merge: 71e0b1c 92ac506
3732 | | / Author: oracleofnj <jared.samet@aya.yale.edu>
3733 | / | Date:   Mon Oct 24 00:55:08 2016 -0400
3734 | | |
3735 | | |     Add File IO, Entry point and Example to LRM
3736 | | |
3737 | * | commit 92ac50626674ad7d1ccd534f24d39892ca8cb3fa
3738 | | | Author: ishaankolluri <ishaankolluri@gmail.com>
3739 | | | Date:   Sun Oct 23 22:24:06 2016 -0400
3740 | | |
3741 | | |     Make small change to data type section
3742 | | |
3743 | * | commit 71e0b1cb20967ecf7698c343d3774588f361e71c
3744 | | | Author: oracleofnj <jared.samet@aya.yale.edu>
3745 | | | Date:   Sun Oct 23 22:58:21 2016 -0400
3746 | | |
3747 | | |     Fix section reference
3748 | | |
3749 | * | commit 6abb290b9aeaf16eaac6bc431862cb780337b4c6
3750 | / / Author: oracleofnj <jared.samet@aya.yale.edu>
3751 | | | Date:   Sun Oct 23 22:34:42 2016 -0400
3752 | | |
3753 | | |     Initial commit for File I/O section
3754 | | |
3755 | * | commit 67b4b65ab85c86f5369138e782c352143aa1cfec
3756 | | | Author: oracleofnj <jared.samet@aya.yale.edu>
3757 | | | Date:   Sun Oct 23 19:12:38 2016 -0400
3758 | | |
3759 | | |     Reduce eye pain
3760 | | |
3761 | * | commit 2824ee9412ceb67f802e77c382b50dfbf98b57e3
3762 | \ \ Merge: 13d0896 f8ae543
3763 | | | Author: Nigel Schuster <Neitsch@users.noreply.github.com>
3764 | | | Date:   Sun Oct 23 19:03:24 2016 -0400
3765 | | |
3766 | | |     Merge pull request #20 from Neitsch/samples
3767 | | |
3768 | | |     Samples from LRM
3769 | | |
3770 | * | commit f8ae5439b7d1ffc27640e2e66cb657184f2cd1b8
3771 | | \ Merge: e0c702d 13d0896
3772 | | / Author: Nigel Schuster <Neitsch@users.noreply.github.com>
3773 | / | Date:   Sun Oct 23 18:23:11 2016 -0400
3774 | | |

```

```

3775 | | | Merge branch 'master' into samples
3776 | | |
3777 * | | commit 13d089696d899b6c691b4c3af603e334fd462f04
3778 |\ \ \ Merge: 9805753 3a2cd60
3779 | | | Author: Jared Samet <jared.samet@aya.yale.edu>
3780 | | | Date: Sun Oct 23 18:20:03 2016 -0400
3781 | | |
3782 | | | Merge pull request #19 from Neitsch/sequence-operator
3783 | | |
3784 | | | Sequence operator
3785 | | |
3786 | * | | commit 3a2cd606ec6fd9bba26107ec9208b34f257d32d4
3787 | |\ \ \ Merge: 57319c4 9805753
3788 | / / / Author: Jared Samet <jared.samet@aya.yale.edu>
3789 | / | | Date: Sun Oct 23 18:16:35 2016 -0400
3790 | | |
3791 | | | Merge branch 'master' into sequence-operator
3792 | | |
3793 | * | | commit 57319c48db49a519a0f2a75f3abed55ece413df9
3794 | | | Author: oracleofnj <jared.samet@aya.yale.edu>
3795 | | | Date: Sun Oct 23 17:11:13 2016 -0400
3796 | | |
3797 | | | Remove intermediate files
3798 | | |
3799 | * | | commit 53824eaef49ae831e43ab5685b9elfcea5fcd678
3800 | | | Author: oracleofnj <jared.samet@aya.yale.edu>
3801 | | | Date: Sun Oct 23 17:10:39 2016 -0400
3802 | | |
3803 | | | Flip precedence of -> and ?: (?: is now lowest)
3804 | | |
3805 | * | | commit 7dedf93f4ed468537317e5105eeb5a2b8f791aad
3806 | | | Author: oracleofnj <jared.samet@aya.yale.edu>
3807 | | | Date: Sun Oct 23 17:05:23 2016 -0400
3808 | | |
3809 | | | Add sequence operator to scanner/parser/AST
3810 | | |
3811 | | * | commit e0c702d18ceb98ce5ec03ffe49ed53534540fb0d
3812 | | | Author: Neitsch <ns3158@columbia.edu>
3813 | | | Date: Sun Oct 23 18:17:58 2016 -0400
3814 | | |
3815 | | | Fixed .gitignore
3816 | | |
3817 | | * | commit e42fe941b3c93b0cda424f01925754e2f12cbe43
3818 | | | Author: Neitsch <ns3158@columbia.edu>
3819 | | | Date: Sun Oct 23 18:05:48 2016 -0400
3820 | | |
3821 | | | Added code in LRM to test code samples
3822 | | |
3823 | | * | commit 9d2cd174bb0cddd3894d3e77ade8626a2700429c
3824 | | \ \ Merge: 167ddd2 9805753
3825 | | _ / / Author: Neitsch <ns3158@columbia.edu>
3826 | / | | Date: Sun Oct 23 17:24:15 2016 -0400
3827 | | |
3828 | | | Merge branch 'master' into samples
3829 | | |
3830 * | | | commit 980575329d6ce70040a7911de284463f8c9ebf2f

```

```

3831 | \ \ \ \ Merge: 3f015ee e0c7aed
3832 | | / / / Author: Nigel Schuster <Neitsch@users.noreply.github.com>
3833 | / | | | Date: Sun Oct 23 17:01:31 2016 -0400
3834 | | | |
3835 | | | | Merge pull request #17 from Neitsch/make-correction
3836 | | | |
3837 | | | | Added simple TCs, Moved Makefile to oasis config
3838 | | | |
3839 | * | | | commit e0c7aed793ba6070ec94baa3bc52f009c5615c20
3840 | | | | Author: Neitsch <ns3158@columbia.edu>
3841 | | | | Date: Sun Oct 23 16:59:33 2016 -0400
3842 | | | |
3843 | | | | Fixed test
3844 | | | |
3845 | * | | | commit ec3d682eeeecb8b1c20f35a746b65e503b7cdeb4
3846 | | \ \ \ Merge: ea05658 3f015ee
3847 | | / / / Author: Nigel Schuster <Neitsch@users.noreply.github.com>
3848 | / | | | Date: Sun Oct 23 16:41:00 2016 -0400
3849 | | | |
3850 | | | | Merge branch 'master' into make-correction
3851 | | | |
3852 | * | | | commit 3f015eedeeda8af3aac5cb728d9e6258d6b7acda
3853 | \ \ \ \ Merge: d4961eb edf3dea
3854 | | | | | Author: Jared Samet <jared.samet@aya.yale.edu>
3855 | | | | | Date: Sun Oct 23 15:52:01 2016 -0400
3856 | | | | |
3857 | | | | | Merge pull request #18 from Neitsch/grammar-bug-fixes
3858 | | | | |
3859 | | | | | Modify grammar to allow [m,n] foo, bar, baz;
3860 | | | | |
3861 | * | | | | commit edf3dead5be0840a9a80f407dcef8b63c898d14a
3862 | | \ \ \ \ Merge: b45718d d4961eb
3863 | | / / / / Author: Jared Samet <jared.samet@aya.yale.edu>
3864 | / | | | | Date: Sun Oct 23 15:44:20 2016 -0400
3865 | | | | |
3866 | | | | | Merge branch 'master' into grammar-bug-fixes
3867 | | | | |
3868 | * | | | | commit b45718decc1b9e3816105722eebb39afa45dbc5a
3869 | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
3870 | | | | | Date: Sun Oct 23 02:27:36 2016 -0400
3871 | | | | |
3872 | | | | | Modify grammar to allow [m,n] foo, bar, baz;
3873 | | | | |
3874 | | * | | | | commit ea056587ed93451542d5a5e67b9b993cba4a170f
3875 | | | | | Author: Neitsch <ns3158@columbia.edu>
3876 | | | | | Date: Sun Oct 23 16:40:24 2016 -0400
3877 | | | | |
3878 | | | | | Moved sequence file
3879 | | | | |
3880 | | * | | | | commit 0ca56a0230a52c254d21724848c52ddc65350997
3881 | | | \ \ \ Merge: 9d1094e 7e558c1
3882 | | | | | Author: Neitsch <ns3158@columbia.edu>
3883 | | | | | Date: Sun Oct 23 16:10:14 2016 -0400
3884 | | | | |
3885 | | | | | Merge
3886 | | | | |

```

```

3887 | | | * | | | commit 7e558c19a9d85c1dea38f9745b2915dcc14418b7
3888 | | | | \ \ \ Merge: 4652c67 d4961eb
3889 | | _|_ / / / Author: Nigel Schuster <Neitsch@users.noreply.github.com>
3890 | / | | | | Date: Sun Oct 23 15:44:20 2016 -0400
3891 | | | | |
3892 | | | | | Merge branch 'master' into make-correction
3893 | | | | |
3894 * | | | | | commit d4961eb9497ffe2abacb91b8ab5d09ebf2b714a9
3895 | \ \ \ \ \ Merge: 143fcba 0e0bda5
3896 | / / / / / Author: Nigel Schuster <Neitsch@users.noreply.github.com>
3897 | / | | | | Date: Sun Oct 23 15:43:16 2016 -0400
3898 | | | | |
3899 | | | | | Merge pull request #15 from Neitsch/functions-doc
3900 | | | | |
3901 | | | | | Functions doc
3902 | | | | |
3903 | * | | | | commit 0e0bda54deefa312a3da9323d2ba2ff58dfdb79
3904 | | \ \ \ \ \ Merge: cfe827d 143fcba
3905 | | / / / / / Author: Nigel Schuster <Neitsch@users.noreply.github.com>
3906 | / | | | | Date: Sun Oct 23 15:05:42 2016 -0400
3907 | | | | |
3908 | | | | | Merge branch 'master' into functions-doc
3909 | | | | |
3910 | * | | | | commit cfe827d4173debla3fff5719704eae3ff131a615
3911 | | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
3912 | | | | | Date: Fri Oct 21 20:50:51 2016 -0400
3913 | | | | |
3914 | | | | | Completed initial functions section doc
3915 | | | | |
3916 | * | | | | commit 1b610ac614b03b24e829084a6d245b0d99fad01f
3917 | | \ \ \ \ \ Merge: acb9b93 7542b5d
3918 | | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
3919 | | | | | Date: Thu Oct 20 13:50:22 2016 -0400
3920 | | | | |
3921 | | | | | Merge
3922 | | | | |
3923 | | * | | | | commit 7542b5d1866554d70bd312ae649cd1d86f459f12
3924 | | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
3925 | | | | | Date: Thu Oct 20 11:16:35 2016 -0400
3926 | | | | |
3927 | | | | | Added dimension section
3928 | | | | |
3929 | | * | | | | commit 995cf83b2b9a53b6312a0f3f468321c3d9d763c0
3930 | | | | _|_ / Author: Nigel Schuster <nigel.schusters@googlemail.com>
3931 | | | / | | | Date: Wed Oct 19 12:28:09 2016 -0400
3932 | | | | |
3933 | | | | | Started working on Functions
3934 | | | | |
3935 | * | | | | commit acb9b935dc9006e0ae878622402e28fad06907fa
3936 | | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
3937 | | | | | Date: Thu Oct 20 13:40:00 2016 -0400
3938 | | | | |
3939 | | | | | Changed subsection header
3940 | | | | |
3941 | * | | | | commit b95d039ea41668404b9005f1a7eb19e686abb071
3942 | | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>

```

```

3943 | | | | | Date: Thu Oct 20 11:16:35 2016 -0400
3944 | | | | |
3945 | | | | | Added dimension section
3946 | | | | |
3947 | * | | | | commit 71b93bbd5cc99d7ebc3a8c5446b06d5b13ab96ac
3948 | | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
3949 | | | | | Date: Wed Oct 19 12:28:09 2016 -0400
3950 | | | | |
3951 | | | | | Started working on Functions
3952 | | | | |
3953 | | | * | | commit 9d1094e62f36b4170d996fffb07dcb5ee769050f2
3954 | | | | | Author: Neitsch <ns3158@columbia.edu>
3955 | | | | | Date: Sun Oct 23 15:00:35 2016 -0400
3956 | | | | |
3957 | | | | | Added simple TCs, Moved Makefile to oasis config
3958 | | | | |
3959 | | | * | | commit 0a28413f410b7cf1f464b83ed99396ab988481d6
3960 | | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
3961 | | | | | Date: Fri Oct 21 20:50:51 2016 -0400
3962 | | | | |
3963 | | | | | Completed initial functions section doc
3964 | | | | |
3965 | | | * | | commit 0797f328f6f6f6e696a72b1222d4cab38f8e8507
3966 | | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
3967 | | | | | Date: Thu Oct 20 13:40:00 2016 -0400
3968 | | | | |
3969 | | | | | Changed subsection header
3970 | | | | |
3971 | | | * | | commit 9df31f7322f58cbc1b14ea9740b102750e0321ae
3972 | | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
3973 | | | | | Date: Thu Oct 20 11:16:35 2016 -0400
3974 | | | | |
3975 | | | | | Added dimension section
3976 | | | | |
3977 | | | * | | commit 8939903db79ed9de1a8e4b97fc02e79b4be54b61
3978 | | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
3979 | | | | | Date: Wed Oct 19 12:28:09 2016 -0400
3980 | | | | |
3981 | | | | | Started working on Functions
3982 | | | | |
3983 | | | * | | commit cae3b37031dc87451ff03790eb6321dce42c516b
3984 | | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
3985 | | | | | Date: Thu Oct 20 11:16:35 2016 -0400
3986 | | | | |
3987 | | | | | Added dimension section
3988 | | | | |
3989 | | | * | | commit 049c95d512ce6e8e48d4b2c838fd06df40b619
3990 | | _/ / / | Author: Nigel Schuster <nigel.schusters@googlemail.com>
3991 | / | | | | Date: Wed Oct 19 12:28:09 2016 -0400
3992 | | | | |
3993 | | | | | Started working on Functions
3994 | | | | |
3995 | | | * | | commit 167ddd2addae79549eb6d887b98bee35cf2cc4cf
3996 | | | | | Author: Neitsch <ns3158@columbia.edu>
3997 | | | | | Date: Sun Oct 23 17:18:35 2016 -0400
3998 | | | | |

```

```

3999 | | | | | Removed test output
4000 | | | | |
4001 | | | | * commit 84d20b5448f24b98f308c1fb319f213de230d4f1
4002 | | | | / Author: Neitsch <ns3158@columbia.edu>
4003 | | | | Date: Sun Oct 23 16:01:00 2016 -0400
4004 | | | |
4005 | | | | Comparing sample code with correctly parsed code in samples_comp
4006 | | | |
4007 | | | | * commit 4652c6727f848dd573a44509fb498729b28b3080
4008 | | | | / Author: Neitsch <ns3158@columbia.edu>
4009 | | | | / Date: Sun Oct 23 15:00:35 2016 -0400
4010 | | | |
4011 | | | | Added simple TCs, Moved Makefile to oasis config
4012 | | | |
4013 | * | | | commit 143fcba457b49d2c4ab1495d17b830534554508e
4014 | \ \ \ Merge: 660de8c a726236
4015 | | | | Author: Jared Samet <jared.samet@aya.yale.edu>
4016 | | | | Date: Sat Oct 22 23:23:10 2016 -0400
4017 | | | |
4018 | | | | Merge pull request #16 from Neitsch/more-AST
4019 | | | |
4020 | | | | Hook up scanner and parser
4021 | | | |
4022 | * | | | commit a7262368b391c2031c66cfcf1d9fd6224749be69
4023 | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
4024 | | | | Date: Sat Oct 22 20:51:27 2016 -0400
4025 | | | |
4026 | | | | Add comments and sample program
4027 | | | |
4028 | | | | * Added multiline comments using microC code
4029 | | | | * Added one-line comments but didn't test it yet
4030 | | | | * Added example file sequence.xtnd which is the program from proposal
4031 | | | |
4032 | | | | To try:
4033 | | | |
4034 | | | | $ make
4035 | | | | $ ./jsonify < sequence.xtnd
4036 | | | |
4037 | | | | paste output into https://jqplay.org/
4038 | | | |
4039 | * | | | commit 8db40989c4efe4ecefdb8409d45fe8347adc0c79
4040 | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
4041 | | | | Date: Sat Oct 22 19:44:48 2016 -0400
4042 | | | |
4043 | | | | Fix minor grammar bug
4044 | | | |
4045 | | | | * foo[:4,:5] was not being accepted
4046 | | | | * Added rules to lslice and rslice to allow this
4047 | | | |
4048 | * | | | commit 80754c3f3072609502c902d6d26c2e6c33885a84
4049 | / / / Author: oracleofnj <jared.samet@aya.yale.edu>
4050 | | | | Date: Sat Oct 22 18:19:27 2016 -0400
4051 | | | |
4052 | | | | Hook up scanner and parser
4053 | | | |
4054 | | | | * Added hand coded jsonification

```



```

4055 | | |      * Didn't look like Yojson would work with our types
4056 | | |      * Currently not working on any input besides EOF
4057 | | |      * Added temporary makefile
4058 | | |
4059 * | | commit 660de8c848406e8f5fffe099610b6147bc553cad
4060 | / / Author: Jared Samet <jared.samet@aya.yale.edu>
4061 | | Date:   Sat Oct 22 13:54:32 2016 -0400
4062 | |
4063 | |      Add stuff to the grammar, minor corrections (#14)
4064 | |
4065 | |      * Started AST
4066 | |
4067 | |      * Made scanner
4068 | |
4069 | |      * Fixed bug in scanner
4070 | |
4071 | |      * Update scanner.mll
4072 | |
4073 | |      * Add stuff to the grammar, minor corrections
4074 | |
4075 | |      Operators, in order of precedence (lowest first)
4076 | |      * LOGOR ||
4077 | |      * LOGAND &&
4078 | |      * EQ NOTEQ LT GT LTEQ GTEQ == != < > <= >=
4079 | |      * PLUS MINUS BITOR BITXOR + - | ^
4080 | |      * TIMES DIVIDE MOD LSHIFT RSHIFT BITAND * / % << >> &
4081 | |      * POWER **
4082 | |      * BITNOT LOGNOT NEG ~ ! -
4083 | |      * I took the precedence from Golang
4084 | |
4085 | |      String and char literals
4086 | |      New token ASN which is = (as opposed to EQ ==)
4087 | |
4088 | |      Range literals
4089 | |      * {1, 2, 3; 4, 5, 6} is a 2x3 range
4090 | |      * {foo(1); bar(2)} should work too
4091 | |      * The semantic analysis phase will be responsible
4092 | |      *   for ensuring that the dimensions make sense
4093 | |      * Had to add mandatory parentheses to switch(cond) to
4094 | |      *   avoid shift/reduce conflicts but C has that anyway
4095 | |
4096 | |      * Allow general expr slicing, imports and globals
4097 | |      * New HASH token #
4098 | |      * New IMPORT and GLOBAL tokens ("import", "global")
4099 | |
4100 | |      Expr slicing:
4101 | |      * rls_sel no longer -> epsilon
4102 | |      * replaced by expr -> expr rhs_sel, expr -> ID, expr -> HASH expr
4103 | |
4104 | |      Import:
4105 | |      * import "othermodule";
4106 | |
4107 | |      Global variables:
4108 | |      * global PI := 3.14159;
4109 | |      * global [3,5] foo; is legal but has no effect
4110 | |

```

```

4111 | |      Program structure:
4112 | |      * imports followed by globals followed by functions
4113 | |      * This should theoretically be mix-and-matchable without much more work
4114 | |
4115 | |      Size inference:
4116 | |      * foo := "Hello, world!" right now results in [1,1] foo == "H"
4117 | |      * Semantic not grammar issue
4118 | |
4119 | |      * Fix error in col_list production
4120 | |
4121 | |      * Remove intermediate files
4122 | |
4123 | |      * Remove intermediate files
4124 | |
4125 | |      * Rework AST
4126 | |
4127 * |      commit a15772cba8b1fd16e7d21da07ec9b89d34b3062e
4128 |\ \      Merge: d8794e9 dee63c7
4129 | | /      Author: Ishaan Kolluri <ishaankolluri@users.noreply.github.com>
4130 | / |      Date:   Thu Oct 20 13:38:08 2016 -0400
4131 | |
4132 | |      Merge pull request #10 from ishaankolluri/LRM
4133 | |
4134 | |      Initial LRM Commit
4135 | |
4136 | *      commit dee63c70c8c9e7396e255662c302f3d77a7bf7db
4137 | |\      Merge: 4d763cb dc93dbf
4138 | | |      Author: Ishaan Kolluri <ishaankolluri@users.noreply.github.com>
4139 | | |      Date:   Thu Oct 20 13:26:28 2016 -0400
4140 | | |
4141 | | |      Merge pull request #1 from Neitsch/grammar-doc
4142 | | |
4143 | | |      Grammar import
4144 | | |
4145 | | *      commit dc93dbfa68ff7ee65eb38ce8a26baf081da0fe30
4146 | | /      Author: Nigel Schuster <nigel.schusters@googlemail.com>
4147 | | |      Date:   Thu Oct 20 13:18:29 2016 -0400
4148 | | |
4149 | | |      Grammar import
4150 | | |
4151 | *      commit 4d763cbd2505f38028dc9ba820c02aea2b66a380
4152 | |      Author: Ishaan Kolluri <ishaankolluri@gmail.com>
4153 | |      Date:   Thu Oct 20 12:21:21 2016 -0400
4154 | |
4155 | |      Made refactor and edits to intro section of LRM
4156 | |
4157 | *      commit e7443cc478f120ea4aff5db91f7dbfa55e94c4b4
4158 | |\      Merge: 02a5c17 40c2a5a
4159 | | |      Author: Ishaan Kolluri <ishaankolluri@gmail.com>
4160 | | |      Date:   Thu Oct 20 11:46:54 2016 -0400
4161 | | |
4162 | | |      Merging
4163 | | |
4164 | | *      commit 40c2a5aa75310086dbcfab58eaale3ce86ccbe72
4165 | | /      Author: ishaankolluri <ishaankolluri@gmail.com>
4166 | / |      Date:   Wed Oct 19 03:43:06 2016 -0400

```

```

4167 | |
4168 | | Initial LRM Commit part 1
4169 | |
4170 | * commit 02a5c175a1032b34fbf0f1726596270d934aec11
4171 | / Author: Ishaan Kolluri <ishaankolluri@gmail.com>
4172 | Date: Tue Oct 18 18:38:21 2016 -0400
4173 |
4174 | Added LRM initial info
4175 |
4176 * commit d8794e96f4544ed0af9678ecb55e83e1e4b46c25
4177 | \ Merge: 5111202 70aalb9
4178 | | Author: Ishaan Kolluri <ishaankolluri@users.noreply.github.com>
4179 | | Date: Mon Oct 17 19:47:42 2016 -0400
4180 | |
4181 | | Merge pull request #9 from Neitsch/documentation
4182 | |
4183 | | Added PDF Latex template
4184 | |
4185 | * commit 70aalb9846e135d1e8e3f2159148d1bc4e9ea0d8
4186 | / Author: Nigel Schuster <nigel.schusters@googlemail.com>
4187 | Date: Sun Oct 16 13:36:23 2016 -0400
4188 |
4189 | Added PDF Latex template
4190 |
4191 * commit 5111202a9f0992d7b5a3f934f4109552064df4e1
4192 | Author: Jared Samet <jared.samet@aya.yale.edu>
4193 | Date: Fri Oct 14 19:59:45 2016 -0400
4194 |
4195 | Added a bunch of stuff to the grammar: (#8)
4196 |
4197 | * Cleaned up the existing token list
4198 | ** Removed a couple duplicates/unused that menhir flagged
4199 | ** Added PLUS MINUS TIMES DIVIDE MOD
4200 | ** Added SWITCH CASE DEFAULT
4201 | ** Added UNDERSCORE (for [_ ,2] in function declaration)
4202 | ** Separated LITERAL into LIT_INT and LIT_FLOAT (dimensions can
4203 | ** only be integers)
4204 | ** Added a couple associativity/precedence rules that
4205 | ** are undoubtedly incomplete
4206 |
4207 | * Made hopefully all the optional / repetition cases consistent
4208 | ** Function declarations [] foo() {} [] bar() {} ....
4209 | ** Function parameters () ([] foo) ([]foo, []bar), ...
4210 | ** Statements stmt1; stmt2; ...
4211 | ** Function arguments (), (foo), (foo, bar), ...
4212 | ** Switch cases (>=1 required) case foo1: bar1; case foo2: bar2; ...
4213 | ** Case conditions (>=1 required) case foo1, foo2, ...
4214 |
4215 | * Added a few expression types
4216 | ** switch [expr] {case foo_1a, foo_1b: bar1; case foo_2: bar2}
4217 | *** intended to work as in Go: if condition is present, evaluates
4218 | *** to first case where foo == expr. if condition is absent, evaluates
4219 | *** to first case where foo is truthy.
4220 | ** function calls foo(x,y,z)
4221 | ** expression in parentheses
4222 | ** literals

```

```

4223 |
4224 |     * Slicing
4225 |     ** Refactored slice_val into lslice_val and rslice_val
4226 |     ** lslice_val can be any expr
4227 |     ** rslice_val can be any expr or [expr]
4228 |     ** both types of selections and slices are now consistent until
4229 |     **     you get down to val
4230 |
4231 |     * Function return dimensions
4232 |     ** Added _ to the allowed return dimension (for [_ ,2])
4233 |
4234 | * commit da967e421b3f6887acd7c0d659f13d60b1898b6a
4235 | Author: Nigel Schuster <Neitsch@users.noreply.github.com>
4236 | Date:   Wed Oct 12 13:24:50 2016 -0400
4237 |
4238 |     CFG Grammar (#6)
4239 |
4240 |     Added simple parser that has most of our rules.
4241 |
4242 | * commit fea4e4b262513b011775899f35d17e29cb7a1642
4243 | Author: Nigel Schuster <Neitsch@users.noreply.github.com>
4244 | Date:   Sat Oct 8 11:42:39 2016 -0400
4245 |
4246 |     There is no need to constantly build all branches. (#2)
4247 |
4248 |     Only building master.
4249 |
4250 | * commit 7a5ccfc7ada9ddefba61d5da995a7949d7fd1e9d
4251 | Author: Jared Samet <jared.samet@aya.yale.edu>
4252 | Date:   Sat Oct 8 11:31:31 2016 -0400
4253 |
4254 |     Added greeting and newlines (#4)
4255 |
4256 | * commit 10b17f7772e4f7521d248b9656a917df781d96a4
4257 | Author: Nigel Schuster <Neitsch@users.noreply.github.com>
4258 | Date:   Sat Oct 8 11:31:08 2016 -0400
4259 |
4260 |     Imported microc (#5)
4261 |
4262 |     * Imported microc
4263 |     * Fixed sources
4264 |
4265 | * commit 726456fcc25a785cc26d90845b8c0c6292fec61f
4266 | Author: Ishaan Kolluri <ishaankolluri@users.noreply.github.com>
4267 | Date:   Tue Sep 20 09:45:07 2016 -0400
4268 |
4269 |     [test] Add sample greeting to repo (#3)
4270 |
4271 |     * Added sample greeting
4272 |
4273 |     * Add double semicolon notation
4274 |
4275 | * commit 9a2183d3439d67d696ec6babcb6c2be233c2f3bd
4276 | Author: Nigel Schuster <nigel.schusters@googlemail.com>
4277 | Date:   Thu Sep 15 18:44:00 2016 -0400
4278 |

```

```

4279 |     Added merlin config
4280 |
4281 * commit 163e1761ac0f8113e017511f456029282e665d8b
4282 | Author: Nigel Schuster <nigel.schusters@googlemail.com>
4283 | Date:   Wed Sep 14 18:51:53 2016 -0400
4284 |
4285 |     Moved whole build to script
4286 |
4287 * commit d401eeaab71e5e927932c403be099e24700cdc8f
4288 | Author: Nigel Schuster <nigel.schusters@googlemail.com>
4289 | Date:   Wed Sep 14 18:43:58 2016 -0400
4290 |
4291 |     Added oasis opam package
4292 |
4293 * commit ba7fd9c3ccaa617d0709b395da84b43cca2c1860
4294 | Author: Nigel Schuster <nigel.schusters@googlemail.com>
4295 | Date:   Wed Sep 14 18:38:58 2016 -0400
4296 |
4297 |     Added ocaml configure (maybe this helps travis)
4298 |
4299 * commit a461eae15fdf84b3a580af6cfa46451ab490b31
4300 | Author: Nigel Schuster <nigel.schusters@googlemail.com>
4301 | Date:   Wed Sep 14 18:26:10 2016 -0400
4302 |
4303 |     Configuring opam environment for travis
4304 |
4305 * commit ba2df2f48c360588df163b22ad674297e3ae8b16
4306 | Author: Nigel Schuster <nigel.schusters@googlemail.com>
4307 | Date:   Wed Sep 14 18:19:26 2016 -0400
4308 |
4309 |     Added ocaml native compiler to apt package list
4310 |
4311 * commit a8e5958abf68181b0a7b4e7995a2cd7044de3bd4
4312 | Author: Nigel Schuster <nigel.schusters@googlemail.com>
4313 | Date:   Wed Sep 14 17:24:36 2016 -0400
4314 |
4315 |     Added some more (possibly necessary opam packages)
4316 |
4317 * commit c54f5e367d0d985d54bf89630e51a59203f91065
4318 | Author: Nigel Schuster <nigel.schusters@googlemail.com>
4319 | Date:   Wed Sep 14 17:18:32 2016 -0400
4320 |
4321 |     Missed opam option
4322 |
4323 * commit b10adf0f2964eff28de9f33618806e7d614573fc
4324 | Author: Nigel Schuster <nigel.schusters@googlemail.com>
4325 | Date:   Wed Sep 14 17:13:57 2016 -0400
4326 |
4327 |     Fixed opam install
4328 |
4329 * commit 124f7f37083351df7beaed11cc77d503blff0d9b
4330 | Author: Nigel Schuster <nigel.schusters@googlemail.com>
4331 | Date:   Wed Sep 14 17:08:09 2016 -0400
4332 |
4333 |     Fixed YAML error
4334 |

```

```

4335 * commit 4909fa82e010b27f9dc0f2489ebf5cad790e4e3
4336 | Author: Nigel Schuster <nigel.schusters@googlemail.com>
4337 | Date:   Wed Sep 14 17:03:54 2016 -0400
4338 |
4339 |     Using avsm source
4340 |
4341 * commit 4b24046db1b50e97bac103170056c77be7273155
4342 | Author: Nigel Schuster <nigel.schusters@googlemail.com>
4343 | Date:   Wed Sep 14 16:58:33 2016 -0400
4344 |
4345 |     Allow sudo
4346 |
4347 * commit e7b50dbb8b2c463fa67f9ce07b6a8413f905b259
4348 | Author: Nigel Schuster <nigel.schusters@googlemail.com>
4349 | Date:   Wed Sep 14 16:56:57 2016 -0400
4350 |
4351 |     Fixed setup order
4352 |
4353 * commit f6d7ac43f5f27fc4c641667bfe8e24ea3b0d7117
4354 | Author: Nigel Schuster <nigel.schusters@googlemail.com>
4355 | Date:   Wed Sep 14 16:50:02 2016 -0400
4356 |
4357 |     Manually installing apt packages
4358 |
4359 * commit f4084ab464f4f3097d11e0582328f351c40d9516
4360 | Author: Nigel Schuster <nigel.schusters@googlemail.com>
4361 | Date:   Wed Sep 14 16:40:55 2016 -0400
4362 |
4363 |     Test commit
4364 |
4365 * commit d7c5e9a4dadd09c6d7ad97fc727f61aff1e88779
4366 | Author: Nigel Schuster <Neitsch@users.noreply.github.com>
4367 | Date:   Wed Sep 14 13:15:43 2016 -0400
4368 |
4369 |     Initial commit

```