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

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.

```
    sudo apt−get install −y ocaml m4 llvm opam oasis ounit
    opam init
    opam install llvm.3.8 ocamlfind
    eval ‘opam config env‘
```

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

```
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.

```
cd Extend/
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:

```
  ./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.

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

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.

```
foo([m,n] arg){
  return m * n; \\ m and n initialized through arg1
}

bar([1,1] arg){
  return 0; \\ 1 by 1 ranges should be primitive data types. If arg is not, a
      runtime error will be thrown
}
```

Primitive Data Types

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

```
bar := 5; \\ Assigns bar to a number
foo := "Hello World"; \\ Assigns foo to a string
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 and 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: «, », &, |, ˆ

Boolean Operators

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

String Concatenation

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

```
    "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(conditional, expr_if_true, expr_if_false)`

The Switch Expression

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

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

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.

```
  [1,2] foo; \\ Declares range of dimensions 1 by 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 ,.

```
  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.

```
foo[0,2] \\ The cell in the first row, third column
foo[0,:] \\ The range of cells in the first row
foo[0,[1]] \\ The range in the column that is 1 column right of the left-hand-
    side cell.
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.

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

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.

```
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:

```
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.

```
extern "stdlib.o" {
  toString(a);
  extend_open(a,b);
  extend_sin(a);
  print(a,b);
}
```

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.

```
    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.

```
    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.

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

File I/O

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

```
extern "stdlib.o" {
  extend_open(a,b);
  extend_read(a,b);
  extend_close(a);
  print(a,b);
}

[1,1] main(args){
  return print(1, extend_read(extend_open("test_file.txt", "r"),5)) -> 0;
}
```

Plotting

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

```
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:

```
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.

```
isNumber(x) {
  return type(x) == "Number";
}

sum_column([m,1] rng) {
  /* Returns the sum of the values in the column, skipping any values that are non-
      numeric */
  [m,1] running_sum;
  running_sum[0,0] = #rng;
  running_sum[1:,0] = running_sum[[-1],] + (isNumber(#rng) ? #rng : 0);
  return running_sum[-1];
}
```

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:

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

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

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.

```
global x := "I’m a global";

foo() {
  y := x; // Scope of x is entire function
  x := "In here I’m a local";
  return y; // Returns "In here I’m a local"
}

bar(x) {
  return x; // Parameters mask globals; returns argument
}

baz() {
  return x; // Returns "I’m a global"
}
```

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 a 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:

```
legal_ranges() {
  r1 := {"Don’t"; "Panic"}; // two rows, one column
  r2 := {"Don’t", "Think", "Twice"}; // one row, three columns
  r3 := {1,2,3;4,5,6;7,8,9}; // three rows, three columns
  r4 := {"Hello";0,1,2,3,4}; // two rows, five columns
  r5 := {{{{{1}}}}}; // one row, one column
  r7 := {−1.5,−2.5,{−2,"nested"},−3.5}; // one row, four columns
  return 0;
}
```

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

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

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. !empty evaluates to empty. It has equal precedence with 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. empty == empty evaluates to 1. Strings are compared by value.
!=	Not equals	x != y is equivalent to !(x == y).
<	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 empty.
>	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 empty, evaluates to 0 or empty respectively. Otherwise, if the second expression is truthy, falsey, or empty, evaluates to 1, 0, or empty respectively.
||	Short-circuit Logical OR	If the first expression is truthy or empty, evaluates to 1 or empty respectively. Otherwise, if the second expression is truthy, falsey, or empty, evaluates to 1, 0, or empty respectively.

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

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:

```
switch_example(foo) {
  return switch (foo) {
    case 2: "foo is 2";
    case 3,4: "foo is 3 or 4";
    default: "none of the above";
  };
}

alternate_format(foo) {
  return switch {
    case foo == 2:
      "foo is 2";
    case foo == 3, foo == 4:
      "foo is 3 or 4";
    default:
      "none of the above";
  };
}
```

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-expr`s is truthy, or `default-expr` if none of the `case-expr`s apply. If `switch-expr` is present, the `switch` expression evaluates to the `match-expr` for the first case where one of the `case-expr`s is equal (with equality defined as for the == operator) to `switch-expr`, or `default-expr` if none of the `case-expr`s 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
`size(expr)`	`Dimensions`	Evaluates to a Range consisting of one row and two columns; the first cell contains the number of rows of `expr` and the second contains the number of columns. If `expr` is a Number, a String, or Empty, both cells will contain 1.
`type(expr)`	`Value Type`	Evaluates to "Number", "String", "Range", or "Empty".
`row()`	`Row Location`	No arguments; returns the row of the cell that is being calculated
`column()`	`Column Location`	No arguments; returns the column of the cell that is being calculated
`+`	`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

```
selection_examples() {
  foo[0,2] /* This evaluates to the cell value in the first row and third column. */
  foo[0,:] /* Evaluates to the range of cells in the first row of foo. */
  foo[:,2] /* Evaluates to the range of cells in the third column of foo. */
  foo[:,[1]] /* The internal brackets denote RELATIVE notation.
  In this case, 1 column right of the column of the left—hand—side cell. */

  foo[3,] /* Equivalent to foo[3,[0]] if foo has more than one column
  or foo[3,0] if foo has one column */

  foo[5:, 7:] /* All cells starting from the 6th row and 8th column to the bottom
      right */

  foo[[1]:[2], 0:[7]]
  /* Selects the rows between the 1st and 2nd row after LHS row, and
     all the columns up to the 7th column to the right of the LHS column */

  /* In this example, each cell of bar would be equal to the cell
   * in foo in the equivalent location plus 1. */
  [5,5] foo;
  [5,5] bar := #foo + 1; // #foo = foo[[0],[0]]

  /* In this example, bar would be a 3x5 range where in each row,
   * the value in bar is equal to the value in foo in the same column.
   * In other words, each row of bar would be a copy of foo. */
  [1,5] foo; // foo has 1 row, 5 columns
  [3,5] bar := #foo; // #foo = foo[0,[0]]

  /* In this example, the values of baz would be
   * 11, 12, 13 in the first row;
   * 21, 22, 23 in the second row;
   * 31, 32, 33 in the third row. */
  foo := {1,2,3}; // 1 row, 3 columns
  bar := {10;20;30}; // 3 rows, 1 column
  [3,3] baz := #foo + #bar; // Equivalent to foo[0,[0]] + bar[[0],0]
}
```

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:

```
foo(arg) {
  return arg;
}
```

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.

```
[2, 5] foo; // Declares foo as a range with 2 rows and 5 columns
[m, n] bar; // Declares bar as a range with m rows and n columns
[3, 3] ham, eggs, spam; // Declares ham, eggs and spam as distinct 3x3 ranges
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).

```
[5, 2] foo, bar, baz; // Declares foo, bar, and baz as distinct 5x2 ranges
foo[0,0] = 42; // Assigns the expression 42 to the first cell of the first row of foo
foo[0,1] = foo[0,0] * 2; // Assigns (foo[0,0] * 2) to the 2nd cell of the 1st row of foo
bar = 3.14159; // Assigns pi to every cell of every row of bar
baz[1:-1,0:1] = 2.71828; // Assigns e to cells (1,0) through (3,1), inclusive, of baz

/* The next line assigns foo[[-1],0] + 2 to every cell in
   both columns of foo, besides the first row */
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.

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

```
number_of_cells([m,n] arg) {
  return m*n; // m and n are initialized with the dimensions of arg
}

die_unless_primitive([1,1] arg) {
  return 0; // If arg is not a primitive value, a runtime error will occur
}

num_cells_if_column_vector([m,1] arg) {
  // If arg has one column, return number of cells; otherwise runtime error
  return m;
}

die_unless_square([m,m] arg) {
  return 0; // Runtime error if number of rows != number of columns
}

num_cells_if_same_size([m,n] arg1, [m,n] arg2) {
  // If arguments are the same size, return # of cells, otherwise runtime error
  return m*n;
}
```

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:

```
foo([m,n] arg1, [m,n] arg2) {
  [m,n] bar := #arg1 + #arg2;
  return bar;
}
```

This function normalizes a column vector to have unit norm:

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

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:

```
maybeCircular(truth_value) {
  x := x;
  return truth_value ? x : 0;
}
```

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:

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

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:

```
value_p extend_foo(value_p arg1, value_p arg2) {
  /* function body here; */
}

value_p extend_bar() {
  /* function body here; */
}
```

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:

```
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.

```
isNumber(x) {
  return type(x) == "Number";
}

sum_column([m,1] rng) {
  /* Returns the sum of the values in the column, skipping any values that are non-numeric */
  [m,1] running_sum;
  running_sum[0,0] = #rng;
  running_sum[1:,0] = running_sum[[-1],] + (isNumber(#rng) ? #rng : 0);
  return running_sum[-1];
}
```

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:

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

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

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.

```
global x := "I’m a global";

foo() {
  y := x; // Scope of x is entire function
  x := "In here I’m a local";
  return y; // Returns "In here I’m a local"
}

bar(x) {
  return x; // Parameters mask globals; returns argument
}

baz() {
  return x; // Returns "I’m a global"
}
```

3.8 Standard Library Reference

3.9 Example Program

```
import "./samples/stdlib.xtnd";

main([1,n] args) {
  /* Get a working copy */
  return 0;
}
```

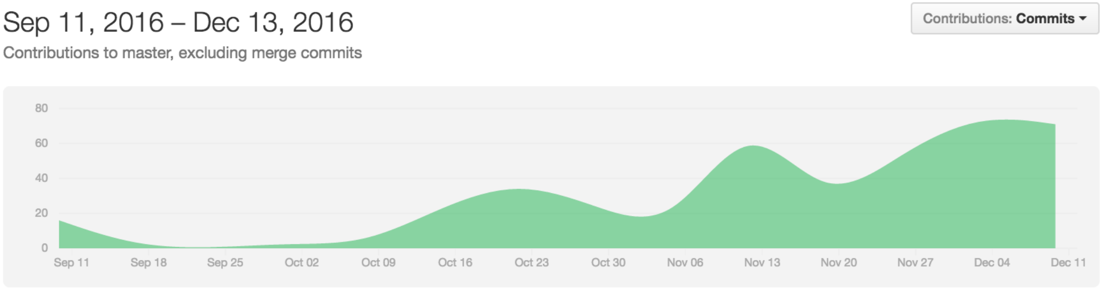
4. Project Plan

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

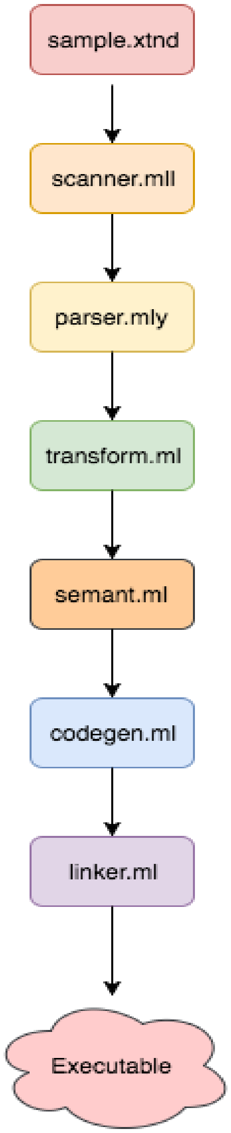
Our development and documentation were all done entirely through version control to maximize independent productivity. New features were introduced to the master branch through pull requests, and the team used this as a platform to peer review code to maximize code quality before such features entered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

```
{
  open Lexing
  open Parser
  open String

  exception SyntaxError of string
  let syntax_error lexbuf = raise (SyntaxError("Invalid character: " ^ Lexing.lexeme
      lexbuf))
}

let digit = ['0'-'9']
let exp = 'e'('+'|'-')?['0'-'9']+
let flt = (digit)+ ('.' (digit)* exp?|exp)
let id = ['a'-'z' 'A'-'Z']['a'-'z' 'A'-'Z' '0'-'9' '_']*


rule token = parse
  ['\n']               { new_line lexbuf; token lexbuf }
| [' ' '\t' '\r']      { token lexbuf }   (* Whitespace *)
| "/*"                 { multiline_comment lexbuf }
| "//"                 { oneline_comment lexbuf }
| '"'                  { read_string (Buffer.create 17) lexbuf }
| '['             { LSQBRACK }
| ']'             { RSQBRACK }
| '('             { LPAREN }
| ')'             { RPAREN }
| '{'             { LBRACE }
| '}'             { RBRACE }
| ":="            { GETS }
| '='             { ASN }
| ':'             { COLON }
| ','             { COMMA }
| "->"            { PRECEDES }
| '?'             { QUESTION }
| "=="            { EQ }
| "!="            { NOTEQ }
| '<'             { LT }
| '>'             { GT }
| "<="            { LTEQ }
| ">="            { GTEQ }
| ';'             { SEMI }
| ’!’             { LOGNOT }
| "&&"            { LOGAND }
| "||"            { LOGOR }
| ’~’             { BITNOT }
| ’&’             { BITAND }
| ’|’             { BITOR }
| ’^’             { BITXOR }
| ’_’             { UNDERSCORE }
| ’+’             { PLUS }
| ’−’             { MINUS }
| ’∗’             { TIMES }
| ’/’             { DIVIDE }
| ’%’             { MOD }
| "∗∗"            { POWER }
| "<<"            { LSHIFT }
| ">>"            { RSHIFT }
| ’#’             { HASH }
| "if"            { IF }
| "empty"         { EMPTY }
| "size"          { SIZE }
| "type"          { TYPE }
| "row"           { ROW }
| "column"        { COLUMN }
| "switch"        { SWITCH }
| "case"          { CASE }
| "default"       { DEFAULT }
| "return"        { RETURN }
| "import"        { IMPORT }
| "global"        { GLOBAL }
| "extern"        { EXTERN }
| "debug"         { DEBUG }
| digit+ as lit   { LIT_INT(int_of_string lit) }
| flt as lit      { LIT_FLOAT(float_of_string lit) }
| id as lit       { ID(lit) }
| eof             { EOF }
| _               { syntax_error lexbuf }

and multiline_comment = parse
  "∗/" { token lexbuf }
| ’\n’ { new_line lexbuf; multiline_comment lexbuf }
| _    { multiline_comment lexbuf }

and oneline_comment = parse
  ’\n’ { new_line lexbuf; token lexbuf }
| _    { oneline_comment lexbuf }

(∗ read_string mostly taken from:
https://realworldocaml.org/v1/en/html/parsing−with−ocamllex−and−menhir.html ∗)
and read_string buf =
  parse
  | ’"’       { LIT_STRING (Buffer.contents buf) }
  | ’\n’      { new_line lexbuf; Buffer.add_char buf ’\n’; read_string buf lexbuf }
  | ’\\’ ’n’  { Buffer.add_char buf ’\n’; read_string buf lexbuf }
  | ’\\’ ’r’  { Buffer.add_char buf ’\r’; read_string buf lexbuf }
  | ’\\’ ’t’  { Buffer.add_char buf ’\t’; read_string buf lexbuf }
  | ’\\’ ([^’\\’ ’n’ ’r’ ’t’] as lxm)
```

```
    { Buffer.add_char buf lxm; read_string buf lexbuf }
  | [^ ’"’ ’\\’]+
    { Buffer.add_string buf (Lexing.lexeme lexbuf);
      read_string buf lexbuf
    }
  | _         { syntax_error lexbuf }
  | eof       { raise (Failure("unterminated string")) }
```

7.2 parser.mly

```
/* Ocamlyacc parser for Extend */

%{
open Ast
%}

%token LSQBRACK RSQBRACK LPAREN RPAREN LBRACE RBRACE HASH
%token COLON COMMA QUESTION IF GETS ASN SEMI PRECEDES UNDERSCORE
%token SWITCH CASE DEFAULT SIZE TYPE ROW COLUMN
%token PLUS MINUS TIMES DIVIDE MOD POWER LSHIFT RSHIFT
%token EQ NOTEQ GT LT GTEQ LTEQ
%token LOGNOT LOGAND LOGOR
%token BITNOT BITXOR BITAND BITOR
%token EMPTY RETURN IMPORT GLOBAL EXTERN
%token DEBUG
%token <int> LIT_INT
%token <float> LIT_FLOAT
%token <string> LIT_STRING
%token <string> ID
%token EOF

%right QUESTION
%left PRECEDES
%left LOGOR
%left LOGAND
%left EQ NOTEQ LT GT LTEQ GTEQ
%left PLUS MINUS BITOR BITXOR
%left TIMES DIVIDE MOD LSHIFT RSHIFT BITAND
%right POWER
%right BITNOT LOGNOT NEG
%left LSQBRACK

%start program
%type <Ast.raw_program> program

%%

program:
    program_piece EOF {  let (imp, glob, fnc, ext) = $1 in (List.rev imp, List.rev glob, List.rev fnc, List.rev ext) }

program_piece:
    /* nothing */ {([],[],[],[])}
  | program_piece import      { let (imp, glob, fnc, ext) = $1 in ($2 :: imp, glob, fnc, ext) }
  | program_piece global      { let (imp, glob, fnc, ext) = $1 in (imp, $2 :: glob, fnc, ext) }
  | program_piece func_decl   { let (imp, glob, fnc, ext) = $1 in (imp, glob, $2 :: fnc, ext) }
  | program_piece extern      { let (imp, glob, fnc, ext) = $1 in (imp, glob, fnc, $2 :: ext) }

import:
    IMPORT LIT_STRING SEMI {$2}

global:
    GLOBAL varinit {$2}

extern:
    EXTERN LIT_STRING LBRACE opt_extern_list RBRACE {(Library($2, $4))}

opt_extern_list:
    /* nothing */ { [] }
  | extern_list { List.rev $1 }

extern_list:
    extern_fn { [$1] }
  | extern_list extern_fn { $2 :: $1 }

extern_fn:
    ID LPAREN func_param_list RPAREN SEMI
    { {
      extern_fn_name = $1;
      extern_fn_params = $3;
      extern_fn_libname = "";
      extern_ret_val = (None, None);
    } }
  | ret_dim ID LPAREN func_param_list RPAREN SEMI
    { {
      extern_fn_name = $2;
      extern_fn_params = $4;
      extern_fn_libname = "";
      extern_ret_val = $1;
    } }

func_decl:
    ID LPAREN func_param_list RPAREN LBRACE opt_stmt_list ret_stmt RBRACE
    { {
      name = $1;
      params = $3;
      body = $6;
      raw_asserts = [];
      ret_val = ((None, None), $7)
    } }
  | ret_dim ID LPAREN func_param_list RPAREN LBRACE opt_stmt_list ret_stmt RBRACE
    { {
      name = $2;
      params = $4;
      body = $7;
      raw_asserts = [];
      ret_val = ($1, $8);
    } }

opt_stmt_list:
    /* nothing */ { [] }
  | stmt_list { List.rev $1 }

stmt_list:
    stmt { [$1] }
  | stmt_list stmt { $2 :: $1 }

stmt:
    varinit { $1 } |  assign { $1 }

ret_stmt:
    RETURN expr SEMI {$2}

varinit:
    var_list SEMI { Varinit((None, None), List.rev $1) }
  | dim var_list SEMI { Varinit($1, List.rev $2) }

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

varassign:
    /* nothing */ { None }
  | GETS expr { Some $2 }

assign:
    ID lhs_sel ASN expr SEMI { Assign($1, $2, Some $4) }

expr:
    expr rhs_sel        { Selection($1, $2) }
  | HASH ID             { Selection(Id($2), (None, None)) }
  | op_expr             { $1 }
  | ternary_expr        { $1 }
  | switch_expr         { $1 }
  | func_expr           { $1 }
  | range_expr          { $1 }
  | DEBUG LPAREN expr RPAREN { Debug($3) }
  | expr PRECEDES expr  { Precedence($1, $3) }
  | LPAREN expr RPAREN  { $2 }
  | ID                  { Id($1) }
  | LIT_INT             { LitInt($1) }
  | LIT_FLOAT           { LitFlt($1) }
  | LIT_STRING          { LitString($1) }
  | EMPTY               { Empty }

op_expr:
    expr PLUS expr      { BinOp($1, Plus, $3) }
  | expr MINUS expr     { BinOp($1, Minus, $3) }
  | expr TIMES expr     { BinOp($1, Times, $3) }
  | expr DIVIDE expr    { BinOp($1, Divide, $3) }
  | expr MOD expr       { BinOp($1, Mod, $3) }
  | expr POWER expr     { BinOp($1, Pow, $3) }
  | expr LSHIFT expr    { BinOp($1, LShift, $3) }
  | expr RSHIFT expr    { BinOp($1, RShift, $3) }
  | expr LOGAND expr    { BinOp($1, LogAnd, $3) }
  | expr LOGOR expr     { BinOp($1, LogOr, $3) }
  | expr BITXOR expr    { BinOp($1, BitXor, $3) }
  | expr BITAND expr    { BinOp($1, BitAnd, $3) }
  | expr BITOR expr     { BinOp($1, BitOr, $3) }
  | expr EQ expr        { BinOp($1, Eq, $3) }
  | expr NOTEQ expr     { UnOp(LogNot,(BinOp($1, Eq, $3))) }
  | expr GT expr        { BinOp($1, Gt, $3) }
  | expr LT expr        { BinOp($1, Lt, $3) }
  | expr GTEQ expr      { BinOp($1, GtEq, $3) }
  | expr LTEQ expr      { BinOp($1, LtEq, $3) }
  | SIZE LPAREN expr RPAREN { UnOp(SizeOf, $3) }
  | TYPE LPAREN expr RPAREN { UnOp(TypeOf, $3) }
  | ROW LPAREN RPAREN       { UnOp(Row, Empty)}
  | COLUMN LPAREN RPAREN    { UnOp(Column, Empty)}
  | MINUS expr %prec NEG    { UnOp(Neg, $2) }
  | LOGNOT expr             { UnOp(LogNot, $2) }
  | BITNOT expr             { UnOp(BitNot, $2) }

ternary_expr:
    IF LPAREN expr COMMA expr COMMA expr RPAREN { Ternary($3, $5, $7) }
  | expr QUESTION expr COLON expr %prec QUESTION { Ternary($1, $3, $5) }

switch_expr:
    SWITCH LPAREN switch_cond RPAREN LBRACE default_case_list RBRACE { Switch($3, fst $6, snd $6) }
  | SWITCH LBRACE default_case_list RBRACE { Switch(None, fst $3, snd $3) }

switch_cond:
    /* nothing */ { None }
  | expr { Some $1 }

default_case_list:
    case_list {(List.rev $1, Empty)}
  | case_list default_expr {(List.rev $1, $2)}

case_list:
    case_stmt { [$1] }
  | case_list case_stmt { $2 :: $1 }

case_stmt:
    CASE case_expr_list COLON expr SEMI { (List.rev $2, $4) }

default_expr:
    DEFAULT COLON expr SEMI { $3 }

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

func_expr:
    ID LPAREN opt_arg_list RPAREN { Call($1, $3) }

range_expr:
    LBRACE row_list RBRACE { allow_range_literal (LitRange(List.rev $2)) }
```

```

row_list:
    col_list {[List.rev $1]}
  | row_list SEMI col_list {List.rev $3 :: $1}

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

opt_arg_list:
    /* nothing */ {[]}
  | arg_list { List.rev $1 }

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

lhs_sel:
    /* nothing */                         { (None, None) }
/* commented out: LSQBRACK lslice RSQBRACK { (Some $2, None) } */
  | LSQBRACK lslice COMMA lslice RSQBRACK { (Some $2, Some $4) }

rhs_sel:
    LSQBRACK rslice RSQBRACK              { (Some $2, None) }
  | LSQBRACK rslice COMMA rslice RSQBRACK { (Some $2, Some $4) }

lslice:
  /* commented out: nothing production { (None, None) } */
    lslice_val                            { (Some $1, None) }
  | lslice_val COLON lslice_val           { (Some $1, Some $3) }
  | lslice_val COLON                      { (Some $1, Some DimensionEnd) }
  | COLON lslice_val                      { (Some DimensionStart, Some $2) }
  | COLON                                 { (Some DimensionStart, Some DimensionEnd) }

rslice:
    /* nothing */                         { (None, None) }
  | rslice_val                            { (Some $1, None) }
  | rslice_val COLON rslice_val           { (Some $1, Some $3) }
  | rslice_val COLON                      { (Some $1, Some DimensionEnd) }
  | COLON rslice_val                      { (Some DimensionStart, Some $2) }
  | COLON                                 { (Some DimensionStart, Some DimensionEnd) }

lslice_val:
    expr { Abs($1) }

rslice_val:
    expr { Abs($1) }
  | LSQBRACK expr RSQBRACK { Rel($2) }

func_param_list:
    /* nothing */ { [] }
  | func_param_int_list { List.rev $1 }

func_param_int_list:
    func_sin_param { [$1] }
  | func_param_int_list COMMA func_sin_param { $3 :: $1 }
```

```

func_sin_param:
    ID { ((None, None), $1) }
  | dim ID { ($1, $2) }

dim:
    LSQBRACK expr RSQBRACK { (Some $2, None) }
  | LSQBRACK expr COMMA expr RSQBRACK { (Some $2, Some $4) }

ret_dim:
    LSQBRACK ret_sin RSQBRACK { ($2, None) }
  | LSQBRACK ret_sin COMMA ret_sin RSQBRACK { ($2,$4) }

ret_sin:
    expr { Some $1 }
  | UNDERSCORE { Some Wild }
```

7.3 ast.ml

```
type op       = Plus | Minus | Times | Divide | Mod | Pow |
                LShift | RShift | BitOr | BitAnd | BitXor |
                Eq | Gt | GtEq | Lt | LtEq | LogAnd | LogOr
type unop     = Neg | LogNot | BitNot | SizeOf | TypeOf | Row | Column | Truthy

type expr     = LitInt of int |
                LitFlt of float |
                LitString of string |
                LitRange of (expr list) list |
                Id of string |
                Empty |
                Wild |
                BinOp of expr * op * expr |
                UnOp of unop * expr |
                Ternary of expr * expr * expr |
                Switch of expr option * case list * expr |
                Call of string * expr list |
                Selection of expr * sel |
                ReducedTernary of string * string * string |
                Precedence of expr * expr |
                Debug of expr
and  index    = Abs of expr |
                Rel of expr |
                DimensionStart |
                DimensionEnd
and  slice    = index option * index option
and  sel      = slice option * slice option
and  case     = expr list * expr

type dim      = expr option * expr option
type var      = dim * string
type assign   = string * sel * expr option
type init     = string * expr option
type stmt     = Assign of assign |
                Varinit of dim * init list

```

```
type raw_func = {
    name: string;
    params: var list;
    body: stmt list;
    raw_asserts: expr list;
    ret_val: dim * expr;
}

type extern_func = {
    extern_fn_name: string;
    extern_fn_params: var list;
    extern_fn_libname: string;
    extern_ret_val: dim;
}

type library  = Library of string * extern_func list
type raw_program = string list * stmt list * raw_func list * library list

(* Desugared types below *)
module StringMap = Map.Make(String)
type formula  = {
  formula_row_start: index;
  formula_row_end: index option;
  formula_col_start: index;
  formula_col_end: index option;
  formula_expr: expr;
}

type dim_expr = DimInt of int
              | DimId of string

type variable = {
  var_rows: dim_expr;
  var_cols: dim_expr;
  var_formulas: formula list;
}

type func_decl = {
  func_params: var list;
  func_body: variable StringMap.t;
  func_asserts: expr list;
  func_ret_val: dim * expr;
}

type program = (variable StringMap.t) * (func_decl StringMap.t) * (extern_func StringMap.t)

type listable = Inits of init list |
                Vars of var list |
                Stmts of stmt list |
                RawFuncs of raw_func list |
                Externs of extern_func list |
                Libraries of library list |
                Exprs of expr list |
                Rows of (expr list) list |
                Strings of string list |
                Cases of case list |
                Formulas of formula list

exception IllegalRangeLiteral of string
exception TransformedAway of string

let quote_string str =
  let escape_characters = Str.regexp "[\n \t \r \\ \"]" in
  let replace_fn s = match Str.matched_string s with
    "\n" -> "\\n"   |
    "\t" -> "\\t"   |
    "\r" -> "\\r"   |
    "\\" -> "\\\\"  |
    "\"" -> "\\\""  |
    _    -> Str.matched_string s in
  "\"" ^ Str.global_substitute escape_characters replace_fn str ^ "\""

let string_of_op o = "\"" ^ (match o with
    Plus -> "+" | Minus -> "-" | Times -> "*" | Divide -> "/" | Mod -> "%" | Pow -> "**" |
    LShift -> "<<" | RShift -> ">>" | BitOr -> "|" | BitAnd -> "&" | BitXor -> "^" |
    Eq -> "==" | Gt -> ">" | GtEq -> ">=" | Lt -> "<" | LtEq -> "<=" |
    LogAnd -> "&& " | LogOr -> "||" ) ^ "\""

let string_of_unop = function
    Neg -> "\"-\"" | LogNot -> "\"!\"" | BitNot -> "\"~\"" | Truthy -> "\"truthy\"" |
    SizeOf -> "\"size\"" | TypeOf -> "\"type\"" | Row -> "\"row\"" | Column -> "\"column\""

let rec string_of_expr = function
    LitInt(l) ->          "{\"LitInt\":" ^ string_of_int l ^ "}"
  | LitFlt(l) ->          "{\"LitFlt\":" ^ string_of_float l ^ "}"
  | LitString(s) ->       "{\"LitString\":" ^ quote_string s ^ "}"
  | LitRange(rowlist) ->  "{\"LitRange\": " ^ string_of_list (Rows rowlist) ^ "}"
  | Id(s) ->              "{\"Id\": " ^ quote_string s ^ "}"
  | Empty ->              "\"Empty\""
  | Wild ->               "\"Wild\""
  | BinOp(e1, o, e2) ->   "{\"BinOp\": {" ^
                            "\"expr1\": " ^ string_of_expr e1 ^ ", " ^
                            "\"operator\": " ^ string_of_op o ^ ", " ^
                            "\"expr2\": " ^ string_of_expr e2 ^ "}}"
  | UnOp(o, e) ->         "{\"UnOp\": {" ^
                            "\"operator\": " ^ string_of_unop o ^ ", " ^
                            "\"expr\": " ^ string_of_expr e ^ "}}"
  | Ternary(c, e1, e2) -> "{\"Ternary\": {" ^
                            "\"condition\": " ^ string_of_expr c ^ ", " ^
                            "\"ifExpr\": " ^ string_of_expr e1 ^ ", " ^
                          "\"elseExpr\": " ^ string_of_expr e2 ^ "}}"
  | ReducedTernary(s1, s2, s3) -> "{\"ReducedTernary\": {" ^
                            "\"truthiness\": " ^ quote_string s1 ^ ", " ^
                            "\"true_values\": " ^ quote_string s2 ^ ", " ^
                            "\"false_values\": " ^ quote_string s3 ^ "}}"
  | Switch(eo, cases, dflt) ->  "{\"Switch\": {" ^
                                "\"condition\": " ^
                                  (match eo with None -> "null" | Some e -> string_of_expr e) ^ ", " ^
                                "\"cases\": " ^ string_of_list (Cases cases) ^ ", " ^
                                "\"defaultExpr\": " ^ string_of_expr dflt ^ "}}"
  | Call(f, arguments) -> "{\"Call\": {" ^
                            "\"function\": " ^ quote_string f ^ ", " ^
                            "\"arguments\": " ^ string_of_list (Exprs arguments) ^ "}}"
  | Selection(e, s) ->    "{\"Selection\": {" ^
                            "\"expr\": " ^ string_of_expr e ^ ", " ^
                            "\"slices\": " ^ string_of_sel s ^ "}}"
  | Precedence(e1, e2) -> "{\"Precedence\": { " ^
                            "\"prior_expr\": " ^ string_of_expr e1 ^ ", " ^
                          "\"dependent_expr\": " ^ string_of_expr e2 ^ "}}"
  | Debug(e) -> string_of_expr e

and string_of_case (el, e) =
    "{\"Cases\": " ^ string_of_list (Exprs el) ^ ", " ^
     "\"expr\": " ^ string_of_expr e ^ "}"

and string_of_sel (s1, s2) =
    "{\"slice1\": " ^ string_of_slice s1 ^ ", \"slice2\": " ^ string_of_slice s2 ^ "}"

and string_of_slice = function
    None -> "null"
  | Some (start_idx, end_idx) -> "{\"start\": " ^ string_of_index start_idx ^ ", \"end\": " ^ string_of_index end_idx ^ "}"

and string_of_index = function
    None -> "null"
  | Some(Abs(e)) -> "{\"Absolute\": " ^ string_of_expr e ^ "}"
  | Some(Rel(e)) -> "{\"Relative\": " ^ string_of_expr e ^ "}"
  | Some(DimensionStart) -> "\"DimensionStart\""
  | Some(DimensionEnd) -> "\"DimensionEnd\""

and string_of_dim (d1,d2) = "{\"d1\": " ^ (match d1 with None -> "null" | Some e -> string_of_expr e) ^ ", " ^
                             "\"d2\": " ^ (match d2 with None -> "null" | Some e -> string_of_expr e) ^ "}"

and string_of_var (d, s) = "{\"Dimensions\": " ^ string_of_dim d ^ ", " ^
                            "\"VarName\": " ^ quote_string s ^ "}"

and string_of_assign (s, selection, eo) =
    "{\"VarName\": " ^ quote_string s ^ ", " ^
     "\"Selection\": " ^ string_of_sel selection ^ ", " ^
    "\"expr\": " ^ (match eo with None -> "null" | Some e -> string_of_expr e) ^ "}"

and string_of_varinit (d, inits) =
  "{\"Dimensions\": " ^ string_of_dim d ^
    ",\"Initializations\": " ^ string_of_list (Inits inits) ^ "}"

and string_of_init (s, eo) =
    "{\"VarName\": " ^ quote_string s ^ ", " ^
     "\"expr\": " ^ (match eo with None -> "null" | Some e -> string_of_expr e) ^ "}"

and string_of_stmt = function
    Assign(a) -> "{\"Assign\": " ^ string_of_assign a ^ "}"
  | Varinit(d, inits) -> "{\"Varinit\": " ^ string_of_varinit (d, inits) ^ "}"

and string_of_range (d, e) = "{\"Dimensions\": " ^ string_of_dim d ^ ", " ^
                              "\"expr\": " ^ string_of_expr e ^ "}"

and string_of_raw_func fd =
    "{\"Name\": " ^ quote_string fd.name ^ "," ^
     "\"Params\": " ^ string_of_list (Vars fd.params) ^ "," ^
     "\"Stmts\": " ^ string_of_list (Stmts fd.body) ^ "," ^
     "\"Assertions\": " ^ string_of_list (Exprs fd.raw_asserts) ^ "," ^
     "\"ReturnVal\": " ^ string_of_range fd.ret_val ^ "}"

and string_of_extern_func fd =
  "{\"Name\": " ^ quote_string fd.extern_fn_name ^ "," ^
  "\"Params\": " ^ string_of_list (Vars fd.extern_fn_params) ^ "," ^
  "\"Library\": " ^ quote_string fd.extern_fn_libname ^ "," ^
  "\"ReturnDim\": " ^ string_of_dim fd.extern_ret_val ^ "}"

and string_of_library (Library(lib_name, lib_fns)) =
  "{\"LibraryName\": " ^ quote_string lib_name ^ "," ^
  "\"ExternalFunctions\": " ^ string_of_list (Externs lib_fns) ^ "}"

and string_of_dimexpr = function
    DimInt(i) -> string_of_int i
  | DimId(s) -> quote_string s

and string_of_formula f =
  "{\"RowStart\": " ^ string_of_index (Some f.formula_row_start) ^ "," ^
  "\"RowEnd\": " ^ string_of_index (f.formula_row_end) ^ "," ^
  "\"ColumnStart\": " ^ string_of_index (Some f.formula_col_start) ^ "," ^
  "\"ColumnEnd\": " ^ string_of_index (f.formula_col_end) ^ "," ^
  "\"Formula\": " ^ string_of_expr f.formula_expr ^ "}"

and string_of_list l =
  let stringrep = (match l with
    Inits (il) -> List.map string_of_init il
  | Vars(vl) -> List.map string_of_var vl
  | Stmts(sl) -> List.map string_of_stmt sl
  | RawFuncs(fl) -> List.map string_of_raw_func fl
  | Externs(efl) -> List.map string_of_extern_func efl
  | Libraries(libl) -> List.map string_of_library libl
  | Exprs(el) -> List.map string_of_expr el
  | Rows(rl) -> List.map (fun (el : expr list) -> string_of_list (Exprs el)) rl
  | Strings(sl) -> List.map quote_string sl
  | Cases(cl) -> List.map string_of_case cl
  | Formulas(fl) -> List.map string_of_formula fl)
  in "[" ^ String.concat ", " stringrep ^ "]"

let string_of_raw_program (imp, glb, fs, exts) =
    "{\"Program\": {" ^
      "\"Imports\": " ^ string_of_list (Strings imp) ^ "," ^
      "\"Globals\": " ^ string_of_list (Stmts glb) ^ "," ^
      "\"ExternalLibraries\": " ^ string_of_list (Libraries exts) ^ "," ^
      "\"Functions\": " ^ string_of_list (RawFuncs fs) ^ "}}"

let string_of_variable v =
  "{\"Rows\": " ^ string_of_dimexpr v.var_rows ^ "," ^
  "\"Columns\": " ^ string_of_dimexpr v.var_cols ^ "," ^
  "\"Formulas\": " ^ string_of_list (Formulas v.var_formulas) ^ "}"

let string_of_map value_desc val_printing_fn m =
  let f_key_val_list k v l = (
    "{\"" ^ value_desc ^ "Name\": " ^ quote_string k ^ ", " ^
    "\"" ^ value_desc ^ "Def\": " ^ val_printing_fn v ^ "}"
  ) :: l in
  "[" ^ String.concat ", " (List.rev (StringMap.fold f_key_val_list m [])) ^ "]"

let string_of_funcdecl f =
  "{\"Params\": " ^ string_of_list (Vars f.func_params) ^ "," ^
  "\"Variables\": " ^ string_of_map "Variable" string_of_variable f.func_body ^ "," ^
  "\"Assertions\": " ^ string_of_list (Exprs f.func_asserts) ^ "," ^
  "\"ReturnVal\": " ^ string_of_range f.func_ret_val ^ "}"

let string_of_program (glb, fs, exts) =
  "{\"Program\": {" ^
    "\"Globals\": " ^ string_of_map "Variable" string_of_variable glb ^ "," ^
    "\"Functions\": " ^ string_of_map "Function" string_of_funcdecl fs ^ "," ^
    "\"ExternalFunctions\": " ^ string_of_map "ExternalFunctions" string_of_extern_func exts ^ "}}"

let allow_range_literal = function
    LitRange(rowlist) ->
      let rec check_range_literal rl =
        List.for_all (fun exprs -> List.for_all check_basic_expr exprs) rl
      and check_basic_expr = function
          LitInt(_) | UnOp(Neg, LitInt(_)) | LitFlt(_) | UnOp(Neg, LitFlt(_)) | LitString(_) | Empty -> true
        | LitRange(rl) -> check_range_literal rl
        | _ -> false in

      if check_range_literal rowlist then LitRange(rowlist)
      else raise(IllegalRangeLiteral(string_of_expr (LitRange(rowlist))))
  | e -> raise(IllegalRangeLiteral(string_of_expr e))
```

7.4 transform.ml

```
open Ast
open Lexing
open Parsing

exception IllegalExpression of string;;
exception DuplicateDefinition of string;;
exception UnknownVariable of string;;
exception UnknownFunction of string;;
exception WrongNumberArgs of string;;
exception LogicError of string;;

module StringSet = Set.Make (String);;
let importSet = StringSet.empty;;

let builtin_signatures = [("cos", 1); ("column", 0); (*("printf", 2);*) ("toString",
    1)]

let idgen =
  (* from http://stackoverflow.com/questions/10459363/side-effects-and-top-level-expressions-in-ocaml*)
  let count = ref (-1) in
  fun prefix -> incr count; "_tmp_" ^ prefix ^ string_of_int !count;;

let expand_file filename =
  let print_error_location filename msg lexbuf =
    let pos = lexbuf.lex_curr_p in
    prerr_endline ("Syntax error in \"" ^ filename ^ "\": " ^ msg) ;
    prerr_endline ("Line " ^ (string_of_int pos.pos_lnum) ^ " at character " ^ (string_of_int (pos.pos_cnum - pos.pos_bol))) in

  let rec expand_imports processed_imports globals fns exts dir = function
      [] -> ([], globals, fns, exts)
    | import :: imports ->
      (* print_endline "---------";
      print_endline ("Working on: " ^ import) ;
      print_endline ("Already processed:"); *)
      (* StringSet.iter (fun a -> print_endline a) processed_imports; *)
      let in_chan = open_in import in
      let lexbuf = (Lexing.from_channel (in_chan)) in
      let (file_imports, file_globals, file_functions, file_externs) =
        try Parser.program Scanner.token lexbuf
        with
          Parsing.Parse_error -> print_error_location import "" lexbuf ; exit(-1)
        | Scanner.SyntaxError(s) -> print_error_location import s lexbuf ; exit(-1)
      in
      let file_imports = List.map (fun file -> dir ^ "/" ^ file) file_imports in
      let new_proc = StringSet.add import processed_imports and _ = close_in in_chan in
      (* print_endline ("Now I'm done with: ") ; *)
      (* StringSet.iter (fun a -> print_endline a) new_proc; *)
      let first_im_hearing_about imp = not (StringSet.mem imp new_proc || List.mem imp imports) in
      let new_imports = StringSet.elements (StringSet.of_list (List.filter first_im_hearing_about file_imports)) in
      (* print_endline ("First I'm hearing about:") ; *)
      (* List.iter print_endline new_imports; *)
      expand_imports new_proc (globals @ file_globals) (fns @ file_functions) (exts @ file_externs) (Filename.dirname import) (imports @ new_imports) in
  expand_imports StringSet.empty [] [] [] (Filename.dirname filename) [filename]

let expand_expressions (imports, globals, functions, externs) =
  let lit_zero = LitInt(0) in let abs_zero = Abs(lit_zero) in
  let lit_one  = LitInt(1) in let abs_one  = Abs(lit_one)  in
  let one_by_one = (Some lit_one, Some lit_one) in
  let zero_comma_zero = (Some (Some abs_zero, Some abs_one),
                         Some (Some abs_zero, Some abs_one)) in
  let entire_dimension = (Some DimensionStart, Some DimensionEnd) in
  let entire_range = (Some entire_dimension, Some entire_dimension) in

  let expand_expr expr_loc = function
    (* Create a new variable for all expressions on the LHS to hold the result;
       return the new expression and whatever new statements are necessary to create the new variable *)
      Empty     -> raise (IllegalExpression("Empty not allowed in " ^ expr_loc))
    | Wild      -> raise (IllegalExpression("wild - this shouldn't be possible"))
    | LitString(s) -> raise (IllegalExpression("String literal " ^ quote_string s ^ " not allowed in " ^ expr_loc))
    | LitRange(rl) -> raise (IllegalExpression("Range literal " ^ string_of_list (Rows rl) ^ " not allowed in " ^ expr_loc))
    | e         -> let new_id = idgen expr_loc in (
        Id(new_id),
        [Varinit (one_by_one, [(new_id, None)]);
         Assign (new_id, zero_comma_zero, Some e)]) in

  let expand_index index_loc = function
    (* Expand one index of a slice if necessary. *)
      Abs(e) -> let (new_e, new_stmts) = expand_expr index_loc e in
      (Abs(new_e), new_stmts)
    | DimensionStart -> (DimensionStart, [])
    | DimensionEnd -> (DimensionEnd, [])
    | Rel(_) -> raise (IllegalExpression("relative - this shouldn't be possible")) in

  let expand_slice slice_loc = function
    (* Expand one or both sides as necessary. *)
      None -> (entire_dimension, [])
    | Some (Some (Abs(e)), None) ->
      let (start_e, start_stmts) = expand_expr (slice_loc ^ "_start") e in
      ((Some (Abs(start_e)), None), start_stmts)
    | Some (Some idx_start, Some idx_end) ->
      let (new_start, new_start_exprs) = expand_index (slice_loc ^ "_start") idx_start in
      let (new_end, new_end_exprs) = expand_index (slice_loc ^ "_end") idx_end in
      ((Some new_start, Some new_end), new_start_exprs @ new_end_exprs)
    | Some (Some _, None) | Some (None, _) -> raise (IllegalExpression("Illegal slice - this shouldn't be possible")) in

  let expand_assign asgn_loc (var_name, (row_slice, col_slice), formula) =
    (* expand_assign: Take an Assign and return a list of more
       atomic statements, with new variables replacing any
       complex expressions in the selection slices and with single
       index values desugared to expr:expr+1. *)
    try
      let (new_row_slice, row_exprs) = expand_slice (asgn_loc ^ "_" ^ var_name ^ "_row") row_slice in
      let (new_col_slice, col_exprs) = expand_slice (asgn_loc ^ "_" ^ var_name ^ "_col") col_slice in
      Assign(var_name, (Some new_row_slice, Some new_col_slice), formula) :: (row_exprs @ col_exprs)
    with IllegalExpression(s) ->
      raise (IllegalExpression("Illegal expression (" ^ s ^ ") in " ^
                               string_of_assign (var_name, (row_slice, col_slice), formula))) in

  let expand_init (r, c) (v, e) =
    Varinit((Some r, Some c), [(v, None)]) ::
    match e with
      None -> []
    | Some e -> [Assign (v, entire_range, Some e)] in

  let expand_dimension dim_loc = function
      None -> expand_expr dim_loc (LitInt(1))
    | Some e -> expand_expr dim_loc e in

  let expand_varinit fname ((row_dim, col_dim), inits) =
    (* expand_varinit: Take a Varinit and return a list of more atomic
       statements. Each dimension will be given a temporary ID, which
       will be declared as [1,1] _tmpXXX; the formula for tmpXXX will be
       set as a separate assignment; the original variable will be
       declared as [_tmpXXX, _tmpYYY] var; and the formula assignment
       will be applied to [:,:]. *)
    try
      let (row_e, row_stmts) = expand_dimension (fname ^ "_" ^ (String.concat "_" (List.map fst inits)) ^ "_row_dim") row_dim in
      let (col_e, col_stmts) = expand_dimension (fname ^ "_" ^ (String.concat "_" (List.map fst inits)) ^ "_col_dim") col_dim in
      row_stmts @ col_stmts @ List.concat (List.map (expand_init (row_e, col_e)) inits)
    with IllegalExpression(s) ->
      raise (IllegalExpression("Illegal expression (" ^ s ^ ") in " ^
                               string_of_varinit ((row_dim, col_dim), inits))) in

  let expand_stmt fname = function
    Assign(a) -> expand_assign fname a
  | Varinit(d, inits) -> expand_varinit fname (d, inits) in

  let expand_stmt_list fname stmts = List.concat (List.map (expand_stmt fname) stmts) in

  let expand_params fname params =
    let needs_sizevar = function
        ((None, None), _) -> false
      | _ -> true in
    let params_with_sizevar = List.map (fun x -> (idgen (fname ^ "_" ^ (snd x) ^ "_size"), x)) (List.filter needs_sizevar params) in
    let expanded_args = List.map (fun (sv, ((rv, cv), s)) -> ((sv, s), [((sv, abs_zero), rv); ((sv, abs_one), cv)])) params_with_sizevar in
    let (sizes, inits) = (List.map fst expanded_args, List.concat (List.map snd expanded_args)) in
    let add_item (varset, (assertlist, initlist)) ((sizevar, pos), var) =
      (match var with
         Some Id(s) ->
         if StringSet.mem s varset then
           (* We've seen this variable before; don't initialize it, just assert it *)
           (varset, (BinOp(Id(s), Eq, Selection(Id(sizevar), (Some(Some(pos), None), None))) :: assertlist, initlist))
         else
           (* We're seeing a string for the first time; don't assert it, just create it *)
           (StringSet.add s varset, (assertlist,
                                     Assign(s, zero_comma_zero, Some (Selection(Id(sizevar), (Some(Some(pos), None), None)))) ::
                                     Varinit(one_by_one, [(s, None)]) ::
                                     initlist))
       | Some LitInt(i) -> (* Seeing a number; don't do anything besides create an assertion *)
         (varset, (BinOp(LitInt(i), Eq, Selection(Id(sizevar), (Some(Some(pos), None), None))) :: assertlist, initlist))
       | Some e -> raise (IllegalExpression("Illegal expression (" ^ string_of_expr e ^ ") in function signature"))
       | _ -> raise (IllegalExpression("Cannot supply a single dimension in function signature"))) in
    let (rev_assertions, rev_inits) = snd (List.fold_left add_item (StringSet.empty, ([], [])) inits) in
    let create_sizevar (sizevar,arg) = [
      Varinit(one_by_one, [(sizevar, None)]);
      Assign(sizevar, entire_range, Some(UnOp(SizeOf,Id(arg))))] in
    (List.concat (List.map create_sizevar sizes), List.rev rev_assertions, List.rev rev_inits) in

  let expand_function f =
    let (new_sizevars, assertions, size_inits) = expand_params f.name f.params in
    let new_retval_id = idgen (f.name ^ "_retval") in
    let new_retval = Id(new_retval_id) in
    let retval_inits = [Varinit (one_by_one, [(new_retval_id, None)]);
                        Assign (new_retval_id, zero_comma_zero, Some (snd f.ret_val))] in
    {
      name = f.name;
      params = f.params;
      raw_asserts = assertions;
      body = new_sizevars @ size_inits @ retval_inits @ expand_stmt_list f.name f.body;
      ret_val = (fst f.ret_val, new_retval)
    } in
  (imports, expand_stmt_list "global" globals, List.map expand_function functions, externs);;

let map_of_list list_of_tuples =
  (*  map_of_list: Take a list of the form [("foo", 2); ("bar", 3)]
      and create a StringMap using the first value of the tuple as
      the key and the second value of the tuple as the value. Raises
      an exception if the key appears more than once in the list. *)
  let rec aux acc = function
      [] -> acc
    | t :: ts ->
      if (StringMap.mem (fst t) acc) then raise(DuplicateDefinition(fst t))
      else aux (StringMap.add (fst t) (snd t) acc) ts in
  aux StringMap.empty list_of_tuples

let create_maps (imports, globals, functions, externs) =
  let vd_of_vi = function
    (*  vd_of_vi--- Take a bare Varinit from the previous transformations
        and return a (string, variable) pair    *)
      Varinit((Some r, Some c), [(v, None)]) -> (v, {
        var_rows = (match r with
              LitInt(i) -> DimInt(i)
            | Id(s) -> DimId(s)
            | _ -> raise (LogicError("Unrecognized expression for rows of " ^ v)));
        var_cols = (match c with
              LitInt(i) -> DimInt(i)
            | Id(s) -> DimId(s)
            | _ -> raise (LogicError("Unrecognized expression for rows of " ^ v)));
        var_formulas = [];
      })
    | _ -> raise (LogicError("Unrecognized format for post-desugaring Varinit")) in

  let add_formula m = function
       Varinit(_,_) -> m
     | Assign(var_name, (Some (Some row_start, row_end), Some (Some col_start, col_end)), Some e) ->
       if StringMap.mem var_name m
       then (let v = StringMap.find var_name m in
             StringMap.add var_name {v with var_formulas = v.var_formulas @ [{
                 formula_row_start = row_start;
                 formula_row_end = row_end;
                 formula_col_start = col_start;
                 formula_col_end = col_end;
                 formula_expr = e;
               }]} m)
       else raise (UnknownVariable(string_of_stmt (Assign(var_name, (Some (Some row_start, row_end), Some (Some col_start, col_end)), Some e))))
     | Assign(a) -> raise (LogicError("Unrecognized format for post-desugaring Assign: " ^ string_of_stmt (Assign(a)))) in

  let vds_of_stmts stmts =
    let is_varinit = function Varinit(_,_) -> true | _ -> false in
    let varinits = List.filter is_varinit stmts in
    let vars_just_the_names = map_of_list (List.map vd_of_vi varinits) in
    List.fold_left add_formula vars_just_the_names stmts in

  let fd_of_raw_func f = (f.name, {
      func_params = f.params;
      func_body = vds_of_stmts f.body;
      func_ret_val = f.ret_val;
      func_asserts = f.raw_asserts;
    }) in

  let tupleize_library (Library(lib_name, lib_fns)) =
    List.map (fun ext_fn -> (ext_fn.extern_fn_name, {ext_fn with extern_fn_libname = lib_name})) lib_fns in

  (vds_of_stmts globals,
   map_of_list (List.map fd_of_raw_func functions),
   map_of_list (List.concat (List.map tupleize_library externs)))

let single_formula e = {
  formula_row_start = DimensionStart;
  formula_row_end = Some DimensionEnd;
  formula_col_start = DimensionStart;
  formula_col_end = Some DimensionEnd;
  formula_expr = e;
}

let ternarize_exprs (globals, functions, externs) =
  let rec ternarize_expr lhs_var = function
```

```
    BinOp(e1, LogAnd, e2) ->
    let (new_e1, new_e1_vars) = ternarize_expr lhs_var e1 in
    let (new_e2, new_e2_vars) = ternarize_expr lhs_var e2 in
    (Ternary(UnOp(Truthy,new_e1), UnOp(Truthy,new_e2), LitInt(0)), new_e1_vars @ new_e2_vars)
  | BinOp(e1, LogOr, e2) ->
    let (new_e1, new_e1_vars) = ternarize_expr lhs_var e1 in
    let (new_e2, new_e2_vars) = ternarize_expr lhs_var e2 in
    (Ternary(UnOp(Truthy,new_e1), LitInt(1), UnOp(Truthy,new_e2)), new_e1_vars @ new_e2_vars)
  | BinOp(e1, op, e2) ->
    let (new_e1, new_e1_vars) = ternarize_expr lhs_var e1 in
    let (new_e2, new_e2_vars) = ternarize_expr lhs_var  e2 in
    (BinOp(new_e1, op, new_e2), new_e1_vars @ new_e2_vars)
  | UnOp(op, e) ->
    let (new_e, new_e_vars) = ternarize_expr lhs_var e in
    (UnOp(op, new_e), new_e_vars)
  | Ternary(cond, e1, e2) ->
    let (new_cond, new_cond_vars) = ternarize_expr lhs_var cond in
    let (new_e1, new_e1_vars) = ternarize_expr lhs_var e1 in
    let (new_e2, new_e2_vars) = ternarize_expr lhs_var e2 in
    (Ternary(new_cond, new_e1, new_e2), new_cond_vars @ new_e1_vars @ new_e2_vars)
  | Call(fname, args) ->
    let new_args_and_vars = List.map (ternarize_expr lhs_var) args in
    (Call(fname, (List.map fst new_args_and_vars)), List.concat (List.map snd new_args_and_vars))
  | Selection(e, (sl1, sl2)) ->
    let (new_e, new_e_vars) = ternarize_expr lhs_var e in
    let (new_sl1, new_sl1_vars) = ternarize_slice lhs_var sl1 in
    let (new_sl2, new_sl2_vars) = ternarize_slice lhs_var sl2 in
    (Selection(new_e, (new_sl1, new_sl2)), new_e_vars @ new_sl1_vars @ new_sl2_vars)
  | Precedence(e1, e2) ->
    let (new_e1, new_e1_vars) = ternarize_expr lhs_var e1 in
    let (new_e2, new_e2_vars) = ternarize_expr lhs_var e2 in
    (Precedence(new_e1, new_e2), new_e1_vars @ new_e2_vars)
  | Switch(cond, cases, dflt) ->
    ternarize_switch lhs_var cases dflt cond
  | LitRange(rowlist) -> let (lhs_varname, _) = lhs_var in formulize_litrange lhs_varname rowlist
  | Debug(e) ->
    let (new_e, new_e_vars) = ternarize_expr lhs_var e in
    (Debug(new_e), new_e_vars)
  | e -> (e, [])
and formulize_litrange lhs_varname rowlist =
  let new_range_id = idgen (lhs_varname ^ "_litrange") in
  let num_rows = List.length rowlist in
  let num_cols = List.fold_left max 0 (List.map List.length rowlist) in
  let formulize_expr r c = function
      LitRange(rl) -> formulize_litrange (new_range_id ^ "_" ^ string_of_int r ^ "_" ^ string_of_int c) rl
    | e -> (e, []) in
  let formulize_row rownum col_exprs =
    let col_formulas_and_vars = List.mapi (fun c e -> formulize_expr rownum c e) col_exprs in
    let create_formula colnum e = {
      formula_row_start = Abs(LitInt(rownum)); formula_row_end = None;
      formula_col_start = Abs(LitInt(colnum)); formula_col_end = None;
      formula_expr = e;
    } in
    (List.mapi create_formula (List.map fst col_formulas_and_vars), List.concat (List.map snd col_formulas_and_vars)) in
  let formulas_and_vars = List.mapi formulize_row rowlist in
  let range_var = {
    var_rows = DimInt(num_rows); var_cols = DimInt(num_cols);
    var_formulas = List.concat (List.map fst formulas_and_vars);
  } in
  (Id(new_range_id),
   (new_range_id, range_var) ::
   List.concat (List.map snd formulas_and_vars))
and ternarize_switch lhs_var cases dflt cond =
  let (new_cond_expr, new_cond_vars) = (match cond with
        Some cond_expr ->
        let (lhs_varname, lhs_vardef) = lhs_var in
        let new_id = idgen (lhs_varname ^ "_switch_cond") in
        let (new_e, new_e_vars) = ternarize_expr lhs_var cond_expr in
        (Some (Selection(Id(new_id),(Some(Some(Rel(LitInt(0)))),None),Some(Some(Rel(LitInt(0))),None)))),
         (new_id, {lhs_vardef with var_formulas = [single_formula new_e]}) ::
         new_e_vars)
      | None ->
        (None,[])
  ) in
  let new_cases_and_vars = List.map (ternarize_case lhs_var new_cond_expr) cases in
  let new_cases = List.map fst new_cases_and_vars in
  let new_case_vars = List.concat (List.map snd new_cases_and_vars) in
  let (new_dflt, new_dflt_vars) = ternarize_expr lhs_var dflt in
  let rec combine_everything = function
      [] -> new_dflt
    | (combined_cases, e) :: more_cases -> Ternary(combined_cases, e, combine_everything more_cases) in
  (combine_everything new_cases, new_cond_vars @ new_case_vars @ new_dflt_vars)
and ternarize_case lhs_var cond (conds, e) =
  let new_conds_and_vars = List.map (ternarize_expr lhs_var) conds in
  let new_conds = List.map fst new_conds_and_vars in
  let new_cond_vars = List.concat (List.map snd new_conds_and_vars) in
  let (new_e, new_e_vars) = ternarize_expr lhs_var e in
  let unify_case_cond_and_switch_cond case_cond = function
      None -> case_cond
    | Some switch_cond -> BinOp(switch_cond,Eq,case_cond) in
  let rec unify_switch_cond_and_case_conds switch_cond = function
      [case_cond] -> unify_case_cond_and_switch_cond case_cond switch_cond
    | case_cond :: case_conds ->
      let (combined_expr, _) = ternarize_expr lhs_var
          (BinOp(unify_case_cond_and_switch_cond case_cond switch_cond, LogOr, unify_switch_cond_and_case_conds switch_cond case_conds)) in
      combined_expr
    | [] -> raise(LogicError("Empty case condition list")) in
  ((unify_switch_cond_and_case_conds cond new_conds, new_e),new_cond_vars @ new_e_vars)
and ternarize_slice lhs_var = function
    None -> (None, [])
  | Some (i1, i2) ->
```

```
      let (new_i1, new_i1_vars) = ternarize_index lhs_var i1 in
      let (new_i2, new_i2_vars) = ternarize_index lhs_var i2 in
      (Some (new_i1, new_i2), new_i1_vars @ new_i2_vars)
  and ternarize_index lhs_var = function
      Some Abs(e) ->
      let (new_e, new_e_vars) = ternarize_expr lhs_var e in
      (Some(Abs(new_e)), new_e_vars)
    | Some Rel(e) ->
      let (new_e, new_e_vars) = ternarize_expr lhs_var e in
      (Some(Rel(new_e)), new_e_vars)
    | i -> (i, []) in
  let ternarize_formula lhs_var f =
    let (new_expr, new_vars) = ternarize_expr lhs_var f.formula_expr in
    ({f with formula_expr = new_expr}, new_vars) in
  let ternarize_variable varname vardef =
    let new_formulas_and_vars = List.map (ternarize_formula (varname, vardef)) vardef.var_formulas in
    ({vardef with var_formulas = List.map fst new_formulas_and_vars}, List.concat (List.map snd new_formulas_and_vars)) in
  let ternarize_variables fn_name m =
    let new_variables_and_maps = StringMap.mapi (fun varname vardef -> ternarize_variable (fn_name ^ "_" ^ varname) vardef) m in
    let add_item var_name (orig_var, new_vars) l = ((var_name, orig_var) :: fst l, new_vars :: snd l) in
    let combined_list = StringMap.fold add_item new_variables_and_maps ([],[]) in
    map_of_list (List.rev (fst combined_list) @ List.concat (snd combined_list)) in
  let ternarize_function fn_name fn_def = {fn_def with func_body = ternarize_variables fn_name fn_def.func_body} in
  (ternarize_variables "global" globals, StringMap.mapi ternarize_function functions, externs)

let reduce_ternaries (globals, functions, externs) =
  let rec reduce_expr lhs_var = function
    | BinOp(e1, op, e2) ->
      let (new_e1, new_e1_vars) = reduce_expr lhs_var e1 in
      let (new_e2, new_e2_vars) = reduce_expr lhs_var  e2 in
      (BinOp(new_e1, op, new_e2), new_e1_vars @ new_e2_vars)
    | UnOp(op, e) ->
      let (new_e, new_e_vars) = reduce_expr lhs_var e in
      (UnOp(op, new_e), new_e_vars)
    | Ternary(cond, e1, e2) -> reduce_ternary lhs_var cond e1 e2
    | Call(fname, args) ->
      let new_args_and_vars = List.map (reduce_expr lhs_var) args in
      (Call(fname, (List.map fst new_args_and_vars)), List.concat (List.map snd new_args_and_vars))
    | Selection(e, (sl1, sl2)) ->
      let (new_e, new_e_vars) = reduce_expr lhs_var e in
      let (new_sl1, new_sl1_vars) = reduce_slice lhs_var sl1 in
      let (new_sl2, new_sl2_vars) = reduce_slice lhs_var sl2 in
      (Selection(new_e, (new_sl1, new_sl2)), new_e_vars @ new_sl1_vars @ new_sl2_vars)
    | Precedence(e1, e2) ->
      let (new_e1, new_e1_vars) = reduce_expr lhs_var e1 in
      let (new_e2, new_e2_vars) = reduce_expr lhs_var e2 in
      (Precedence(new_e1, new_e2), new_e1_vars @ new_e2_vars)
    | Debug(e) ->
      let (new_e, new_e_vars) = reduce_expr lhs_var e in
      (Debug(new_e), new_e_vars)
    | e -> (e, [])
  and reduce_ternary lhs_var cond e1 e2 =
    let (new_cond, new_cond_vars) = reduce_expr lhs_var cond in
    let (new_true_e, new_true_vars) = reduce_expr lhs_var e1 in
    let (new_false_e, new_false_vars) = reduce_expr lhs_var e2 in
    let (lhs_varname, lhs_vardef) = lhs_var in
    let new_cond_id = idgen (lhs_varname ^ "_truthiness") in
    let new_true_id = idgen (lhs_varname ^ "_values_if_true") in
    let new_false_id = idgen (lhs_varname ^ "_values_if_false") in
    (ReducedTernary(new_cond_id, new_true_id, new_false_id),
     (new_cond_id, {lhs_vardef with var_formulas = [single_formula (UnOp(Truthy, new_cond))]}) ::
     (new_true_id, {lhs_vardef with var_formulas = [single_formula new_true_e]}) ::
     (new_false_id, {lhs_vardef with var_formulas = [single_formula new_false_e]}) ::
     (new_cond_vars @ new_true_vars @ new_false_vars))
  and reduce_slice lhs_var = function
      None -> (None, [])
    | Some (i1, i2) ->
      let (new_i1, new_i1_vars) = reduce_index lhs_var i1 in
      let (new_i2, new_i2_vars) = reduce_index lhs_var i2 in
      (Some (new_i1, new_i2), new_i1_vars @ new_i2_vars)
  and reduce_index lhs_var = function
      Some Abs(e) ->
      let (new_e, new_e_vars) = reduce_expr lhs_var e in
      (Some(Abs(new_e)), new_e_vars)
    | Some Rel(e) ->
      let (new_e, new_e_vars) = reduce_expr lhs_var e in
      (Some(Rel(new_e)), new_e_vars)
    | i -> (i, []) in
  let reduce_formula lhs_var f =
    let (new_expr, new_vars) = reduce_expr lhs_var f.formula_expr in
    ({f with formula_expr = new_expr}, new_vars) in
  let reduce_variable varname vardef =
    let new_formulas_and_vars = List.map (reduce_formula (varname, vardef)) vardef.var_formulas in
    ({vardef with var_formulas = List.map fst new_formulas_and_vars}, List.concat (List.map snd new_formulas_and_vars)) in
  let reduce_variables fn_name m =
    let new_variables_and_maps = StringMap.mapi (fun varname vardef -> reduce_variable (fn_name ^ "_" ^ varname) vardef) m in
    let add_item var_name (orig_var, new_vars) l = ((var_name, orig_var) :: fst l, new_vars :: snd l) in
    let combined_list = StringMap.fold add_item new_variables_and_maps ([],[]) in
    map_of_list (List.rev (fst combined_list) @ List.concat (snd combined_list)) in
  let reduce_function fn_name fn_def = {fn_def with func_body = reduce_variables fn_name fn_def.func_body} in
  (reduce_variables "global" globals, StringMap.mapi reduce_function functions, externs)

let check_semantics (globals, functions, externs) =
  let fn_signatures = map_of_list
      (builtin_signatures @
       (StringMap.fold (fun s f l -> (s, List.length f.func_params) :: l) functions []) @
       (StringMap.fold (fun s f l -> (s, List.length f.extern_fn_params) :: l) externs
            [])) in
  let check_function fname f =
    if StringMap.mem fname externs then raise(DuplicateDefinition(fname ^ "() is defined as both an external and local function")) else ();
    let locals = f.func_body in
    let params = List.map snd f.func_params in
    List.iter
      (fun param ->
         if StringMap.mem param locals then raise(DuplicateDefinition(param ^ " is defined multiple times in " ^ fname ^ "()"))
         else ())
      params ;
    let check_call called_fname num_args =
      match called_fname with
          "printf" -> ()
        | _ -> if (not (StringMap.mem called_fname fn_signatures)) then raise(UnknownFunction(called_fname))
            else let signature_args = StringMap.find called_fname fn_signatures in
            if num_args != signature_args then raise(WrongNumberArgs(
                "In " ^ fname ^ "(), the function " ^ called_fname ^ "() was called with " ^ string_of_int num_args ^ " arguments " ^
                "but the signature specifies " ^ string_of_int signature_args
              ))
            else () in
    let rec check_expr = function
        BinOp(e1,_,e2) -> check_expr e1 ; check_expr e2
      | UnOp(_, e) -> check_expr e
      | Ternary(cond, e1, e2) -> check_expr cond ; check_expr e1 ; check_expr e2
      | ReducedTernary(s1, s2, s3) -> check_expr (Id(s1)) ; check_expr (Id(s2)) ; check_expr (Id(s3))
      | Id(s) -> if (List.mem s params || StringMap.mem s locals || StringMap.mem s globals) then () else raise(UnknownVariable(fname ^ "(): " ^ s))
      | Switch(Some e, cases, dflt) -> check_expr e ; List.iter check_case cases ; check_expr dflt
      | Switch(None, cases, dflt) -> List.iter check_case cases ; check_expr dflt
      | Call(called_fname, args) ->
        check_call called_fname (List.length args) ;
        List.iter check_expr args
      | Selection(e, (sl1, sl2)) -> check_expr e ; check_slice sl1 ; check_slice sl2
      | Precedence(e1, e2) -> check_expr e1 ; check_expr e2
      | Debug(e) -> check_expr e;
      | LitInt(_) | LitFlt(_) | LitRange(_) | LitString(_) | Empty | Wild -> ()
    and check_case (conds, e) = List.iter check_expr conds ; check_expr e
    and check_slice = function
        None -> ()
      | Some (i1, i2) -> check_index i1 ; check_index i2
    and check_index = function
        Some Abs(e) -> check_expr e
      | Some Rel(e) -> check_expr e
      | _ -> () in
    let check_formula f =
      check_index (Some f.formula_row_start) ;
      check_index f.formula_row_end ;
      check_index (Some f.formula_col_start) ;
      check_index f.formula_col_end ;
      check_expr f.formula_expr in
    let check_dim = function
        DimInt(1) -> ()
      | DimInt(i) -> raise(IllegalExpression("This is not going to work right"))
      | DimId(s) -> check_expr (Id(s)) in
    let check_variable v =
      check_dim v.var_rows ;
      check_dim v.var_cols ;
      List.iter check_formula v.var_formulas in

    StringMap.iter (fun _ v -> check_variable v) f.func_body ;
    check_expr (snd f.func_ret_val)

  in StringMap.iter check_function functions

let create_ast filename =
  let ast_imp_res = expand_file filename in
  let ast_expanded = expand_expressions ast_imp_res in
  let ast_mapped = create_maps ast_expanded in check_semantics ast_mapped ;
  let ast_ternarized = ternarize_exprs ast_mapped in
  let ast_reduced = reduce_ternaries ast_ternarized in check_semantics ast_reduced ;
  ast_reduced
```

7.5 codegen.ml

```
(* Extend code generator *)

open Ast
open CodeGenTypes
exception NotImplemented
exception LogicError of string

type symbol = LocalVariable of int | GlobalVariable of int | FunctionParameter of int | ExtendFunction of int
and  symbolTable = symbol StringMap.t
and  symbolTableType = Locals | Globals | ExtendFunctions

let helper_functions = Hashtbl.create 10
let runtime_functions = Hashtbl.create 20

let index_map table_type m =
  let add_item key _ (accum_map, accum_idx) =
    let index_val = match table_type with Locals -> LocalVariable(accum_idx) | Globals -> GlobalVariable(accum_idx) | ExtendFunctions -> ExtendFunction(accum_idx) in
    (StringMap.add key index_val accum_map, accum_idx + 1) in
  StringMap.fold add_item m (StringMap.empty, 0)

let (=>) struct_ptr elem = (fun val_name builder ->
    let the_pointer = Llvm.build_struct_gep struct_ptr elem "the_pointer" builder in
    Llvm.build_load the_pointer val_name builder);;

(* from http://stackoverflow.com/questions/243864/what-is-the-ocaml-idiom-equivalent-to-pythons-range-function without the infix *)
let zero_until i =
  let rec aux n acc =
    if n < 0 then acc else aux (n-1) (n :: acc)
  in aux (i-1) []

let create_runtime_functions ctx bt the_module =
  let add_runtime_func fname returntype arglist =
    let the_func = Llvm.declare_function fname (Llvm.function_type returntype arglist) the_module
    in Hashtbl.add runtime_functions fname the_func in
  add_runtime_func "strlen" bt.long_t [|bt.char_p|];
  add_runtime_func "strcmp" bt.long_t [|bt.char_p; bt.char_p|];
  add_runtime_func "pow" bt.float_t [|bt.float_t; bt.float_t|] ;
  add_runtime_func "lrint" bt.int_t [|bt.float_t|] ;
  add_runtime_func "llvm.memcpy.p0i8.p0i8.i64" bt.void_t [|bt.char_p; bt.char_p; bt.long_t; bt.int_t; bt.bool_t|] ;
  add_runtime_func "getVal" bt.value_p [|bt.var_instance_p; bt.int_t; bt.int_t|] ;
  add_runtime_func "clone_value" bt.value_p [|bt.value_p;|] ;
  (* add_runtime_func "freeMe" (Llvm.void_type ctx) [|bt.extend_scope_p;|] ; *)
  add_runtime_func "getSize" bt.value_p [|bt.var_instance_p;|] ;
  add_runtime_func "get_variable" bt.var_instance_p [|bt.extend_scope_p; bt.int_t|] ;
  add_runtime_func "null_init" (Llvm.void_type ctx) [|bt.extend_scope_p|] ;
  add_runtime_func "debug_print" (Llvm.void_type ctx) [|bt.value_p ; bt.char_p|] ;
  add_runtime_func "new_string_go_all_the_way" bt.value_p [|bt.char_p|] ;
  ()

let create_helper_functions ctx bt the_module =
  let create_def_bod fname rtype argtypes =
    let fn_def = Llvm.define_function fname (Llvm.function_type rtype (Array.of_list argtypes)) the_module in
    let fn_bod = Llvm.builder_at_end ctx (Llvm.entry_block fn_def) in
    (fn_def, fn_bod) in

  (* let create_is_subrange_1x1 fname =
    let is_index_one fn builder idx =
      let the_value = ((Llvm.param fn 0) => (subrange_field_index idx)) "the_value" builder in
      let the_bool = Llvm.build_icmp Llvm.Icmp.Eq the_value (Llvm.const_int bt.int_t 1) "the_bool" builder in
      the_bool in
    let (fn_def, fn_bod) = create_def_bod fname bt.bool_t [bt.subrange_p] in
    let one_row = is_index_one fn_def fn_bod SubrangeRows in
    let one_col = is_index_one fn_def fn_bod SubrangeCols in
    let one_by_one = Llvm.build_and one_row one_col "one_by_one" fn_bod in
    let _ = Llvm.build_ret one_by_one fn_bod in
    Hashtbl.add helper_functions fname fn_def in
 *)
  let create_new_string fname =
    let (fn_def, fn_bod) = create_def_bod fname bt.string_p [bt.char_p] in
    let the_string_ptr = Llvm.build_malloc bt.string_t "the_string_ptr" fn_bod in
    let src_char_ptr = Llvm.param fn_def 0 in
    let dst_char_ptr_ptr = Llvm.build_struct_gep the_string_ptr (string_field_index StringCharPtr) "dst_char_ptr_ptr" fn_bod in
    let string_len = Llvm.build_call (Hashtbl.find runtime_functions "strlen") [|src_char_ptr|] "string_len" fn_bod in
    let extra_byte = Llvm.build_add string_len (Llvm.const_int bt.long_t 1) "extra_byte" fn_bod in
    let strlen_ptr = Llvm.build_struct_gep the_string_ptr (string_field_index StringLen) "strlen_ptr" fn_bod in
    let refcount_ptr = Llvm.build_struct_gep the_string_ptr (string_field_index StringRefCount) "refcount" fn_bod in
    let dst_char_ptr = Llvm.build_array_malloc bt.char_t extra_byte "dst_char_ptr" fn_bod in
    let _ = Llvm.build_store dst_char_ptr dst_char_ptr_ptr fn_bod in
    let _ = Llvm.build_call (Hashtbl.find runtime_functions "llvm.memcpy.p0i8.p0i8.i64")
        [| dst_char_ptr ; src_char_ptr ; extra_byte ; (Llvm.const_int bt.int_t 0) ; (Llvm.const_int bt.bool_t 0) |]
        "" fn_bod in
    let _ = Llvm.build_store string_len strlen_ptr fn_bod in
    let _ = Llvm.build_store (Llvm.const_int bt.int_t 1) refcount_ptr fn_bod in
    let _ = Llvm.build_ret the_string_ptr fn_bod in
    Hashtbl.add helper_functions fname fn_def in

  (* let create_box_native_string_list fname =
    let (fn_def, fn_bod) = create_def_bod fname bt.string_p_p [bt.int_t; bt.char_p_p] in
    let argc = Llvm.param fn_def 0 in
    let argv = Llvm.param fn_def 1 in
    let ret_val = Llvm.build_array_malloc bt.string_p argc "ret_val" fn_bod in
    let i_ptr = Llvm.build_alloca bt.int_t "i_ptr" fn_bod in
    let _ = Llvm.build_store (Llvm.const_int bt.int_t 0) i_ptr fn_bod in (* i = 0; *)
    let pred_bb = Llvm.append_block ctx "while_pred" fn_def in
    let body_bb = Llvm.append_block ctx "while_body" fn_def in
    let merge_bb = Llvm.append_block ctx "merge" fn_def in
    let _ = Llvm.build_br pred_bb fn_bod in
    let pred_builder = Llvm.builder_at_end ctx pred_bb in
    let i_val = Llvm.build_load i_ptr "i" pred_builder in
    let pred_bool = Llvm.build_icmp Llvm.Icmp.Slt i_val argc "i_lt_argc" pred_builder in (* i < argc *)
    let _ = Llvm.build_cond_br pred_bool body_bb merge_bb pred_builder in
    let body_builder = Llvm.builder_at_end ctx body_bb in
    let i_val = Llvm.build_load i_ptr "i" body_builder in
    let argv_i_addr = Llvm.build_in_bounds_gep argv [|i_val|] "argv_i_addr" body_builder in
    let argv_i = Llvm.build_load argv_i_addr "argv_i" body_builder in
    let ns_ptr = Llvm.build_call (Hashtbl.find helper_functions "new_string") [|argv_i|] "ns_ptr" body_builder in
    let dst = Llvm.build_in_bounds_gep ret_val [|i_val|] "dst" body_builder in
    let _ = Llvm.build_store ns_ptr dst body_builder in
    let i_plus_1 = Llvm.build_add i_val (Llvm.const_int bt.int_t 1) "i_plus_1" body_builder in
    let _ = Llvm.build_store i_plus_1 i_ptr body_builder in
    let _ = Llvm.build_br pred_bb body_builder in
    let merge_builder = Llvm.builder_at_end ctx merge_bb in
    let _ = Llvm.build_ret ret_val merge_builder in
    Hashtbl.add helper_functions fname fn_def in *)

  let create_box_value_string fname =
    let (fn_def, fn_bod) = create_def_bod fname bt.value_p [bt.string_p] in
    let str = Llvm.param fn_def 0 in
    let ret_val = Llvm.build_malloc bt.value_t "" fn_bod in
    let sp = Llvm.build_struct_gep ret_val (value_field_index String) "str_pointer" fn_bod in
    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
    let _ = Llvm.build_store str sp fn_bod in
    let _ = Llvm.build_ret ret_val fn_bod in
    Hashtbl.add helper_functions fname fn_def in

  (* let create_box_value_float fname =
    let (fn_def, fn_bod) = create_def_bod fname bt.value_p [bt.float_t] in
    let str = Llvm.param fn_def 0 in
    let ret_val = Llvm.build_malloc bt.value_t "" fn_bod in
    let sp = Llvm.build_struct_gep ret_val (value_field_index Number) "num_pointer"
        fn_bod in
    let _ = Llvm.build_store (Llvm.const_int bt.char_t (value_field_flags_index Number
        )) (Llvm.build_struct_gep ret_val (value_field_index Flags) "" fn_bod) fn_bod
        in
    let _ = Llvm.build_store str sp fn_bod in
    let _ = Llvm.build_ret ret_val fn_bod in
    Hashtbl.add helper_functions fname fn_def in *)


  (* let create_box_single_value fname =
    let (fn_def, fn_bod) = create_def_bod fname bt.subrange_p [bt.value_p] in
    let value = Llvm.param fn_def 0 in
    let subrange = Llvm.build_malloc bt.subrange_t "" fn_bod in
    let var_instance = Llvm.build_malloc bt.var_instance_t "" fn_bod in
    let rp = Llvm.build_struct_gep subrange (subrange_field_index BaseRangePtr) "
        range_p" fn_bod in
    let vp = Llvm.build_struct_gep var_instance (var_instance_field_index Values) "
        value_p" fn_bod in
    let _ = Llvm.build_store value vp fn_bod in
    let _ = Llvm.build_store var_instance rp fn_bod in
    let _ = Llvm.build_store (Llvm.const_int bt.int_t 0) (Llvm.build_struct_gep
        subrange (subrange_field_index BaseOffsetCol) "" fn_bod) fn_bod in
    let _ = Llvm.build_store (Llvm.const_int bt.int_t 0) (Llvm.build_struct_gep
        subrange (subrange_field_index BaseOffsetRow) "" fn_bod) fn_bod in
    let _ = Llvm.build_store (Llvm.const_int bt.int_t 1) (Llvm.build_struct_gep
        subrange (subrange_field_index SubrangeRows) "" fn_bod) fn_bod in
    let _ = Llvm.build_store (Llvm.const_int bt.int_t 1) (Llvm.build_struct_gep
        subrange (subrange_field_index SubrangeCols) "" fn_bod) fn_bod in
    let _ = Llvm.build_store (Llvm.const_int bt.int_t 1) (Llvm.build_struct_gep
        var_instance (var_instance_field_index Rows) "" fn_bod) fn_bod in
    let _ = Llvm.build_store (Llvm.const_int bt.int_t 1) (Llvm.build_struct_gep
        var_instance (var_instance_field_index Cols) "" fn_bod) fn_bod in
    let _ = Llvm.build_ret subrange fn_bod in
    Hashtbl.add helper_functions fname fn_def in *)

    (* create_is_subrange_1x1 "is_subrange_1x1"; *)
    (*create_get_val "get_val";
    create_deref_subrange "deref_subrange";*)
    create_new_string "new_string";
    (* create_box_native_string_list "box_native_string_list"; *)
    create_box_value_string "box_value_string";
    (* create_box_single_value "box_single_value"; *)
    (* create_box_value_float "box_value_float";  *)
    ()

```

```
let translate (globals, functions, externs) =

  (* LLVM Boilerplate *)
  let context = Llvm.global_context () in
  let base_module = Llvm.create_module context "Extend" in
  let base_types = setup_types context in

  (* Declare the runtime functions that we need to call *)
  create_runtime_functions context base_types base_module ;
  create_helper_functions context base_types base_module ;

  (* Build function_llvalues, which is a StringMap from function name to llvalue.
   * It includes both functions from external libraries, such as the standard library,
   * and functions declared within Extend. *)
  let declare_library_function fname func accum_map =
    let llvm_ftype = Llvm.function_type base_types.value_p (Array.of_list (List.map (fun a -> base_types.value_p) func.extern_fn_params)) in
    let llvm_fn = Llvm.declare_function fname llvm_ftype base_module in
    StringMap.add fname llvm_fn accum_map in
  let library_functions = StringMap.fold declare_library_function externs StringMap.empty in
  let define_user_function fname func =
    let llvm_fname = "extend_" ^ fname in
    let llvm_ftype = Llvm.function_type base_types.value_p (Array.of_list (List.map (fun a -> base_types.value_p) func.func_params)) in
    let llvm_fn = Llvm.define_function llvm_fname llvm_ftype base_module in
    (func, llvm_fn) in
  let extend_functions = StringMap.mapi define_user_function functions in
  let function_llvalues = StringMap.fold StringMap.add (StringMap.map snd extend_functions) library_functions in

  (* Build the global symbol table *)
  let (global_symbols, num_globals) = index_map Globals globals in
  let (extend_fn_numbers, num_extend_fns) = index_map ExtendFunctions extend_functions in

  (* Create the global array that will hold each function's array of var_defns. *)
  let vardefn_ptr = Llvm.const_pointer_null base_types.var_defn_p in
  let vardefn_array = Array.make (StringMap.cardinal extend_functions) vardefn_ptr in
  let array_of_vardefn_ptrs = Llvm.define_global "array_of_vardefn_ptrs" (Llvm.const_array base_types.var_defn_p vardefn_array) base_module in

  (* Create the pointer to the global scope object *)
  let global_scope_loc = Llvm.define_global "global_scope_loc" (Llvm.const_pointer_null base_types.extend_scope_p) base_module in

  let main_def = Llvm.define_function "main" (Llvm.function_type base_types.int_t [|base_types.int_t; base_types.char_p_p|]) base_module in
  let main_bod = Llvm.builder_at_end context (Llvm.entry_block main_def) in

  (* Create the array of value_ps that will contain the responses to TypeOf(val) *)
  let null_val_ptr = Llvm.const_pointer_null base_types.value_p in
  let null_val_array = Array.make (Array.length int_to_type_array) null_val_ptr in
  let array_of_typeof_val_ptrs = Llvm.define_global "array_of_val_ptrs" (Llvm.const_array base_types.value_p null_val_array) base_module in
  let create_typeof_string i s =
    let sp = Llvm.build_global_stringptr s "global_typeof_stringptr" main_bod in
    let vp = Llvm.build_call (Hashtbl.find runtime_functions "new_string_go_all_the_way") [|sp|] "global_typeof_string" main_bod in
    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
    let _ = Llvm.build_store vp vp_dst main_bod in
    () in
  Array.iteri create_typeof_string int_to_type_array ;

  (* Look these two up once and for all *)
  (* let deepCopy = Hashtbl.find runtime_functions "deepCopy" in *)
  (* let freeMe = Hashtbl.find runtime_functions "freeMe" in *)
  let getVal = Hashtbl.find runtime_functions "getVal" in (*getVal retrieves the value of a variable instance for a specific x and y*)
  let getVar = Hashtbl.find runtime_functions "get_variable" in (*getVar retrieves a variable instance based on the offset. It instanciates the variable if it does not exist yet*)

  (* build_formula_function takes a symbol table and an expression, builds the LLVM function, and returns the llvalue of the function *)
  let build_formula_function (varname, formula_idx) symbols formula_expr =
    let form_decl = Llvm.define_function ("formula_fn_" ^ varname ^ "_num_" ^ (string_of_int formula_idx)) base_types.formula_call_t base_module in
    let builder_at_top = Llvm.builder_at_end context (Llvm.entry_block form_decl) in
    let local_scope = Llvm.param form_decl 0 in
    let global_scope = Llvm.build_load global_scope_loc "global_scope" builder_at_top in

    (* Some repeated stuff to avoid cut & paste *)
    let empty_type = (Llvm.const_int base_types.char_t (value_field_flags_index Empty)) in
    let number_type = (Llvm.const_int base_types.char_t (value_field_flags_index Number)) in
    let string_type = (Llvm.const_int base_types.char_t (value_field_flags_index String)) in
    let range_type = (Llvm.const_int base_types.char_t (value_field_flags_index Range)) in
    let make_block blockname =
      let new_block = Llvm.append_block context blockname form_decl in
      let new_builder = Llvm.builder_at_end context new_block in
      (new_block, new_builder) in
    let store_number value_ptr store_builder number_llvalue =
      let sp = Llvm.build_struct_gep value_ptr (value_field_index Number) "num_pointer" store_builder in
      let _ = Llvm.build_store number_type (Llvm.build_struct_gep value_ptr (value_field_index Flags) "" store_builder) store_builder in
      ignore (Llvm.build_store number_llvalue sp store_builder) in
    let store_empty value_ptr store_builder =
      ignore (Llvm.build_store empty_type (Llvm.build_struct_gep value_ptr (value_field_index Flags) "" store_builder) store_builder) in

    let make_truthiness_blocks blockprefix ret_val =
      let (merge_bb, merge_builder) = make_block (blockprefix ^ "_merge") in

      let (make_true_bb, make_true_builder) = make_block (blockprefix ^ "_true") in
      let _ = store_number ret_val make_true_builder (Llvm.const_float base_types.float_t 1.0) in
      let _ = Llvm.build_br merge_bb make_true_builder in

      let (make_false_bb, make_false_builder) = make_block (blockprefix ^ "_false") in
      let _ = store_number ret_val make_false_builder (Llvm.const_float base_types.float_t 0.0) in
      let _ = Llvm.build_br merge_bb make_false_builder in

      let (make_empty_bb, make_empty_builder) = make_block (blockprefix ^ "_empty") in
      let _ = store_empty ret_val make_empty_builder  in
      let _ = Llvm.build_br merge_bb make_empty_builder in

      (make_true_bb, make_false_bb, make_empty_bb, merge_builder) in

    let rec build_expr old_builder exp = match exp with
        LitInt(i) -> let vvv = Llvm.const_float base_types.float_t (float_of_int i) in
        let ret_val = Llvm.build_malloc base_types.value_t "int_ret_val" old_builder in
        let _ = store_number ret_val old_builder vvv in
        (ret_val, old_builder)
      | LitFlt(f) -> let vvv = Llvm.const_float base_types.float_t f in
        let ret_val = Llvm.build_malloc base_types.value_t "flt_ret_val" old_builder in
        let _ = store_number ret_val old_builder vvv in
        (ret_val, old_builder)
      | Empty ->
        let ret_val = Llvm.build_malloc base_types.value_t "empty_ret_val" old_builder in
        let _ = store_empty ret_val old_builder in
        (ret_val, old_builder)
      | Debug(e) ->
        let (ret_val, new_builder) = build_expr old_builder e in
        let _ = Llvm.build_call (Hashtbl.find runtime_functions "debug_print") [|ret_val; Llvm.const_pointer_null base_types.char_p|] "" new_builder in
        (ret_val, new_builder)
      | Id(name) ->
        (
          match (try StringMap.find name symbols with Not_found -> raise(LogicError("Something went wrong with your semantic analysis - " ^ name ^ " not found"))) with
            LocalVariable(i) ->
            let llvm_var = Llvm.build_call getVar [|local_scope; Llvm.const_int base_types.int_t i|] "llvm_var" old_builder in
            (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)
          | GlobalVariable(i) ->
            let llvm_var = Llvm.build_call getVar [|global_scope; Llvm.const_int base_types.int_t i|] "llvm_var" old_builder in
            (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)
          | FunctionParameter(i) ->
            let paramarray = (local_scope => (scope_field_type_index FunctionParams)) "paramarray" old_builder in
            let param_addr = Llvm.build_in_bounds_gep paramarray [|Llvm.const_int
                base_types.int_t i|] "param_addr" old_builder in
            let param = Llvm.build_load param_addr "param" old_builder in
            (Llvm.build_call (Hashtbl.find runtime_functions "clone_value") [|param|]
                "function_param_ret_val" old_builder, old_builder)
          | ExtendFunction(i) -> raise(LogicError("Something went wrong with your
              semantic analyis - function " ^ name ^ " used as variable in RHS for " ^
              varname))
        )
      | Selection(expr, sel) -> build_expr old_builder expr
      | Precedence(a,b) -> let (_, new_builder) = build_expr old_builder a in
          build_expr new_builder b
      | LitString(str) ->
        let boxxx = Llvm.build_call
            (Hashtbl.find helper_functions "new_string")
            (Array.of_list [
                Llvm.build_global_stringptr str "glob_str" old_builder
              ]) "boxed_str" old_builder in
        let boxx = Llvm.build_call
            (Hashtbl.find helper_functions "box_value_string")
            (Array.of_list [boxxx]) "box_value_str" old_builder
        in (boxx, old_builder)
      | Call(fn,exl) -> (*TODO: Call needs to be reviewed. Possibly switch call
          arguments to value_p*)
        let build_one_expr (arg_list, intermediate_builder) e =
          let (arg_val, next_builder) = build_expr intermediate_builder e in
          (arg_val :: arg_list, next_builder) in
        let (reversed_arglist, call_builder) = List.fold_left build_one_expr ([],
            old_builder) exl in
        let args = Array.of_list (List.rev reversed_arglist) in
        let result = Llvm.build_call (
          StringMap.find fn function_llvalues
          ) args "call_ret_val" call_builder in
        (result, call_builder)
      | BinOp(expr1,op,expr2) -> (
          let (val1, builder1) = build_expr old_builder expr1 in
          let (val2, int_builder) = build_expr builder1 expr2 in
          let bit_shift = (Llvm.const_int base_types.char_t 4) in
          let expr1_type = (val1 => (value_field_index Flags)) "expr1_type"
              int_builder in
          let expr2_type = (val2 => (value_field_index Flags)) "expr2_type"
              int_builder in
          let expr1_type_shifted = Llvm.build_shl expr1_type bit_shift "
              expr_1_type_shifted" int_builder in
          let combined_type = Llvm.build_add expr1_type_shifted expr2_type "
              combined_type" int_builder in
          let number_number = Llvm.const_add (Llvm.const_shl number_type bit_shift)
              number_type in
          let string_string = Llvm.const_add (Llvm.const_shl string_type bit_shift)
              string_type in
          let empty_empty = Llvm.const_add (Llvm.const_shl empty_type bit_shift)
              empty_type in
          let range_range = Llvm.const_add (Llvm.const_shl range_type bit_shift)
              range_type in
          let build_simple_binop oppp int_builder =
            (let ret_val = Llvm.build_malloc base_types.value_t "binop_minus_ret_val"
                int_builder in
              let _ = Llvm.build_store
                  (
                    Llvm.const_int
                    base_types.char_t
                    (value_field_flags_index Empty)
                  ) (
                    Llvm.build_struct_gep
                    ret_val
                    (value_field_index Flags)
                    ""
                    int_builder
                  )
                  int_builder
              in
              let bailout = (Llvm.append_block context "" form_decl) in
              let bbailout = Llvm.builder_at_end context bailout in
              let (numnum_bb, numnum_builder) = make_block "numnum" in
              let numeric_val_1 = (val1 => (value_field_index Number)) "number_one"
                  numnum_builder in
              let numeric_val_2 = (val2 => (value_field_index Number)) "number_two"
                  numnum_builder in
              let numeric_res = oppp numeric_val_1 numeric_val_2 "numeric_res"
                  numnum_builder in
              let _ = Llvm.build_store
                  numeric_res (
                    Llvm.build_struct_gep
                    ret_val
                    (value_field_index Number)
                    ""
                    numnum_builder
                  )
                  numnum_builder in
              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)
                    ""
                    numnum_builder
                  )
                  numnum_builder in
              let _ = Llvm.build_br bailout numnum_builder in
              let _ = Llvm.build_cond_br (Llvm.build_icmp Llvm.Icmp.Eq combined_type
                  number_number "" int_builder) numnum_bb bailout int_builder in
               (ret_val, bbailout)
           )
           and build_simple_int_binop oppp int_builder =
             (let ret_val = Llvm.build_malloc base_types.value_t "binop_minus_ret_val"
                  int_builder in
               let _ = Llvm.build_store
                   (
                   Llvm.const_int
                   base_types.char_t
                   (value_field_flags_index Empty)
                 ) (
                   Llvm.build_struct_gep
                   ret_val
                   (value_field_index Flags)
                   ""
                   int_builder
                 )
                 int_builder
             in
             let bailout = (Llvm.append_block context "" form_decl) in
             let bbailout = Llvm.builder_at_end context bailout in
             let (numnum_bb, numnum_builder) = make_block "numnum" in
             let roundfl x = Llvm.build_call (Hashtbl.find runtime_functions "lrint") [|x|] "" numnum_builder in
             let numeric_val_1 = roundfl ((val1 => (value_field_index Number)) "number_one" numnum_builder) in
             let numeric_val_2 = roundfl ((val2 => (value_field_index Number)) "number_two" numnum_builder) in
             let numeric_res = oppp numeric_val_1 numeric_val_2 "numeric_res" numnum_builder in
             let _ = Llvm.build_store
                 (Llvm.build_sitofp numeric_res base_types.float_t "" numnum_builder )
                 (
                   Llvm.build_struct_gep
                   ret_val
                   (value_field_index Number)
                   ""
                   numnum_builder
                 )
                 numnum_builder in
             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)
                   ""
                   numnum_builder
                 )
                 numnum_builder in
             let _ = Llvm.build_br bailout numnum_builder in
             let _ = Llvm.build_cond_br (Llvm.build_icmp Llvm.Icmp.Eq combined_type number_number "" int_builder) numnum_bb bailout int_builder in
              (ret_val, bbailout)
           ) in
        let build_boolean_op numeric_comparator string_comparator int_builder =
          let ret_val = Llvm.build_malloc base_types.value_t "binop_gt_ret_val" int_builder in
          let (make_true_bb, make_false_bb, make_empty_bb, merge_builder) =
              make_truthiness_blocks "binop_eq" ret_val in

          let (numnum_bb, numnum_builder) = make_block "numnum" in
          let numeric_val_1 = (val1 => (value_field_index Number)) "number_one" numnum_builder in
          let numeric_val_2 = (val2 => (value_field_index Number)) "number_two" numnum_builder in
          let numeric_greater = Llvm.build_fcmp numeric_comparator numeric_val_1 numeric_val_2 "numeric_greater" numnum_builder in
          let _ = Llvm.build_cond_br numeric_greater make_true_bb make_false_bb numnum_builder in

          let (strstr_bb, strstr_builder) = make_block "strstr" in
          let str_p_1 = (val1 => (value_field_index String)) "string_one" strstr_builder in
          let str_p_2 = (val2 => (value_field_index String)) "string_two" strstr_builder in
          let char_p_1 = (str_p_1 => (string_field_index StringCharPtr)) "char_p_one" strstr_builder in
          let char_p_2 = (str_p_2 => (string_field_index StringCharPtr)) "char_p_two" strstr_builder in
          let strcmp_result = Llvm.build_call (Hashtbl.find runtime_functions "strcmp") [|char_p_1; char_p_2|] "strcmp_result" strstr_builder in
          let string_greater = Llvm.build_icmp string_comparator strcmp_result (Llvm.const_null base_types.long_t) "string_greater" strstr_builder in
          let _ = Llvm.build_cond_br string_greater make_true_bb make_false_bb strstr_builder in

          let switch_inst = Llvm.build_switch combined_type make_empty_bb 2 int_builder in (* Incompatible ===> default to empty *)
          Llvm.add_case switch_inst number_number numnum_bb;
          Llvm.add_case switch_inst string_string strstr_bb;
          (ret_val, merge_builder) in
        match op with
          Minus -> build_simple_binop Llvm.build_fsub int_builder
        | Plus ->
            let result = Llvm.build_malloc base_types.value_t "" int_builder
            and stradd = (Llvm.append_block context "" form_decl)
            and numadd = (Llvm.append_block context "" form_decl)
            and bailout = (Llvm.append_block context "" form_decl)
            and numorstrorother = (Llvm.append_block context "" form_decl)
            and strorother = (Llvm.append_block context "" form_decl)
            in
            let bstradd = Llvm.builder_at_end context stradd
            and bnumadd = Llvm.builder_at_end context numadd
            and bnumorstrorother = Llvm.builder_at_end context numorstrorother
            and bstrorother = Llvm.builder_at_end context strorother
            and bbailout = Llvm.builder_at_end context bailout
            and _ = Llvm.build_store
                (
                  Llvm.const_int
                  base_types.char_t
                  (value_field_flags_index Empty)
                ) (
                  Llvm.build_struct_gep
                  result
                  (value_field_index Flags)
                  ""
                  int_builder
                )
                int_builder
            in
            (*let _ = Llvm.build_cond_br pred_bool body_bb merge_bb pred_builder in
                *)
            let isnumber = Llvm.build_icmp
                Llvm.Icmp.Eq
                (
                  Llvm.build_load
                  (
                    Llvm.build_struct_gep
                    val1
                    (value_field_index Flags)
                    ""
                    bnumorstrorother
                  ) "" bnumorstrorother
                ) (
                  Llvm.const_int
                  base_types.char_t
                  (value_field_flags_index Number)
                )
                ""
                bnumorstrorother
            and isstring = Llvm.build_icmp
                Llvm.Icmp.Eq
                (
                  Llvm.build_load
                  (
                    Llvm.build_struct_gep
                    val1
                    (value_field_index Flags)
                    ""
                    bstrorother
                  )
                  ""
                  bstrorother
                ) (
                  Llvm.const_int
                  base_types.char_t
                  (value_field_flags_index String)
                )
                ""
                bstrorother
            and isnumorstring = Llvm.build_icmp
                Llvm.Icmp.Eq
                (
                  Llvm.build_load
                  (
                    Llvm.build_struct_gep
                    val1
                    (value_field_index Flags)
                    ""
                    int_builder
                  )
                  ""
                  int_builder
                ) (
                  Llvm.build_load
                  (
                    Llvm.build_struct_gep
                    val2
                    (value_field_index Flags)
                    ""
                    int_builder
                  )
                  ""
                  int_builder
                )
                ""
                int_builder
            and _ = Llvm.build_store (
                Llvm.build_fadd
                (
                  Llvm.build_load
                  (
                    Llvm.build_struct_gep
                    val1
                    (value_field_index Number)
                    ""
                    bnumadd
                  )
                  ""
                  bnumadd
                ) (
                  Llvm.build_load
                  (
                    Llvm.build_struct_gep
                    val2
                    (value_field_index Number)
                    ""
                    bnumadd
                  )
                  ""
                  bnumadd
                )
                ""
                bnumadd
              ) (
                Llvm.build_struct_gep
                result
                (value_field_index Number)
                ""
                bnumadd
              )
              bnumadd
            and _ = Llvm.build_store (
                Llvm.const_int base_types.char_t (value_field_flags_index Number)
              ) (
                Llvm.build_struct_gep
                result
                (value_field_index Flags)
                ""
                bnumadd
              )
              bnumadd
            and str1 = Llvm.build_load
            (
              Llvm.build_struct_gep
              val1
              (value_field_index String)
              ""
              bstradd
            ) "" bstradd
            and str2 = Llvm.build_load
            (
              Llvm.build_struct_gep
              val2
              (value_field_index String)
              ""
              bstradd
            ) "" bstradd
            and newstr =
            (
              Llvm.build_malloc base_types.string_t "" bstradd
            )
            in
            let len1 = Llvm.build_load (
              Llvm.build_struct_gep
              str1
              (string_field_index StringLen)
              ""
              bstradd
            ) "" bstradd
            and len2 = Llvm.build_load (
              Llvm.build_struct_gep
              str2
              (string_field_index StringLen)
              ""
              bstradd
            ) "" bstradd
            and p1 = Llvm.build_load (
              Llvm.build_struct_gep
              str1
              (string_field_index StringCharPtr)
              ""
              bstradd
            ) "" bstradd
            and p2 = Llvm.build_load (
              Llvm.build_struct_gep
              str2
              (string_field_index StringCharPtr)
              ""
              bstradd
            ) "" bstradd
            and dst_char_ptr_ptr = (
              Llvm.build_struct_gep
              newstr
              (string_field_index StringCharPtr)
              ""
              bstradd
            )
            and _ = Llvm.build_store (
              Llvm.const_int base_types.char_t (value_field_flags_index String)
            ) (
              Llvm.build_struct_gep
              result
              (value_field_index Flags)
              ""
              bstradd
            ) bstradd
            and _ = Llvm.build_store newstr (
              Llvm.build_struct_gep
              result
              (value_field_index String)
              ""
              bstradd
            )
            bstradd in
            let fullLen = Llvm.build_nsw_add (Llvm.build_nsw_add len1 len2 "" bstradd) (Llvm.const_int base_types.long_t 1) "" bstradd
            and extra_byte2 = (Llvm.build_add len2 (Llvm.const_int base_types.long_t 1) "" bstradd) in
            let dst_char = Llvm.build_array_malloc base_types.char_t (Llvm.build_trunc fullLen base_types.int_t "" bstradd) "" bstradd in
            let dst_char2 = Llvm.build_in_bounds_gep dst_char [|len1|] "" bstradd in
            let _ = Llvm.build_call
              (Hashtbl.find runtime_functions "llvm.memcpy.p0i8.p0i8.i64")
              [|dst_char; p1; len1; (Llvm.const_int base_types.int_t 0); (Llvm.const_int base_types.bool_t 0)|]
              ""
              bstradd
            and _ = Llvm.build_call
              (Hashtbl.find runtime_functions "llvm.memcpy.p0i8.p0i8.i64")
              [|dst_char2; p2; extra_byte2; (Llvm.const_int base_types.int_t 0); (Llvm.const_int base_types.bool_t 0)|]
              ""
              bstradd
            and _ = Llvm.build_store dst_char dst_char_ptr_ptr bstradd
            in
            let _ = Llvm.build_store (Llvm.build_nsw_add fullLen (Llvm.const_int base_types.long_t (-1)) "" bstradd) (Llvm.build_struct_gep newstr (string_field_index StringLen) "" bstradd) bstradd
            in
            let _ = Llvm.build_cond_br isnumorstring numorstrorother bailout int_builder
            and _ = Llvm.build_cond_br isnumber numadd strorother bnumorstrorother
            and _ = Llvm.build_cond_br isstring stradd bailout bstrorother
            and _ = Llvm.build_br bailout bstradd
            and _ = Llvm.build_br bailout bnumadd
            in
            (result, bbailout)
```

```
          | Times -> build_simple_binop Llvm.build_fmul int_builder
          | Eq ->
            (* let _ = Llvm.build_call (Hashtbl.find runtime_functions "debug_print") [|val1; Llvm.build_global_stringptr "Eq operator - value 1" "" old_builder|] "" int_builder in
            let _ = Llvm.build_call (Hashtbl.find runtime_functions "debug_print") [|val2; Llvm.build_global_stringptr "Eq operator - value 2" "" old_builder|] "" int_builder in *)
            let ret_val = Llvm.build_malloc base_types.value_t "binop_eq_ret_val" int_builder in
            let (make_true_bb, make_false_bb, _, merge_builder) = make_truthiness_blocks "binop_eq" ret_val in

            let (numnum_bb, numnum_builder) = make_block "numnum" in
            let numeric_val_1 = (val1 => (value_field_index Number)) "number_one" numnum_builder in
            let numeric_val_2 = (val2 => (value_field_index Number)) "number_two" numnum_builder in
            let numeric_equality = Llvm.build_fcmp Llvm.Fcmp.Oeq numeric_val_1 numeric_val_2 "numeric_equality" numnum_builder in
            let _ = Llvm.build_cond_br numeric_equality make_true_bb make_false_bb numnum_builder in

            let (strstr_bb, strstr_builder) = make_block "strstr" in
            let str_p_1 = (val1 => (value_field_index String)) "string_one" strstr_builder in
            let str_p_2 = (val2 => (value_field_index String)) "string_two" strstr_builder in
            let char_p_1 = (str_p_1 => (string_field_index StringCharPtr)) "char_p_one" strstr_builder in
            let char_p_2 = (str_p_2 => (string_field_index StringCharPtr)) "char_p_two" strstr_builder in
            let strcmp_result = Llvm.build_call (Hashtbl.find runtime_functions "strcmp") [|char_p_1; char_p_2|] "strcmp_result" strstr_builder in
            let string_equality = Llvm.build_icmp Llvm.Icmp.Eq strcmp_result (Llvm.const_null base_types.long_t) "string_equality" strstr_builder in
            let _ = Llvm.build_cond_br string_equality make_true_bb make_false_bb strstr_builder in

            let (rngrng_bb, rngrng_builder) = make_block "rngrng" in
            (* TODO: Make this case work *)
            let _ = Llvm.build_br make_false_bb rngrng_builder in

            let switch_inst = Llvm.build_switch combined_type make_false_bb 4 int_builder in (* Incompatible ===> default to false *)
            Llvm.add_case switch_inst number_number numnum_bb;
            Llvm.add_case switch_inst string_string strstr_bb;
            Llvm.add_case switch_inst range_range rngrng_bb;
            Llvm.add_case switch_inst empty_empty make_true_bb; (* Nothing to check in this case, just return true *)
            (ret_val, merge_builder)
          | Gt -> build_boolean_op Llvm.Fcmp.Ogt Llvm.Icmp.Sgt int_builder
          | GtEq -> build_boolean_op Llvm.Fcmp.Oge Llvm.Icmp.Sge int_builder
          | Lt -> build_boolean_op Llvm.Fcmp.Olt Llvm.Icmp.Slt int_builder
          | LtEq -> build_boolean_op Llvm.Fcmp.Ole Llvm.Icmp.Sle int_builder
          | 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))
          | Divide-> build_simple_binop Llvm.build_fdiv int_builder
          | Mod-> build_simple_binop Llvm.build_frem int_builder
          | Pow-> (
            let powcall numeric_val_1 numeric_val_2 valname b =
              Llvm.build_call (Hashtbl.find runtime_functions "pow") [|numeric_val_1;
                  numeric_val_2|] "" b in
            build_simple_binop powcall int_builder)
          | LShift-> build_simple_int_binop Llvm.build_shl int_builder
          | RShift-> build_simple_int_binop Llvm.build_lshr int_builder
          | BitOr-> build_simple_int_binop Llvm.build_or int_builder
          | BitAnd-> build_simple_int_binop Llvm.build_and int_builder
          | BitXor-> build_simple_int_binop Llvm.build_xor int_builder
        )
      | UnOp(SizeOf,expr) -> let vvv = Llvm.const_float base_types.float_t 0.0 in
        let ret_val = Llvm.build_malloc base_types.value_t "unop_size_ret_val"
            old_builder in
        let sp = Llvm.build_struct_gep ret_val (value_field_index Number) "num_pointer
            " old_builder in
        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
        let _ = Llvm.build_store vvv sp old_builder in
        (ret_val, old_builder)
      | UnOp(Truthy, expr) ->
        let ret_val = Llvm.build_malloc base_types.value_t "unop_truthy_ret_val"
            old_builder in
        let (expr_val, expr_builder) = build_expr old_builder expr in

        let (truthy_bb, falsey_bb, empty_bb, merge_builder) = make_truthiness_blocks "
            binop_eq" ret_val in

        let expr_flags = (expr_val => (value_field_index Flags)) "expr_flags"
            expr_builder in
        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
        let is_empty = Llvm.build_zext is_empty_bool base_types.char_t "is_empty"
            expr_builder in
        let is_empty_two = Llvm.build_shl is_empty (Llvm.const_int base_types.char_t
            1) "is_empty_two" expr_builder in
        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
        let the_number = (expr_val => (value_field_index Number)) "the_number"
            expr_builder in
        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
        let is_numeric_zero_bool = Llvm.build_and is_zero is_number "
            is_numeric_zero_bool" expr_builder in
        let is_numeric_zero = Llvm.build_zext is_numeric_zero_bool base_types.char_t "
            is_numeric_zero" expr_builder in
        let switch_num = Llvm.build_add is_empty_two is_numeric_zero "switch_num"
            expr_builder in
        let switch_inst = Llvm.build_switch switch_num empty_bb 2 expr_builder in
        Llvm.add_case switch_inst (Llvm.const_int base_types.char_t 0) truthy_bb; (* empty << 1 + is_zero == 0 ===> truthy *)
        Llvm.add_case switch_inst (Llvm.const_int base_types.char_t 1) falsey_bb; (* empty << 1 + is_zero == 1 ===> falsey *)
        (ret_val, merge_builder)
      | UnOp(LogNot, expr) ->
        let (truth_val, truth_builder) = build_expr old_builder (UnOp(Truthy, expr)) in
        let the_number = (truth_val => (value_field_index Number)) "the_number" truth_builder in
        let not_the_number = Llvm.build_fsub (Llvm.const_float base_types.float_t 1.0) the_number "not_the_number" truth_builder in
        let sp = Llvm.build_struct_gep truth_val (value_field_index Number) "num_pointer" truth_builder in
        let _ = Llvm.build_store not_the_number sp truth_builder in
        (truth_val, truth_builder)
      | UnOp(Neg, expr) ->
        let ret_val = Llvm.build_malloc base_types.value_t "unop_truthy_ret_val" old_builder in
        let _ = store_empty ret_val old_builder in
        let (expr_val, expr_builder) = build_expr old_builder expr in
        let expr_type = (expr_val => (value_field_index Flags)) "expr_type" expr_builder in
        let is_number = Llvm.build_icmp Llvm.Icmp.Eq expr_type number_type "is_number" expr_builder in
        let (finish_bb, finish_builder) = make_block "finish" in

        let (number_bb, number_builder) = make_block "number" in
        let the_number = (expr_val => (value_field_index Number)) "the_number" number_builder in
        let minus_the_number = Llvm.build_fneg the_number "minus_the_number" number_builder in
        let _ = store_number ret_val number_builder minus_the_number in
        let _ = Llvm.build_br finish_bb number_builder in

        let _ = Llvm.build_cond_br is_number number_bb finish_bb expr_builder in
        (ret_val, finish_builder)
      | UnOp(BitNot, expr) -> print_endline "Unsupported Unop" ; print_endline (Ast.string_of_expr exp); raise NotImplemented
      | UnOp(TypeOf, expr) ->
        let (expr_val, expr_builder) = build_expr old_builder expr in
        let expr_type = (expr_val => (value_field_index Flags)) "expr_type" expr_builder in
        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
        let vp_to_clone = Llvm.build_load vp_to_clone_loc "vp_to_clone" expr_builder in
        let ret_val = Llvm.build_call (Hashtbl.find runtime_functions "clone_value") [|vp_to_clone|] "typeof_ret_val" expr_builder in
        (ret_val, expr_builder)
      | UnOp(Row, expr) -> print_endline "Unsupported Unop" ; print_endline (Ast.string_of_expr exp); raise NotImplemented
      | UnOp(Column, expr) -> print_endline "Unsupported Unop" ; print_endline (Ast.string_of_expr exp); raise NotImplemented
      | ReducedTernary(cond_var, true_var, false_var) ->
```

```
        let ret_val_addr = Llvm.build_alloca base_types.value_p "tern_ret_val_addr" old_builder in
        let (cond_val, _) = build_expr old_builder (Id(cond_var)) in (* Relying here on the fact that Id() doesn’t change the builder *)
        let merge_bb = Llvm.append_block context "merge" form_decl in
        let merge_builder = Llvm.builder_at_end context merge_bb in
        let ret_val = Llvm.build_load ret_val_addr "tern_ret_val" merge_builder in

        let truthy_bb = Llvm.append_block context "truthy" form_decl in
        let truthy_builder = Llvm.builder_at_end context truthy_bb in
        let (truthy_val, _) = build_expr truthy_builder (Id(true_var)) in (* Relying here on the fact that Id() doesn’t change the builder *)
        let _ = Llvm.build_store truthy_val ret_val_addr truthy_builder in
        let _ = Llvm.build_br merge_bb truthy_builder in

        let falsey_bb = Llvm.append_block context "falsey" form_decl in
        let falsey_builder = Llvm.builder_at_end context falsey_bb in
        let (falsey_val, _) = build_expr falsey_builder (Id(false_var)) in (* Relying here on the fact that Id() doesn’t change the builder *)
        let _ = Llvm.build_store falsey_val ret_val_addr falsey_builder in
        let _ = Llvm.build_br merge_bb falsey_builder in

        let empty_bb = Llvm.append_block context "empty" form_decl in
        let empty_builder = Llvm.builder_at_end context empty_bb in
        let ret_val_empty = Llvm.build_malloc base_types.value_t "tern_empty" empty_builder in
        let _ = store_empty ret_val_empty empty_builder in
        let _ = Llvm.build_store ret_val_empty ret_val_addr empty_builder in
        let _ = Llvm.build_br merge_bb empty_builder in

        let expr_flags = (cond_val => (value_field_index Flags)) "expr_flags" old_builder in
        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" old_builder) in
        let is_empty = Llvm.build_zext is_empty_bool base_types.char_t "is_empty" old_builder in
        let is_empty_two = Llvm.build_shl is_empty (Llvm.const_int base_types.char_t 1) "is_empty_two" old_builder in
        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" old_builder in
        let the_number = (cond_val => (value_field_index Number)) "the_number" old_builder in
        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
        let is_numeric_zero_bool = Llvm.build_and is_zero is_number "is_numeric_zero_bool" old_builder in
        let is_numeric_zero = Llvm.build_zext is_numeric_zero_bool base_types.char_t "is_numeric_zero" old_builder in
        let switch_num = Llvm.build_add is_empty_two is_numeric_zero "switch_num" old_builder in
        let switch_inst = Llvm.build_switch switch_num empty_bb 2 old_builder in
        Llvm.add_case switch_inst (Llvm.const_int base_types.char_t 0) truthy_bb; (* empty << 1 + is_zero == 0 ===> truthy *)
        Llvm.add_case switch_inst (Llvm.const_int base_types.char_t 1) falsey_bb; (*
           empty << 1 + is_zero == 1 ===> falsey *)
        (ret_val, merge_builder)
      | unknown_expr -> print_endline (string_of_expr unknown_expr);raise NotImplemented in
    let (ret_value_p, final_builder) = build_expr builder_at_top formula_expr in
    let _ = Llvm.build_ret ret_value_p final_builder in
    form_decl in

  (*build formula creates a formula declaration in a separate method from the function it belongs to*)
  let build_formula (varname, idx) formula_array element symbols =
    let storage_addr = Llvm.build_in_bounds_gep formula_array [|Llvm.const_int base_types.int_t idx|] "" main_bod in
    let getStarts = function (* Not really just for starts *)
        Abs(LitInt(1)) | Abs(LitInt(0)) | DimensionStart | DimensionEnd -> (1, -1)
      | Abs(Id(s)) ->
        (match StringMap.find s symbols with
           LocalVariable(i) | GlobalVariable(i) -> (0, i)
         | _ -> raise(TransformedAway("Error in " ^ varname ^ ": The LHS expresssions should always either have dimension length 1 or be the name of a variable in their own scope.")))
      | _ -> 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
    let getEnds = function
        Some x -> let (b, c) = getStarts x in (b, c, 0)
      | None -> (0, -1, 1) in
    let (fromStartRow, rowStartVarnum) = getStarts element.formula_row_start in
    let (fromStartCol, colStartVarnum) = getStarts element.formula_col_start in
    let (toEndRow, rowEndVarnum, isSingleRow) = getEnds element.formula_row_end in
    let (toEndCol, colEndVarnum, isSingleCol) = getEnds element.formula_col_end in

    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
    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
    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
    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
    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

    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
    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
    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
  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
  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

  let form_decl = build_formula_function (varname, idx) symbols element.formula_expr in
  let _ = Llvm.build_store form_decl (Llvm.build_struct_gep storage_addr (formula_field_index FormulaCall) "" main_bod) main_bod in
  () in

(* Builds a var_defn struct for each variable *)
let build_var_defn defn varname va symbols =
  let numForm = List.length va.var_formulas in
  let formulas = Llvm.build_array_malloc base_types.formula_t (Llvm.const_int base_types.int_t numForm) "" main_bod in
  (*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*)
  let getDefn = function
      DimId(a) -> (match StringMap.find a symbols with LocalVariable(i) -> i | GlobalVariable(i) -> i | _ -> raise(TransformedAway("Error in " ^ varname ^ ": The LHS expresssions should always either have dimension length 1 or be the name of a variable in their own scope.")))
    | DimInt(1) -> 1
    | DimInt(_) -> print_endline "Non1Dim" ; raise(NotImplemented) in
  let _ = (match va.var_rows with
        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
      | DimInt(_) -> print_endline "Non1Dim" ; raise(NotImplemented)
      | DimId(a) -> (
          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 ();
          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 ();
          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
        )
    ) in
  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
  and _ = Llvm.build_store formulas (Llvm.build_struct_gep defn (var_defn_field_index Formulas) "" main_bod) main_bod
  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
  List.iteri (fun idx elem -> build_formula (varname, idx) formulas elem symbols) va.var_formulas in

(* Creates a scope object and inserts the necessary instructions into main to
    populate the var_defns, and
 * into the function specified by builder to populate the scope object. *)
let build_scope_obj
    fname (* The function name, or "globals" *)
    symbols (* The symbols to use when creating the functions *)
    vars (* The variables to build definitions and formula−functions for *)
    static_location_ptr (* The copy of the global pointer used in main *)
    var_defns_loc (* The copy of the global pointer used in the local function *)
    num_params (* How many parameters the function takes *)
    builder (* The LLVM builder for the local function *)
  =
  let cardinal = Llvm.const_int base_types.int_t (StringMap.cardinal vars) in
  let build_var_defns =
    let static_var_defns = Llvm.build_array_malloc base_types.var_defn_t cardinal (
        fname ^ "_static_var_defns") main_bod in
    let _ = Llvm.build_store static_var_defns static_location_ptr main_bod in
    let add_variable varname va (sm, count) =
      let fullname = fname ^ "_" ^ varname in
      let defn = (Llvm.build_in_bounds_gep static_var_defns [|Llvm.const_int
          base_types.int_t count|] (fullname ^ "_defn") main_bod) in
      let _ = build_var_defn defn fullname va symbols in
      (StringMap.add varname count sm, count + 1) in
    ignore (StringMap.fold add_variable vars (StringMap.empty, 0)) in

  let var_defns = Llvm.build_load var_defns_loc (fname ^ "_global_defn_ptr_loc")
      builder in
  let var_insts = Llvm.build_array_malloc base_types.var_instance_p cardinal "
      var_insts" builder in
  let scope_obj = Llvm.build_malloc base_types.extend_scope_t "scope_obj" builder in

  (*Store variable definition and instance*)
  let _ = Llvm.build_store var_defns (Llvm.build_struct_gep scope_obj (
      scope_field_type_index VarDefn) "" builder) builder in
  let _ = Llvm.build_store var_insts (Llvm.build_struct_gep scope_obj (
      scope_field_type_index VarInst) "" builder) builder in
  let _ = Llvm.build_store cardinal (Llvm.build_struct_gep scope_obj (
      scope_field_type_index VarNum) "" builder) builder in
  let _ = Llvm.build_store (Llvm.const_int base_types.int_t 0) (Llvm.
      build_struct_gep scope_obj (scope_field_type_index ScopeRefCount) "" builder)
      builder in
  let paramarray = if num_params > 0 then Llvm.build_array_malloc base_types.value_p
       (Llvm.const_int base_types.int_t num_params) "paramarray" builder else Llvm.
      const_pointer_null (Llvm.pointer_type base_types.value_p) in
  let _ = Llvm.build_store paramarray (Llvm.build_struct_gep scope_obj (
      scope_field_type_index FunctionParams) "" builder) builder in
  let copy_fn_arg i =
    let param_addr = Llvm.build_in_bounds_gep paramarray [|Llvm.const_int base_types
        .int_t i|] (fname ^ "_param_" ^ string_of_int i ^ "_loc") builder in
    ignore (Llvm.build_store (Llvm.param (StringMap.find fname function_llvalues) i)
         param_addr builder) in
  List.iter copy_fn_arg (zero_until num_params);
  let _ = Llvm.build_call (Hashtbl.find runtime_functions "null_init") [|scope_obj|]
       "" builder in
  build_var_defns ; scope_obj in
(* End of build_scope_obj *)

let build_function fname (fn_def, fn_llvalue) =
  (* Build the symbol table for this function *)
  let (local_indices, num_locals) = index_map Locals fn_def.func_body in
  let add_param (st, idx) param_name =
    let new_st = StringMap.add param_name (FunctionParameter(idx)) st in
    (new_st, idx + 1) in
  let (params_and_globals, _) = List.fold_left add_param (global_symbols, 0) (List.map snd fn_def.func_params) in
  let symbols = StringMap.fold StringMap.add local_indices params_and_globals in
  let fn_idx = match StringMap.find fname extend_fn_numbers with ExtendFunction(i) -> i | _ -> raise(LogicError(fname ^ " not in function table")) in
  let builder = Llvm.builder_at_end context (Llvm.entry_block fn_llvalue) in
  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
  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

  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

  let ret = snd fn_def.func_ret_val in
  match ret with
    Id(name) ->
    (
      match (try StringMap.find name symbols with Not_found -> raise(LogicError("Something went wrong with your semantic analysis - " ^ name ^ " not found"))) with
        LocalVariable(i) ->
        let llvm_var = Llvm.build_call getVar [|scope_obj; Llvm.const_int base_types.int_t i|] "return_variable" builder in
        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
        ignore (Llvm.build_ret llvm_retval builder)
      | _ -> print_endline (string_of_expr ret); raise(TransformedAway("Error in " ^ fname ^ ": The return value should always have been transformed into a local variable"))
    )
  | _ -> print_endline (string_of_expr ret); raise(TransformedAway("Error in " ^ fname ^ ": The return value should always have been transformed into a local variable")) in
(* End of build_function *)

(* Build the global scope object *)
let vardefn_p_p = Llvm.build_alloca base_types.var_defn_p "v_p_p" main_bod in
let global_scope_obj = build_scope_obj "globals" global_symbols globals vardefn_p_p vardefn_p_p 0 main_bod in
let _ = Llvm.build_store global_scope_obj global_scope_loc main_bod in

(*iterates over function definitions*)
StringMap.iter build_function extend_functions ;

(* Define the LLVM entry point for the program *)
let extend_entry_point = StringMap.find "main" function_llvalues in
let inp = Llvm.build_alloca base_types.value_t "input_arg" main_bod in
let _ = Llvm.build_call extend_entry_point (Array.of_list [inp]) "" main_bod in
let _ = Llvm.build_ret (Llvm.const_int base_types.int_t 0) main_bod in

base_module

let build_this ast_mapped =
let modu = (translate ast_mapped) in
let _ = Llvm_analysis.assert_valid_module modu in
modu
```

7.6 linker.ml

```
module StringSet = Set.Make(String)
let link xtndOut ast compiler outputFile =
  let tmpFilenameLL = Filename.temp_file "" ".ll"
  and tmpFilenameC = Filename.temp_file "" ".o"
  and getExterns (_,_,extern) =
    StringSet.elements
      (Ast.StringMap.fold
        (fun key value store -> StringSet.add value.Ast.extern_fn_libname store)
        extern
        StringSet.empty) in
  let tmpChan = open_out tmpFilenameLL in
  output_string tmpChan xtndOut; close_out tmpChan;
  let call1 = (String.concat " " ("llc-3.8" :: "-filetype=obj" :: tmpFilenameLL :: "-o" :: tmpFilenameC :: []))
  and call2 = (String.concat " " (compiler :: "-O -o" :: outputFile :: tmpFilenameC :: (getExterns ast) @ ["runtime.o"])) ^ " -lm" in
  let resc1 = Sys.command call1 in
  if resc1 == 0 then (
    Sys.remove tmpFilenameLL;
    let resc2 = Sys.command call2 in
      Sys.remove tmpFilenameC;
      if resc2 == 0 then () else raise Not_found
    )
  else (Sys.remove tmpFilenameC;raise Not_found)
```

7.7 main.ml

```
open Ast;;

let print_ast = ref false
let interpret_ast = ref false
let compile_ast = ref false
let link = ref false
let output = ref "./out"
let compiler = ref "gcc"

let the_ast = ref (StringMap.empty, StringMap.empty, StringMap.empty)
let just_one_please = ref false

let speclist = [
                ("-p", Arg.Set print_ast, "Print the AST");
                ("-i", Arg.Set interpret_ast, "Interpret the program");
                ("-c", Arg.Set compile_ast, "Compile the program");
                ("-l", Arg.Set link, "Link the program");
]

let usage_message = "Welcome to Extend!\n\nUsage: extend <options> <source-file>\n\nOptions are:"

let parse_ast filename =
  if !just_one_please
  then print_endline "Any files after the first one are ignored."
  else just_one_please := true ; the_ast := (Transform.create_ast filename);;

Arg.parse speclist parse_ast usage_message;
if not !just_one_please then Arg.usage speclist usage_message else ();
if !print_ast then print_endline (string_of_program !the_ast) else ();
if !interpret_ast then (Interpreter.interpret !the_ast; ()) else ();
if !compile_ast then
  let compiled = (Llvm.string_of_llmodule (Codegen.translate !the_ast))
  in
    if not (!link) then print_endline compiled
    else Linker.link compiled !the_ast !compiler !output
else ();
```

7.8 lib.c

```
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
#include<string.h>
#include<stdbool.h>
/* #include <sys/time.h> */
#include <time.h>
#include "runtime.h"

#define MAX_FILES 255
FILE *open_files[1 + MAX_FILES] = {NULL};
int open_num_files = 0;

value_p print(value_p whatever, value_p text) {
  if(!assertSingleString(text)) return new_val();
  if(!assertText(text)) return new_val();
  printf("%s", text->str->text);
  return new_val();
}

value_p printv(value_p whatever, value_p text) {
  printf("%s", text->str->text);
  return new_val();
}

value_p printd(value_p whatever, value_p text) {
  printf("%f\n", text->numericVal);
  value_p result = malloc(sizeof(struct value_t));
  return result;
}
```

```

value_p to_string(value_p val) {
    if(assertSingleNumber(val)) {
      double possible_num = val->numericVal;
      int rounded_int = (int) lrint(possible_num);
      char *converted_str;
      if (fabs(possible_num - rounded_int) < FLOAT_CUTOFF) {
        int size = snprintf(NULL, 0, "%d", rounded_int);
        converted_str = malloc(size + 1);
        sprintf(converted_str, "%d", rounded_int);
      } else {
        int size = snprintf(NULL, 0, "%f", possible_num);
        converted_str = malloc(size + 1);
        sprintf(converted_str, "%f", possible_num);
      }
      value_p result = box_value_string(new_string(converted_str));
      return result;
    }
    else if(assertSingleString(val)) return val;
    else if(val->flags == FLAG_EMPTY) {
      value_p _new = new_val();
      setString(_new, "empty", 5);
      return _new;
    }
    // If the struct does not hold a string or number, return empty?
    return new_val();
}

#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);}
FUNC(sin)
FUNC(cos)
FUNC(tan)
FUNC(acos)
FUNC(asin)
FUNC(atan)
FUNC(sinh)
FUNC(cosh)
FUNC(tanh)
FUNC(exp)
FUNC(log)
FUNC(log10)
FUNC(sqrt)
FUNC(ceil)
FUNC(fabs)
FUNC(floor)

value_p extend_get_stdin() {
  if (open_num_files + 1 > MAX_FILES) {
    return new_val();
  } else {
    open_num_files++;
    open_files[open_num_files] = stdin;
    return new_number((double) open_num_files);
  }
}
```

```c

value_p extend_get_stdout() {
  if (open_num_files + 1 > MAX_FILES) {
    return new_val();
  } else {
    open_num_files++;
    open_files[open_num_files] = stdout;
    return new_number((double) open_num_files);
  }
}

value_p extend_get_stderr() {
  if (open_num_files + 1 > MAX_FILES) {
    return new_val();
  } else {
    open_num_files++;
    open_files[open_num_files] = stderr;
    return new_number((double) open_num_files);
  }
}

value_p extend_open(value_p filename, value_p mode){
  FILE *val;
  if (   !assertSingleString(filename)
      || !assertSingleString(mode)
      || open_num_files + 1 > MAX_FILES) {
        return new_val();
  }
  val = fopen(filename->str->text, mode->str->text);
  if(val == NULL) return new_val();
  open_num_files++;
  open_files[open_num_files] = val;
  return new_number((double) open_num_files);
}

value_p extend_close(value_p file_handle) {
  if(!assertSingleNumber(file_handle)) {
    // Per the LRM this is actually supposed to crash the program.
    fprintf(stderr, "EXITING - Attempted to close something that was not a valid file pointer\n");
    exit(-1);
  }
  int fileNum = (int) file_handle->numericVal;

  if (fileNum > open_num_files || open_files[fileNum] == NULL) {
    // Per the LRM this is actually supposed to crash the program.
    fprintf(stderr, "EXITING - Attempted to close something that was not a valid file pointer\n");
    exit(-1);
  }
  fclose(open_files[fileNum]);
  open_files[fileNum] = NULL; // Empty the container for the pointer.
  return new_val(); // asssuming it was an open valid handle, close() is just supposed to return empty
}

```

```
value_p extend_read(value_p file_handle, value_p num_bytes){
  if(!assertSingleNumber(file_handle) || !assertSingleNumber(num_bytes)) return new_val();
  int max_bytes;
  int fileNum = (int) file_handle->numericVal;
  if (fileNum > open_num_files || open_files[fileNum] == NULL)  return new_val();
  FILE *f = open_files[fileNum];
  max_bytes = (int) num_bytes->numericVal;
  if (max_bytes == 0) {
    long cur_pos = ftell(f);
    fseek(f, 0, SEEK_END);
    long end_pos = ftell(f);
    fseek(f, cur_pos, SEEK_SET);
    max_bytes = end_pos - cur_pos;
  }
  char *buf = malloc(sizeof(char) * (max_bytes + 1));
  int bytes_read = fread(buf, sizeof(char), max_bytes, f);
  buf[bytes_read] = 0;
  value_p result = box_value_string(new_string(buf));
  free(buf);
  return result;
  //edge case: how to return the entire contents of the file if n == empty?
}

value_p extend_readline(value_p file_handle) {
  int i=0, buf_size = 256;
  char next_char;
  if (!assertSingleNumber(file_handle)) return new_val();
  int fileNum = (int) file_handle->numericVal;
  FILE *f = open_files[fileNum];
  if (fileNum > open_num_files || open_files[fileNum] == NULL) {
    return new_val();
  }
  char *buf = (char *) malloc (buf_size * sizeof(char));
  while ((next_char = fgetc(f)) != ’\n’) {
    buf[i++] = next_char;
    if (i == buf_size - 2) {
      buf_size *= 2;
      char *new_buf = (char *) malloc (buf_size * sizeof(char));
      memcpy(new_buf, buf, i);
      free(buf);
      buf = new_buf;
    }
  }
  buf[i] = ’\0’;
  value_p result = box_value_string(new_string(buf));
  free(buf);
  return result;
}

value_p extend_write(value_p file_handle, value_p buffer){
  if(!assertSingleNumber(file_handle) || !assertSingleString(buffer)) return new_val();
  int fileNum = (int) file_handle->numericVal;
  if (fileNum > open_num_files || open_files[fileNum] == NULL) {
    // Per the LRM this is actually supposed to crash the program.
    fprintf(stderr, "EXITING - Attempted to write to something that was not a valid file pointer\n");
    exit(-1);
  }
  fwrite(buffer->str->text, 1, buffer->str->length, open_files[fileNum]);
  // TODO: make this return empty once compiler handles Id(s)
  // RN: Use the return value to close the file
  return new_number((double) fileNum);
}

value_p extend_current_hour() {
  time_t ltime;
  struct tm info;
  ltime = time(&ltime);
  localtime_r(&ltime, &info);
  return new_number((double) info.tm_hour);
}
```

7.9 runtime.c

```
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
#include<string.h>
#include<stdbool.h>
#include "runtime.h"

void debug_print(value_p val, char *which_value) {
  char *flag_meanings[4] = {"Empty", "Number", "String", "Subrange"};
  fprintf(stderr, "------Everything you ever wanted to know about %s:------\n", which_value == NULL ? "some anonymous variable" : which_value);
  fprintf(stderr, "Memory address: %p\n", val);
  if (val == NULL) {
    fprintf(stderr, "------------Nice try asking me to dereference a null pointer\n------------");
    return;
  }
  fprintf(stderr, "Flags: %d (%s)\n", val->flags, flag_meanings[val->flags]);
  fprintf(stderr, "NumericVal: %f\n", val->numericVal);
  fprintf(stderr, "String contents: Probably safer not to check that pointer (%p) blindly\n", val->str);
  if (val->flags == FLAG_STRING && val->str != NULL) {
    fprintf(stderr, "It says it's a string and it's not a NULL pointer though, so here you go:\n");
    fprintf(stderr, "String refcount: %d\n", val->str->refs);
    fprintf(stderr, "String length: %ld\n", val->str->length);
    fprintf(stderr, "String char* memory address: %p\n", val->str->text);
    if (val->str->text == NULL) {
      fprintf(stderr, "Not going to print the contents of NULL!\n");
    } else {
      fprintf(stderr, "String char* contents:\n%s\n", val->str->text);
    }
  }
  fprintf(stderr, "Subrange contents: Probably safer not to check that pointer (%p) blindly either\n", val->subrange);
  fprintf(stderr, "-------That’s all I’ve got to say about %s:-------\n", which_value == NULL ? "some anonymous variable" : which_value);
}

void debug_print_formula(struct ExtendFormula *fdef) {
  fprintf(stderr, "-------Everything you ever wanted to know about your favorite formula:-------\n");
  fprintf(stderr, "RowStart varnum: %d %d\n", fdef->rowStart_varnum, fdef->fromFirstRow);
  fprintf(stderr, "RowEnd varnum: %d %d\n", fdef->rowEnd_varnum, fdef->toLastRow);
  fprintf(stderr, "ColStart varnum: %d %d\n", fdef->colStart_varnum, fdef->fromFirstCol);
  fprintf(stderr, "ColEnd varnum: %d %d\n", fdef->colEnd_varnum, fdef->toLastCol);
}

void debug_print_res_formula(struct ResolvedFormula *rdef) {
  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);
}

void debug_print_vardefn(struct var_defn *pdef) {
  fprintf(stderr, "-------Everything you ever wanted to know about var defn %s:-------\n", pdef->name);
  fprintf(stderr, "Row varnum: %d\n", pdef->rows_varnum);
  fprintf(stderr, "Col varnum: %d\n", pdef->cols_varnum);
  fprintf(stderr, "Num formulas: %d\n", pdef->numFormulas);
  fprintf(stderr, "Formula defs: \n");
  int i;
  for (i=0; i < pdef->numFormulas; i++) {
    debug_print_formula(pdef->formulas + i);
  }
  fprintf(stderr, "Is 1x1: %d\n", pdef->isOneByOne);
}

void debug_print_varinst(struct var_instance *inst) {
  fprintf(stderr, "-------Everything you ever wanted to know about var %s:-------\n", inst->name);
  fprintf(stderr, "Rows: %d\n", inst->rows);
  fprintf(stderr, "Cols: %d\n", inst->cols);
  fprintf(stderr, "Num formulas: %d\n", inst->numFormulas);
  fprintf(stderr, "*****Formulas:*****\n");
  int i;
  for (i = 0; i < inst->numFormulas; i++) {
    debug_print_res_formula(inst->formulas + i);
  }
  fprintf(stderr, "**** End of Formulas *** \n");
  fprintf(stderr, "~~~~~~~~Cells:~~~~~~~\n");
  for (i = 0; i < inst->rows * inst->cols; i++) {
    printf("%s[%d,%d]: Status=%d\n", inst->name, i / inst->cols, i % inst->cols, inst->status[i]);
    if (inst->status[i] == CALCULATED) {
      printf("%s[%d,%d] Value:\n", inst->name, i / inst->cols, i % inst->cols);
      debug_print(inst->values[i], inst->name);
    }
  }
  fprintf(stderr, "~~~ End of Cells: ~~~\n");
```

```
}

double setNumeric(value_p result, double val) {
  result->flags = FLAG_NUMBER;
  return (result->numericVal = val);
}

char* setString(value_p result, char *str, int length) {
  result->flags = FLAG_STRING;
  result->str = malloc(sizeof(struct string_t));
  result->str->length = length;
  return (result->str->text = str);
}

double setFlag(value_p result, double flag_num) {
  return (result->flags = flag_num);
}

int assertSingle(value_p value) {
  /* TODO: dereference 1 by 1 subrange */
  return !(value->flags == FLAG_SUBRANGE);
}

int assertSingleNumber(value_p p) {
  if (!assertSingle(p)) {
    return 0;
  }
  return (p->flags == FLAG_NUMBER);
}

int assertText(value_p my_val) {
  return (my_val->flags == FLAG_STRING);
}

int assertSingleString(value_p p) {
  if (!assertSingle(p)) {
    return 0;
  }
  return (p->flags == FLAG_STRING);
}

int assertEmpty(value_p p) {
  if (!assertSingle(p)) {
    return 0;
  }
  return (p->flags == FLAG_EMPTY);
}

value_p new_val() {
  value_p empty_val = malloc(sizeof(struct value_t));
  setFlag(empty_val, FLAG_EMPTY);
  return empty_val;
}

value_p new_number(double val) {
  value_p new_v = malloc(sizeof(struct value_t));
```

```c
  setFlag(new_v, FLAG_NUMBER);
  setNumeric(new_v, val);
  return new_v;
}

value_p new_string_go_all_the_way(char *s) {
  if (s == NULL) return new_val();
  value_p new_v = malloc(sizeof(struct value_t));
  setFlag(new_v, FLAG_STRING);
  string_p new_str = malloc(sizeof(struct string_t));
  long len = strlen(s);
  new_str->text = malloc(len+1);
  strcpy(new_str->text, s);
  new_str->length = len;
  new_str->refs = 1;
  new_v->str = new_str;
  return new_v;
}

struct ExtendScope *global_scope;

void null_init(struct ExtendScope *scope_ptr) {
  int i;
  for(i = 0; i < scope_ptr->numVars; i++)
    scope_ptr->vars[i] = NULL;
}

int getIntFromOneByOne(struct ExtendScope *scope_ptr, int varnum) {
  struct var_instance *inst = get_variable(scope_ptr, varnum);
  if (inst->rows != 1 || inst->cols != 1) {
    fprintf(stderr, "The variable you claimed (%s) was one by one is actually %d by %d.\n", inst->name, inst->rows, inst->cols);
    debug_print_varinst(inst);
    exit(-1);
  }
  value_p val = getVal(inst, 0, 0);
  if (!assertSingleNumber(val)) {
    fprintf(stderr, "The variable you claimed (%s) was a number isn’t.\n", inst->name);
    debug_print(val, inst->name);
    exit(-1);
  }
  return (int) lrint(val->numericVal);
}

struct var_instance *instantiate_variable(struct ExtendScope *scope_ptr, struct var_defn def) {
  struct var_instance *inst = malloc(sizeof(struct var_instance));
  if(def.isOneByOne) {
    inst->rows = 1;
    inst->cols = 1;
  } else {
    inst->rows = getIntFromOneByOne(scope_ptr, def.rows_varnum);
    inst->cols = getIntFromOneByOne(scope_ptr, def.cols_varnum);
  }
  // TODO: do the same thing for each FormulaFP to turn an ExtendFormula into a
```

```
      ResolvedFormula
  inst->numFormulas = def.numFormulas;
  inst->closure = scope_ptr;
  inst->name = def.name;
  int size = inst->rows * inst->cols;
  inst->values = malloc(sizeof(value_p) * size);
  memset(inst->values, 0, sizeof(value_p) * size);
  inst->status = malloc(sizeof(char) * size);
  memset(inst->status, 0, sizeof(char) * size);
  inst->formulas = malloc(sizeof(struct ResolvedFormula) * inst->numFormulas);
  //debug_print_vardefn(&def);
  //debug_print_varinst(inst);
  int i;
  for(i = 0; i < inst->numFormulas; i++) {

    // Set the formula function pointer to the pointer from the definition
    inst->formulas[i].formula = def.formulas[i].formula;

    if (def.isOneByOne) {
      inst->formulas[i].rowStart = 0;
      inst->formulas[i].rowEnd = 1;
      inst->formulas[i].colStart = 0;
      inst->formulas[i].colEnd = 1;
    } else {
      if(def.formulas[i].fromFirstRow) {
        inst->formulas[i].rowStart = 0;
      } else {
        inst->formulas[i].rowStart = getIntFromOneByOne(scope_ptr, def.formulas[i].rowStart_varnum);
        if (inst->formulas[i].rowStart < 0) {
          inst->formulas[i].rowStart += inst->rows;
        }
        if (inst->formulas[i].rowStart < 0 || inst->formulas[i].rowStart >= inst->rows) {
          //Doesn't matter, but will never get called
        }
      }
      if (def.formulas[i].isSingleRow) {
        inst->formulas[i].rowEnd = inst->formulas[i].rowStart + 1;
      } else if (def.formulas[i].toLastRow) {
        inst->formulas[i].rowEnd = inst->rows;
      } else {
        inst->formulas[i].rowEnd = getIntFromOneByOne(scope_ptr, def.formulas[i].rowEnd_varnum);
      }
      if(def.formulas[i].fromFirstCol) {
        inst->formulas[i].colStart = 0;
      } else {
        inst->formulas[i].colStart = getIntFromOneByOne(scope_ptr, def.formulas[i].colStart_varnum);
        if (inst->formulas[i].colStart < 0) {
          inst->formulas[i].colStart += inst->cols;
        }
        if (inst->formulas[i].colStart < 0 || inst->formulas[i].colStart >= inst->cols) {
          //Doesn't matter, but will never get called
        }
      }
      if (def.formulas[i].isSingleCol) {
        inst->formulas[i].colEnd = inst->formulas[i].colStart + 1;
      } else if (def.formulas[i].toLastCol) {
        inst->formulas[i].colEnd = inst->cols;
      } else {
        inst->formulas[i].colEnd = getIntFromOneByOne(scope_ptr, def.formulas[i].colEnd_varnum);
      }
    }
  }

  scope_ptr->refcount++;
  return inst;
}

struct var_instance *get_variable(struct ExtendScope *scope_ptr, int varnum) {
  if (varnum >= scope_ptr->numVars) {
    fprintf(stderr, "Runtime error: Asked for nonexistant variable number\n");
    exit(-1);
  }
  if (scope_ptr->vars[varnum] == NULL) {
    scope_ptr->vars[varnum] = instantiate_variable(scope_ptr, scope_ptr->defns[varnum]);
  }
  return scope_ptr->vars[varnum];
}

char assertInBounds(struct var_instance *defn, int r, int c) {
  return (
    r >= 0 && r < defn->rows &&
    c >= 0 && c < defn->cols
  );
}

value_p calcVal(struct var_instance *inst, int r, int c) {
  int i;
  for (i = 0; i < inst->numFormulas; i++) {
    if (
      r >= inst->formulas[i].rowStart && r < inst->formulas[i].rowEnd &&
      c >= inst->formulas[i].colStart && c < inst->formulas[i].colEnd
    ) {
      return (inst->formulas[i].formula)(inst->closure, r, c);
    }
  }
  return new_val();
}

void setRange(value_p val, struct var_instance *inst) {
  subrange_p sr = malloc(sizeof(struct subrange_t));
  sr->offsetCol = 0;
  sr->offsetRow = 0;
  sr->subrangeCol = inst->cols;
  sr->subrangeRow = inst->rows;
  sr->range = inst;
  val->subrange = sr;
  val->flags = FLAG_SUBRANGE;
}

value_p getSize(struct var_instance *inst) {
  value_p res = malloc(sizeof(struct value_t));
  setNumeric(res, 1); /*TODO*/
  return res;
}

value_p deepCopy(value_p value) {
  value_p _new = new_val();
  if(value->flags == FLAG_EMPTY) {}
  else if(value->flags == FLAG_STRING) {
    _new->flags = FLAG_STRING;
    _new->str = malloc(sizeof(struct string_t));
    memcpy(_new->str->text, value->str->text, value->str->length);
    _new->str->length = value->str->length;
  }
  else if(value->flags == FLAG_NUMBER) {
    _new->flags = FLAG_NUMBER;
    _new->numericVal = value->numericVal;
  }
  else if(value->flags == FLAG_SUBRANGE) {
    struct var_instance *v = malloc(sizeof(struct subrange_t));
    int cols = value->subrange->subrangeCol;
    int rows = value->subrange->subrangeRow;
    v->name = "COPYCAT";
    v->formulas = NULL;
    v->status = malloc(sizeof(char *) * rows * cols);
    v->values = malloc(sizeof(value_p) * rows * cols);
    v->closure = NULL;
    int i,j;
    for(i = 0; i < rows; i++) {
      for(j = 0; j < cols; j++) {
        int offset = i * rows + j;
        *(v->status + offset) = CALCULATED;
        /*TODO: eval lazzzy*/
        *(v->values + offset) = getVal(value->subrange->range, i + value->subrange->
            offsetRow, j + value->subrange->offsetCol);
      }
    }
    setRange(_new, v);
  }
  return _new;
}

value_p clone_value(value_p old_value) {
  value_p new_value = (value_p) malloc(sizeof(struct value_t));
  new_value->flags = old_value->flags;
  switch (new_value->flags) {
    case FLAG_EMPTY:
      break;
    case FLAG_NUMBER:
      new_value->numericVal = old_value->numericVal;
      break;
    case FLAG_STRING:
      new_value->str = old_value->str;
      new_value->str->refs++;
      break;
    case FLAG_SUBRANGE:
      new_value->subrange = (subrange_p) malloc(sizeof(struct subrange_t));
      memcpy(new_value->subrange, old_value->subrange, sizeof(struct subrange_t));
      new_value->subrange->range->closure->refcount++; /* Not sure about this one */
      break;
    default:
      fprintf(stderr, "clone_value(%p): Illegal value of flags: %c\n", old_value,
          new_value->flags);
      exit(-1);
      break;
  }
  return new_value;
}

void delete_string_p(string_p old_string) {
  old_string->refs--;
  if (old_string->refs == 0) {
    /* free(old_string); */
  }
}

void delete_subrange_p(subrange_p old_subrange) {
  old_subrange->range->closure->refcount--;
  free(old_subrange);
}

void delete_value(value_p old_value) {
  switch (old_value->flags) {
    case FLAG_EMPTY:
      break;
    case FLAG_NUMBER:
      break;
    case FLAG_STRING:
      delete_string_p(old_value->str); /* doesn't do anything besides decrement the
          ref count now */
      break;
    case FLAG_SUBRANGE:
      delete_subrange_p(old_value->subrange);
      break;
    default:
      fprintf(stderr, "delete_value(%p): Illegal value of flags: %c\n", old_value,
          old_value->flags);
      exit(-1);
      break;
  }
}

value_p getVal(struct var_instance *inst, int r, int c) {
  /* If we're going to return new_val() then we have to
   * do clone_value(). Otherwise the receiver won't know
   * whether or not they can free the value_p they get back.
   * I think this should return, dangerously, return NULL if it's
   * invalid, and the callers will have to be careful to check the value.
   * The alternative is to always clone_value - safer, but much slower
   * and makes our memory issues even bigger.
   * Right now there are only a few places that call this. */

  if(!assertInBounds(inst, r, c)) return NULL;
  int cell_number = r * inst->cols + c;
  char cell_status = inst->status[cell_number];
  switch(cell_status) {
    case NEVER_EXAMINED:
      inst->status[cell_number] = IN_PROGRESS;
      inst->values[cell_number] = calcVal(inst, r, c);
      inst->status[cell_number] = CALCULATED;
      break;
    case IN_PROGRESS:
      fprintf(stderr, "EXITING - Circular reference in %s[%d,%d]\n", inst->name, r, c);
      exit(-1);
      break;
    case CALCULATED:
      if (inst->values[cell_number] == NULL) {
        fprintf(stderr, "Supposedly, %s[%d,%d] was already calculated, but there is a null pointer there.\n", inst->name, r, c);
        fprintf(stderr, "Attempting to print contents of the variable instance where this occurred:\n");
        fflush(stdout);
        debug_print_varinst(inst);
        exit(-1);
      }
      break;
    default:
      fprintf(stderr, "Unrecognized cell status %d (row %d, col %d)!\n", cell_status, r, c);
      fprintf(stderr, "Attempting to print contents of the variable instance where this occurred:\n");
      fflush(stdout);
      debug_print_varinst(inst);
      exit(-1);
      break;
  }
  return inst->values[cell_number];
//  char *status = inst->status + offset;
//  value_p return_val;
//  if(*status & IN_PROGRESS) {
//    /* TODO: Circular dependency. Possibly throw? */
//    return_val = new_val();
//  } else if ((~(*status)) & CALCULATED) { /* value not calculated */
//    value_p val = calcVal(inst, x, y);
//    inst->values[offset] = val;
//    *status = (*status && !IN_PROGRESS) | CALCULATED;
//    return_val = val;
//  } else {
//    return_val = inst->values[offset];
//  }
//  while(return_val->flags == FLAG_SUBRANGE && return_val->subrange->subrangeRow == 1 && return_val->subrange->subrangeCol == 1) {
```

```
//    return_val = getVal(return_val->subrange->range, return_val->subrange->offsetRow, return_val->subrange->offsetCol);
//  }
// // debug_print_varinst(inst);
//  return return_val;
}
```

7.10 stdlib.xtnd

```
extern "stdlib.o" {
  extend_get_stdin();
  extend_get_stdout();
  extend_get_stderr();
  extend_readline(file_handle);
  extend_write(file_handle, str);
  extend_current_hour();
  to_string(x);
}

global STDIN := extend_get_stdin();
global STDOUT := extend_get_stdout();
global STDERR := extend_get_stderr();
```

8. Tests and Output

helloworld.xtnd

```
extern "stdlib.o" {
  print(a,b);
    printv(a,b);
}

[1,1] main(args) {
  foo := printv(1,"Hello World\n") -> 0;
  return foo;
}
```

helloworld.xtnd - Expected Output

```
Hello World
```

test-access-cell.xtnd

```
extern "stdlib.o" {
  print(a,b);
  to_string(a);
}

main([1,n] args) {
  [2,2] foo := "string";
  bar := foo[1,1];
  return print(1,to_string(bar)) -> print(1, "\n") -> 0;
}
```

test-access-cell.xtnd - Expected Output

```
string
```

test-access-column-cell.xtnd

```
extern "stdlib.o" {
  print(a,b);
  to_string(a);
}

main([1,n] args) {
  [4,1] foo := "string";
  return print(1,to_string( foo[1,0])+"\n") -> 0;
}
```

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

```
string
```

test-access-hashtag-multi-dim.xtnd

```
extern "stdlib.o" {
  print(a,b);
  to_string(a);
}

main([1,n] args) {
  [4,4] foo := "string";
  return print(1,to_string( #foo)+"\n") -> 0;
}
```

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

```
string
```

test-access-hashtag-single-dim.xtnd

```
extern "stdlib.o" {
  print(a,b);
  to_string(a);
}

main([1,n] args) {
  [1,1] foo := "string";
  return print(1,to_string( #foo)+"\n") -> 0;
}
```

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

```
string
```

test-access-relative-range.xtnd

```
extern "stdlib.o" {
  print(a,b);
  to_string(a);
}

main([1,n] args) {
  [4,4] foo := "string";
  return print(1,to_string( foo[,[1]])+"\n") -> 0;
}
```

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

```
string
```

test-acos.xtnd

```
extern "stdlib.o" {
  extend_acos(a);
  printd(a,b);
}

[1,1] main(args) {
  return printd(1, extend_acos(0.0)) -> 0;
}
```

test-acos.xtnd - Expected Output

```
1.570796
```

test-addition.xtnd

```
extern "stdlib.o" {
  print(a,b);
  to_string(a);
}

main([1,1] args){
  return print(1,to_string( 5 + 7)+"\n") -> 0;
}
```

test-addition.xtnd - Expected Output

```
12
```

test-addition-empty.xtnd

```
extern "stdlib.o" {
  print(a,b);
  to_string(a);
}

main([1,1] args){
  return print(1,to_string( empty + 5)+"\n") -> 0;
}
```

test-addition-empty.xtnd - Expected Output

```
empty
```

test-asin.xtnd

```
extern "stdlib.o" {
  extend_asin(a);
  printd(a,b);
}

[1,1] main([1,n] args) {
  return printd(1, extend_asin(0.5)) -> 0;
}
```

test-asin.xtnd - Expected Output

```
0.523599
```

test-atan.xtnd

```
extern "stdlib.o" {
  extend_atan(a);
  printd(a,b);
}

[1,1] main([1,n] args) {
  return printd(1, extend_atan(45.0)) -> 0;
}
```

test-atan.xtnd - Expected Output

```
1.548578
```

test-basic-func.xtnd

```
extern "stdlib.o" {
  print(a,b);
  to_string(a);
}

main([1,n] args) {
  foo := 2;
  bar := 3;
  foobar := foo + bar;
  return print(1,to_string( 0)+"\n") -> 0;
}
```

test-basic-func.xtnd - Expected Output

```
0
```

test-bitwise-and.xtnd

```
extern "stdlib.o" {
  print(a,b);
  to_string(a);
}

main([1,1] args){
  return print(1,to_string( 23 & 12)+"\n") -> 0;
}
```

test-bitwise-and.xtnd - Expected Output

```
4
```

test-bitwise-and-empty.xtnd

```
extern "stdlib.o" {
  print(a,b);
  to_string(a);
}

main([1,1] args){
  return print(1,to_string( empty & 4)+"\n") -> 0;
}
```

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

```
empty
```

test-bitwise-left.xtnd

```
extern "stdlib.o" {
  print(a,b);
  to_string(a);
}

main([1,1] args){
  return print(1,to_string( 14 << 2)+"\n") -> 0;
}
```

test-bitwise-left.xtnd - Expected Output

```
56
```

test-bitwise-left-empty.xtnd

```
extern "stdlib.o" {
  print(a,b);
  to_string(a);
}

main([1,1] args){
  return print(1,to_string( empty >> 1)+"\n") -> 0;
}
```

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

```
empty
```

test-bitwise-or.xtnd

```
extern "stdlib.o" {
  print(a,b);
  to_string(a);
}

main([1,1] args){
  return print(1,to_string( 14 | 12)+"\n") -> 0;
}
```

test-bitwise-or.xtnd - Expected Output

```
14
```

test-bitwise-or-empty.xtnd

```
extern "stdlib.o" {
  print(a,b);
  to_string(a);
}

main([1,1] args){
  return print(1,to_string( empty | 2)+"\n") -> 0;
}
```

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

```
empty
```

test-bitwise-right.xtnd

```
extern "stdlib.o" {
  print(a,b);
  to_string(a);
}

main([1,1] args){
  return print(1,to_string( 12 >> 2)+"\n") -> 0;
}
```

test-bitwise-right.xtnd - Expected Output

```
3
```

test-bitwise-right-empty.xtnd

```
extern "stdlib.o" {
  print(a,b);
  to_string(a);
}

main([1,1] args){
  return print(1,to_string( empty >> 2)+"\n") -> 0;
}
```

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

```
empty
```

test-bitwise-xor.xtnd

```
extern "stdlib.o" {
  print(a,b);
  to_string(a);
}

main([1,1] args){
  return print(1,to_string( 14 ^ 12)+"\n") -> 0;
}
```

test-bitwise-xor.xtnd - Expected Output

```
2
```

test-bitwise-xor-empty.xtnd

```
extern "stdlib.o" {
  print(a,b);
  to_string(a);
}

main([1,1] args){
  return print(1,to_string( empty ^ 2)+"\n") -> 0;
}
```

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

```
empty
```

test-boolean-equals.xtnd

```
extern "stdlib.o" {
  print(a,b);
  to_string(a);
}

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

test-boolean-equals.xtnd - Expected Output

```
0
```

test-boolean-equals-both-empty.xtnd

```
extern "stdlib.o" {
  print(a,b);
  to_string(a);
}

main([1,1] args){
  return print(1,to_string( empty == empty)+"\n") -> 0;
}
```

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

```
1
```

test-boolean-equals-harder.xtnd

```
extern "stdlib.o" {
  printv(a,b);
  printd(a,b);
}

main([1,1] args){
  return
    printv(1, "True cases for ==\n") ->
    printd(1,  (5 == 5)) ->
    printd(1,  (5 == 5.0)) ->
    printd(1,  (0.5 == 5e-1)) ->
    printd(1,  (50 == 5e1)) ->
    printd(1,  2 + 2 == 4) ->
    printd(1,  "foo" == "foo") ->
    printd(1,  "" == "") ->
    printd(1,  empty == empty) ->
    printd(1,  empty == !empty) ->
    printd(1,  !"foo" == !"bar") ->
    printd(1, (2 ? 3 : 4) == ("foo" ? 3 : "not 4") ) ->

    printv(1, "\nFalse cases for ==\n") ->
    printd(1,  (5 == 6)) ->
    printd(1,  (5 == 5.01)) ->
    printd(1,  (0.5 == 5e-2)) ->
    printd(1,  (50 == 5e2)) ->
    printd(1,  2 + 2 == 5) ->
    printd(1,  "foo" == "bar") ->
    printd(1,  "" == "foo") ->
    printd(1,  "" == empty) ->
    printd(1,  2 == empty) ->
    printd(1,  empty == 2) ->
    printd(1, (2 ? 3 : 4) == ("foo" ? "not 3" : 4) ) ->

    printv(1, "\nTrue cases for !=\n") ->
    printd(1,  (5 != 6)) ->
    printd(1,  (5 != 5.01)) ->
    printd(1,  (0.5 != 5e-2)) ->
    printd(1,  (50 != 5e2)) ->
    printd(1,  2 + 2 != 5) ->
    printd(1,  "foo" != "bar") ->
    printd(1,  "" != "foo") ->
    printd(1,  "" != empty) ->
    printd(1,  2 != empty) ->
    printd(1,  empty != 2) ->
    printd(1, (2 ? 3 : 4) != ("foo" ? "not 3" : 4) ) ->

    printv(1, "\nFalse cases for !=\n") ->
    printd(1,  (5 != 5)) ->
    printd(1,  (5 != 5.0)) ->
    printd(1,  (0.5 != 5e-1)) ->
    printd(1,  (50 != 5e1)) ->
    printd(1,  2 + 2 != 4) ->
    printd(1,  "foo" != "foo") ->
    printd(1,  "" != "") ->
    printd(1,  empty != empty) ->
    printd(1,  empty != !empty) ->
    printd(1,  !"foo" != !"bar") ->
    printd(1, (2 ? 3 : 4) != ("foo" ? 3 : "not 4") ) ->

    0;
}
```

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

```
True cases for ==
1.000000
1.000000
1.000000
1.000000
1.000000
1.000000
1.000000
1.000000
1.000000
1.000000
1.000000

False cases for ==
0.000000
0.000000
0.000000
0.000000
0.000000
0.000000
0.000000
0.000000
0.000000
0.000000
0.000000

True cases for !=
1.000000
1.000000
1.000000
```

```
1.000000
1.000000
1.000000
1.000000
1.000000
1.000000
1.000000
1.000000

False cases for !=
0.000000
0.000000
0.000000
0.000000
0.000000
0.000000
0.000000
0.000000
0.000000
0.000000
0.000000
```

test-boolean-equals-one-empty.xtnd

```
extern "stdlib.o" {
  print(a,b);
  to_string(a);
}

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

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

```
0
```

test-boolean-logical-not-equals.xtnd

```
extern "stdlib.o" {
  print(a,b);
  to_string(a);
}

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

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

```
1
```

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

```
extern "stdlib.o" {
  print(a,b);
  to_string(a);
}
```

```

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

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

```
0
```

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

```
extern "stdlib.o" {
  print(a,b);
  to_string(a);
}

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

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

```
1
```

test-calling-func-from-import.xtnd

```
extern "stdlib.o" {
  print(a,b);
  to_string(a);
}

import "../../samples/gcd_func.xtnd";

main([1,n] args){
  return print(1,to_string( gcd(70, 55))+"\n") -> 0;
}
```

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

```
5
```

test-ceil.xtnd

```
extern "stdlib.o" {
  extend_ceil(a);
  printd(a,b);
}

[1,1] main([1,n] args) {
  return printd(1, extend_ceil(10.45)) -> 0;
}
```

test-ceil.xtnd - Expected Output

```
11.000000
```

test-cos.xtnd

```
extern "stdlib.o" {
  extend_cos(a);
  printd(a,b);
}

[1,1] main([1,n] args) {
  return printd(1, extend_cos(45.0)) -> 0;
}
```

test-cos.xtnd - Expected Output

```
0.525322
```

test-cosh.xtnd

```
extern "stdlib.o" {
  extend_cosh(a);
  printd(a,b);
}

[1,1] main([1,n] args) {
  return printd(1, extend_cosh(45.0)) -> 0;
}
```

test-cosh.xtnd - Expected Output

```
17467135528742547456.000000
```

test-division.xtnd

```
extern "stdlib.o" {
  print(a,b);
  to_string(a);
}

main([1,1] args){
  /* Should evaluate to 4 */
  return print(1,to_string( 20 / 5)+"\n") -> 0;
}
```

test-division.xtnd - Expected Output

```
4
```

test-division-empty.xtnd

```
extern "stdlib.o" {
  print(a,b);
  to_string(a);
}

main([1,n] args){
  /* Should return empty */
  return print(1,to_string( empty / 5)+"\n") -> 0;
}
```

test-division-empty.xtnd - Expected Output

```
empty
```

test-exp.xtnd

```
extern "stdlib.o" {
  extend_exp(a);
  printd(a,b);
}

[1,1] main([1,n] args) {
  return printd(1, extend_exp(2.0)) -> 0;
}
```

test-exp.xtnd - Expected Output

```
7.389056
```

test-fabs.xtnd

```
extern "stdlib.o" {
  extend_fabs(a);
  printd(a,b);
}

[1,1] main([1,n] args) {
  return printd(1, extend_fabs(-45.0)) -> 0;
}
```

test-fabs.xtnd - Expected Output

```
45.000000
```

test-file-close.xtnd

```
extern "stdlib.o" {
  extend_open(a,b);
  extend_close(a);
  print(a,b);
}

[1,1] main(args){
  return extend_close(extend_open("testcases/assets/test_file.txt", "r")) -> print(1,"Made it this far\n") -> 0;
}
```

test-file-close.xtnd - Expected Output

```
Made it this far
```

test-file-read.xtnd

```
extern "stdlib.o" {
  extend_open(a,b);
  extend_read(a,b);
  extend_close(a);
  print(a,b);
}

[1,1] main(args){
  return print(1, extend_read(extend_open("testcases/assets/test_file.txt", "r"),5)) -> 0;
}
```

test-file-read.xtnd - Expected Output

```
This
```

test-file-write.xtnd

```
extern "stdlib.o" {
  extend_open(a,b);
  extend_write(a,b);
  extend_close(a);
  print(a,b);
}

[1,1] main(args){
  return extend_close(extend_write(extend_open("testcases/assets/test_file_write.out",
       "w"), "Hello"))
          -> print(1,"Made it this far\n")
          -> 0;
}
```

test-file-write.xtnd - Expected Output

```
Made it this far
```

test-floor.xtnd

```
extern "stdlib.o" {
  extend_floor(a);
  printd(a,b);
}

[1,1] main([1,n] args) {
  return printd(1, extend_floor(10.45)) -> 0;
}
```

test-floor.xtnd - Expected Output

```
10.000000
```

test-func-params.xtnd

```
extern "stdlib.o" {
  print(a,b);
  to_string(a);
}

main([1,n] args) {
  return print(1,to_string( foo("string"))+"\n") -> 0;
}
[1,1] foo([1,1] arg) {
  return arg;
}
```

test-func-params.xtnd - Expected Output

```
string
```

test-func-params-omit-dim.xtnd

```
extern "stdlib.o" {
  print(a,b);
  to_string(a);
}

main([1,n] args) {
  return print(1,to_string( foo("string"))+"\n") -> 0;
}
foo([1,1] arg) {
  return arg;
}
```

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

```
string
```

test-global-hello.xtnd

```
extern "stdlib.o" {
  print(a,b);
    printv(a,b);
}

bar() {
  foo := 5;
  return 2;
}

global foo := printv(1,"Hello Globals!\n") -> 0;

[1,1] main(args) {
  return foo;
}
```

test-global-hello.xtnd - Expected Output

```
Hello Globals!
```

test-global-masking.xtnd

```
extern "stdlib.o" {
  print(a,b);
    printv(a,b);
}

bar() {
  foo := 5;
  return 2;
}

global foo := printv(1,"Hello Globals!\n") -> 0;

[1,1] main(args) {
  foo := printv(1,"Hello Locals!\n") -> 0;
  return foo;
}
```

test-global-masking.xtnd - Expected Output

```
Hello Locals!
```

test-globals-between-imports.xtnd

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

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

```
Hello
```

test-greater-than.xtnd

```
extern "stdlib.o" {
  print(a,b);
  to_string(a);
}

main([1,1] args){
  return print(1,to_string( 6 > 5)+"\n") -> 0;
}
```

test-greater-than.xtnd - Expected Output

```
1
```

test-greater-than-empty.xtnd

```
extern "stdlib.o" {
  print(a,b);
  to_string(a);
}

main([1,1] args){
  return print(1,to_string( empty > 5)+"\n") -> 0;
}
```

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

```
empty
```

test-greater-than-or-equal.xtnd

```
extern "stdlib.o" {
  print(a,b);
  to_string(a);
}

main([1,1] args){
  return print(1,to_string( 7 >= 7)+"\n") -> 0;
}
```

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

```
1
```

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

```
extern "stdlib.o" {
  print(a,b);
  to_string(a);
}

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

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

```
empty
```

test-less-than.xtnd

```
extern "stdlib.o" {
  print(a,b);
  to_string(a);
}

main([1,1] args){
  return print(1,to_string( 6 < 7)+"\n") -> 0;
}
```

test-less-than.xtnd - Expected Output

```
1
```

test-less-than-empty.xtnd

```
extern "stdlib.o" {
  print(a,b);
  to_string(a);
}

main([1,1] args){
  return print(1,to_string( empty > 5)+"\n") -> 0;
}
```

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

```
empty
```

test-less-than-or-equal.xtnd

```
extern "stdlib.o" {
  print(a,b);
  to_string(a);
}

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

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

```
0
```

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

```
extern "stdlib.o" {
  print(a,b);
  to_string(a);
}

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

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

```
empty
```

test-log.xtnd

```
extern "stdlib.o" {
  extend_log(a);
  printd(a,b);
}

[1,1] main([1,n] args) {
  return printd(1, extend_log(10.0)) -> 0;
}
```

test-log.xtnd - Expected Output

```
2.302585
```

test-log10.xtnd

```
extern "stdlib.o" {
  extend_log10(a);
  printd(a,b);
}

[1,1] main([1,n] args) {
  return printd(1, extend_log10(100.0)) -> 0;
}
```

test-log10.xtnd - Expected Output

```
2.000000
```

test-logical-and.xtnd

```
extern "stdlib.o" {
  print(a,b);
  to_string(a);
}

main([1,1] args){
  return print(1,to_string( 1 && 6)+"\n") -> 0;
}
```

test-logical-and.xtnd - Expected Output

```
1
```

test-logical-and-empty.xtnd

```
extern "stdlib.o" {
  print(a,b);
  to_string(a);
}

main([1,1] args){
  return print(1,to_string( empty && 1)+"\n") -> 0;
}
```

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

```
empty
```

test-logical-not.xtnd

```
extern "stdlib.o" {
  print(a,b);
  to_string(a);
}

main([1,1] args){
  return print(1,to_string( !5)+"\n") -> 0;
}
```

test-logical-not.xtnd - Expected Output

```
0
```

test-logical-not-empty.xtnd

```
extern "stdlib.o" {
  print(a,b);
  to_string(a);
}

main([1,1] args){
  return print(1,to_string( !empty)+"\n") -> 0;
}
```

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

```
empty
```

test-logical-or.xtnd

```
extern "stdlib.o" {
  print(a,b);
  to_string(a);
}

main([1,1] args){
  return print(1,to_string( 5 || 6)+"\n") -> 0;
}
```

test-logical-or.xtnd - Expected Output

```
1
```

test-logical-or-empty.xtnd

```
extern "stdlib.o" {
  print(a,b);
  to_string(a);
}

main([1,1] args){
  return print(1,to_string( empty || 4)+"\n") -> 0;
}
```

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

```
empty
```

test-modulo.xtnd

```
extern "stdlib.o" {
  print(a,b);
  to_string(a);
}

main([1,n] args){
  /* Should return 1 */
  return print(1,to_string( 5 % 4)+"\n") -> 0;
}
```

test-modulo.xtnd - Expected Output

```
1
```

test-modulo-empty.xtnd

```
extern "stdlib.o" {
  print(a,b);
  to_string(a);
}

main([1,n] args){
  /* Should return empty */
  return print(1,to_string( empty % 5)+"\n") -> 0;
}
```

test-modulo-empty.xtnd - Expected Output

```
empty
```

test-multiple-imports.xtnd

```
import "../../testcases/assets/string.xtnd";
import "../../testcases/assets/string.xtnd";
```

test-multiple-imports.xtnd - Expected Output

```
Hello
```

test-multiplication.xtnd

```
extern "stdlib.o" {
  print(a,b);
  to_string(a);
}

main([1,n] args){
  /* Should evaluate to 35 */
  return print(1,to_string( 7 * 5)+"\n") -> 0;
}
```

test-multiplication.xtnd - Expected Output

```
35
```

test-multiplication-empty.xtnd

```
extern "stdlib.o" {
  print(a,b);
  to_string(a);
}

main([1,n] args){
  /* Should evaluate to empty */
  return print(1,to_string( empty * 5)+"\n") -> 0;
}
```

test-multiplication-empty.xtnd - Expected Output

```
empty
```

test-parse-error.xtnd

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

test-parse-error.xtnd - Expected Output

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

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

```
[1,1] main(args){
/* This is a comment spanning multiple lines.


















20 of them, in fact. */
  foo := 5/5;
  bar := $$$$
  return foo;
}
```

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

```
Syntax error in "./testcases/inputs_regression/
    test_parse_error_after_multiline_comment.xtnd": Invalid character: $
Line 23 at character 10
```

test-parse-error-comment.xtnd

```
[1,1] main(args){
  foo := 5/5;
  /* Test comment */ foo := 5$5;
  return foo;
}
```

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

```
Syntax error in "./testcases/inputs_regression/test_parse_error_comment.xtnd": Invalid
     character: $
Line 3 at character 30
```

test-parse-error-missing-semicolon.xtnd

```
main([1,1] args){
  x := switch() {
    case 1 > 2: 100;
    case 3 > 0: 200
  };
  return printf(1,toString(x)+"\n") -> 0;
}
```

test-parse-error-missing-semicolon.xtnd - Expected Output

```
Syntax error in "./testcases/inputs_regression/test_parse_error_missing_semicolon.xtnd
    ":
Line 5 at character 2
```

test-parse-error-newlines.xtnd

```
[1,1] main(args){
  foo := 5/5;
  bar := "\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n"$5;
  return foo;
}
```

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

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

test-parse-error-string.xtnd

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

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

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

test-power.xtnd

```
extern "stdlib.o" {
  print(a,b);
  to_string(a);
}

main([1,n] args){
  /* Should return 216 */
  return print(1,to_string( 6**3)+"\n") -> 0;
}
```

test-power.xtnd - Expected Output

```
216
```

test-power-empty.xtnd

```
extern "stdlib.o" {
  print(a,b);
  to_string(a);
}

main([1,n] args){
  /* Should return empty */
  return print(1,to_string( empty**5)+"\n") -> 0;
}
```

test-power-empty.xtnd - Expected Output

```
empty
```

test-print-empty.xtnd

```
extern "stdlib.o" {
  print(a,b);
  to_string(a);
}

main([1,n] args) {
  foo := empty;
  return print(1,to_string( foo)+"\n") -> 0;
}
```

test-print-empty.xtnd - Expected Output

```
empty
```

test-print-nums.xtnd

```
extern "stdlib.o" {
  print(a,b);
  to_string(a);
}

main([1,n] args) {
  foo := 1;
  return print(1,to_string( foo)+"\n") -> 0;
}
```

test-print-nums.xtnd - Expected Output

```
1
```

test-print-str.xtnd

```
extern "stdlib.o" {
  print(a,b);
  to_string(a);
}

main([1,n] args) {
  foo := "string";
  return print(1,to_string( foo)+"\n") -> 0;
}
```

test-print-str.xtnd - Expected Output

```
string
```

test-ref-between-globals.xtnd

```
extern "stdlib.o" {
  print(a,b);
  to_string(a);
}

global [2,2] foo;
global [2,2] bar;
main([1,n] args) {
  foo := 1;
  bar := foo;
  return print(1,to_string( bar)+"\n") -> 0;
}
```

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

```
1
```

test-short-circuiting-and.xtnd

```
extern "stdlib.o" {
  print(a,b);
}

main([1,1] args){
  return 0 && print(1,"FAIL\n") -> print(1,"PASS\n") -> 0;
}
```

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

```
PASS
```

test-short-circuiting-and2.xtnd

```
extern "stdlib.o" {
  print(a,b);
}

main([1,1] args){
  return 1 && print(1,"PASS1\n") -> print(1,"PASS2\n") -> 0;
}
```

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

```
PASS1
PASS2
```

test-short-circuiting-or.xtnd

```
extern "stdlib.o" {
  print(a,b);
}

main([1,1] args){
  return 0 || print(1,"PASS1\n") -> print(1,"PASS2\n") -> 0;
}
```

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

```
PASS1
PASS2
```

test-short-circuiting-or2.xtnd

```
extern "stdlib.o" {
  print(a,b);
}

main([1,1] args){
  return 1 || print(1,"FAIL\n") -> print(1,"PASS\n") -> 0;
}
```

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

```
PASS
```

test-sin.xtnd

```
extern "stdlib.o" {
  extend_sin(a);
  printd(a,b);
}

[1,1] main([1,n] args) {
  return printd(1, extend_sin(45.0)) -> 0;
}
```

test-sin.xtnd - Expected Output

```
0.850904
```

test-sin-through-function.xtnd

```
extern "stdlib.o" {
  extend_sin(a);
  printd(a,b);
}

internal_sin(x,y,z) {
  return extend_sin(z);
}

[1,1] main([1,n] args) {
  return printd(1, internal_sin(1,2,45.0)) -> 0;
}
```

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

```
0.850904
```

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

```
extern "stdlib.o" {
  extend_sin(a);
  printd(a,b);
}

global theta := 45.0;

internal_sin(x,y,z) {
  return extend_sin(z);
}

[1,1] main([1,n] args) {
  return printd(1, internal_sin(1,2,theta)) -> 0;
}
```

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

```
0.850904
```

test-single-import.xtnd

```
extern "stdlib.o" {
  print(a,b);
  to_string(a);
}

import "../../samples/gcd_func.xtnd";

main([1,n] args) {
  return print(1, to_string(gcd(70, 55)) + "\n") -> 0;
}
```

test-single-import.xtnd - Expected Output

```
5
```

test-sinh.xtnd

```
extern "stdlib.o" {
  extend_sinh(a);
  printd(a,b);
}

[1,1] main([1,n] args) {
  return printd(1, extend_sinh(45.0)) -> 0;
}
```

test-sinh.xtnd - Expected Output

```
17467135528742547456.000000
```

test-sqrt.xtnd

```
extern "stdlib.o" {
  extend_sqrt(a);
  printd(a,b);
}

[1,1] main([1,n] args) {
  return printd(1, extend_sqrt(9.0)) -> 0;
}
```

test-sqrt.xtnd - Expected Output

```
3.000000
```

test-string-concatenation.xtnd

```
extern "stdlib.o" {
    printv(a,b);
}

[1,1] main(args) {
  foo :=
    printv(1,"Hello " + "World\n") ->
    printv(1,"Hello " + "World" + "\n") ->
    printv(1,("Hello " + "World") + ("" + "\n")) ->
     0;
  return foo;
}
```

test-string-concatenation.xtnd - Expected Output

```
Hello World
Hello World
Hello World
```

test-subtraction.xtnd

```
extern "stdlib.o" {
  print(a,b);
  to_string(a);
}

main([1,1] args){
  return print(1,to_string( 7 - 5)+"\n") -> 0;
}
```

test-subtraction.xtnd - Expected Output

```
2
```

test-subtraction-empty.xtnd

```
extern "stdlib.o" {
  print(a,b);
  to_string(a);
}

main([1,1] args){
  return print(1,to_string( empty - 2)+"\n") -> 0;
}
```

test-subtraction-empty.xtnd - Expected Output

```
empty
```

test-switch-v1.xtnd

```
extern "stdlib.o" {
  print(a,b);
  to_string(a);
}

main([1,1] args){
  x := switch(1) {
    case 1: 100;
    case 2: 200;
    default: 300;
  };
  return print(1,to_string(x)+"\n") -> 0;
}
```

test-switch-v1.xtnd - Expected Output

```
100
```

test-switch-v10.xtnd

```
extern "stdlib.o" {
  printd(a,b);
  to_string(a);
}

main([1,1] args){
  x := switch {
    case 0: 100;
    case "also true": 200;
    default: 99;
  };
  return printd(1,x) -> 0;
}
```

test-switch-v10.xtnd - Expected Output

```
200.000000
```

test-switch-v11.xtnd

```
extern "stdlib.o" {
  printd(a,b);
  to_string(a);
}

main([1,1] args){
  x := switch {
    case 0: 100;
    default: 99;
  };
  return printd(1,x) -> 0;
}
```

test-switch-v11.xtnd - Expected Output

```
99.000000
```

test-switch-v2.xtnd

```
extern "stdlib.o" {
  print(a,b);
  to_string(a);
}

main([1,1] args){
  x := switch(2) {
    case 1: 100;
    case 2: 200;
    default: 300;
  };
  return print(1,to_string(x)+"\n") -> 0;
}
```

test-switch-v2.xtnd - Expected Output

```
200
```

test-switch-v3.xtnd

```
extern "stdlib.o" {
  print(a,b);
  to_string(a);
}

main([1,1] args){
  x := switch(3) {
    case 1: 100;
    case 2: 200;
    default: 300;
  };
  return print(1,to_string(x)+"\n") -> 0;
}
```

test-switch-v3.xtnd - Expected Output

```
300
```

test-switch-v4.xtnd

```
extern "stdlib.o" {
  print(a,b);
  to_string(a);
}

main([1,1] args){
  x := switch(2) {
    case 1, 2: 100;
    default: 300;
  };
  return print(1,to_string(x)+"\n") -> 0;
}
```

test-switch-v4.xtnd - Expected Output

```
100
```

test-switch-v5.xtnd

```
extern "stdlib.o" {
  print(a,b);
  to_string(a);
}

main([1,1] args){
  x := switch(3) {
    case 1, 2: 100;
    default: 300;
  };
  return print(1,to_string(x)+"\n") -> 0;
}
```

test-switch-v5.xtnd - Expected Output

```
300
```

test-switch-v6.xtnd

```
extern "stdlib.o" {
  print(a,b);
  to_string(a);
}

main([1,1] args){
  x := switch(3) {
    case 1, 2: 100;
    case 0, 3: 200;
    default: 300;
  };
  return print(1,to_string(x)+"\n") -> 0;
}
```

test-switch-v6.xtnd - Expected Output

```
200
```

test-switch-v7.xtnd

```
extern "stdlib.o" {
  print(a,b);
  to_string(a);
}

main([1,1] args){
  x := switch(4) {
    case 1, 2: 100;
    case 0, 3: 200;
  };
  return print(1,to_string(x)+"\n") -> 0;
}
```

test-switch-v7.xtnd - Expected Output

```
empty
```

test-switch-v8.xtnd

```
extern "stdlib.o" {
  print(a,b);
  to_string(a);
}

main([1,1] args){
  x := switch() {
    case 1 > 2: 100;
    case 3 > 0: 200;
  };
  return print(1,to_string(x)+"\n") -> 0;
}
```

test-switch-v8.xtnd - Expected Output

```
200
```

test-switch-v9.xtnd

```
extern "stdlib.o" {
  printd(a,b);
  to_string(a);
}

main([1,1] args){
  x := switch {
    case "true": 100;
    case "also true": 200;
  };
  return printd(1,x) -> 0;
}
```

test-switch-v9.xtnd - Expected Output

```
100.000000
```

test-tan.xtnd

```
extern "stdlib.o" {
  extend_tan(a);
  printd(a,b);
}

[1,1] main([1,n] args) {
  return printd(1, extend_tan(45.0)) -> 0;
}
```

test-tan.xtnd - Expected Output

```
1.619775
```

test-tanh.xtnd

```
extern "stdlib.o" {
  extend_tanh(a);
  printd(a,b);
}

[1,1] main([1,n] args) {
  return printd(1, extend_tanh(45.0)) -> 0;
}
```

test-tanh.xtnd - Expected Output

```
1.000000
```

test-ternary-conditional.xtnd

```
extern "stdlib.o" {
  print(a,b);
  to_string(a);
}

main([1,1] args){
  return print(1,to_string(5 ? 2 : 3) + "\n") -> 0;
}
```

test-ternary-conditional.xtnd - Expected Output

```
2
```

test-ternary-conditional-empty.xtnd

```
extern "stdlib.o" {
  print(a,b);
  to_string(a);
}

main([1,1] args){
  return print(1,to_string( empty ? 5 : 6)+"\n") -> 0;
}
```

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

```
empty
```

test-unary-negation.xtnd

```
extern "stdlib.o" {
  print(a,b);
  to_string(a);
}

main([1,n] args){
  /* Should return -33 */
  return print(1,to_string( -33)+"\n") -> 0;
}
```

test-unary-negation.xtnd - Expected Output

```
-33
```

test-unary-negation-empty.xtnd

```
extern "stdlib.o" {
  print(a,b);
  to_string(a);
}

main([1,n] args){
  return print(1,to_string( -empty)+"\n") -> 0;
}
```

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

```
empty
```

9. Git Logs

```
*   commit 8297f339f2e0d82ff2aa0b48840103441af0e7a5
|\  Merge: d87b73c 37470e9
| | Author: Jared Samet <jared.samet@aya.yale.edu>
| | Date:   Wed Dec 14 12:01:47 2016 -0500
| |
| |     Merge pull request #85 from ExtendLang/put-lt-back
| |
| |     Put back LT, comment out sys/time.h
| |
| * commit 37470e97bb50de7678bb6ea084f11bb4169c8367
|/  Author: oracleofnj <jared.samet@aya.yale.edu>
|   Date:   Wed Dec 14 11:14:06 2016 -0500
|
|       Put back LT, comment out sys/time.h
|
*   commit d87b73c920b3943e1eaf83b0b4437e115ffca0a8
|\  Merge: 61bc9b6 d126e3c
| | Author: Jared Samet <jared.samet@aya.yale.edu>
| | Date:   Wed Dec 14 10:51:58 2016 -0500
| |
| |     Merge pull request #82 from ExtendLang/hard-to-repro-bug
| |
| |     Half the time it works
| |
| * commit d126e3ca94535cdbdeb917737b0464f1b83f1483
| | Author: oracleofnj <jared.samet@aya.yale.edu>
| | Date:   Wed Dec 14 00:51:00 2016 -0500
| |
| |     Try with time.h instead of sys/time.h
| |
| * commit a5356125e9e9c01e933ba553cd4ba6d168c48a25
| | Author: oracleofnj <jared.samet@aya.yale.edu>
| | Date:   Wed Dec 14 00:48:35 2016 -0500
| |
| |     Remove lrints
| |
| * commit e8448532121cc49d4b8cb3b1b436d3268f643753
| | Author: oracleofnj <jared.samet@aya.yale.edu>
| | Date:   Wed Dec 14 00:34:37 2016 -0500
| |
| |     Initialize all variables and remove pointer math; bug appears fixed
| |
| * commit 4c1a421a814d132a57e1cda50e4b1c9c3709dd6d
| | Author: oracleofnj <jared.samet@aya.yale.edu>
| | Date:   Tue Dec 13 22:55:07 2016 -0500
| |
| |     Some formula is weird
| |
| *   commit 5dbd4090286655786d4fdfe259f055e696eaa7b4
| |\  Merge: 879eaf3 37f5ce2
| | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | Date:   Tue Dec 13 22:43:19 2016 -0500
| | |
| | |     Merge branch ’hard-to-repro-bug’ of https://github.com/ExtendLang/Extend into hard-to-repro-bug
| | |
| | *   commit 37f5ce24fa1ee119ae55c4aa16799960e6a1ed3a
| | |\  Merge: e20f7e4 a1cfc5a
| | | | Author: Jared Samet <jared.samet@aya.yale.edu>
| | | | Date:   Tue Dec 13 22:42:40 2016 -0500
| | | |
| | | |     Merge pull request #83 from ExtendLang/rounding-for-read
| | | |
| | | |     Added rounding at several places
| | | |
| | | * commit a1cfc5a6513b5e67e2c99707834108b78b82ede3
| | |/  Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | |   Date:   Tue Dec 13 22:34:21 2016 -0500
| | |
| | |       Added rounding at several places
| | |
| * | commit 879eaf3795a5b000e4bfa45cb7cd100fef32fe16
| |/  Author: oracleofnj <jared.samet@aya.yale.edu>
| |   Date:   Tue Dec 13 22:43:17 2016 -0500
| |
| |       Testing
| |
| * commit e20f7e4b4a34eb102fd10408acb0c05db53a0ef7
|/  Author: oracleofnj <jared.samet@aya.yale.edu>
|   Date:   Tue Dec 13 21:36:13 2016 -0500
|
|       Half the time it works
|
*   commit 61bc9b646c9a47d9895c436fe070671cfd2dccdf
|\  Merge: ae5b8a8 4a810df
| | Author: Nigel Schuster <Neitsch@users.noreply.github.com>
| | Date:   Tue Dec 13 20:33:27 2016 -0500
| |
| |     Merge pull request #81 from ExtendLang/fix-em-all
| |
| |     Corrected testcase outputs
| |
| * commit 4a810dff997bbb03a5093ff83a4a5ac0ce51decb
|/  Author: Nigel Schuster <nigel.schusters@googlemail.com>
|   Date:   Tue Dec 13 19:34:29 2016 -0500
|
|       Corrected testcase outputs
|
*   commit ae5b8a8e130a1b0e20f70d446097115be22018aa
|\  Merge: 8e6e9ba 70b2704
```

```
| | Author: Jared Samet <jared.samet@aya.yale.edu>
| | Date:   Tue Dec 13 19:08:43 2016 -0500
| |
| |     Merge pull request #80 from ExtendLang/select
| |
| |     Select
| |
| * commit 70b270444f13c33a41366d8bfa0be971f11b0b30
| | Author: oracleofnj <jared.samet@aya.yale.edu>
| | Date:   Tue Dec 13 19:02:32 2016 -0500
| |
| |     No C99
| |
| *   commit 15fd76223412a2a9c25f2854300f80248e93f5f4
| |\  Merge: 7a93885 8e6e9ba
| |/  Author: oracleofnj <jared.samet@aya.yale.edu>
|/|   Date:   Tue Dec 13 18:42:21 2016 -0500
| |
| |       Merge branch ’master’ into select
| |
* |   commit 8e6e9ba23e7d8f401efc0d9e6fc52ca15eac78bc
|\ \  Merge: 8146d04 a483282
| | | Author: Jared Samet <jared.samet@aya.yale.edu>
| | | Date:   Tue Dec 13 18:42:05 2016 -0500
| | |
| | |     Merge pull request #78 from ExtendLang/unop-unary-minus
| | |
| | |     Unop TypeOf
| | |
| | * commit 7a93885edd8c186e3487dc57efdd00ad156f10dd
| | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | Date:   Tue Dec 13 18:41:49 2016 -0500
| | |
| | |     Calculate all formula indices
| | |
| | * commit 07e63dc3ffeceecfc6e598f5faa1435c4c4a4bd7
| | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | Date:   Tue Dec 13 18:19:58 2016 -0500
| | |
| | |     Properly build instantiate var
| | |
| | * commit 1a291292bfb2a95aa54d796c741ba622262c3137
| | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | Date:   Tue Dec 13 17:24:16 2016 -0500
| | |
| | |     Replace bools with chars for compatibility between C and LLVM
| | |
| | * commit 12e78a30f51a2d6034bbfdf313b624383fa94885
| |/  Author: oracleofnj <jared.samet@aya.yale.edu>
| |   Date:   Tue Dec 13 17:17:54 2016 -0500
| |
| |       Added debug output
| |
| *   commit a4832821fbe532bbf2fcbc38270c9821bcf44ec0
| |\  Merge: f8c9b43 8146d04
| |/  Author: oracleofnj <jared.samet@aya.yale.edu>
```

```
|/|   Date:   Tue Dec 13 16:13:30 2016 —0500
| |
| |       Merge branch ’master’ into unop—unary—minus
| |
* |   commit 8146d046adbf93a34157048b3c4b1777207334fc
|\ \  Merge: dcd5766 94afc93
| | | Author: Jared Samet <jared.samet@aya.yale.edu>
| | | Date:   Tue Dec 13 16:12:17 2016 —0500
| | |
| | |     Merge pull request #75 from ExtendLang/fix—more—tc
| | |
| | |     Fix more tc
| | |
| * | commit 94afc93afdbe4eb32322f9e8951ef250f1870263
| | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | Date:   Tue Dec 13 16:02:06 2016 —0500
| | |
| | |     Corrected expected TC
| | |
| * | commit f6f82767ec790d01dd5b6e7e05509fceaec322ec
|/ /  Author: Nigel Schuster <nigel.schusters@googlemail.com>
| |   Date:   Tue Dec 13 16:00:59 2016 —0500
| |
| |       Fixed string.xtnd file
| |
* |   commit dcd5766e9b71ae3445135a1c731dd35cdafd24d3
|\ \  Merge: 23328f1 d9abfc0
| | | Author: Nigel Schuster <Neitsch@users.noreply.github.com>
| | | Date:   Tue Dec 13 15:44:38 2016 —0500
| | |
| | |     Merge pull request #74 from ExtendLang/fix—tc
| | |
| | |     Fixed ternary testcase
| | |
| * |   commit d9abfc09658cf505f5f9b9239953cb7b1347bef7
| |\ \  Merge: 324779a 23328f1
| |/ /  Author: Nigel Schuster <Neitsch@users.noreply.github.com>
|/| |   Date:   Tue Dec 13 15:38:38 2016 —0500
| | |
| | |       Merge branch ’master’ into fix—tc
| | |
| * | commit 324779a30d19b74050584d4ef2bd536d3f1b4943
| | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | Date:   Tue Dec 13 15:32:26 2016 —0500
| | |
| | |     Corrected expected value
| | |
| * | commit fafe2e602f3a091d77ddbb90887680bcefc9bd95
| | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | Date:   Tue Dec 13 15:29:00 2016 —0500
| | |
| | |     Fixed string tc
| | |
| * | commit 022f05c5b303b6c567d38393517803276c90b157
| | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | Date:   Tue Dec 13 15:23:59 2016 —0500
| | |
| | |     Fixed testcase
| | |
| | * commit f8c9b435857d699209cfe033171f124ce78e3dae
| | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | Date:   Tue Dec 13 16:13:09 2016 —0500
| | |
| | |     Make TypeOf work
| | |
| | *   commit bfe1c07e3b3ec6d5237043eb50ce73f2fac80fb9
| | |\  Merge: 50ed49c 23328f1
| |_|/  Author: oracleofnj <jared.samet@aya.yale.edu>
|/| |   Date:   Tue Dec 13 15:39:45 2016 —0500
| | |
| | |       Merge branch ’master’ into unop—unary—minus
| | |
* | |   commit 23328f10c3bf28e27a7e6a081fa91eea27611b5b
|\ \ \  Merge: 6ad8512 b12fe37
| | | | Author: Nigel Schuster <Neitsch@users.noreply.github.com>
| | | | Date:   Tue Dec 13 15:37:18 2016 —0500
| | | |
| | | |     Merge pull request #73 from ExtendLang/and—or—xor
| | | |
| | | |     And or xor
| | | |
| * | | commit b12fe378c00d0782b36cc270a1388b2bb9094a52
| | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | | Date:   Tue Dec 13 15:18:57 2016 —0500
| | | |
| | | |     Implemented and, or and xor
| | | |
| * | | commit 90cbaa089e26f712bdb0321c66604d5ab83d4a75
| | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | | Date:   Tue Dec 13 15:16:31 2016 —0500
| | | |
| | | |     Added left and right shift
| | | |
| * | |   commit 571ee7e663823d3cdcc388c11c3ef4903cc07625
| |\ \ \  Merge: aeab40d e377567
| | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | | | Date:   Tue Dec 13 14:56:05 2016 —0500
| | | | |
| | | | |     Merge branch ’power’ of https://github.com/ExtendLang/Extend into power
| | | | |
| | * | |   commit e377567e16ab447e156073e63e80ddd17545de28
| | |\ \ \  Merge: 71f395d 6ad8512
| |_|/ / /  Author: Nigel Schuster <Neitsch@users.noreply.github.com>
|/| | / /   Date:   Tue Dec 13 14:53:28 2016 —0500
| | |/ /
| | | |         Merge branch ’master’ into power
| | | |
* | | |   commit 6ad85121a7782e13817dfd58a7b7308cccf9bf0d
|\ \ \ \  Merge: 668a0eb 6a04209
| | | | | Author: Jared Samet <jared.samet@aya.yale.edu>
| | | | | Date:   Tue Dec 13 14:53:11 2016 —0500
| | | | |
| | | | |     Merge pull request #69 from ExtendLang/unop—unary—minus
| | | | |
| | | | |     Unop unary minus
| | | | |
| | * | | commit aeab40daf88e72d6bd61d80553c17c0c2db73930
| | |/ /  Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | |   Date:   Tue Dec 13 14:55:57 2016 —0500
| | | |
| | | |       Removed unneccessary level of indirection
| | | |
| | * | commit 71f395dcefc8f064d9cd0700da3345fc9658e083
| |/ /  Author: Nigel Schuster <nigel.schusters@googlemail.com>
|/| |   Date:   Tue Dec 13 14:46:27 2016 —0500
| | |
| | |       Power to the people of Extend
| | |
| | * commit 50ed49cab80e64801ddef4b2242de0aabf886912
| |/  Author: oracleofnj <jared.samet@aya.yale.edu>
| |   Date:   Tue Dec 13 15:38:04 2016 —0500
| |
| |       Merging in main
| |
| *   commit 6a04209cc7ebbdb2a96fc67b66fed983b10c44c8
| |\  Merge: edb0ecc 668a0eb
| |/  Author: oracleofnj <jared.samet@aya.yale.edu>
|/|   Date:   Tue Dec 13 14:45:46 2016 —0500
| |
| |       Fix merge conflict
| |
* |   commit 668a0eb548dbab171c854c1e08c24628cc5f268c
|\ \  Merge: f873242 866b68f
| | | Author: Jared Samet <jared.samet@aya.yale.edu>
| | | Date:   Tue Dec 13 14:37:19 2016 —0500
| | |
| | |     Merge pull request #68 from ExtendLang/mod—div
| | |
| | |     Modulo and division
| | |
| * | commit 866b68f65b6e94cb063a1fb88918f62ebba68bb6
| | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | Date:   Tue Dec 13 14:32:18 2016 —0500
| | |
| | |     Added modulo and division operation
| | |
| * | commit 84dfc3315078447c8df4b64a2f00cda7bef47cf1
| | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | Date:   Tue Dec 13 14:26:25 2016 —0500
| | |
| | |     Crunched some code
| | |
| * |   commit f4d5a8114d50acf5683d58629eca82a776c32480
| |\ \  Merge: fc94112 f873242
| |/ /  Author: Nigel Schuster <nigel.schusters@googlemail.com>
|/| |   Date:   Tue Dec 13 14:22:12 2016 —0500
| | |
| | |       Merge branch ’master’ into simplification
| | |
| * | commit fc941122a7b42a66c6db83c23e9f8abfce9c5a25
| | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | Date:   Tue Dec 13 14:20:35 2016 -0500
| | |
| | |     Added multiplication
| | |
| | * commit edb0eccb71dc27ceacae7362802b9d3562dd1f9b
| | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | Date:   Tue Dec 13 14:43:32 2016 -0500
| | |
| | |     Add unary minus
| | |
| | *   commit 46d5aa6b0827bf684873459b2ea347f43294c488
| | |\  Merge: 76210eb f873242
| |_|/  Author: oracleofnj <jared.samet@aya.yale.edu>
|/| |   Date:   Tue Dec 13 14:26:35 2016 -0500
| | |
| | |       Merge branch ’master’ into unop-typeof
| | |
* | |   commit f873242d23ab48c0e4230881d4524d53bdca5b52
|\ \ \  Merge: 4afd78e 6c26c2c
| | | | Author: Nigel Schuster <Neitsch@users.noreply.github.com>
| | | | Date:   Tue Dec 13 14:21:26 2016 -0500
| | | |
| | | |     Merge pull request #65 from ExtendLang/subtraction
| | | |
| | | |     Addition of Subtraction
| | | |
| * | |   commit 6c26c2cae72e3780299f447d1258e27dc799e1a3
| |\ \ \  Merge: bd90241 4afd78e
| |/ / /  Author: Nigel Schuster <Neitsch@users.noreply.github.com>
|/| | |   Date:   Tue Dec 13 14:19:07 2016 -0500
| | | |
| | | |       Merge branch ’master’ into subtraction
| | | |
| * | |   commit bd90241ce837eb554b67204b86d9db76c9888a0f
| |\ \ \  Merge: 4042259 c7d4162
| | | | | Author: Nigel Schuster <Neitsch@users.noreply.github.com>
| | | | | Date:   Tue Dec 13 14:14:17 2016 -0500
| | | | |
| | | | |     Merge branch ’master’ into subtraction
| | | | |
| * | | | commit 4042259e9960bc7d4e9fd50a45c8f94231770339
| | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | | | Date:   Tue Dec 13 14:13:09 2016 -0500
| | | | |
| | | | |     Added subtraction
| | | | |
| * | | |   commit 82a3db27594ad2acc9f8a275a26b8fd01937adde
| |\ \ \ \  Merge: 1e1f973 cc40008
| | | |/ /  Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | |/| |   Date:   Tue Dec 13 14:11:31 2016 -0500
| | | | |
| | | | |       Merge branch ’master’ into subtraction
| | | | |
| * | | | commit 1e1f9734f1e8b482c8cd7074014aa4e5c506e19c
| | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | | | Date:   Tue Dec 13 13:44:36 2016 −0500
| | | | |
| | | | |     Subtraction
| | | | |
| | | | * commit 76210ebd86f466b3e2c3e837b5aa9d1c4776d56b
| |_|_|/  Author: oracleofnj <jared.samet@aya.yale.edu>
|/| | |   Date:   Tue Dec 13 14:26:18 2016 −0500
| | | |
| | | |       Start on it
| | | |
* | | |   commit 4afd78e4535d88d3c131af4c4dfd5a1812ec17e9
|\ \ \ \  Merge: c7d4162 d4d4388
| |_|_|/  Author: Nigel Schuster <Neitsch@users.noreply.github.com>
|/| | |   Date:   Tue Dec 13 14:18:55 2016 −0500
| | | |
| | | |       Merge pull request #64 from ExtendLang/refactor−boolean−binops
| | | |
| | | |       Change Lt, Lte in grammar; implement GTE
| | | |
| * | |   commit d4d4388775ad5c1ce66fa729f5161d324c0da91c
| |\ \ \  Merge: 663f399 c7d4162
| |/ / /  Author: Nigel Schuster <Neitsch@users.noreply.github.com>
|/| | |   Date:   Tue Dec 13 14:15:58 2016 −0500
| | | |
| | | |       Merge branch ’master’ into refactor−boolean−binops
| | | |
* | | |   commit c7d41623a82f3819c7a2104a8dabd1ca17c2a5d6
|\ \ \ \  Merge: cc40008 97821c8
| |_|_|/  Author: Jared Samet <jared.samet@aya.yale.edu>
|/| | |   Date:   Tue Dec 13 14:02:13 2016 −0500
| | | |
| | | |       Merge pull request #63 from ExtendLang/more−binops
| | | |
| | | |       More binops
| | | |
| | * | commit 663f399e023f753aca1c87d758cf4cbdd167e466
| | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | Date:   Tue Dec 13 14:12:57 2016 −0500
| | | |
| | | |     Remove wildcard from BinOp pattern match
| | | |
| | * | commit 1bf6bedffc1776bc66e7dbc413bd74d293778ead
| | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | Date:   Tue Dec 13 14:09:47 2016 −0500
| | | |
| | | |     Add TransformedAway exception for LogAnd and LogOr
| | | |
| | * | commit 952778e9205d92208cfb48b6b8c92c2c697cfb40
| |/ /  Author: oracleofnj <jared.samet@aya.yale.edu>
| | |   Date:   Tue Dec 13 14:01:54 2016 −0500
| | |
| | |       Change Lt, Lte in grammar; implement GTE
| | |
| * | commit 97821c8291bc07c3607f98179832fb05a89f6e8d
| | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | Date:   Tue Dec 13 13:47:52 2016 –0500
| | |
| | |     GT
| | |
| * | commit e0a883a96e63eea084f350d58e12a8d56be8e157
|/ /  Author: oracleofnj <jared.samet@aya.yale.edu>
| |   Date:   Tue Dec 13 13:37:57 2016 –0500
| |
| |       Remove NotEq from AST since != is parsed to UnOp(LogNot,BinOp(Eq,...))
| |
* |   commit cc40008a9c11d7b084feb72657cf2a69fb5dea62
|\ \  Merge: a656f57 7123ebc
| | | Author: Nigel Schuster <Neitsch@users.noreply.github.com>
| | | Date:   Tue Dec 13 12:49:33 2016 –0500
| | |
| | |     Merge pull request #60 from ExtendLang/addition2
| | |
| | |     String concatenation
| | |
| * |   commit 7123ebcb45ebe8a8dc4a924a17760a8eaf016473
| |\ \  Merge: eb134b3 a656f57
| |/ /  Author: Nigel Schuster <Neitsch@users.noreply.github.com>
|/| |   Date:   Tue Dec 13 12:41:09 2016 –0500
| | |
| | |       Merge branch ’master’ into addition2
| | |
* | |   commit a656f57efd95a86d97de787aa6694f65f39e265a
|\ \ \  Merge: 81533f4 a64cc15
| |_|/  Author: Jared Samet <jared.samet@aya.yale.edu>
|/| |   Date:   Tue Dec 13 12:38:12 2016 –0500
| | |
| | |       Merge pull request #61 from ExtendLang/debug–unop
| | |
| | |       Add Debug expr
| | |
| * | commit a64cc15c6f95beaf1f4fe85112c4869a2b9b12fb
|/ /  Author: oracleofnj <jared.samet@aya.yale.edu>
| |   Date:   Tue Dec 13 12:14:45 2016 –0500
| |
| |       Add Debug expr
| |
| * commit eb134b35a63c3f58eb744c921a9235b11399fba3
| | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | Date:   Tue Dec 13 12:29:53 2016 –0500
| |
| |     Moved testcases
| |
| * commit 044c6bdda6cba9100614b3c20ed355bdecbf2095
| | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | Date:   Tue Dec 13 12:29:07 2016 –0500
| |
| |     Fixed off by one error
| |
| * commit 59858a0952c52893b5b5245b158b426bbb5c4a4d
| | Author: oracleofnj <jared.samet@aya.yale.edu>
| | Date:   Tue Dec 13 11:33:12 2016 –0500
| |
| |     Whoops no space
| |
| * commit 0426f3411166f44f628bcb94bdde53ec4a6bb5ba
| | Author: oracleofnj <jared.samet@aya.yale.edu>
| | Date:   Tue Dec 13 11:30:26 2016 –0500
| |
| |     Add test case
| |
| *   commit 49ffa8688e78b8add3b67dbb36fe505f759eac17
| |\  Merge: 3cdaa5a 81533f4
| |/  Author: Nigel Schuster <Neitsch@users.noreply.github.com>
|/|   Date:   Tue Dec 13 11:19:14 2016 –0500
| |
| |       Merge branch ’master’ into addition2
| |
* |   commit 81533f4fc65b6303d3bc21bf755814053e9212ec
|\ \  Merge: 200b8b6 64d1760
| | | Author: Jared Samet <jared.samet@aya.yale.edu>
| | | Date:   Tue Dec 13 11:13:44 2016 –0500
| | |
| | |     Merge pull request #59 from ExtendLang/equal–rights
| | |
| | |     Equal rights
| | |
| * | commit 64d1760ba8810a2d57de49736980530e5d803441
| | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | Date:   Tue Dec 13 11:04:55 2016 –0500
| | |
| | |     Wake up please, GitHub
| | |
| * | commit 840aeaf2a7afd1b88756ad9bb391d7763921c76e
| | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | Date:   Tue Dec 13 10:48:03 2016 –0500
| | |
| | |     Remove usage demonstration
| | |
| * | commit 61ff4395ca4fef30147df4e8a7733f3eb07d3ed3
| | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | Date:   Tue Dec 13 03:26:35 2016 –0500
| | |
| | |     Add string equality and test cases
| | |
| * | commit f3112e98f1a16888694bae635e3c2d8707c83469
| | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | Date:   Tue Dec 13 01:57:10 2016 –0500
| | |
| | |     Reduce cut & paste
| | |
| * | commit 08ce677a9c4db365fefdf59a91c89b8c8bae5d73
| | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | Date:   Tue Dec 13 01:35:46 2016 –0500
| | |
| | |     Remove obsolete testing file
| | |
| * |   commit ae8a07e0ccd6e9959d83a800af7e68ef7267e39c
| |\ \  Merge: 862b38c 6090713
| | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | Date:   Tue Dec 13 01:23:26 2016 —0500
| | | |
| | | |     Merge branch ’print_value_p’ into equal—rights
| | | |
| | * | commit 6090713b9c7b49a8a58308cb4caaaef6a1817aa9
| | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | Date:   Tue Dec 13 01:22:47 2016 —0500
| | | |
| | | |     Use correct printf specifier
| | | |
| * | |   commit 862b38c5f0190732e311a967863ff8862e92490c
| |\ \ \  Merge: 50281b1 5e913ad
| | |/ /  Author: oracleofnj <jared.samet@aya.yale.edu>
| | | |   Date:   Tue Dec 13 01:19:14 2016 —0500
| | | |
| | | |       Merge branch ’print_value_p’ into equal—rights
| | | |
| | * | commit 5e913ad2d9e42cc06417b1de641cb812d712e52a
| |/ /  Author: oracleofnj <jared.samet@aya.yale.edu>
|/| |   Date:   Tue Dec 13 01:16:07 2016 —0500
| | |
| | |       Add debug_print; remove print statement that was causing us to falsely pass test cases from to_string; show usage in UnOp(Neg)
| | |
| * | commit 50281b1d3047296e7f3c5ed362bd4126a37dd0a2
| | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | Date:   Tue Dec 13 00:47:28 2016 —0500
| | |
| | |     Numeric equality
| | |
| * | commit 0f76aa4eac383b95c8c9b05c7a1073b18c4e53b4
|/ /  Author: oracleofnj <jared.samet@aya.yale.edu>
| |   Date:   Mon Dec 12 22:30:15 2016 —0500
| |
| |       Remove print flags
| |
* |   commit 200b8b6bf132a7d80f6545ab1a8063b95f8b8477
|\ \  Merge: 8834635 7e7276b
| | | Author: Nigel Schuster <Neitsch@users.noreply.github.com>
| | | Date:   Mon Dec 12 22:16:15 2016 —0500
| | |
| | |     Merge pull request #57 from ExtendLang/addition2
| | |
| | |     Addition
| | |
| | * commit 3cdaa5a45e4644004a3cb274fd43c0c56ecd38a4
| | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | Date:   Tue Dec 13 11:12:41 2016 —0500
| | |
| | |     String addition
| | |
| | * commit da7c5433dc305f65f46f6bacec96c351aeaef60c
| |/  Author: Nigel Schuster <nigel.schusters@googlemail.com>
| |   Date:   Mon Dec 12 12:43:31 2016 -0500
| |
| |       Setting flag for addition
| |
| *   commit 7e7276ba7b8c816cc7c3c3d8af46948e96b14a16
| |\  Merge: 8aa125f 8834635
| |/  Author: Nigel Schuster <nigel.schusters@googlemail.com>
|/|   Date:   Mon Dec 12 12:37:35 2016 -0500
| |
| |       Merge branch ’master’ into addition2
| |
* |   commit 88346350bff1466a192f24dbb77f572a329c7204
|\ \  Merge: 6ed303e 53ae9e0
| | | Author: Nigel Schuster <Neitsch@users.noreply.github.com>
| | | Date:   Mon Dec 12 10:18:51 2016 -0500
| | |
| | |     Merge pull request #55 from ExtendLang/runtime
| | |
| | |     Extracted runtime into seperate file
| | |
| * |   commit 53ae9e05b3affda0148b3112b8b6362d15cabbc0
| |\ \  Merge: d1e196d 6ed303e
| |/ /  Author: Nigel Schuster <Neitsch@users.noreply.github.com>
|/| |   Date:   Mon Dec 12 10:06:24 2016 -0500
| | |
| | |       Merge branch ’master’ into runtime
| | |
* | |   commit 6ed303e17033f6525a77d6640b2da8c6bf806f7b
|\ \ \  Merge: ecc620e ae49ce6
| | | | Author: Jared Samet <jared.samet@aya.yale.edu>
| | | | Date:   Mon Dec 12 09:43:57 2016 -0500
| | | |
| | | |     Merge pull request #56 from ExtendLang/truthy-fix
| | | |
| | | |     Truthy fix
| | | |
| * | | commit ae49ce6bba03f66e939fb33bdb7b6a9af58f1696
| | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | Date:   Mon Dec 12 01:15:29 2016 -0500
| | | |
| | | |     Remove extra file
| | | |
| * | | commit 7fe6a2234a976d5d85333147b0016d4099f71f5d
|/ / /  Author: oracleofnj <jared.samet@aya.yale.edu>
| | |   Date:   Mon Dec 12 01:11:53 2016 -0500
| | |
| | |       Falsey fix
| | |
* | |   commit ecc620eba7af7a55e6da7a222e6ef6a2a08c461d
|\ \ \  Merge: a875b41 4c8caa5
| | | | Author: Nigel Schuster <Neitsch@users.noreply.github.com>
| | | | Date:   Mon Dec 12 00:17:06 2016 -0500
| | | |
| | | |     Merge pull request #54 from ExtendLang/final-draft-for-real
| | | |
| | | |     Final draft for real
| | | |
| * | |   commit 4c8caa516cd2dbf01475669aa1ca048ee1c90130
| |\ \ \  Merge: 04d3b57 a875b41
| |/ / /  Author: Jared Samet <jared.samet@aya.yale.edu>
|/| | |   Date:   Mon Dec 12 00:09:16 2016 -0500
| | | |
| | | |       Merge branch ’master’ into final-draft-for-real
| | | |
| * | |   commit 04d3b57dd8f38dafb9edb7999ed203e2c9921f26
| |\ \ \  Merge: 718ecd3 39025b0
| | | | | Author: Nigel Schuster <Neitsch@users.noreply.github.com>
| | | | | Date:   Mon Dec 12 00:00:29 2016 -0500
| | | | |
| | | | |     Merge pull request #39 from ExtendLang/more-lrm-ed
| | | | |
| | | | |     More lrm edits
| | | | |
| | * | | commit 39025b056938eb96d3643228d86ff1fc93a9ce24
| | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | | | Date:   Sun Dec 11 23:59:18 2016 -0500
| | | | |
| | | | |     Fixed examples, made small corrections
| | | | |
| | * | | commit a0ed757bb534a010db452bef1409ec536cb17637
| | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | | | Date:   Sat Dec 10 23:31:38 2016 -0500
| | | | |
| | | | |     Edited explanation for row() and column()
| | | | |
| | * | | commit 7c50ef28d766680fd30067187d01d27fda12e7fa
| | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | | | Date:   Sat Dec 10 23:27:07 2016 -0500
| | | | |
| | | | |     Added info for strings
| | | | |
| | * | | commit 738e41b433a38008360696f2ece253a91fafdbe6
| | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | | | Date:   Sat Dec 10 23:24:20 2016 -0500
| | | | |
| | | | |     Added boolean example
| | | | |
| | * | | commit 5377fdff3f9824062de5a599b59ea07f5e2d2d4b
| |/ / /  Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | |   Date:   Sat Dec 10 23:19:26 2016 -0500
| | | |
| | | |       Added arithmetic example
| | | |
| * | | commit 718ecd3b1c641c7b5816718c142423fa900d07ca
| | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | Date:   Sat Dec 10 03:09:18 2016 -0500
| | | |
| | | |     Some changes to LRM; add if(a,b,c)
| | | |
| | * | commit d1e196da9518ddc18fd6df57a52c91aef3e71fe8
| |/ /  Author: Nigel Schuster <nigel.schusters@googlemail.com>
|/| |   Date:   Mon Dec 12 00:23:13 2016 -0500
```

```
| | |
| | |       Extracted runtime into seperate file
| | |
* | |   commit a875b4172c1d79172be671c297c09718b244dbdb
|\ \ \  Merge: b95d14f 616dd34
| | | | Author: Jared Samet <jared.samet@aya.yale.edu>
| | | | Date:   Sun Dec 11 23:51:30 2016 -0500
| | | |
| | | |     Merge pull request #53 from ExtendLang/truthy
| | | |
| | | |     Truthy
| | | |
| * | |   commit 616dd34bebf74dfb7b74191141880b7cb68a7019
| |\ \ \  Merge: 0fa8255 b95d14f
| |/ / /  Author: oracleofnj <jared.samet@aya.yale.edu>
|/| | |   Date:   Sun Dec 11 23:15:54 2016 -0500
| | | |
| | | |       Merge branch ’master’ into truthy
| | | |
* | | |   commit b95d14f5dd690433f23be5a71e0affebce6b1829
|\ \ \ \  Merge: 2a905c7 6dea96f
| | | | | Author: Jared Samet <jared.samet@aya.yale.edu>
| | | | | Date:   Sun Dec 11 21:02:28 2016 -0500
| | | | |
| | | | |     Merge pull request #50 from ExtendLang/builder-hotfix
| | | | |
| | | | |     So many builders
| | | | |
| | * | | commit 0fa8255b29077c769514973ad17fa7eb7fec9351
| | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | | Date:   Sun Dec 11 23:14:42 2016 -0500
| | | | |
| | | | |     Apparently still needs some work
| | | | |
| | * | | commit 78584d7808785b7a8e7253788bdfc0240e96cfd9
| | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | | Date:   Sun Dec 11 23:09:07 2016 -0500
| | | | |
| | | | |     Thanks a lot Travis
| | | | |
| | * | | commit b5673d22f67f84f159cd1c28288a195be81aada3
| | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | | Date:   Sun Dec 11 22:51:52 2016 -0500
| | | | |
| | | | |     TERRRRRRRRR NARRRRRRR EEEEEEEEEEEEEEEEE
| | | | |
| | * | | commit b81bc1bab064a134c831685c9d76c47a4b2cd69c
| |/ / /  Author: oracleofnj <jared.samet@aya.yale.edu>
| | | |   Date:   Sun Dec 11 22:04:25 2016 -0500
| | | |
| | | |       Maybe Truthy
| | | |
| * | | commit 6dea96f0c2d42f131990b37e515f2e6df198d2e0
|/ / /  Author: oracleofnj <jared.samet@aya.yale.edu>
| | |   Date:   Sun Dec 11 20:40:47 2016 -0500
| | |
| | |       So many builders
| | |
* | |   commit 2a905c7b5b89a9a2d490bda9782e1e0e54fb8cb3
|\ \ \  Merge: 18fc1be 2bc6c85
| | | | Author: Jared Samet <jared.samet@aya.yale.edu>
| | | | Date:   Sun Dec 11 19:15:47 2016 —0500
| | | |
| | | |     Merge pull request #47 from ExtendLang/function—parameter
| | | |
| | | |     Function parameter
| | | |
| * | | commit 2bc6c85a240f72c3bc040bb2d54e4bb375b1720b
| | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | Date:   Sun Dec 11 19:11:33 2016 —0500
| | | |
| | | |     Add combined test case
| | | |
| * | |   commit 860a11b3e4368c6f21337ccd055df4c00f502b0f
| |\ \ \  Merge: 8c3499e 18fc1be
| |/ / /  Author: oracleofnj <jared.samet@aya.yale.edu>
|/| | |   Date:   Sun Dec 11 19:04:35 2016 —0500
| | | |
| | | |       Merge branch ’master’ into function—parameter
| | | |
| * | | commit 8c3499e3dd670ff1e16911b823cff1933452c71f
| | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | Date:   Sun Dec 11 19:03:39 2016 —0500
| | | |
| | | |     Remove extraneous printlines
| | | |
| * | | commit 99418c0215673e964074547dced2e0f69a432b06
| | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | Date:   Sun Dec 11 19:02:31 2016 —0500
| | | |
| | | |     Make function parameters work
| | | |
| * | | commit 387559b44107284c1056db47378bf9b79458cb93
| | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | Date:   Sun Dec 11 18:39:00 2016 —0500
| | | |
| | | |     First attempt
| | | |
| | | * commit 8aa125f1d5292eb3f7aa41be186d761248134535
| | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | | Date:   Sun Dec 11 20:15:52 2016 —0500
| | | |
| | | |     Made som rpgroess
| | | |
| | | * commit 6c00a72a2912c7f38af73fc49070f63e05243ca6
| |_|/  Author: Nigel Schuster <nigel.schusters@googlemail.com>
|/| |   Date:   Sun Dec 11 18:45:46 2016 —0500
| | |
| | |       Some progress
| | |
* | |   commit 18fc1beab2196a2a45d6ce4cdf2b528f2545c03e
|\ \ \  Merge: f1dd8a5 f7e9be8
```

```
| |/ /  Author: Nigel Schuster <Neitsch@users.noreply.github.com>
|/| |   Date:   Sun Dec 11 18:08:11 2016 -0500
| | |
| | |       Merge pull request #45 from ExtendLang/empty
| | |
| | |       Implemented empty, small flag setting fix
| | |
| * |   commit f7e9be8c20044c73ea00ce076bd075a24e809dfa
| |\ \  Merge: 18db166 f1dd8a5
| |/ /  Author: Nigel Schuster <Neitsch@users.noreply.github.com>
|/| |   Date:   Sun Dec 11 16:30:05 2016 -0500
| | |
| | |       Merge branch ’master’ into empty
| | |
* | |   commit f1dd8a5ca9976def254e61530ed91c650c09070b
|\ \ \  Merge: 3c4681d 50366f4
| | | | Author: Jared Samet <jared.samet@aya.yale.edu>
| | | | Date:   Sun Dec 11 16:18:44 2016 -0500
| | | |
| | | |     Merge pull request #46 from ExtendLang/actually-make-global-scope
| | | |
| | | |     Actually make global scope
| | | |
| * | | commit 50366f43d98220a48d90a6faa884c70e68778627
| | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | Date:   Sun Dec 11 15:38:05 2016 -0500
| | | |
| | | |     Make sure locals are properly masking globals
| | | |
| * | | commit 046c7cc0ca4ad773ac83b7677dd7f14e3d8c9337
| | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | Date:   Sun Dec 11 15:30:53 2016 -0500
| | | |
| | | |     Make globals work, fix bug
| | | |
| * | | commit a844a4635358aeba7edd4f6faa83871036c1fee4
| | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | Date:   Sun Dec 11 15:14:09 2016 -0500
| | | |
| | | |     So close
| | | |
| * | | commit 67849f0177f0c67cd19a61121b9d971bab6a0474
|/ / /  Author: oracleofnj <jared.samet@aya.yale.edu>
| | |   Date:   Sun Dec 11 15:01:52 2016 -0500
| | |
| | |       Make the global scope object
| | |
| * |   commit 18db1667adb6a80cc1d0bbb4b6b657ec6db0200e
| |\ \  Merge: 393d02c 3c4681d
| |/ /  Author: Nigel Schuster <Neitsch@users.noreply.github.com>
|/| |   Date:   Sun Dec 11 15:05:42 2016 -0500
| | |
| | |       Merge branch ’master’ into empty
| | |
* | |   commit 3c4681d6a7aaf164d403b3148822d8992783754d
|\ \ \  Merge: abcffd0 7be1001
```

```
| | | | Author: Jared Samet <jared.samet@aya.yale.edu>
| | | | Date:   Sun Dec 11 13:31:12 2016 —0500
| | | |
| | | |     Merge pull request #44 from ExtendLang/float—display—hotfix
| | | |
| | | |     Floating point math hotfix
| | | |
| * | |   commit 7be1001d43b1082b4cc2fa6801f619f99f0679c8
| |\ \ \  Merge: 556da44 abcffd0
| |/ / /  Author: Jared Samet <jared.samet@aya.yale.edu>
|/| | |   Date:   Sun Dec 11 13:26:55 2016 —0500
| | | |
| | | |       Merge branch ’master’ into float—display—hotfix
| | | |
* | | |   commit abcffd04216b35160b67c1f7ca29de8020a57774
|\ \ \ \  Merge: d65aad4 0ad195e
| | | | | Author: Jared Samet <jared.samet@aya.yale.edu>
| | | | | Date:   Sun Dec 11 13:19:05 2016 —0500
| | | | |
| | | | |     Merge pull request #42 from ExtendLang/encapsulate—build—scope
| | | | |
| | | | |     Encapsulate a little more of building the scope
| | | | |
| * | | |   commit 0ad195e79e1bce957f7d265e7365de0ffc718988
| |\ \ \ \  Merge: 9caf464 d65aad4
| |/ / / /  Author: oracleofnj <jared.samet@aya.yale.edu>
|/| | | |   Date:   Sun Dec 11 12:42:42 2016 —0500
| | | | |
| | | | |       Merge branch ’master’ into encapsulate—build—scope
| | | | |
| * | | | commit 9caf4645cf9a747101bed8801917dda953b3df5e
| | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | | Date:   Sun Dec 11 12:41:40 2016 —0500
| | | | |
| | | | |     Encapsulate a little more of building the scope
| | | | |
| | * | | commit 556da44ef36ec5b4b93aade0ef2bb2372f0cd0a4
| |/ / /  Author: oracleofnj <jared.samet@aya.yale.edu>
|/| | |   Date:   Sun Dec 11 13:18:15 2016 —0500
| | | |
| | | |       Floating point math hotfix
| | | |
| | * | commit 393d02c61efdaa9e246509fef8b3b61303bf0b96
| |/ /  Author: Nigel Schuster <nigel.schusters@googlemail.com>
|/| |   Date:   Sun Dec 11 14:25:02 2016 —0500
| | |
| | |       Implemented empty, small flag setting fix
| | |
* | |   commit d65aad46b60dde24fdc2740ec1e26676fa03a2e0
|\ \ \  Merge: 9cee2fc 0f5a6ba
| |/ /  Author: Jared Samet <jared.samet@aya.yale.edu>
| | |   Date:   Sun Dec 11 12:09:28 2016 —0500
| | |
| | |       Merge pull request #40 from ExtendLang/make—global—scope
| | |
| | |       Make global scope
| | |
| * |   commit 0f5a6ba3e6c1e3e29a4ed02f70cf418ce059bd4b
| |\ \  Merge: 56b58d9 9cee2fc
| |/ /  Author: oracleofnj <jared.samet@aya.yale.edu>
|/| |   Date:   Sun Dec 11 12:04:05 2016 -0500
| | |
| | |       Merge branch 'master' into make-global-scope
| | |
* | | commit 9cee2fc6bb72ad0cf074bbeb16faa4759778f5d1
| | | Author: kevinye1 <kevinye1@users.noreply.github.com>
| | | Date:   Sun Dec 11 10:07:36 2016 -0500
| | |
| | |     Testcases (#38)
| | |
| | |     * Updated testcases with to_string method
| | |
| | |     * Merged with master
| | |
| * | commit 56b58d9b121d5b8e26a25bdb45572b9de5ed89bb
| | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | Date:   Sun Dec 11 12:01:28 2016 -0500
| | |
| | |     Encapsulate build_var_defns
| | |
| * | commit f25e5b377975ab5473d0592720ded03b7bce259f
| | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | Date:   Sun Dec 11 11:43:19 2016 -0500
| | |
| | |     Only construct var_defns once
| | |
| * | commit f3f4bef9eb2db0313d80dfbe6bdbf83f3a42076d
| | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | Date:   Sun Dec 11 00:45:44 2016 -0500
| | |
| | |     Make global variable to hold vardefns
| | |
| * | commit a8f4ad93aa0dd3365b6661a16035f40a8a1c5587
| | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | Date:   Sat Dec 10 21:28:18 2016 -0500
| | |
| | |     Isolate the part of building a scope for reuse with global variables
| | |
| * | commit 58f7a4d3c4aece7e9f12809f50479a2b8267f01d
|/ /  Author: Nigel Schuster <nigel.schusters@googlemail.com>
| |   Date:   Sat Dec 10 18:05:01 2016 -0500
| |
| |       Performing copy before returning, so that memory can be freed with alloca
| |
* |   commit c0e56aa3b9853d9e44a27109f8c2666d752afe7b
|\ \  Merge: ef0e5e7 a4b35df
| | | Author: Nigel Schuster <Neitsch@users.noreply.github.com>
| | | Date:   Sat Dec 10 17:07:00 2016 -0500
| | |
| | |     Merge pull request #37 from ExtendLang/dereference
| | |
| | |     Dereferencing 1x1 subrange
| | |
| * | commit a4b35df069c5c5243c499cd7aaeda27973678cb6
| | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | Date:   Sat Dec 10 16:42:17 2016 —0500
| | |
| | |     Removed obsolete methods
| | |
| * |   commit cf08a8cc461f1aa54f7f408c78bbf0e20269ccca
| |\ \  Merge: ce833d4 ef0e5e7
| |/ /  Author: Nigel Schuster <Neitsch@users.noreply.github.com>
|/| |   Date:   Sat Dec 10 16:36:20 2016 —0500
| | |
| | |       Merge branch ’master’ into dereference
| | |
* | |   commit ef0e5e7bc6c971c65c8f48460b27479aebb56075
|\ \ \  Merge: 0177dc2 38ba6e6
| | | | Author: Nigel Schuster <Neitsch@users.noreply.github.com>
| | | | Date:   Sat Dec 10 16:36:03 2016 —0500
| | | |
| | | |     Merge pull request #36 from ExtendLang/comp—warn
| | | |
| | | |     Suppressing compiler warnings
| | | |
| * | | commit 38ba6e64fbe18f17f1611c695655b96670cda5b2
| | |/  Author: Nigel Schuster <nigel.schusters@googlemail.com>
| |/|   Date:   Sat Dec 10 13:18:39 2016 —0500
| | |
| | |       Suppressing compiler warnings
| | |
* | |   commit 0177dc290c5e56e428fa135e0b0b9a4495c1e05f
|\ \ \  Merge: 127f99d e259556
| | | | Author: Nigel Schuster <Neitsch@users.noreply.github.com>
| | | | Date:   Sat Dec 10 16:35:50 2016 —0500
| | | |
| | | |     Merge pull request #35 from ExtendLang/linker
| | | |
| | | |     Linker
| | | |
| * | | commit e2595560863fe9fff107818f713b5e8508d0c768
| | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | | Date:   Sat Dec 10 13:53:12 2016 —0500
| | | |
| | | |     Removed nodefaultlibs directive
| | | |
| * | | commit 09c39619d89f65a19d537336c3b3680553f732a0
| | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | | Date:   Sat Dec 10 13:50:19 2016 —0500
| | | |
| | | |     Modified linker to work for travis
| | | |
| * | | commit 36d662a599754648d00c4d773fb16ea1a59d335f
| | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | | Date:   Sat Dec 10 13:37:27 2016 —0500
| | | |
| | | |     Attempt to link math
| | | |
| * | | commit 2d4564a4f9aae99f67f72baf514b7095b452ee08
| | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | | Date:   Sat Dec 10 13:22:14 2016 -0500
| | | |
| | | |     Linking math library
| | | |
| * | | commit 9deac9bbd385607cbd5d80a97e929bc52f7f9633
| | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | | Date:   Sat Dec 10 13:06:17 2016 -0500
| | | |
| | | |     Modified compile script. Removed debug output
| | | |
| * | | commit d35607b7b7830c502d4af4d1d9c65c70bfa37d96
| | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | | Date:   Sat Dec 10 13:04:30 2016 -0500
| | | |
| | | |     Simpler testscript
| | | |
| * | | commit d37dac2f40ea9f0013dd69b6ce40197fe4ec82ab
| | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | | Date:   Sat Dec 10 12:36:45 2016 -0500
| | | |
| | | |     Fixed duplicate import issue
| | | |
| * | | commit 31c26bccd855332e33505e8f887727069a450a00
| |/ /  Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | |   Date:   Sat Dec 10 12:30:29 2016 -0500
| | |
| | |       Added cmd args to link file
| | |
* | |   commit 127f99dc5c44b0cc673b7c254847c54b8a23610b
|\ \ \  Merge: b2e881d a350720
| | | | Author: Nigel Schuster <Neitsch@users.noreply.github.com>
| | | | Date:   Sat Dec 10 16:35:41 2016 -0500
| | | |
| | | |     Merge pull request #34 from ExtendLang/rel-import
| | | |
| | | |     Import path switch
| | | |
| * | | commit a350720f73027f0b5d99b73492d4326081324d84
| |/ /  Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | |   Date:   Sat Dec 10 11:40:50 2016 -0500
| | |
| | |       Switched import style from root directory to relative path
| | |
* | |   commit b2e881df8c2d033bbe299575b6d9092ba7ab51ea
|\ \ \  Merge: 6a8f836 90e39b0
| |/ /  Author: Nigel Schuster <Neitsch@users.noreply.github.com>
|/| |   Date:   Sat Dec 10 16:35:31 2016 -0500
| | |
| | |       Merge pull request #33 from ExtendLang/ts-fix
| | |
| | |       Testscript fix
| | |
| * | commit 90e39b0ba32729808fa7f835ce8adf9784ec5fb4
|/ /  Author: Nigel Schuster <nigel.schusters@googlemail.com>
```

```
| |   Date:   Sat Dec 10 11:24:19 2016 -0500
| |
| |       Fixed issue in testscript that might report false results when it fails early
| |
| * commit ce833d4433f8b0e2ea4782478a3b6d61e85ec0d5
|/  Author: Nigel Schuster <nigel.schusters@googlemail.com>
|   Date:   Sat Dec 10 16:14:34 2016 -0500
|
|       Dereferencing 1x1 subrange
|
*   commit 6a8f8363850840afc597a0b97853ae66a8709683
|\  Merge: eac9e77 fc886a9
| | Author: Jared Samet <jared.samet@aya.yale.edu>
| | Date:   Fri Dec 9 18:29:22 2016 -0500
| |
| |     Merge pull request #24 from ExtendLang/final-draft-lrm
| |
| |     Almost the final draft!
| |
| *   commit fc886a96416ec9513edbe1afac1c94d39b41ae02
| |\  Merge: eac9e77 cda63cb
|/ /  Author: oracleofnj <jared.samet@aya.yale.edu>
| |   Date:   Fri Dec 9 18:23:52 2016 -0500
| |
| |       Merge branch ’final-draft-lrm’
| |
| *   commit cda63cb3e7aeb7efa06b4af110a9064709b384ac
| |\  Merge: fb18949 eac9e77
| |/  Author: oracleofnj <jared.samet@aya.yale.edu>
|/|   Date:   Fri Dec 9 18:23:24 2016 -0500
| |
| |       Fix merge conflict
| |
* |   commit eac9e77c743d4730ef72b6fa7f4a72304808e35d
|\ \  Merge: 90fc58e fe825f4
| | | Author: Nigel Schuster <Neitsch@users.noreply.github.com>
| | | Date:   Fri Dec 9 18:04:08 2016 -0500
| | |
| | |     Merge pull request #29 from ExtendLang/refactor
| | |
| | |     Made new execution strategy work
| | |
| * | commit fe825f42d95f8d620d082764118657e3791b33ff
| | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | Date:   Fri Dec 9 17:55:39 2016 -0500
| | |
| | |     Compact last bit
| | |
| * | commit b02dbbe930e7ef9bf5a4e454a2e367aa0b50fce1
| | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | Date:   Fri Dec 9 17:49:00 2016 -0500
| | |
| | |     Give formula functions names
| | |
| * | commit edd7aa40ee048a31d84a1f14c73a96d0722c242b
| | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | Date:   Fri Dec 9 17:40:57 2016 -0500
| | |
| | |     Removed artifcats
| | |
| * | commit 9b49e202be310550b9391dc6de3ad1aee961f53a
| | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | Date:   Fri Dec 9 17:37:59 2016 -0500
| | |
| | |     Fixed I/O testcases
| | |
| * |   commit a4ad4b19eae1f9bfaeae27c3a6810b26da856fee
| |\ \  Merge: b07398b ed01567
| | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | | Date:   Fri Dec 9 17:18:13 2016 -0500
| | | |
| | | |     Merge
| | | |
| | * | commit ed01567417fae9a616e32e1d08614fa08424a6bd
| | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | Date:   Fri Dec 9 17:17:06 2016 -0500
| | | |
| | | |     Make sizeof not break tests
| | | |
| * | | commit b07398bd13daed37fc65f0317100f816940257c9
| |/ /  Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | |   Date:   Fri Dec 9 17:17:19 2016 -0500
| | |
| | |       Added macro for function definition
| | |
| * | commit a0a7054e9ce6d82ee760a64a0e3cac3d1ae89c85
| | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | Date:   Fri Dec 9 17:01:20 2016 -0500
| | |
| | |     Use symbol table
| | |
| * |   commit 56fd61b31aa673bc532a40300d61349e0205bb53
| |\ \  Merge: 38aedba dfb702e
| | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | Date:   Fri Dec 9 16:11:10 2016 -0500
| | | |
| | | |     Merge branch ’refactor’ of https://github.com/ExtendLang/Extend into refactor
| | | |
| | * | commit dfb702e023849665c7fc46b0488c85cc8c46c26f
| | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | | Date:   Fri Dec 9 16:01:08 2016 -0500
| | | |
| | | |     Converted more to value_p from subrange_p
| | | |
| | * | commit e963186b1712b1531c8982c35e2ead98e2da3907
| | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | | Date:   Fri Dec 9 15:42:35 2016 -0500
| | | |
| | | |     Made example TC work
| | | |
| * | | commit 38aedba9bfcd2f48a1868fe8f256d9e512dcc0cc
| |/ /  Author: oracleofnj <jared.samet@aya.yale.edu>
| | |   Date:   Fri Dec 9 16:10:35 2016 -0500
| | |
| | |       Create symbol table
| | |
| * | commit eb76234747c9b70a41891853ddfd4730f1950a2c
| | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | Date:   Fri Dec 9 11:14:58 2016 -0500
| | |
| | |     Made Hello World work again
| | |
| * | commit 08aeb70eac43c1fd2e1b8e9f6ead7be11fbc9c12
| | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | Date:   Fri Dec 9 02:13:09 2016 -0500
| | |
| | |     Done for the night
| | |
| * | commit cb39114298fe29c6652e422d86d663482b2d2961
| | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | Date:   Fri Dec 9 01:35:36 2016 -0500
| | |
| | |     More refactoring
| | |
| * | commit 7974bbdf6d9be11657ebf61e534f331efd920ff2
| | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | Date:   Thu Dec 8 23:53:31 2016 -0500
| | |
| | |     Banish the term extern
| | |
| * | commit 49af9724790d0c38ccfcf336c07014f2165948ed
| | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | Date:   Thu Dec 8 23:45:30 2016 -0500
| | |
| | |     Add a couple comments
| | |
| * | commit 0fbf461200018cf1345a9b3ee2b5efd37189c2ce
| | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | Date:   Thu Dec 8 21:52:24 2016 -0500
| | |
| | |     Get my bearings
| | |
| * | commit 5ecb59941a875361553eb25544f92aaaf5231ca0
| | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | Date:   Thu Dec 8 19:47:51 2016 -0500
| | |
| | |     Added some documentation
| | |
| * | commit 65066fc265e8c4e89272ce050c4424a4873b0674
| | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | Date:   Thu Dec 8 12:18:57 2016 -0500
| | |
| | |     Added name display for variable
| | |
| * |   commit f985cc8022350d02fd0c3091aa8e10b85f334c38
| |\ \  Merge: 4b58ce9 a65c24e
| | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | | Date:   Wed Dec 7 12:14:59 2016 –0500
| | | |
| | | |     Merge branch ’finish–transformations’ into get–val–rev
| | | |
| | * |   commit a65c24ed8b30ed639f67913bdd1ffef1b31b0437
| | |\ \  Merge: 29d02d9 90fc58e
| |_|/ /  Author: oracleofnj <jared.samet@aya.yale.edu>
|/| | |   Date:   Tue Dec 6 16:14:10 2016 –0500
| | | |
| | | |       Merge branch ’master’ into finish–transformations
| | | |
| | * |   commit 29d02d923ffa6af98aacbd2375d6e533f6035ccf
| | |\ \  Merge: 0e8398f 52e7a8a
| | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | | Date:   Mon Dec 5 13:34:27 2016 –0500
| | | | |
| | | | |     Fix merge conflict – keep expr_loc
| | | | |
| | * | | commit 0e8398f46809eb8f11d393a663f0096966d2573e
| | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | | Date:   Sun Dec 4 13:23:27 2016 –0500
| | | | |
| | | | |     Transform all LHS expressions including integers to IDs; check for strings or range literals and disallow
| | | | |
| | * | |   commit 36f5848cd9bd8421e6a61e5b79553c5b0aeeac8e
| | |\ \ \  Merge: 330bec3 2ae2b83
| | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | | | Date:   Sat Dec 3 14:07:39 2016 –0500
| | | | | |
| | | | | |     Merge branch ’master’ into finish–transformations
| | | | | |
| | * | | |   commit 330bec3fea9d441236961977886b8c8e7d09df75
| | |\ \ \ \  Merge: 9702e5b 8a60995
| | | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | | | | Date:   Fri Dec 2 13:49:34 2016 –0500
| | | | | | |
| | | | | | |     Merge branch ’master’ into finish–transformations
| | | | | | |
| | * | | | |   commit 9702e5b56018c6187cb0e9421889369f9c867bd6
| | |\ \ \ \ \  Merge: 047cfec 5bdd52c
| | | | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | | | | | Date:   Thu Dec 1 21:14:35 2016 –0500
| | | | | | | |
| | | | | | | |     Merge branch ’master’ into finish–transformations
| | | | | | | |
| | * | | | | | commit 047cfec9cf0a8e645b8b61d52238a107a479db38
| | | | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | | | | | Date:   Thu Dec 1 18:42:04 2016 –0500
| | | | | | | |
| | | | | | | |     Add short circuiting test cases
| | | | | | | |
| | * | | | | |   commit 6acd7f624a1db65cba846ea8745a6457f638ce7c
| | |\ \ \ \ \ \  Merge: 5762112 72360f4
| | | | | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | | | | | | Date:   Thu Dec 1 18:31:33 2016 -0500
| | | | | | | | |
| | | | | | | | |     Merge remote-tracking branch 'origin/fail-silent' into finish-transformations
| | | | | | | | |
| | * | | | | | | commit 57621128490ab146c2c82339fe3edfde4fdb0029
| | | | | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | | | | | | Date:   Thu Dec 1 16:04:06 2016 -0500
| | | | | | | | |
| | | | | | | | |     Get rid of wildcard pattern match in interpreter
| | | | | | | | |
| | * | | | | | |   commit a90a34323a7fe253998bbffb3eee8e8212017587
| | |\ \ \ \ \ \ \  Merge: 85bc21d 7abb394
| | | | | | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | | | | | | | Date:   Thu Dec 1 15:59:40 2016 -0500
| | | | | | | | | |
| | | | | | | | | |     Merge branch 'master' into finish-transformations
| | | | | | | | | |
| | * | | | | | | | commit 85bc21d8c88af6bdce2a7ca4eec8939dad0783a2
| | | | | | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | | | | | | | Date:   Thu Dec 1 15:59:05 2016 -0500
| | | | | | | | | |
| | | | | | | | | |     Remove unnecessary file
| | | | | | | | | |
| | * | | | | | | | commit 81fe5654c032c936924bc103c97d1ae4cd23409c
| | | | | | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | | | | | | | Date:   Thu Dec 1 15:58:40 2016 -0500
| | | | | | | | | |
| | | | | | | | | |     Finish range literals
| | | | | | | | | |
| | * | | | | | | | commit da41520e92a27c3ceeaf63b6fb32bda57d16baf0
| | | | | | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | | | | | | | Date:   Thu Dec 1 14:54:21 2016 -0500
| | | | | | | | | |
| | | | | | | | | |     So close
| | | | | | | | | |
| | * | | | | | | | |   commit 2cabadcf4a488b40a33944794bebe5a3afec6344
| | |\ \ \ \ \ \ \ \  Merge: cf36f70 13cd317
| | | | | | | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | | | | | | | | Date:   Thu Dec 1 11:58:03 2016 -0500
| | | | | | | | | | |
| | | | | | | | | | |     Merge branch 'master' into finish-transformations
| | | | | | | | | | |
| | * | | | | | | | | commit cf36f702053bdb168c51d2ae78d62a013f73cc4d
| | | | | | | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | | | | | | | | Date:   Thu Dec 1 09:47:38 2016 -0500
| | | | | | | | | | |
| | | | | | | | | | |     Sample digits function
| | | | | | | | | | |
| | * | | | | | | | | commit febcff889be83fe6edc63cbffdaefc0d7b02d519
| | | | | | | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | | | | | | | | Date:   Wed Nov 30 15:54:45 2016 -0500
| | | | | | | | | | |
| | | | | | | | | | |     Add oddball formula test case and try out theory for range literal
| | | | | | | | | | |
| | * | | | | | | | | commit 4a1ff4f66645eed55713a02b5d13d330cf65a054
| | | | | | | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | | | | | | | | Date:   Wed Nov 30 14:54:05 2016 -0500
| | | | | | | | | | |
| | | | | | | | | | |     Finish reducing Ternary to ReducedTernary
| | | | | | | | | | |
| | * | | | | | | | | commit 8f0a9816dcc979fcfa348e9e7923b6fa685e1ef7
| | | | | | | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | | | | | | | | Date:   Wed Nov 30 12:35:43 2016 -0500
| | | | | | | | | | |
| | | | | | | | | | |     Working on reducing ternaries
| | | | | | | | | | |
| | * | | | | | | | | commit d3c581201b653c520d2a95b66e59d59dc88684e3
| | | | | | | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | | | | | | | | Date:   Wed Nov 30 02:39:58 2016 -0500
| | | | | | | | | | |
| | | | | | | | | | |     Finish desugaring switch
| | | | | | | | | | |
| | * | | | | | | | | commit 0a22713d988acc28c1e41b556370bcb076782a57
| | | | | | | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | | | | | | | | Date:   Wed Nov 30 00:09:10 2016 -0500
| | | | | | | | | | |
| | | | | | | | | | |     Getting ready to ternarize switch
| | | | | | | | | | |
| | * | | | | | | | | commit 84f016a72cf8ddfc8c1603a8914a7d130a23e7d9
| | | | | | | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | | | | | | | | Date:   Tue Nov 29 21:54:15 2016 -0500
| | | | | | | | | | |
| | | | | | | | | | |     Fix bug in switch() with default case
| | | | | | | | | | |
| | * | | | | | | | | commit d331b7a827290f92b327a315f79a57a294c65185
| | | | | | | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | | | | | | | | Date:   Tue Nov 29 17:33:41 2016 -0500
| | | | | | | | | | |
| | | | | | | | | | |     Give desugaring variables easier-to-read names for debugging purposes
| | | | | | | | | | |
| | * | | | | | | | | commit 36f8de55da1af884e3035da77076dbf4f2058a84
| | | | | | | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | | | | | | | | Date:   Tue Nov 29 16:14:46 2016 -0500
| | | | | | | | | | |
| | | | | | | | | | |     Missed one
| | | | | | | | | | |
| | * | | | | | | | | commit d96da34c254834d2911e6588889f336c25da1748
| | | | | | | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | | | | | | | | Date:   Tue Nov 29 16:13:21 2016 -0500
| | | | | | | | | | |
| | | | | | | | | | |     Transform &&, || into ternary expressions to support proper short-circuit evaluation
| | | | | | | | | | |
| * | | | | | | | | | commit 4b58ce91659ae05d73b6046c41c2fa282dee869a
| | | | | | | | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | | | | | | | | | Date:   Wed Dec 7 12:13:23 2016 -0500
| | | | | | | | | | |
| | | | | | | | | | |     Tried to add more instructions
| | | | | | | | | | |
| * | | | | | | | | | commit 099efe7e1eb8caaf8af5b1a30198ca1b720ad401
| | | | | | | | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | | | | | | | | | Date:   Wed Dec 7 10:48:35 2016 -0500
| | | | | | | | | | |
| | | | | | | | | | |     Making progress on evaluating dimensions
| | | | | | | | | | |
| * | | | | | | | | | commit fa09df7a2ac5d43f34fdc2eaca5a5a7c8fb6c3ee
| | | | | | | | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | | | | | | | | | Date:   Wed Dec 7 09:51:23 2016 -0500
| | | | | | | | | | |
| | | | | | | | | | |     Finally it works
| | | | | | | | | | |
| * | | | | | | | | | commit b265e741e0601cb63dd76cbd739b20631a21638e
| | | | | | | | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | | | | | | | | | Date:   Wed Dec 7 00:41:23 2016 -0500
| | | | | | | | | | |
| | | | | | | | | | |     test commit to look at
| | | | | | | | | | |
| * | | | | | | | | | commit a4554c067dce77cf90af8143310e790643082e89
| | | | | | | | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | | | | | | | | | Date:   Tue Dec 6 23:14:32 2016 -0500
| | | | | | | | | | |
| | | | | | | | | | |     At least it compiles
| | | | | | | | | | |
| * | | | | | | | | | commit 34324847082b94eeded12fca2b820425ee5aabc1
| | | | | | | | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | | | | | | | | | Date:   Tue Dec 6 22:42:22 2016 -0500
| | | | | | | | | | |
| | | | | | | | | | |     Getting closer. Need to add var_defn wrapper in build_formula
| | | | | | | | | | |
| * | | | | | | | | | commit 05145cab5d7bf34581622598567fa14502146e29
| | | | | | | | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | | | | | | | | | Date:   Tue Dec 6 21:10:11 2016 -0500
| | | | | | | | | | |
| | | | | | | | | | |     Minor fix
| | | | | | | | | | |
| * | | | | | | | | | commit 174a7b81a25ac76e94faf513c34700d30e444c01
| | | | | | | | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | | | | | | | | | Date:   Tue Dec 6 11:09:31 2016 -0500
| | | | | | | | | | |
| | | | | | | | | | |     Made partial progress on implementing variable instanciation and such
| | | | | | | | | | |
| * | | | | | | | | | commit 767851d3edbba7585c625da901489d6a51380ab6
| | | | | | | | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | | | | | | | | | Date:   Mon Dec 5 16:18:17 2016 -0500
| | | | | | | | | | |
| | | | | | | | | | |     Finished C side implementation of getVal
| | | | | | | | | | |
| * | | | | | | | | |   commit 6b837d41af7b3872e8d4c3e20da83ce95baf0280
| |\ \ \ \ \ \ \ \ \ \  Merge: 1ce7f83 910bd01
| | | | | | | | | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | | | | | | | | | | Date:   Mon Dec 5 16:06:34 2016 -0500
| | | | | | | | | | | |
| | | | | | | | | | | |     Merge branch 'master' into get-val
| | | | | | | | | | | |
| * | | | | | | | | | | commit 1ce7f83e9cc542daba3b0395f2bebbc8c779e473
| | | | | | | | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | | | | | | | | | Date:   Mon Dec 5 14:18:41 2016 -0500
| | | | | | | | | | | |
| | | | | | | | | | | |     Create patch file
| | | | | | | | | | | |
| * | | | | | | | | | | commit f1b11ee94f4792ff5a6ddf2e837fffe1a3396dbe
| | | | | | | | | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | | | | | | | | | | Date:   Mon Dec 5 12:46:35 2016 -0500
| | | | | | | | | | | |
| | | | | | | | | | | |     Skeleton for get_val
| | | | | | | | | | | |
| | | | | | | | | | | *   commit fb1894999d655ab960f63a01c0c820373b6d8c1a
| | | | | | | | | | | |\  Merge: 4aab3dc 90fc58e
| |_|_|_|_|_|_|_|_|_|_|/  Author: oracleofnj <jared.samet@aya.yale.edu>
|/| | | | | | | | | | |   Date:   Wed Dec 7 23:44:17 2016 -0500
| | | | | | | | | | | |
| | | | | | | | | | | |       Merge branch ’master’ into final-draft-lrm
| | | | | | | | | | | |
* | | | | | | | | | | |   commit 90fc58e52cb1bdec2ea03f5b81c0bbda9c0f954b
|\ \ \ \ \ \ \ \ \ \ \ \  Merge: 910bd01 04c2c65
| |_|/ / / / / / / / / /  Author: Jared Samet <jared.samet@aya.yale.edu>
|/| | | | | | | | | | |   Date:   Mon Dec 5 22:14:41 2016 -0500
| | | | | | | | | | | |
| | | | | | | | | | | |       Merge pull request #23 from ExtendLang/read-empty
| | | | | | | | | | | |
| | | | | | | | | | | |       Read empty
| | | | | | | | | | | |
| * | | | | | | | | | | commit 04c2c655bfe133f98aa03b93739a0545a8dc8e37
| | | | | | | | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | | | | | | | | | Date:   Mon Dec 5 15:53:35 2016 -0500
| | | | | | | | | | | |
| | | | | | | | | | | |     Add slurp by passing 0 max bytes
| | | | | | | | | | | |
| * | | | | | | | | | | commit d8cf3168a0145845d437816b173f36d8e6bbbded
|/ / / / / / / / / / /  Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | | | | | | | |   Date:   Mon Dec 5 14:46:46 2016 -0500
| | | | | | | | | | |
| | | | | | | | | | |       Start handling empty
| | | | | | | | | | |
| | | | | | | | | | * commit 4aab3dcdf8fdef825fc6f517c906c714925fff95
| | | | | | | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | | | | | | | | Date:   Wed Dec 7 23:43:25 2016 -0500
| | | | | | | | | | |
| | | | | | | | | | |     Update PDF
| | | | | | | | | | |
| | | | | | | | | | * commit ed44d27981de0127f73a6cc6f3f32cbf6bcac8ce
| | | | | | | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | | | | | | | | Date:   Wed Dec 7 23:43:01 2016 -0500
| | | | | | | | | | |
| | | | | | | | | | |     Fix failing test cases
| | | | | | | | | | |
| | | | | | | | | | *   commit 9354fa7a2111e943f000edb1409a05f781b14276
| | | | | | | | | | |\  Merge: 0722412 78649f4
| | | | | | | | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | | | | | | | | | Date:   Wed Dec 7 23:06:36 2016 -0500
| | | | | | | | | | | |
| | | | | | | | | | | |     Final draft candidate
| | | | | | | | | | | |
| | | | | | | | | | | * commit 78649f4df74af3ca528e3fe72fafee42fcd9e925
| | | | | | | | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | | | | | | | | | Date:   Wed Dec 7 18:09:46 2016 -0500
| | | | | | | | | | | |
| | | | | | | | | | | |     Almost done
| | | | | | | | | | | |
| | | | | | | | | | | * commit 05ded19fed74d99c33fe3e5333ce190413f11c7c
| | | | | | | | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | | | | | | | | | Date:   Wed Dec 7 15:47:52 2016 -0500
| | | | | | | | | | | |
| | | | | | | | | | | |     More work
| | | | | | | | | | | |
| | | | | | | | | | * | commit 07224121c0ac2f0d2cced8d57d8df81620a3c881
| | | | | | | | | | |/  Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | | | | | | | |   Date:   Wed Dec 7 11:32:11 2016 -0500
| | | | | | | | | | |
| | | | | | | | | | |       Working
| | | | | | | | | | |
| | | | | | | | | | * commit cbb05776c7518c07430d1fb482fd4ee30c70a3e6
| | | | | | | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | | | | | | | | Date:   Wed Dec 7 02:35:06 2016 -0500
| | | | | | | | | | |
| | | | | | | | | | |     Still WIP
| | | | | | | | | | |
| | | | | | | | | | * commit e3c94366de1322d5409194ca08043d6d0cc355f5
| | | | | | | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | | | | | | | | Date:   Wed Dec 7 00:44:22 2016 -0500
| | | | | | | | | | |
| | | | | | | | | | |     WIP
| | | | | | | | | | |
| | | | | | | | | | * commit 18bb1821e80b81c4ec98f834e28f48c838216de8
| | | | | | | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | | | | | | | | Date:   Wed Dec 7 00:35:06 2016 -0500
| | | | | | | | | | |
| | | | | | | | | | |     Still work in progress
| | | | | | | | | | |
| | | | | | | | | | * commit af69b92ab6338b061bf9d8b767b3e2e8b1ab50a4
| | | | | | | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | | | | | | | | Date:   Tue Dec 6 17:23:45 2016 -0500
| | | | | | | | | | |
| | | | | | | | | | |     More updates
| | | | | | | | | | |
| | | | | | | | | | * commit 85a4ccb27d496fbe4c377941e024d0088fae8d1b
| |_|_|_|_|_|_|_|_|/  Author: oracleofnj <jared.samet@aya.yale.edu>
|/| | | | | | | | |   Date:   Tue Dec 6 16:12:31 2016 -0500
| | | | | | | | | |
| | | | | | | | | |       LRM update part 1
| | | | | | | | | |
* | | | | | | | | |   commit 910bd01aacd801d145b0dde2f55a92827c691ae4
|\ \ \ \ \ \ \ \ \ \  Merge: 52e7a8a 88480fb
| |_|_|_|_|_|_|_|_|/  Author: Jared Samet <jared.samet@aya.yale.edu>
|/| | | | | | | | |   Date:   Mon Dec 5 14:27:07 2016 -0500
```

```
| | | | | | | | | |
| | | | | | | | | |       Merge pull request #21 from ExtendLang/fileio
| | | | | | | | | |
| | | | | | | | | |       File IO Stdlib functions
| | | | | | | | | |
| * | | | | | | | | |   commit 88480fbc58a8931e54098dc99ba74f96c9d780bb
| |\ \ \ \ \ \ \ \ \  Merge: bfa906b 52e7a8a
| |/ / / / / / / / /  Author: Nigel Schuster <Neitsch@users.noreply.github.com>
|/| | | | | | | | |   Date:   Mon Dec 5 13:36:28 2016 -0500
| | | | | | | | | |
| | | | | | | | | |       Merge branch 'master' into fileio
| | | | | | | | | |
* | | | | | | | | |   commit 52e7a8a8aa3c5b38ca324a24fa4f0ce15059ab2f
|\ \ \ \ \ \ \ \ \ \  Merge: a5b9066 e4e5e26
| |_|/ / / / / / / /  Author: Jared Samet <jared.samet@aya.yale.edu>
|/| | | | | | | | |   Date:   Mon Dec 5 13:32:54 2016 -0500
| | | | | | | | | |
| | | | | | | | | |       Merge pull request #22 from ExtendLang/rm-micro
| | | | | | | | | |
| | | | | | | | | |       Removed microc
| | | | | | | | | |
| * | | | | | | | | commit e4e5e264cf4369c696f0259c1825552aaee13bc2
|/ / / / / / / / /  Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | | | | | | |   Date:   Mon Dec 5 09:25:17 2016 -0500
| | | | | | | | |
| | | | | | | | |       Removed microc reference implementation
| | | | | | | | |
| * | | | | | | | commit bfa906b7661ba27c0f28bed535033a0f4b5eff06
| | | | | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | | | | | | Date:   Mon Dec 5 13:28:03 2016 -0500
| | | | | | | | |
| | | | | | | | |     Fix off-by-one bug
| | | | | | | | |
| * | | | | | | | commit eb8dd71e3eb2f92d1a82dd19d03b1e090176bbef
| | | | | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | | | | | | Date:   Mon Dec 5 13:20:03 2016 -0500
| | | | | | | | |
| | | | | | | | |     Address issues
| | | | | | | | |
| * | | | | | | |   commit 270da2b340e5f3f2e8240495ff32c4e31e58456b
| |\ \ \ \ \ \ \ \  Merge: b928e98 a5b9066
| |/ / / / / / / /  Author: Ishaan Kolluri <ishaankolluri@users.noreply.github.com>
|/| | | | | | | |   Date:   Mon Dec 5 02:40:59 2016 -0500
| | | | | | | | |
| | | | | | | | |       Merge branch 'master' into fileio
| | | | | | | | |
* | | | | | | | |   commit a5b9066a498c16d2c10664d59e42d71f731959ea
|\ \ \ \ \ \ \ \ \  Merge: 2ae2b83 35e9471
| |_|_|_|_|_|_|_|/  Author: Jared Samet <jared.samet@aya.yale.edu>
|/| | | | | | | |   Date:   Sun Dec 4 14:00:30 2016 -0500
| | | | | | | | |
| | | | | | | | |       Merge pull request #20 from ExtendLang/lhs-all-ids
| | | | | | | | |
| | | | | | | | |       Lhs all ids
| | | | | | | | |
| * | | | | | | | commit 35e947123ee5d51957a79780b78e2cfc816ae0e3
| | | | | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | | | | | | Date:   Sun Dec 4 13:38:44 2016 -0500
| | | | | | | | |
| | | | | | | | |     Put back Id(s) as it was
| | | | | | | | |
| * | | | | | | | commit 641d4546969477380744cdec08904ca7fa061f73
|/ / / / / / / /  Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | | | | |   Date:   Sun Dec 4 13:36:36 2016 -0500
| | | | | | | |
| | | | | | | |       Always transform to ID on LHS, even for LitInts
| | | | | | | |
| * | | | | | | commit b928e9826f816e841ee2ba0a7a59d33a0e1daf0f
| | | | | | | | Author: Ishaan <ishaankolluri@gmail.com>
| | | | | | | | Date:   Mon Dec 5 02:40:10 2016 -0500
| | | | | | | |
| | | | | | | |     Remove bloat
| | | | | | | |
| * | | | | | | commit 894b511acf6e592bbe552b9cf639bc764e2a2342
| | | | | | | | Author: Ishaan <ishaankolluri@gmail.com>
| | | | | | | | Date:   Mon Dec 5 02:32:49 2016 -0500
| | | | | | | |
| | | | | | | |     Added testcase
| | | | | | | |
| * | | | | | | commit 62b8e83bf20e352b8c6ce5523b519e12371b88f5
| | | | | | | | Author: Ishaan <ishaankolluri@gmail.com>
| | | | | | | | Date:   Mon Dec 5 02:30:16 2016 -0500
| | | | | | | |
| | | | | | | |     Added fwrite implementation
| | | | | | | |
| * | | | | | | commit 77a23ae8b083fba49a27adb2a9fdb479258c2b1b
| | | | | | | | Author: Ishaan <ishaankolluri@gmail.com>
| | | | | | | | Date:   Mon Dec 5 01:39:30 2016 -0500
| | | | | | | |
| | | | | | | |     Added read
| | | | | | | |
| * | | | | | | commit 46e9b5893df2c7a7ff36ab9642880517c7ccec3c
| | | | | | | | Author: Ishaan <ishaankolluri@gmail.com>
| | | | | | | | Date:   Mon Dec 5 00:07:16 2016 -0500
| | | | | | | |
| | | | | | | |     Make refactoring changes and new helpers
| | | | | | | |
| * | | | | | | commit f47f2ba753dad63a6232136b2f8be7ceb61c1b44
| | | | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | | | | | Date:   Sun Dec 4 10:30:44 2016 -0500
| | | | | | | |
| | | | | | | |     Add error handling to close() and add a couple test cases
| | | | | | | |
| * | | | | | | commit e95a95a32ec5aca30c1f49fb2aeb48c3b160b935
| | | | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | | | | | Date:   Sun Dec 4 10:07:01 2016 -0500
| | | | | | | |
| | | | | | | |     Add assertSingleNumber and get_number to eliminate more copy & paste
| | | | | | | |
| * | | | | | | commit 543e7209e425928904d9ccbd8299cd518865ffb2
| | | | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | | | | | Date:   Sun Dec 4 09:47:03 2016 -0500
| | | | | | | |
| | | | | | | |     Add new_number() to eliminate some copy and paste
| | | | | | | |
| * | | | | | | commit d7f10c926a7701189e4df2acb12bcac88b86d875
| | | | | | | | Author: Ishaan <ishaankolluri@gmail.com>
| | | | | | | | Date:   Sun Dec 4 02:31:03 2016 -0500
| | | | | | | |
| | | | | | | |     Tentative drafts of fileio functions
| | | | | | | |
| * | | | | | | commit 7d81e43b768ea42ee806a6ad23869f7efacbf8a8
| | | | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | | | | | Date:   Sun Dec 4 00:15:20 2016 -0500
| | | | | | | |
| | | | | | | |     add diagnostic prinfs
| | | | | | | |
| * | | | | | | commit 868d9a4491fe90a050b9875f701ffe6d4d84cef6
| | | | | | | | Author: Ishaan <ishaankolluri@gmail.com>
| | | | | | | | Date:   Sat Dec 3 23:46:01 2016 -0500
| | | | | | | |
| | | | | | | |     Cleanup
| | | | | | | |
| * | | | | | | commit aa1e0148bf45498311b2cb9eb0fe1ebdf38cc07a
| | | | | | | | Author: Ishaan <ishaankolluri@gmail.com>
| | | | | | | | Date:   Sat Dec 3 23:42:46 2016 -0500
| | | | | | | |
| | | | | | | |     Add file pointer array
| | | | | | | |
| * | | | | | | commit 88d05de75293d0073ef4823ffa7f92297b6ccd85
|/ / / / / / /  Author: Ishaan <ishaankolluri@gmail.com>
| | | | | | |   Date:   Sat Dec 3 18:38:34 2016 -0500
| | | | | | |
| | | | | | |       Working on fopen
| | | | | | |
* | | | | | |   commit 2ae2b839a88fad0ed90c1b4b6666e9c37c54a911
|\ \ \ \ \ \ \  Merge: 8a60995 7c78a23
| |_|_|_|_|_|/  Author: Jared Samet <jared.samet@aya.yale.edu>
|/| | | | | |   Date:   Sat Dec 3 14:06:40 2016 -0500
| | | | | | |
| | | | | | |       Merge pull request #15 from ExtendLang/stdlib-fun
| | | | | | |
| | | | | | |       Stdlib fun
| | | | | | |
| * | | | | | commit 7c78a2359f174dd9d3d68a02a23c587186b3e20d
| | | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | | | | Date:   Sat Dec 3 14:02:51 2016 -0500
| | | | | | |
| | | | | | |     Move test_fabs out of regression test suite
| | | | | | |
| * | | | | | commit 0a8055b8851a60af3e6f23d19ebd24f126f1bdf9
| | | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | | | | Date:   Sat Dec 3 13:48:19 2016 -0500
| | | | | | |
| | | | | | |     make test | grep REGRESSION
| | | | | | |
| * | | | | |   commit a24742bca26352fb2f9a255ca1abf4036b975a9d
| |\ \ \ \ \ \  Merge: 5243c5a 8a60995
| |/ / / / / /  Author: Kevin <kevinye1113@gmail.com>
|/| | | | | |   Date:   Fri Dec 2 22:50:43 2016 —0500
| | | | | | |
| | | | | | |       Merged stdlib with master
| | | | | | |
* | | | | | |   commit 8a60995f5d9e25c950066478d4b6f451f0c17a16
|\ \ \ \ \ \ \  Merge: 96a3028 f0d33e2
| | | | | | | | Author: Jared Samet <jared.samet@aya.yale.edu>
| | | | | | | | Date:   Thu Dec 1 23:38:54 2016 —0500
| | | | | | | |
| | | | | | | |     Merge pull request #18 from ExtendLang/parser—error
| | | | | | | |
| | | | | | | |     Parser error
| | | | | | | |
| * | | | | | | commit f0d33e292b39aab8e6d615cb28d6a011697a4115
| | | | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | | | | | Date:   Thu Dec 1 23:18:39 2016 —0500
| | | | | | | |
| | | | | | | |     Move error handling
| | | | | | | |
| * | | | | | | commit 3b24c3a9570d8c03150bf405fcb3c15a407083fd
| | | | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | | | | | Date:   Thu Dec 1 23:16:53 2016 —0500
| | | | | | | |
| | | | | | | |     Adjust test script
| | | | | | | |
| * | | | | | |   commit 60a732fe464f53cb119d5a5d0dca2e10202baf27
| |\ \ \ \ \ \ \  Merge: 5dec6a2 96a3028
| |/ / / / / / /  Author: oracleofnj <jared.samet@aya.yale.edu>
|/| | | | | | |   Date:   Thu Dec 1 22:55:28 2016 —0500
| | | | | | | |
| | | | | | | |       Merge branch ’master’ into parser—error
| | | | | | | |
* | | | | | | |   commit 96a30280911892931866a653d3271e1c8a8eb085
|\ \ \ \ \ \ \ \  Merge: 5bdd52c 7912d5a
| |_|_|_|_|_|_|/  Author: Nigel Schuster <Neitsch@users.noreply.github.com>
|/| | | | | | |   Date:   Thu Dec 1 22:19:21 2016 —0500
| | | | | | | |
| | | | | | | |       Merge pull request #16 from ExtendLang/fail—silent
| | | | | | | |
| | | | | | | |       Reduce testscript output
| | | | | | | |
| * | | | | | |   commit 7912d5ad15e53af8b7fd5d4fa50086489683abe5
| |\ \ \ \ \ \ \  Merge: 72360f4 5bdd52c
| |/ / / / / / /  Author: Nigel Schuster <Neitsch@users.noreply.github.com>
|/| | | | | | /   Date:   Thu Dec 1 21:26:03 2016 —0500
| | |_|_|_|_|/
| |/| | | | |         Merge branch ’master’ into fail—silent
| | | | | | |
| * | | | | | commit 72360f4eb9c1dcf6ef01987c5a3764bb3ffda0d8
| | |_|_|_|/  Author: Nigel Schuster <nigel.schusters@googlemail.com>
| |/| | |     Date:   Thu Dec 1 17:09:08 2016 —0500
| | | | | |
| | | | | |       Minified error output for outputs that have not passed yet
| | | | | |
| | * | | | commit 5dec6a2973176e9aa664adec96fae44e28b8a30b
| | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | | | Date:   Thu Dec 1 22:55:05 2016 -0500
| | | | | |
| | | | | |     Thank you Nigel!!!
| | | | | |
| | * | | | commit 6c3696ce28a857d6cbdca227677d454b0dbd280b
| |/ / / /  Author: oracleofnj <jared.samet@aya.yale.edu>
|/| | | |   Date:   Thu Dec 1 21:59:40 2016 -0500
| | | | |
| | | | |       Figure out why test is failing
| | | | |
* | | | |   commit 5bdd52cc31c181789388acdafba2eb69709658f0
|\ \ \ \ \  Merge: 8c7b6ce 8893255
| | | | | | Author: Jared Samet <jared.samet@aya.yale.edu>
| | | | | | Date:   Thu Dec 1 21:13:45 2016 -0500
| | | | | |
| | | | | |     Merge pull request #17 from ExtendLang/lexbuf-pos
| | | | | |
| | | | | |     Lexbuf pos
| | | | | |
| * | | | | commit 889325556f4f0f43d4a6048fb15ea0b59e02b0fc
| | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | | | Date:   Thu Dec 1 20:35:04 2016 -0500
| | | | | |
| | | | | |     Add a couple test cases
| | | | | |
| * | | | | commit 2868653773273d33ce8d4a5ddb12c02086d7e037
|/ / / / /  Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | |   Date:   Thu Dec 1 20:23:01 2016 -0500
| | | | |
| | | | |       Use lexbuf.lex_curr_p to calculate position
| | | | |
* | | | |   commit 8c7b6cea2bb4fd797d7dda9e2cdcbb819890f23c
|\ \ \ \ \  Merge: 7abb394 2885ac7
| |/ / / /  Author: Ishaan Kolluri <ishaankolluri@users.noreply.github.com>
|/| | | |   Date:   Thu Dec 1 18:59:49 2016 -0500
| | | | |
| | | | |       Merge pull request #11 from ExtendLang/parse_error
| | | | |
| | | | |       Add line numbers to Syntax Errors
| | | | |
| * | | | commit 2885ac7e0d21ca4132922e5e5215155e6bf4c137
| | | | | Author: Ishaan <ishaankolluri@gmail.com>
| | | | | Date:   Thu Dec 1 18:56:15 2016 -0500
| | | | |
| | | | |     Added test case for string
| | | | |
| * | | | commit e9fb1c2dca7114154c79f31f534da8e5cc1d0f7f
| | | | | Author: Ishaan <ishaankolluri@gmail.com>
| | | | | Date:   Thu Dec 1 15:03:29 2016 -0500
| | | | |
| | | | |     Added increment to string buffer and tests
| | | | |
| * | | | commit eb7c1e843628a9714a50a847fca26ce8e4be75c4
| | | | | Author: Ishaan <ishaankolluri@gmail.com>
| | | | | Date:   Thu Dec 1 03:50:53 2016 −0500
| | | | |
| | | | |     Add partial character indexing
| | | | |
| * | | | commit df09aeaa3fb16283f652ab549ab2e246c4d79464
| | | | | Author: Ishaan <ishaankolluri@gmail.com>
| | | | | Date:   Thu Dec 1 00:41:07 2016 −0500
| | | | |
| | | | |     Add expected parse testcase intermediate
| | | | |
| * | | | commit 712a71006cf73e566676393cb09cca2cd84e311e
| | | | | Author: Ishaan <ishaankolluri@gmail.com>
| | | | | Date:   Wed Nov 30 16:42:32 2016 −0500
| | | | |
| | | | |     Added tentative scanner−level line number
| | | | |
| * | | | commit bf4ee6c3bc017d4d2a7140526717edfc6c70a896
|/ / / /  Author: Ishaan <ishaankolluri@gmail.com>
| | | |   Date:   Wed Nov 30 15:53:43 2016 −0500
| | | |
| | | |       Added SyntaxError Exception at scan level
| | | |
* | | |   commit 7abb39400db2ebc10312943f011b363532dbd872
|\ \ \ \  Merge: 13cd317 e0b7fdb
| |_|_|/  Author: Nigel Schuster <Neitsch@users.noreply.github.com>
|/| | |   Date:   Thu Dec 1 14:07:58 2016 −0500
| | | |
| | | |       Merge pull request #14 from ExtendLang/sinner
| | | |
| | | |       Sinner
| | | |
| * | | commit e0b7fdbea24db1a5d6606dfc19848d8755cb931c
| | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | | Date:   Thu Dec 1 14:05:38 2016 −0500
| | | |
| | | |     Rename empty to new_val
| | | |
| * | | commit 6ea8cff367f1465f7903fc249fdd65a94e5fed27
| | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | | Date:   Thu Dec 1 10:10:26 2016 −0500
| | | |
| | | |     Using define instead of magic numbers
| | | |
| * | |   commit cd7d261cc0f4faf9f3fb3178228fce7e81ac551c
| |\ \ \  Merge: 3986f38 13cd317
| |/ / /  Author: Nigel Schuster <nigel.schusters@googlemail.com>
|/| | |   Date:   Thu Dec 1 10:07:10 2016 −0500
| | | |
| | | |       Merge branch ’master’ into sinner
| | | |
* | | |   commit 13cd31794ee3313e0b87a106100fe4e752969a45
|\ \ \ \  Merge: effc20b 4eeed07
| | | | | Author: Nigel Schuster <Neitsch@users.noreply.github.com>
| | | | | Date:   Thu Dec 1 10:06:25 2016 −0500
| | | | |
| | | | |     Merge pull request #13 from ExtendLang/value_p
| | | | |
| | | | |     Moved all function signatures to value_p return value
| | | | |
| * | | | commit 4eeed0719d5cee5cae27e2debf1a26572a2a96ad
| | | | | Author: Ishaan <ishaankolluri@gmail.com>
| | | | | Date:   Thu Dec 1 01:02:56 2016 -0500
| | | | |
| | | | |     Change print return type to empty
| | | | |
| | | * | commit 5243c5a654a69cc7b5239e0236c3318b132fec8e
| | | | | Author: Kevin <kevinye1113@gmail.com>
| | | | | Date:   Fri Dec 2 18:16:36 2016 -0500
| | | | |
| | | | |     Removed magic numbers and add fabs test
| | | | |
| | | * | commit fa42f2718eada7ac73302690aac7a4982e20da4f
| | | | | Author: Kevin <kevinye1113@gmail.com>
| | | | | Date:   Thu Dec 1 00:41:47 2016 -0500
| | | | |
| | | | |     Fixed acos function
| | | | |
| | | * | commit 53d34adc07f06d185d57de7d3ed3901706eeb4f3
| | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | | | Date:   Thu Dec 1 00:29:32 2016 -0500
| | | | |
| | | | |     Moved double values type to numeric
| | | | |
| | | * |   commit f769c61b5da538e4934721de651d2772359fa2c5
| | | |\ \  Merge: 7f0bc86 3986f38
| | | |/ /  Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | |/| |   Date:   Thu Dec 1 00:18:07 2016 -0500
| | | | |
| | | | |       Merge branch ’sinner’ into stdlib-fun
| | | | |
| | * | |   commit 3986f38a3fe7b5cbe849af80e5c00fc7881dc13e
| | |\ \ \  Merge: 4604545 5bd87f9
| | |/ / /  Author: Nigel Schuster <nigel.schusters@googlemail.com>
| |/| | |   Date:   Thu Dec 1 00:17:21 2016 -0500
| | | | |
| | | | |       Merge branch ’value_p’ into sinner
| | | | |
| * | | | commit 5bd87f9233c2b4be9f60cecd0bbab3792df5d8b7
| | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | | | Date:   Wed Nov 30 23:37:58 2016 -0500
| | | | |
| | | | |     Explicitly declaring to link math library
| | | | |
| | * | | commit 4604545c905e2a9a77524cbaeb54a93a9adfb249
| | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | | | Date:   Thu Dec 1 00:12:08 2016 -0500
| | | | |
| | | | |     Consistently using floats
| | | | |
| | * | |   commit 38b9824041ecc7c1b9985cf44436a31f5daa90f3
| | |\ \ \  Merge: e085977 3303575
| | | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | | | | Date:   Wed Nov 30 23:46:14 2016 −0500
| | | | | |
| | | | | |     Merge branch ’value_p’ into sinner
| | | | | |
| | | * | | commit 330357567faf2c6fc8b1c3e769d7aa3e047629de
| | |/ / /  Author: Nigel Schuster <nigel.schusters@googlemail.com>
| |/| | |   Date:   Wed Nov 30 23:37:58 2016 −0500
| | | | |
| | | | |       Explicitly declaring to link math library
| | | | |
| * | | |   commit 31a74ecd91861165dfdcfed780a8d59ba99042e7
| |\ \ \ \  Merge: 206ee5a effc20b
| |/ / / /  Author: Nigel Schuster <nigel.schusters@googlemail.com>
|/| | | |   Date:   Wed Nov 30 23:35:34 2016 −0500
| | | | |
| | | | |       Merge branch ’master’ into value_p
| | | | |
* | | | |   commit effc20bd8f99f841b0ae5a0af19042599db89345
|\ \ \ \ \  Merge: 3a8efbc 3b6d7b7
| |_|_|_|/  Author: Ishaan Kolluri <ishaankolluri@users.noreply.github.com>
|/| | | |   Date:   Wed Nov 30 18:45:52 2016 −0500
| | | | |
| | | | |       Merge pull request #12 from ExtendLang/easy−compile
| | | | |
| | | | |       Added script to compile and link
| | | | |
| * | | | commit 3b6d7b740336c5fefd9a60be7f77351f04ab46de
|/ / / /  Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | |   Date:   Wed Nov 30 17:51:19 2016 −0500
| | | |
| | | |       Added script to compile and link
| | | |
| | | * commit 7f0bc8650a1d6d89ac6ddd1d4862c245e7dfbc7e
| | | | Author: Kevin <kevinye1113@gmail.com>
| | | | Date:   Wed Nov 30 23:04:34 2016 −0500
| | | |
| | | |     Finished remainder of stdlib
| | | |
| | | * commit cd160df84fff7fceb7340b5fad27fe9360c436cd
| | |/  Author: Kevin <kevinye1113@gmail.com>
| | |   Date:   Wed Nov 30 22:50:18 2016 −0500
| | |
| | |       Added more c functions to stdlib
| | |
| | * commit e085977d3edd3785dc682c42161601b6b975cc41
| |/  Author: Nigel Schuster <nigel.schusters@googlemail.com>
| |   Date:   Wed Nov 30 19:59:57 2016 −0500
| |
| |       Made sin function work
| |
| * commit 206ee5ab0f46bc5411cc3834f8107134c6e6e80b
|/  Author: Nigel Schuster <nigel.schusters@googlemail.com>
|   Date:   Wed Nov 30 19:07:28 2016 −0500
|
|       Moved all function signatures to value_p return value
|
*   commit 3a8efbc6575cb2d73b305c087b6f2294cf1c3511
|\  Merge: 5b3edb0 7a2af49
| | Author: Nigel Schuster <Neitsch@users.noreply.github.com>
| | Date:   Mon Nov 28 23:05:28 2016 -0500
| |
| |     Merge pull request #9 from ExtendLang/func-calls
| |
| |     Function calls work now
| |
| * commit 7a2af4937e6abf0621cfabad4ee463dfa7b64e20
| | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | Date:   Mon Nov 28 20:33:53 2016 -0500
| |
| |     Removed another ocaml 4.3 dep
| |
| * commit 468e79f88bcecd2d60c822799c353bc00ee11099
| | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | Date:   Mon Nov 28 19:50:53 2016 -0500
| |
| |     Added ocaml 4.3 as dep for travis (hopefully this works)
| |
| * commit a40876103c19a444466a07019854cf508ccfe276
| | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | Date:   Mon Nov 28 19:35:49 2016 -0500
| |
| |     Fixed String.equal
| |
| * commit 90c3caf1dc0ebf18d2e8709df39be6333266bc56
| | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | Date:   Sun Nov 27 22:52:14 2016 -0500
| |
| |     Fixed interpreter for now
| |
| * commit a18da787c64c95ed8cc31f8f188bb1ef9507453b
| | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | Date:   Sun Nov 27 22:42:27 2016 -0500
| |
| |     Added accidentally created file
| |
| * commit 5647312ef3939b7417783c3ff7620a90d3b002fe
| | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | Date:   Sun Nov 27 22:41:22 2016 -0500
| |
| |     Made extern function calls work
| |
| *   commit 872aa8c4c287ddb35dd5d529df2122e1b38a0fb9
| |\  Merge: 26ef1cc 877336f
| | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | Date:   Sun Nov 27 13:52:44 2016 -0500
| | |
| | |     Merge branch 'func-calls' of https://github.com/ExtendLang/Extend into func-calls
| | |
| | *   commit 877336f09f10848092540a3349bb1a42c9a6ebd9
| | |\  Merge: 374273f 5b3edb0
| |_|/  Author: Nigel Schuster <Neitsch@users.noreply.github.com>
```

```
|/| |   Date:   Sun Nov 27 12:15:11 2016 –0500
| | |
| | |       Merge branch ’master’ into func–calls
| | |
* | |   commit 5b3edb0d0b6e8764cadd9290eab3c9dad7fb782d
|\ \ \  Merge: 442ae91 952aab8
| | | | Author: Nigel Schuster <Neitsch@users.noreply.github.com>
| | | | Date:   Sun Nov 27 12:14:43 2016 –0500
| | | |
| | | |     Merge pull request #8 from ExtendLang/stdlib–template
| | | |
| | | |     Stdlib template
| | | |
| | * | commit 26ef1cc506931d6771cf950fdae016f9f3874bbe
| | |/  Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | |   Date:   Sun Nov 27 13:51:06 2016 –0500
| | |
| | |       Merging list of functions
| | |
| | * commit 374273f2ceb3b69ee489c39c0f3767ab4748f246
| |/  Author: Nigel Schuster <nigel.schusters@googlemail.com>
| |   Date:   Sun Nov 27 12:13:52 2016 –0500
| |
| |       Function calls work now
| |
| *   commit 952aab83190eb5b8e3be609fe44fa0fc0cbe8f2b
| |\  Merge: ac6268f 554fbb2
| | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | Date:   Sun Nov 27 09:54:12 2016 –0500
| | |
| | |     Merge extern
| | |
| | * commit 554fbb28bf7e995117790bd17770af810b1caa37
| | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | Date:   Wed Nov 23 22:28:29 2016 –0500
| | |
| | |     Better error message for WrongNumberArgs
| | |
| | * commit f09e40e405befe48f3b2310c6f4a0aecd48dbcf7
| | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | Date:   Wed Nov 23 12:47:39 2016 –0500
| | |
| | |     Make sequence work
| | |
| | * commit 053980bc62d59316a73fe42efaf1706d8f8d813c
| | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | Date:   Tue Nov 22 16:02:27 2016 –0500
| | |
| | |     Actually commit all the extern stuff
| | |
| * | commit ac6268fd89de976aa5d355e09215f5bab22d128f
| | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | Date:   Sat Nov 26 23:06:00 2016 –0500
| | |
| | |     Boxing ints, added unop sizeof, actually returning subrange not dummy object
| | |
| * | commit ca07be3d9e79a9398b1da5fbc30a22976fbe2246
| | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | Date:   Sat Nov 26 21:27:19 2016 -0500
| | |
| | |     Unboxing hello world to and from subrange
| | |
| * | commit aef6c1964b6e8ca2b0d05d3a5e1efb8a5b0f8159
| | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | Date:   Sat Nov 26 16:55:48 2016 -0500
| | |
| | |     Made Hello World somewhat workable
| | |
| * | commit cfb637efdc1747331c83e6ccfe9f417fb8fc3c1d
| | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | Date:   Fri Nov 25 18:27:37 2016 -0500
| | |
| | |     Fixed faulty setup on call
| | |
| * | commit ebf926a2c3cd85309b6dfac4febc166d381c8ca2
| | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | Date:   Fri Nov 25 17:48:57 2016 -0500
| | |
| | |     Added template in C
| | |
| * | commit 0e0fa23bc1b0fb0bc039fa0d1cbcb5f617f0cc9c
| |/  Author: Nigel Schuster <nigel.schusters@googlemail.com>
| |   Date:   Tue Nov 22 14:36:54 2016 -0500
| |
| |       Added extern in Ast
| |
| * commit aac63be5941852ff37036e5a69d1347ba7744a82
| | Author: oracleofnj <jared.samet@aya.yale.edu>
| | Date:   Mon Nov 21 23:52:25 2016 -0500
| |
| |     Better duplicate definition checking
| |
| * commit 08e2d073c08f1b2067f9e8a130595114815091ae
| | Author: oracleofnj <jared.samet@aya.yale.edu>
| | Date:   Mon Nov 21 23:29:28 2016 -0500
| |
| |     Check assertions before evaluating fn return expression
| |
| * commit 69fa332091dfb61848a9c7b47092c31244e74153
| | Author: oracleofnj <jared.samet@aya.yale.edu>
| | Date:   Mon Nov 21 18:01:23 2016 -0500
| |
| |     Add size assertions
| |
| * commit 22541c427b7883b6fc89841b077ec8c1c9d41c80
| | Author: oracleofnj <jared.samet@aya.yale.edu>
| | Date:   Mon Nov 21 12:48:34 2016 -0500
| |
| |     Fix bug in Call()
| |
| * commit 9a1d24bb89721d76c3987f50d3095b16c182f38e
| | Author: oracleofnj <jared.samet@aya.yale.edu>
| | Date:   Mon Nov 21 12:39:41 2016 -0500
| |
| |     Working on crazy bug
| |
| * commit a485ceef7e987cdef7cd4321d01849bc73342e78
| | Author: oracleofnj <jared.samet@aya.yale.edu>
| | Date:   Sun Nov 20 22:13:46 2016 -0500
| |
| |     Add test case for foo([m, n] arg)
| |
| * commit 10afe9a2fdb645fbb146c724d6281d34d4ba3f8a
| | Author: oracleofnj <jared.samet@aya.yale.edu>
| | Date:   Sun Nov 20 22:07:17 2016 -0500
| |
| |     Expand function signature
| |
| * commit 325e9ba5b275d2d312953e02eb1abca561a31a74
| | Author: oracleofnj <jared.samet@aya.yale.edu>
| | Date:   Sun Nov 20 18:53:52 2016 -0500
| |
| |     Well, this is awkward
| |
| * commit 0a76dc987d319e86ab1e342c5d96bbc624c4bbdb
| | Author: oracleofnj <jared.samet@aya.yale.edu>
| | Date:   Sun Nov 20 18:41:12 2016 -0500
| |
| |     Add check of return value
| |
| * commit 488e34ea0aeebc91443a355a94e9586351d57049
| | Author: oracleofnj <jared.samet@aya.yale.edu>
| | Date:   Sun Nov 20 18:31:39 2016 -0500
| |
| |     Add sample #1
| |
| * commit 93eebc5c3442929d4d2f2a1f99c199538009d707
| | Author: oracleofnj <jared.samet@aya.yale.edu>
| | Date:   Sun Nov 20 18:27:23 2016 -0500
| |
| |     Add semantic checking to make sure functions and variables on RHS exist
| |
| * commit 881f164a15251a781692045ce602dc5153addbb7
|/  Author: oracleofnj <jared.samet@aya.yale.edu>
|   Date:   Sun Nov 20 17:22:40 2016 -0500
|
|       Check RHS slice to ensure end > start, otherwise evaluate to empty
|
*   commit 442ae9121c99adeb30215e524022e1dd046e0e66
|\  Merge: 367bc2b f7f701d
| | Author: Nigel Schuster <Neitsch@users.noreply.github.com>
| | Date:   Sun Nov 20 11:42:54 2016 -0500
| |
| |     Merge pull request #73 from Neitsch/interpreter-global
| |
| |     Added use of global variables to interpreter
| |
| * commit f7f701d56f6e553742adcf396bffd7d0d2021cd2
```

```
|/  Author: Nigel Schuster <nigel.schusters@googlemail.com>
|   Date:   Sun Nov 20 11:30:06 2016 -0500
|
|       Added use of global variables to interpreter, fixed specs for logical or and and testcases with empty
|
*   commit 367bc2b3b07ed593eba6475aa42d3d66bada93c9
|\  Merge: e956238 bdca834
| | Author: Nigel Schuster <Neitsch@users.noreply.github.com>
| | Date:   Sun Nov 20 00:33:17 2016 -0500
| |
| |     Merge pull request #72 from Neitsch/codegen-part-app-fix
| |
| |     Fixed partial function application warning
| |
| *   commit bdca8349c823e69a9065ca2f7efd24d8ba3d3482
| |\  Merge: 9b742d1 e956238
| |/  Author: Nigel Schuster <Neitsch@users.noreply.github.com>
|/|   Date:   Sun Nov 20 00:31:04 2016 -0500
| |
| |       Merge branch 'master' into codegen-part-app-fix
| |
* |   commit e956238e31b3811552c0c4fb1ab6ea10f00107e1
|\ \  Merge: f87cb94 32f2989
| | | Author: Nigel Schuster <Neitsch@users.noreply.github.com>
| | | Date:   Sun Nov 20 00:28:49 2016 -0500
| | |
| | |     Merge pull request #71 from Neitsch/tc-fixes
| | |
| | |     Tc fixes
| | |
| * |   commit 32f2989ad2a169fe36509509b8bf8aeb61e7d8f4
| |\ \  Merge: 05f317a f87cb94
| |/ /  Author: Nigel Schuster <Neitsch@users.noreply.github.com>
|/| |   Date:   Sun Nov 20 00:20:51 2016 -0500
| | |
| | |       Merge branch 'master' into tc-fixes
| | |
* | |   commit f87cb946eff0389be01d20ba5e62ff9882b6b94f
|\ \ \  Merge: 6d73717 842ee5a
| | | | Author: Nigel Schuster <Neitsch@users.noreply.github.com>
| | | | Date:   Sun Nov 20 00:20:35 2016 -0500
| | | |
| | | |     Merge pull request #69 from Neitsch/regression-tests
| | | |
| | | |     Regression tests
| | | |
| * | |   commit 842ee5aa0e12f721fc985689fa2bf1bff66a167b
| |\ \ \  Merge: 214ab9d 6d73717
| |/ / /  Author: Nigel Schuster <Neitsch@users.noreply.github.com>
|/| | |   Date:   Sun Nov 20 00:18:56 2016 -0500
| | | |
| | | |       Merge branch 'master' into regression-tests
| | | |
* | | |   commit 6d73717a9152d479eceeb1778f90baaaca6a9083
|\ \ \ \  Merge: 36f72a1 04e5c4a
```

```
| | | | | Author: Jared Samet <jared.samet@aya.yale.edu>
| | | | | Date:   Sat Nov 19 23:55:35 2016 -0500
| | | | |
| | | | |     Merge pull request #66 from Neitsch/fix-test-cases
| | | | |
| | | | |     Fix test cases
| | | | |
| | * | |   commit 214ab9d5df7fa62828bbb0d1284e761365885a23
| | |\ \ \  Merge: fb31505 5e39ba7
| | | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | | | | Date:   Sat Nov 19 22:10:33 2016 -0500
| | | | | |
| | | | | |     Merge
| | | | | |
| | | * | |   commit 5e39ba7b6f490b9f810c37e8157056ab31151b4c
| | | |\ \ \  Merge: 25263fe 36f72a1
| |_|_|/ / /  Author: Nigel Schuster <nigel.schusters@googlemail.com>
|/| | | | |   Date:   Sat Nov 19 21:55:03 2016 -0500
| | | | | |
| | | | | |       Merge
| | | | | |
| | | * | | commit 25263fe599515a7bc97a37f874d5995371f77b2c
| | | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | | | | Date:   Sat Nov 19 21:50:21 2016 -0500
| | | | | |
| | | | | |     Removed travis from build, removed super verbose output
| | | | | |
| | | * | | commit 0554ad9c6ceb85f2f30929b4b63f1476b9e52780
| | | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | | | | Date:   Sat Nov 19 21:42:28 2016 -0500
| | | | | |
| | | | | |     Using precise lli version
| | | | | |
| | | * | | commit 2825ada0022a5df1aeac7d1871904a16f676cb9c
| | | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | | | | Date:   Fri Nov 18 12:51:33 2016 -0500
| | | | | |
| | | | | |     Added branch to build
| | | | | |
| | | * | | commit aafabb2f0b48af7e1961740edccd63c676fb2595
| | | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | | | | Date:   Fri Nov 18 12:50:56 2016 -0500
| | | | | |
| | | | | |     Verbose output for travis debug
| | | | | |
| | * | | | commit fb315051683e9323f494ec9b4cd42a87103f7ea9
| |/ / / /  Author: Nigel Schuster <nigel.schusters@googlemail.com>
|/| | | |   Date:   Sat Nov 19 22:08:42 2016 -0500
| | | | |
| | | | |       Passing testcases are in separate directory. Output of stats
| | | | |
| | | * | commit 05f317a8560e8cc6f59af840b3e5d97581fec1be
| | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | | | Date:   Sat Nov 19 22:37:36 2016 -0500
| | | | |
| | | | |     Fixed output on TCs
| | | | |
| | | * | commit aa1d974fec852f9d6c2c524eb4e0e585c000d2a8
| | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | | | Date:   Sat Nov 19 22:33:40 2016 –0500
| | | | |
| | | | |     Fixed expected value for ternary
| | | | |
| | | * | commit ab7653af94b6eb3c441ab2aa4cb1a0238cb0103f
| | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | | | Date:   Sat Nov 19 22:32:27 2016 –0500
| | | | |
| | | | |     Fixed import testcases
| | | | |
| | | * | commit 848066cd1ea2e7e580ec4710181b5e36273ea9a8
| | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | | | Date:   Sat Nov 19 22:24:55 2016 –0500
| | | | |
| | | | |     Moved testcase asset to asset folder
| | | | |
| | | * | commit 53c920628b8283b501dd70df16b76c1b668b5a66
| | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | | | Date:   Sat Nov 19 22:21:48 2016 –0500
| | | | |
| | | | |     Corrected use of global variable in test_globals
| | | | |
| | | * | commit 5fe74a8c9883277c3ef1df944623a54f6782030c
| | |/ /  Author: Nigel Schuster <nigel.schusters@googlemail.com>
| |/| |   Date:   Sat Nov 19 22:19:47 2016 –0500
| | | |
| | | |       Fixed expected output for test_access_column_cells
| | | |
| * | | commit 04e5c4a7bc3ea9d79a2cbdffe03011df96b5af6f
| | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | Date:   Sat Nov 19 18:30:32 2016 –0500
| | | |
| | | |     Add more operators to interpreter
| | | |
| * | | commit e4a190c32f50362cae49e27679e8860a1b95f49b
| | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | Date:   Sat Nov 19 17:14:04 2016 –0500
| | | |
| | | |     Add argument to main and remove _expected from filenames
| | | |
| * | |   commit 7cd2b3ae0741a05e19d1f0dc39afb4f3d6be3bff
| |\ \ \  Merge: d1fddfd 36f72a1
| |/ / /  Author: oracleofnj <jared.samet@aya.yale.edu>
|/| | |   Date:   Sat Nov 19 16:53:12 2016 –0500
| | | |
| | | |       Merge branch ’master’ into fix–test–cases
| | | |
| * | |   commit d1fddfd7bf6ba8190ff38aaed5a22cd3b3f85ef4
| |\ \ \  Merge: a9320f3 7b6b79e
| | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | | Date:   Sat Nov 19 16:52:48 2016 –0500
| | | | |
| | | | |     Merge branch ’fix–test–cases’ of https://github.com/Neitsch/plt into fix
```

```
   —test—cases
| | | | |
| | * | |   commit 7b6b79ed80d9356bfea1a038757c746788c41fee
| | |\ \ \  Merge: 24a3625 de262b4
| | | | | | Author: Jared Samet <jared.samet@aya.yale.edu>
| | | | | | Date:   Sat Nov 19 14:31:39 2016 —0500
| | | | | |
| | | | | |     Merge branch ’master’ into fix—test—cases
| | | | | |
| * | | | |   commit a9320f3f249e2b2a342add2c750ee9beb4bec2de
| |\ \ \ \ \  Merge: 24a3625 de262b4
| | |/ / / /  Author: oracleofnj <jared.samet@aya.yale.edu>
| |/| / / /   Date:   Sat Nov 19 14:29:51 2016 —0500
| | |/ / /
| | | | |         Merge branch ’master’ into fix—test—cases
| | | | |
| * | | | commit 24a3625ee0a2ebe90f49563eb355549fe6938905
| | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | | Date:   Sat Nov 19 14:27:48 2016 —0500
| | | | |
| | | | |     Add switch tests
| | | | |
| * | | | commit 75e3f71b55e8f57c4092924b1294a6c717bd223e
| | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | | Date:   Fri Nov 18 20:39:23 2016 —0500
| | | | |
| | | | |     Fix parsing errors in test cases
| | | | |
| | | | * commit 9b742d16a14b81215a51612eb237875017320a97
| |_|_|/  Author: Nigel Schuster <nigel.schusters@googlemail.com>
|/| | |   Date:   Sun Nov 20 00:24:39 2016 —0500
| | | |
| | | |       Fixed partial function application warning
| | | |
* | | |   commit 36f72a190820385610b63dd1c8e9e94a0fc564a9
|\ \ \ \  Merge: de262b4 c46c87b
| |_|/ /  Author: Jared Samet <jared.samet@aya.yale.edu>
|/| | |   Date:   Sat Nov 19 16:49:34 2016 —0500
| | | |
| | | |       Merge pull request #67 from Neitsch/test_cases
| | | |
| | | |       Test cases
| | | |
| * | |   commit c46c87bef9282aae0e4d754d74e77cd1c5399a13
| |\ \ \  Merge: 642ce76 de262b4
| |/ / /  Author: Jared Samet <jared.samet@aya.yale.edu>
|/| | |   Date:   Sat Nov 19 16:47:26 2016 —0500
| | | |
| | | |       Merge branch ’master’ into test_cases
| | | |
* | | |   commit de262b4bbe84505ef0dc460afe5fa4fea8da6ba3
|\ \ \ \  Merge: 7146dce 4e38757
| |_|/ /  Author: Jared Samet <jared.samet@aya.yale.edu>
|/| | |   Date:   Sat Nov 19 14:24:39 2016 —0500
| | | |
| | | |       Merge pull request #60 from Neitsch/box—args
| | | |
| | | |       Box args
| | | |
| * | |   commit 4e3875716c30388817c66fad2df038141b89ca51
| |\ \ \  Merge: faecfa1 7146dce
| |/ / /  Author: Jared Samet <jared.samet@aya.yale.edu>
|/| | |   Date:   Fri Nov 18 16:00:10 2016 -0500
| | | |
| | | |       Merge branch ’master’ into box-args
| | | |
* | | |   commit 7146dce9e2cc8dc459dc51e0dbfeb7dd426551a0
|\ \ \ \  Merge: 124d61e 09cb42f
| |_|_|/  Author: Jared Samet <jared.samet@aya.yale.edu>
|/| | |   Date:   Fri Nov 18 15:59:54 2016 -0500
| | | |
| | | |       Merge pull request #64 from Neitsch/reorg-test
| | | |
| | | |       Reorg test
| | | |
| * | | commit 09cb42f8e184a376bc7a00f69ed57d1d60a30737
| | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | Date:   Fri Nov 18 14:07:39 2016 -0500
| | | |
| | | |     Fix parse difference
| | | |
| * | | commit 39634bbf216c4eb75ed7653687739f3d22bf68a4
| | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | Date:   Fri Nov 18 14:01:21 2016 -0500
| | | |
| | | |     Remove unnecessary files
| | | |
| * | | commit d772725d15628dd48f6542cd70bd234c5d157304
| | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | Date:   Fri Nov 18 14:01:02 2016 -0500
| | | |
| | | |     Make inputs work with interpreter
| | | |
| | * |   commit faecfa1905804786fa0fc08086ef32e37b811964
| | |\ \  Merge: 6f63e89 41a81ce
| | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | | Date:   Fri Nov 18 01:48:44 2016 -0500
| | | | |
| | | | |     Fix merge conflict in box_args
| | | | |
| | | * | commit 41a81cead3637dc09da80d33ba04ee453d873b88
| | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | | Date:   Fri Nov 18 01:40:11 2016 -0500
| | | | |
| | | | |     Move argument boxing into a function
| | | | |
| | | | * commit 642ce765c175416879ca7d17f68f31966a027710
| | | | | Author: Kevin <kevinye1113@gmail.com>
| | | | | Date:   Sat Nov 19 16:39:50 2016 -0500
| | | | |
| | | | |     Fixed helloworld bug
| | | | |
| | | | * commit ac3d7fa3ff769a6b987e3837a05652be00b1057b
| | | | | Author: Kevin <kevinye1113@gmail.com>
| | | | | Date:   Sat Nov 19 16:10:53 2016 -0500
| | | | |
| | | | |     Added corresponding AST result for gcd function
| | | | |
| | | | * commit f483ac7bc551d831ef314f07ecf51eb3c571b184
| | | | | Author: Kevin <kevinye1113@gmail.com>
| | | | | Date:   Fri Nov 18 14:10:32 2016 -0500
| | | | |
| | | | |     Updated print statement for each test
| | | | |
| | | | *   commit f4456f8815d6e4d181920a4800da3257d75edbf7
| | | | |\  Merge: 00aafb7 124d61e
| |_|_|_|/  Author: kevinye1 <kevinye1@users.noreply.github.com>
|/| | | |   Date:   Fri Nov 18 13:17:25 2016 -0500
| | | | |
| | | | |       Merge branch ’master’ into test_cases
| | | | |
* | | | |   commit 124d61e803cd209a0b7edf977473240fa1b450f5
|\ \ \ \ \  Merge: 6f63e89 82cf599
| |/ / / /  Author: Nigel Schuster <Neitsch@users.noreply.github.com>
| | / / /   Date:   Fri Nov 18 12:44:50 2016 -0500
| |/ / /
|/| | |         Merge pull request #61 from Neitsch/reorg-test
| | | |
| | | |         Modify test script to compare interpreter and compiler with expected
| | | |
| * | | commit 82cf59904f569175b2dcd4de51e4faf5ff925d7d
|/ / /  Author: oracleofnj <jared.samet@aya.yale.edu>
| | |   Date:   Fri Nov 18 12:34:57 2016 -0500
| | |
| | |       Modify test script to compare interpreter and compiler with expected
| | |
* | |   commit 6f63e898f49513139ab852300ee219aad742020b
|\ \ \  Merge: 012caaa 088dc45
| |/ /  Author: Jared Samet <jared.samet@aya.yale.edu>
|/| |   Date:   Fri Nov 18 00:48:07 2016 -0500
| | |
| | |       Merge pull request #59 from Neitsch/hello-hello
| | |
| | |       Hello world
| | |
| * |   commit 088dc45e0b59702e41d2a3d4e13ef6eb11f580dc
| |\ \  Merge: f84757b 012caaa
| |/ /  Author: Nigel Schuster <nigel.schusters@googlemail.com>
|/| |   Date:   Fri Nov 18 00:29:45 2016 -0500
| | |
| | |       Merge
| | |
* | |   commit 012caaaf689c35d226b447301bb847b14d645cab
|\ \ \  Merge: 5e63cee 9463afa
| | | | Author: Jared Samet <jared.samet@aya.yale.edu>
| | | | Date:   Fri Nov 18 00:12:40 2016 -0500
| | | |
| | | |     Merge pull request #58 from Neitsch/copy-argv
| | | |
| | | |     Copy argv
| | | |
| * | |   commit 9463afa08811674f462b4a52b8433c8a7c19a7e9
| |\ \ \  Merge: 54858ab bb11d6d
| | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | | Date:   Thu Nov 17 23:12:41 2016 —0500
| | | | |
| | | | |     Merge branch ’copy—argv’ of https://github.com/Neitsch/plt into copy—argv
| | | | |
| * | | | commit 54858ab5bbcc0b1879f4b4d7387d9e7e839e0677
| | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | | Date:   Thu Nov 17 23:11:29 2016 —0500
| | | | |
| | | | |     Add => infix operator to cut down on all the build_struct_gep calls
| | | | |
| | | * | commit f84757bec65346775db43303f5b02754aa155e29
| | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | | | Date:   Fri Nov 18 00:02:34 2016 —0500
| | | | |
| | | | |     Removed unneccessary files
| | | | |
| | | * | commit 18fbff1163bf5660839cc7f2d03ab00299fbec29
| | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | | | Date:   Fri Nov 18 00:01:49 2016 —0500
| | | | |
| | | | |     Removed dummy arg reading, added printing to interpreter — helloworld TC passes
| | | | |
| | | * | commit b866da35db69204510000fd87c33cb6aade0532d
| | |/ /  Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | |   Date:   Thu Nov 17 23:31:42 2016 —0500
| | | |
| | | |       Made hello world work
| | | |
| | * |   commit bb11d6dfa4ba716dfd50347645619d8f79321a66
| | |\ \  Merge: e123652 5e63cee
| |/ / /  Author: Nigel Schuster <Neitsch@users.noreply.github.com>
| | / /   Date:   Thu Nov 17 23:10:24 2016 —0500
| |/ /
|/| |         Merge branch ’master’ into copy—argv
| | |
* | |   commit 5e63ceebeaf8664dfdedb23271f70769609eb4b2
|\ \ \  Merge: cafe20e 4a4a806
| | | | Author: Nigel Schuster <Neitsch@users.noreply.github.com>
| | | | Date:   Thu Nov 17 17:40:31 2016 —0500
| | | |
| | | |     Merge pull request #54 from Neitsch/operation_tests
| | | |
| | | |     Operation tests.
| | | |
| * | |   commit 4a4a8063c0da95694f00b7683f9985f40877cbda
| |\ \ \  Merge: 4b28df2 cafe20e
| |/ / /  Author: Nigel Schuster <Neitsch@users.noreply.github.com>
|/| | |   Date:   Thu Nov 17 17:19:13 2016 —0500
```

```
| | | |
| | | |       Merge branch ’master’ into operation_tests
| | | |
* | | |   commit cafe20e6e0f4dfe020f2c34caafa3f8879a944ee
|\ \ \ \  Merge: d43a87b b728e2e
| | | | | Author: Nigel Schuster <Neitsch@users.noreply.github.com>
| | | | | Date:   Thu Nov 17 17:19:11 2016 —0500
| | | | |
| | | | |     Merge pull request #52 from Neitsch/one—main—arg
| | | | |
| | | | |     Call main() with first argument <empty> in interpreter
| | | | |
| * | | |   commit b728e2e80558871009f78e188adaea92ff9826af
| |\ \ \ \  Merge: e490548 d43a87b
| |/ / / /  Author: Nigel Schuster <Neitsch@users.noreply.github.com>
|/| | | |   Date:   Thu Nov 17 17:16:20 2016 —0500
| | | | |
| | | | |       Merge branch ’master’ into one—main—arg
| | | | |
| * | | |   commit e490548c499782a30071fb192311adea1711422f
| |\ \ \ \  Merge: 79ee3de a4cf367
| | | | | | Author: Jared Samet <jared.samet@aya.yale.edu>
| | | | | | Date:   Thu Nov 17 15:50:35 2016 —0500
| | | | | |
| | | | | |     Merge branch ’master’ into one—main—arg
| | | | | |
| * | | | | commit 79ee3dea137cf7fd4ace406716195cf758cabd75
| | | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | | | Date:   Thu Nov 17 15:18:58 2016 —0500
| | | | | |
| | | | | |     Call main() with first argument <empty> in interpreter
| | | | | |
| | | * | |   commit 4b28df26b724eccc92829ca84bd1ccdbc84c6729
| | | |\ \ \  Merge: 3255e1b d43a87b
| |_|_|/ / /  Author: Nigel Schuster <Neitsch@users.noreply.github.com>
|/| | | | |   Date:   Thu Nov 17 17:17:44 2016 —0500
| | | | | |
| | | | | |       Merge branch ’master’ into operation_tests
| | | | | |
* | | | | |   commit d43a87b8ab2c8e52d42c16af44d90a48301752c7
|\ \ \ \ \ \  Merge: a4cf367 b1238a0
| |_|/ / / /  Author: Nigel Schuster <Neitsch@users.noreply.github.com>
|/| | | | |   Date:   Thu Nov 17 17:15:28 2016 —0500
| | | | | |
| | | | | |       Merge pull request #55 from Neitsch/shell—fix
| | | | | |
| | | | | |       Using bourne shell style redirection:
| | | | | |
| * | | | | commit b1238a0145da88327094dd2fa59a243cc72856c2
| | | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | | | | Date:   Thu Nov 17 17:08:56 2016 —0500
| | | | | |
| | | | | |     Shell is not my strength
| | | | | |
| * | | | | commit a6cc0eafe87931aa48498436441c6351ea1bfe0c
| | | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | | | | Date:   Thu Nov 17 17:05:09 2016 -0500
| | | | | |
| | | | | |     Screw you bourne shell
| | | | | |
| * | | | | commit 51fbe679918ad80440986f53208c9d393a98f2bb
|/ / / / /  Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | | |   Date:   Thu Nov 17 16:59:50 2016 -0500
| | | | |
| | | | |       Using bourne shell style redirection:
| | | | |
| | * | | commit 3255e1b771252f98fbbffc2067744e7761ee4a2e
| | | | | Author: Ishaan <ishaankolluri@gmail.com>
| | | | | Date:   Thu Nov 17 16:32:12 2016 -0500
| | | | |
| | | | |     Modify test suite specs
| | | | |
| | * | | commit f0ab4d8880196a8f693891ef5ab6c8510e9a305d
| | | | | Author: Ishaan <ishaankolluri@gmail.com>
| | | | | Date:   Thu Nov 17 16:30:42 2016 -0500
| | | | |
| | | | |     Moved expected output text files to directory
| | | | |
| | * | | commit 06d330c8e0470daaeefa0c06ad7b89a31f2ea4e5
| |/ / /  Author: Ishaan <ishaankolluri@gmail.com>
|/| | |   Date:   Thu Nov 17 16:07:02 2016 -0500
| | | |
| | | |       75% through operator cases
| | | |
| | * | commit e123652c39885430a3ae0d6773b186cd102d37ac
| | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | Date:   Thu Nov 17 22:28:12 2016 -0500
| | | |
| | | |     Add byte for zero
| | | |
| | * | commit 26a03b7e32173ed2ef947084d16e25eaa82ff921
| |/ /  Author: oracleofnj <jared.samet@aya.yale.edu>
|/| |   Date:   Thu Nov 17 22:24:17 2016 -0500
| | |
| | |       Add new_string function
| | |
* | |   commit a4cf367633dd1dba496a2ec83472d8f513f56aaf
|\ \ \  Merge: 7af929a c4f7437
| |/ /  Author: Jared Samet <jared.samet@aya.yale.edu>
|/| |   Date:   Thu Nov 17 15:50:29 2016 -0500
| | |
| | |       Merge pull request #51 from Neitsch/test-script
| | |
| | |       Test script
| | |
| * | commit c4f74379a669e99fa7941712f9d5f3406bd2a390
| | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | Date:   Thu Nov 17 14:39:38 2016 -0500
| | |
| | |     Removed version specific lli
| | |
| * | commit 7b2236b3671a30fe5c49dcd15de8e1d7ec40fd7d
| | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | Date:   Thu Nov 17 14:35:55 2016 —0500
| | |
| | |     Fixed if no flag is given
| | |
| * | commit e10f6566f1d0a786eb103f542bd5572a7dc049b1
| | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | Date:   Thu Nov 17 14:24:20 2016 —0500
| | |
| | |     Outputting diff only if —p flag is given
| | |
| * | commit 2d2959718512cd1c4ab4d8618595e0fbf5a1afb2
| | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | Date:   Thu Nov 17 14:19:30 2016 —0500
| | |
| | |     Added it as build target
| | |
* | |   commit 7af929a4a10c0b447752ddb02b189ff74e48c7ed
|\ \ \  Merge: 7feb392 6ea43f6
| |/ /  Author: Jared Samet <jared.samet@aya.yale.edu>
| | |   Date:   Thu Nov 17 14:12:19 2016 —0500
| | |
| | |       Merge pull request #50 from Neitsch/test—script
| | |
| | |       Test script
| | |
| * | commit 6ea43f63f5e0e05287b1b7c85d60b25a3fa471d9
| | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | Date:   Thu Nov 17 13:54:55 2016 —0500
| | |
| | |     Added more env variables to avoid copy paste
| | |
| * | commit 05f27a219ca18a846664c65754bf682a6bc3c58d
| | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | Date:   Thu Nov 17 12:45:11 2016 —0500
| | |
| | |     Made simple testscript
| | |
| * | commit aca43c139c3b3a1c925473669f664b41acc7c0df
|/ /  Author: Nigel Schuster <nigel.schusters@googlemail.com>
| |   Date:   Thu Nov 17 11:08:11 2016 —0500
| |
| |       Removed accidentally added files
| |
| * commit 00aafb71be96f422170e12cd38bd29a624cae40f
| | Author: Kevin <kevinye1113@gmail.com>
| | Date:   Fri Nov 18 13:16:08 2016 —0500
| |
| |     Renamed inputs folder
| |
| * commit 99db652ae157dce96aa01290d6a492e4467e63ed
| | Author: Kevin <kevinye1113@gmail.com>
| | Date:   Fri Nov 18 12:51:40 2016 —0500
| |
| |     Renamed expected output extension and created input folder for test cases
| |
| * commit b8028f9b923d3cdd5600088f2582c8b29bc9bc5b
| | Author: Kevin <kevinye1113@gmail.com>
| | Date:   Thu Nov 17 20:27:37 2016 —0500
| |
| |     Removed files from test folder
| |
| * commit c85d9b71b9b5073e2f2e2b2ecaf5253c2ffc1203
| | Author: Kevin <kevinye1113@gmail.com>
| | Date:   Thu Nov 17 20:25:21 2016 —0500
| |
| |     Move testcases to testcases directory
| |
| * commit f17c6b67700996588a2696886faba09bc5484880
| | Author: Kevin Ye <kevinye1113@gmail.com>
| | Date:   Thu Nov 17 20:21:38 2016 —0500
| |
| |     Complete testcases for List/Range/Function/Expression with expected outputs
| |
| * commit 9228eacadf83a6c901ce34947cad67eceb1dabca
|/  Author: Kevin Ye <kevinye1113@gmail.com>
|   Date:   Thu Nov 17 04:52:31 2016 —0500
|
|       Test cases for List of Tests and Range/Function/Expression Tests
|
*   commit 7feb392e1193b63f71f99995db0981330668253a
|\  Merge: 41ef578 6e42afa
| | Author: Ishaan Kolluri <ishaankolluri@users.noreply.github.com>
| | Date:   Thu Nov 17 00:28:53 2016 —0500
| |
| |     Merge pull request #48 from Neitsch/testing_list
| |
| |     Added initial testing list
| |
| *   commit 6e42afa596a96286149edacb4a44a0ff0c6f2a47
| |\  Merge: e40734b 41ef578
| |/  Author: Ishaan Kolluri <ishaankolluri@users.noreply.github.com>
|/|   Date:   Thu Nov 17 00:27:13 2016 —0500
| |
| |       Merge branch ’master’ into testing_list
| |
* |   commit 41ef57814b0e03cb890a26a11021150a1cb6793f
|\ \  Merge: 1570836 3cbf089
| | | Author: Jared Samet <jared.samet@aya.yale.edu>
| | | Date:   Wed Nov 16 17:50:03 2016 —0500
| | |
| | |     Merge pull request #49 from Neitsch/consume—command—line—args
| | |
| | |     Consume command line args
| | |
| * |   commit 3cbf089e698f117fe7c093c0ad965ae75f7f5060
| |\ \  Merge: 2fa73be 1570836
| |/ /  Author: oracleofnj <jared.samet@aya.yale.edu>
|/| |   Date:   Wed Nov 16 17:45:58 2016 —0500
| | |
| | |       Fix merge conflict
| | |
* | |   commit 1570836bf3643d233e52d1ceb1d46e072e37222a
|\ \ \  Merge: 4a1fcac a8fbced
| | | | Author: Nigel Schuster <Neitsch@users.noreply.github.com>
| | | | Date:   Wed Nov 16 16:51:05 2016 -0500
| | | |
| | | |     Merge pull request #45 from Neitsch/doc
| | | |
| | | |     Added a little documentation
| | | |
| * | | commit a8fbcedf43a46b3e398838e15b1f713ba5874ad3
| | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | | Date:   Wed Nov 16 16:38:49 2016 -0500
| | | |
| | | |     Fixed minor syntax error
| | | |
| * | |   commit c2f37c8a801eda2b33c09d24ab762e10c0336168
| |\ \ \  Merge: 92fb7a3 4a1fcac
| |/ / /  Author: Nigel Schuster <nigel.schusters@googlemail.com>
|/| | |   Date:   Wed Nov 16 16:30:43 2016 -0500
| | | |
| | | |       Merge
| | | |
| * | | commit 92fb7a326d3811b20ca2f64938fc46e3879dc090
| | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | | Date:   Tue Nov 15 23:57:37 2016 -0500
| | | |
| | | |     Added a little documentation
| | | |
| | * | commit 2fa73be6cebe75a58364148689a0b496940cccd9
| | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | Date:   Wed Nov 16 16:05:37 2016 -0500
| | | |
| | | |     Set return code to length of argv[1]
| | | |
| | * | commit cd0d156ceeab4830df0f3a30b3265f34d3c07c68
| |/ /  Author: oracleofnj <jared.samet@aya.yale.edu>
|/| |   Date:   Wed Nov 16 15:50:39 2016 -0500
| | |
| | |       Start processing command line args
| | |
| | * commit e40734b736193f27f346eda80f4748132316db43
| | | Author: Ishaan <ishaankolluri@gmail.com>
| | | Date:   Wed Nov 16 23:25:01 2016 -0500
| | |
| | |     Added more test scenarios
| | |
| | * commit bc21af622d3a32bb6c9fe2e86f83129886b8e574
| |/  Author: Ishaan <ishaankolluri@gmail.com>
|/|   Date:   Wed Nov 16 15:54:12 2016 -0500
| |
| |       Added initial testing list
| |
* |   commit 4a1fcacf86ee892bc72582fd2ea3a9dc74d0e63c
|\ \  Merge: 8944b9a f1b481e
| | | Author: Nigel Schuster <Neitsch@users.noreply.github.com>
| | | Date:   Wed Nov 16 13:55:46 2016 -0500
| | |
| | |     Merge pull request #46 from Neitsch/number-type
| | |
| | |     Added number type that defaults to int
| | |
| * | commit f1b481efad2233c37a8722580c655266116dad55
|/ /  Author: Nigel Schuster <nigel.schusters@googlemail.com>
| |   Date:   Wed Nov 16 11:04:44 2016 -0500
| |
| |       Added number type that defaults to int
| |
* |   commit 8944b9a1fb299e43a8de33dfb342243f75f7718a
|\ \  Merge: fa1741a bcbde36
| |/  Author: Nigel Schuster <Neitsch@users.noreply.github.com>
|/|   Date:   Wed Nov 16 00:19:33 2016 -0500
| |
| |       Merge pull request #44 from Neitsch/fix-arg
| |
| |       Using subranges instead of ranges everywhere
| |
| *   commit bcbde3610ba04bc8b125f34a502ad20f1ae80bd8
| |\  Merge: 57b2162 fa1741a
| |/  Author: Nigel Schuster <Neitsch@users.noreply.github.com>
|/|   Date:   Tue Nov 15 23:49:07 2016 -0500
| |
| |       Merge branch ’master’ into fix-arg
| |
* |   commit fa1741a24c0bd75650d06a79fee7ea1401382c0a
|\ \  Merge: 660c049 9407677
| | | Author: Jared Samet <jared.samet@aya.yale.edu>
| | | Date:   Tue Nov 15 23:03:23 2016 -0500
| | |
| | |     Merge pull request #43 from Neitsch/more-llvm-gen-js
| | |
| | |     More llvm gen from JS
| | |
| * | commit 9407677c52644d935175cf5abe25a31127bf1787
| | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | Date:   Tue Nov 15 22:31:03 2016 -0500
| | |
| | |     Add hash table for common functions and add dereference-the-range
| | |
| * | commit 46e1fd5752c75220787e60a50a8c51064e50c7ce
|/ /  Author: oracleofnj <jared.samet@aya.yale.edu>
| |   Date:   Tue Nov 15 21:38:51 2016 -0500
| |
| |       Eliminate some copy & paste
| |
| * commit 57b2162706267d8ae19c759d4ef4b1199f9946e3
|/  Author: Nigel Schuster <nigel.schusters@googlemail.com>
|   Date:   Tue Nov 15 22:39:38 2016 -0500
|
|       Using subranges instead of ranges everywhere
|
*   commit 660c049fb22baac3526cd251b0b366ee3d1b8305
|\  Merge: 8ad5a19 25b23cd
```

```
| | Author: Jared Samet <jared.samet@aya.yale.edu>
| | Date:   Tue Nov 15 20:54:33 2016 —0500
| |
| |     Merge pull request #42 from Neitsch/llvm—gen
| |
| |     Llvm gen
| |
| * commit 25b23cd3df1ebe93a00a330b2e02166d41e4dfea
| | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | Date:   Tue Nov 15 17:23:54 2016 —0500
| |
| |     Fixed column retrieval for 1x1
| |
| * commit 3f022031254b6da7a3278153288a09d052d16fa2
| | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | Date:   Tue Nov 15 17:17:02 2016 —0500
| |
| |     Fixed tests
| |
| *   commit 26b8fcf1a26394ee6c82f447330ef2510aa32dc8
| |\  Merge: e347a87 aed28b3
| | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | Date:   Tue Nov 15 17:15:08 2016 —0500
| | |
| | |     Merge
| | |
| | * commit aed28b3e2ad836558e1ecdb4afb276f5bd022114
| | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | Date:   Tue Nov 15 17:08:07 2016 —0500
| | |
| | |     Add is_subrange_1x1
| | |
| | *   commit cf5cbf0613395995f0ee23f19ab4a1a879364385
| | |\  Merge: c71d469 4b34abd
| | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | Date:   Tue Nov 15 14:51:40 2016 —0500
| | | |
| | | |     Merge branch ’llvm—gen’ of https://github.com/Neitsch/plt into llvm—gen
| | | |
| | | *   commit 4b34abd125b1651df31211d8779b7c549d515ae7
| | | |\  Merge: a80a6d0 8ad5a19
| |_|_|/  Author: Nigel Schuster <Neitsch@users.noreply.github.com>
|/| | |   Date:   Tue Nov 15 14:41:37 2016 —0500
| | | |
| | | |       Merge branch ’master’ into llvm—gen
| | | |
* | | |   commit 8ad5a19ed1d3e25da7ee9995fa1079fab3d08cb5
|\ \ \ \  Merge: d6daff3 3f0362a
| | | | | Author: Jared Samet <jared.samet@aya.yale.edu>
| | | | | Date:   Tue Nov 15 14:33:40 2016 —0500
| | | | |
| | | | |     Merge pull request #40 from Neitsch/interpreter
| | | | |
| | | | |     Interpreter
| | | | |
| * | | |   commit 3f0362aff5a44f3c39f0d991c5cc11b85dc7c47e
| |\ \ \ \  Merge: d5f4024 d6daff3
| |/ / / /  Author: Jared Samet <jared.samet@aya.yale.edu>
|/| | | |   Date:   Tue Nov 15 14:28:44 2016 –0500
| | | | |
| | | | |       Merge branch ’master’ into interpreter
| | | | |
* | | | |   commit d6daff3a9c877c8277f1bd86f080f0defa597ead
|\ \ \ \ \  Merge: 443a818 6afe599
| | | | | | Author: Jared Samet <jared.samet@aya.yale.edu>
| | | | | | Date:   Sun Nov 13 20:26:14 2016 –0500
| | | | | |
| | | | | |     Merge pull request #41 from Neitsch/LRM_String_Update
| | | | | |
| | | | | |     Added changes relating to strings.
| | | | | |
| * | | | | commit 6afe5996ee757b714490685718f04027e04532ab
| | | | | | Author: Ishaan Kolluri <ishaankolluri@gmail.com>
| | | | | | Date:   Sun Nov 13 18:44:58 2016 –0500
| | | | | |
| | | | | |     Added changes relating to strings.
| | | | | |
| | | | * | commit c71d469cf691b624960ee4657697d623f0055f55
| | | | |/  Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | |   Date:   Tue Nov 15 14:51:19 2016 –0500
| | | | |
| | | | |       Replace String.equal with =
| | | | |
| | | | * commit a80a6d0053c71eb473fbbe5b3cdcaca004be4b48
| | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | | Date:   Tue Nov 15 14:41:07 2016 –0500
| | | | |
| | | | |     Add compile option to main
| | | | |
| | | * | commit e347a876c8713deff609324bae48d55ba8a69f04
| | | |/  Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | |   Date:   Tue Nov 15 17:12:26 2016 –0500
| | | |
| | | |       Using more generic flag for values
| | | |
| | | *   commit c0c95a24323efbbe0abab2632671275c6ba0854a
| | | |\  Merge: aa61ac9 d5f4024
| | | |/  Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | |/|   Date:   Tue Nov 15 14:16:13 2016 –0500
| | | |
| | | |       Merge
| | | |
| | * | commit d5f4024aab33d3307ddea75005f9885dc03395dc
| | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | | Date:   Tue Nov 15 13:44:44 2016 –0500
| | | |
| | | |     Moved failing TCs
| | | |
| | * | commit 42fd9ef88b59f80351a1170a3c5fc3d393ad55b6
| | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | Date:   Tue Nov 15 12:21:57 2016 –0500
| | | |
| | | |     Fix bug in import
| | | |
| | * | commit 9c567c954d5e4f576f1ffaec3c5f152e86761c65
| | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | | Date:   Tue Nov 15 11:11:30 2016 -0500
| | | |
| | | |     Working on imports, fixed most testcases
| | | |
| | * | commit cf1ebf94b4dfacc0163e479d2ed6979146b47e2e
| | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | Date:   Sun Nov 13 23:09:30 2016 -0500
| | | |
| | | |     Rewrite main to take options; fix bug where import didn’t know about first filename
| | | |
| | * | commit 0a5d484dba0b5e288e3b2bad4d6a4c044e1f8da9
| | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | Date:   Sun Nov 13 18:45:29 2016 -0500
| | | |
| | | |     Revert "Generating function header"
| | | |
| | | |     This reverts commit f83a0bcc6ca8d58d79bdfb4b49471c6031e8d201.
| | | |
| | * |   commit 137d7e2b95b7340894b1d826dc38ef91a958e731
| | |\ \  Merge: 118bfc5 bf1d8bb
| | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | | Date:   Sun Nov 13 18:39:33 2016 -0500
| | | | |
| | | | |     Merge branch ’interpreter’ of https://github.com/Neitsch/plt into interpreter
| | | | |
| | * | | commit 118bfc54ba67ecba87442b26112557c59868220e
| | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | | Date:   Sun Nov 13 18:38:34 2016 -0500
| | | | |
| | | | |     Allow single slice on RHS; make hashtag work
| | | | |
| | * | | commit 3addcc8b8c5029a7635a4b7edb5d718917618304
| | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | | Date:   Sun Nov 13 14:38:11 2016 -0500
| | | | |
| | | | |     Make size(expr) an operator instead of built-in function
| | | | |
| | * | | commit 9a74e14abc798cab3268420451b7b9f3a9a2b7aa
| | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | | Date:   Sun Nov 13 14:22:44 2016 -0500
| | | | |
| | | | |     Changing size() to be an operator
| | | | |
| | | | * commit aa61ac98d42571ab91a54ff06cb80be3deaa76b7
| | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | | | Date:   Tue Nov 15 09:31:42 2016 -0500
| | | | |
| | | | |     Allocating scope object
| | | | |
| | | | * commit 574953833ac17b8d9c86faa967568ee8a27c7c22
| | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | | | Date:   Sun Nov 13 21:59:28 2016 -0500
| | | | |
| | | | |     Added main function
| | | | |
| | | | * commit e376270c9b5af196cac37f5cd2f64071b9768ace
| | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | | | Date:   Sun Nov 13 17:55:41 2016 -0500
| | | | |
| | | | |     Added type arguments for functions
| | | | |
| | | | * commit 5cfb5199f5effd2ce86663bfce7ba388a169bb6f
| | | |/  Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | |   Date:   Sun Nov 13 17:26:23 2016 -0500
| | | |
| | | |       Set more types up
| | | |
| | | *   commit bf1d8bb2b3965e4aeab26a1f4d1c91569c2eb6f2
| | | |\  Merge: f83a0bc d6d2eaa
| | | |/  Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | |/|   Date:   Sun Nov 13 15:30:35 2016 -0500
| | | |
| | | |       Merge branch ’interpreter’ of https://github.com/Neitsch/plt into interpreter
| | | |
| | * | commit d6d2eaa5440fbb39958b104a520197857514ea06
| | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | Date:   Sun Nov 13 00:08:41 2016 -0500
| | | |
| | | |     Add closure to interpreter_variable
| | | |
| | * | commit 64fba825d3c3d6eb216bfa6228cab91c8985d048
| | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | Date:   Sat Nov 12 22:38:39 2016 -0500
| | | |
| | | |     Added bsearch to show logic bug
| | | |
| | * | commit 66ffdb1cdd48dfdb93653a30e07ea47573fbb42c
| | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | Date:   Sat Nov 12 19:21:07 2016 -0500
| | | |
| | | |     Add alpha version of function calls
| | | |
| | * | commit 376b29ae1f4b72fedb438e13b5fd1b3373c827bb
| | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | Date:   Sat Nov 12 17:17:23 2016 -0500
| | | |
| | | |     Add string as value type
| | | |
| | * | commit 08c61ee7724a15c9f95f1afbaabfe69aff63c2a4
| | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | Date:   Sat Nov 12 17:14:47 2016 -0500
| | | |
| | | |     Clean up discrepancies
| | | |
| | | * commit f83a0bcc6ca8d58d79bdfb4b49471c6031e8d201
| | |/  Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | |   Date:   Sun Nov 13 15:30:28 2016 -0500
| | |
| | |       Generating function header
| | |
| | * commit a18d5fc571d7ad30c4720b2a2ab5a58865726120
| | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | Date:   Tue Nov 8 11:38:22 2016 -0500
| | |
| | |     Fix bug with x[-1]
| | |
| | * commit 962f81284eea2537d5b92451fb5494e52c1600a1
| | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | Date:   Mon Nov 7 23:27:08 2016 -0500
| | |
| | |     Refactor scope for interpreter; resolve variables on demand; make selections work properly
| | |
| | * commit 47bbef19ea8c13e555d5751a1ae4f369b8d0007d
| | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | Date:   Sun Nov 6 22:05:55 2016 -0500
| | |
| | |     Minor adjustments to interpreter to work with mapped AST
| | |
| | * commit fddc6bc63b746a5aa19fb96fc1100a9da62ef94f
| | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | Date:   Sun Nov 6 18:32:17 2016 -0500
| | |
| | |     Eliminate extraneous nulls in JSON
| | |
| | * commit ffddb17fb3d6995f44b24199a1fa6a45289ec8a3
| | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | Date:   Sun Nov 6 18:15:40 2016 -0500
| | |
| | |     Turn statement and function lists into StringMaps
| | |
| | |     * Lists of statements are organized into StringMaps, mapping
| | |     * variable names to (dimensions, [formula assignments])
| | |     * Each dimension is either an integer or an ID (no "Some")
| | |     * For each formula assignment, row_start, row_end, col_start, col_end
| | |     * are now either integers or IDs
| | |     *
| | |     * Lists of functions are organized into StringMaps, mapping function
| | |     * names to (parameters, StringMap{variable name: def}, return val)
| | |     *
| | |     * Throws an exception if duplicate variable definitions or function
| | |     * definitions are encountered
| | |     *
| | |     * Throws an exception if an assignment to an unknown variable
| | |     * is encountered
| | |
| | * commit 6810003e25ead7a1cb013849cb94600ab501dae8
| | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | Date:   Sat Nov 5 19:47:57 2016 -0400
| | |
| | |     Fix pattern matching warning
| | |
| | * commit 7107a46504b717448dcd1f51a969fe5d0fc289e8
| | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | Date:   Sat Nov 5 18:01:34 2016 -0400
| | |
| | |     Add function to check range literals for legality at parse time
| | |
| | * commit 80b13d19d1c44db5109f6bcaf2a484175a0340a6
| | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | Date:   Sat Nov 5 15:13:10 2016 -0400
| | |
| | |     Handle selections better
| | |
| | * commit 6cbb009ff6e6d1cee972a5cf79520797c685a52f
| | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | Date:   Fri Nov 4 15:48:58 2016 -0400
| | |
| | |     Count to 1,000,000 using tail-recursive versions of List.map and cartesian product
| | |
| | * commit 9b2252d163cb36dee96cc654b3b33a2adb96825e
| | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | Date:   Fri Nov 4 15:25:13 2016 -0400
| | |
| | |     Show enter and exit
| | |
| | * commit 3585e432acf0c6aef34585cad6ef491f1e0aa1cd
| | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | Date:   Fri Nov 4 02:21:38 2016 -0400
| | |
| | |     See how high it can count recursively
| | |
| | * commit 38cf541bfd31e7fe9458c287c39c639ddda2792f
| | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | Date:   Fri Nov 4 02:15:50 2016 -0400
| | |
| | |     Get the easy parts of the interpreter working
| | |
| | * commit 5d81d6ebc492cd13db7ea52a35f104512a5873c9
| | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | Date:   Thu Nov 3 17:17:51 2016 -0400
| | |
| | |     Start working on interpreter
| | |
| | * commit 0078cee2ecbb431212849db233909580a2bb8c41
| | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | Date:   Tue Nov 1 23:40:57 2016 -0400
| | |
| | |     Got a non-tail-recursive version of topological sort working
| | |
| | * commit 85df17519a57bf08f21d20c97c7c768e294673ac
| | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | Date:   Tue Nov 1 15:39:10 2016 -0400
| | |
| | |     Irrelevant highlighting thing
| | |
| | * commit 84c719aaaf45a702e2ad81ffb08b52505d780b3f
| | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | Date:   Tue Nov 1 14:39:49 2016 -0400
| | |
| | |     Rearrange nested functions
| | |
| | * commit 557dc4e32897ad2a47585d6a64cee6f649813000
| | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | Date:   Tue Nov 1 13:50:52 2016 -0400
| | |
| | |     Add circular import test case
| | |
| | * commit c47679884ae0408cce7352f2b7c12032596d1813
| | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | Date:   Tue Nov 1 13:35:46 2016 -0400
| | |
| | |     Fix syntax errors
| | |
| | *   commit af5a31da8b2c2dbf99b1f7238aa75f5424a4917a
| | |\  Merge: 02ca24f d451cc4
| | | | Author: Jared Samet <jared.samet@aya.yale.edu>
| | | | Date:   Tue Nov 1 13:31:49 2016 -0400
| | | |
| | | |     Merge pull request #37 from Neitsch/import-rec
| | | |
| | | |     Recursively looking up dependencies
| | | |
| | | *   commit d451cc42d244b6fbab30c782cb2c2a4005be539b
| | | |\  Merge: 6a28c05 e673dca
| | | | | Author: Jared Samet <jared.samet@aya.yale.edu>
| | | | | Date:   Tue Nov 1 13:31:33 2016 -0400
| | | | |
| | | | |     Merge pull request #38 from Neitsch/import-load
| | | | |
| | | | |     Loading data from all imports
| | | | |
| | | | * commit e673dca67cec24791211af2b9f80791cb14de39d
| | | |/  Author: Neitsch <ns3158@columbia.edu>
| | | |   Date:   Mon Oct 31 15:56:43 2016 -0400
| | | |
| | | |       Loading data from all imports
| | | |
| | | * commit 6a28c05cd7632d70aedb70fdcfdc8d264e3e3479
| | | | Author: Neitsch <ns3158@columbia.edu>
| | | | Date:   Mon Oct 31 15:40:41 2016 -0400
| | | |
| | | |     Recursively looking up dependencies
| | | |
| | * |   commit 02ca24fac16c38f7b7f2dcbe2010ec04af49e007
| | |\ \  Merge: 3f28289 6fa0e39
| | | | | Author: Jared Samet <jared.samet@aya.yale.edu>
| | | | | Date:   Tue Nov 1 13:30:47 2016 -0400
| | | | |
| | | | |     Merge pull request #39 from Neitsch/wild-exc
| | | | |
| | | | |     Raising exceptions on certain values
| | | | |
| | | * | commit 6fa0e392019110045c14f1cdddc007e2bf5533ec
| | | |/  Author: Neitsch <ns3158@columbia.edu>
| | | |   Date:   Mon Oct 31 16:43:17 2016 -0400
| | | |
| | | |       Raising exceptions on certain values
| | | |
| | * |   commit 3f28289a73cf2fe837b2b44eaab7b105bb2b81fb
| | |\ \  Merge: 7d70af2 4eaef3b
| | | |/  Author: Jared Samet <jared.samet@aya.yale.edu>
| | |/|   Date:   Mon Oct 31 11:53:10 2016 -0400
| | | |
| | | |       Merge pull request #36 from Neitsch/import-arrange
| | | |
| | | |       Added unsorted function, globals and imports
| | | |
| | | * commit 4eaef3b64c7b0404fea7876990e621d4f6ded907
| | | | Author: Neitsch <ns3158@columbia.edu>
| | | | Date:   Mon Oct 31 11:01:00 2016 -0400
| | | |
| | | |     Removed obsolete parts
| | | |
| | | * commit 7d7b1e51de024be3acc6f276df2027ea80c8b35f
| | |/  Author: Neitsch <ns3158@columbia.edu>
| | |   Date:   Mon Oct 31 10:59:12 2016 -0400
| | |
| | |       Added unsorted function, globals and imports
| | |
| | * commit 7d70af2345c2cc0d09944e19f7517b0378635ce7
| | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | Date:   Sun Oct 30 15:23:04 2016 -0400
| | |
| | |     Add some explanatory comments
| | |
| | * commit 40d6b169baf47be6839cc792f7fa17e79b83dab6
| | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | Date:   Sun Oct 30 15:03:32 2016 -0400
| | |
| | |     More expansion samples
| | |
| | * commit af9b01c245de91965ee64957c7ce5112e8e059db
| | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | Date:   Sun Oct 30 14:48:44 2016 -0400
| | |
| | |     Refactor expansion code
| | |
| | * commit 903bc3fe44b93382ae8211e1828a66526954058a
| | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | Date:   Sun Oct 30 00:19:10 2016 -0400
| | |
| | |     Add test output
| | |
| | * commit 68b7b0340a9b6e21ca71140207c46a00e36d3396
| | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | Date:   Sun Oct 30 00:17:02 2016 -0400
| | |
| | |     Add test case
| | |
| | * commit a8bdf33619977487b629fc12bb99a66a5a5c41ed
| | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | Date:   Sun Oct 30 00:04:05 2016 -0400
| | |
| | |     Add LHS slice expansion
| | |
| | * commit 4ee6fdf258900ecbae3a8d1efb66e1019c188ce8
| | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | Date:   Sat Oct 29 17:36:17 2016 -0400
| | |
| | |     Add output
| | |
| | * commit 2b8bced61855a2361384912a47b3459c98828e2c
| |/  Author: oracleofnj <jared.samet@aya.yale.edu>
|/|   Date:   Sat Oct 29 17:27:22 2016 -0400
| |
| |       Expand dimension expressions
| |
* |   commit 443a818dc4fd9ec42da4afd6e733dca35cc198e8
|\ \  Merge: 022e8cd 9ba3c65
| |/  Author: Ishaan Kolluri <ishaankolluri@users.noreply.github.com>
| |   Date:   Wed Oct 26 16:31:51 2016 -0400
| |
| |       Merge pull request #35 from ishaankolluri/master
| |
| |       Add UNIs
| |
| * commit 9ba3c657b5a67bc5d2fc69be83b2e73809ab96ce
| | Author: Ishaan Kolluri <ishaankolluri@gmail.com>
| | Date:   Wed Oct 26 16:31:00 2016 -0400
| |
| |     Add UNIs
| |
* |   commit 022e8cdff18e452b0d3c518836141f30f93a9356
|\ \  Merge: 0bd9c4a 808aae5
| |/  Author: Ishaan Kolluri <ishaankolluri@users.noreply.github.com>
| |   Date:   Wed Oct 26 16:25:57 2016 -0400
| |
| |       Merge pull request #34 from ishaankolluri/master
| |
| |       One more edit
| |
| * commit 808aae5ee501ea7b86371ffe3b03ca489af02c07
|/  Author: Ishaan Kolluri <ishaankolluri@gmail.com>
|   Date:   Wed Oct 26 16:22:10 2016 -0400
|
|       Added change to precedence operators
|
*   commit 0bd9c4af5b3575b5c8f60973b8a2e9593ec207ee
|\  Merge: e7020ec fb2b382
| | Author: Jared Samet <jared.samet@aya.yale.edu>
| | Date:   Wed Oct 26 15:59:53 2016 -0400
| |
| |     Merge pull request #33 from Neitsch/final-slicing-comments
| |
| |     Thats all for now folks
| |
| * commit fb2b382bd13dcdd469a3c2d7a6cce429f41ea632
|/  Author: oracleofnj <jared.samet@aya.yale.edu>
|   Date:   Wed Oct 26 15:54:11 2016 -0400
|
|       Thats all for now folks
|
*   commit e7020ec688f7a86453b383714d8b5db98c699d2a
|\  Merge: 4b7984a 4683f14
| | Author: Jared Samet <jared.samet@aya.yale.edu>
| | Date:   Wed Oct 26 15:00:11 2016 -0400
| |
| |     Merge pull request #32 from Neitsch/final-lrm-edits
| |
| |     Flesh out switch expressions, add precedence
| |
| * commit 4683f14715a99d5cdb53089a3bd247035847f3c5
|/  Author: oracleofnj <jared.samet@aya.yale.edu>
|   Date:   Wed Oct 26 14:48:41 2016 -0400
|
|       Flesh out switch expressions, add precedence
|
*   commit 4b7984ac504b295e16f8bc77db8024742c6543ab
|\  Merge: 0c42b9c 3d587c5
| | Author: Jared Samet <jared.samet@aya.yale.edu>
| | Date:   Wed Oct 26 11:15:03 2016 -0400
| |
| |     Merge pull request #31 from Neitsch/more-lrm-edits
| |
| |     Incorporate requested edits and a few more clarifications
| |
| * commit 3d587c5997c06337022a69c4a064ee1f33ca6bd2
|/  Author: oracleofnj <jared.samet@aya.yale.edu>
|   Date:   Wed Oct 26 11:10:15 2016 -0400
|
|       Incorporate requested edits and a few more clarifications
|
*   commit 0c42b9cf523a95903a4464420864a25a638c8bfb
|\  Merge: 63fb02b cd81040
| | Author: Jared Samet <jared.samet@aya.yale.edu>
| | Date:   Wed Oct 26 09:22:08 2016 -0400
| |
| |     Merge pull request #30 from ishaankolluri/LRM_update
| |
| |     Added changes to first half of LRM
| |
| * commit cd81040e2c5ba5da130c04bc67eaef13fa6522db
|/  Author: ishaankolluri <ishaankolluri@gmail.com>
|   Date:   Wed Oct 26 03:30:20 2016 -0400
|
|       Added changes to first half of LRM
|
*   commit 63fb02be16a805fd845fb89632ca258ddbeedeb6
|\  Merge: 5932551 0941e96
```

```
| | Author: Jared Samet <jared.samet@aya.yale.edu>
| | Date:   Wed Oct 26 02:13:17 2016 -0400
| |
| |     Merge pull request #29 from Neitsch/lrm-edits
| |
| |     Lrm edits
| |
| * commit 0941e967272126c6417f458450c611109792e6b8
| | Author: oracleofnj <jared.samet@aya.yale.edu>
| | Date:   Wed Oct 26 02:04:47 2016 -0400
| |
| |     Rebuild PDF
| |
| * commit cb0406921e69f750eca4de178dc2eb8d0733f90f
| | Author: oracleofnj <jared.samet@aya.yale.edu>
| | Date:   Wed Oct 26 02:04:01 2016 -0400
| |
| |     Add built in functions
| |
| * commit 4abf638392a694426f3ca02fb7fcd8ce5b5d3098
| | Author: oracleofnj <jared.samet@aya.yale.edu>
| | Date:   Wed Oct 26 01:56:38 2016 -0400
| |
| |     Add built in functions
| |
| * commit 7661925a96608a3ca00325d343f3feb4c6a0803a
|/  Author: oracleofnj <jared.samet@aya.yale.edu>
|   Date:   Wed Oct 26 00:04:22 2016 -0400
|
|       Initial comments
|
*   commit 59325515eae265feca08ec854902c0e14fcd0565
|\  Merge: b978f00 cc66297
| | Author: Nigel Schuster <Neitsch@users.noreply.github.com>
| | Date:   Tue Oct 25 21:30:40 2016 -0400
| |
| |     Merge pull request #28 from Neitsch/func-doc-fix
| |
| |     Fixed mistakes in functions part of the doc
| |
| * commit cc66297fd6785a81de7d638a838cc2e4e6645c54
|/  Author: Nigel Schuster <nigel.schusters@googlemail.com>
|   Date:   Tue Oct 25 20:14:27 2016 -0400
|
|       Fixed mistakes in functions part of the doc
|
*   commit b978f0021545f18d4481658176c1118d7186e624
|\  Merge: 2e1ea60 125a5bb
| | Author: Ishaan Kolluri <ishaankolluri@users.noreply.github.com>
| | Date:   Tue Oct 25 13:04:05 2016 -0400
| |
| |     Merge pull request #27 from ishaankolluri/master
| |
| |     Removed AUX file
| |
| * commit 125a5bb4578448db78b45022651c818e6e07c842
```

```
|/  Author: Ishaan Kolluri <ishaankolluri@gmail.com>
|   Date:   Tue Oct 25 12:49:38 2016 —0400
|
|       Removed AUX file
|
*   commit 2e1ea607d20d2b7c0ce2b283c77720b067e197a9
|\  Merge: eb24036 84b03ee
| | Author: Jared Samet <jared.samet@aya.yale.edu>
| | Date:   Tue Oct 25 11:30:35 2016 —0400
| |
| |     Merge pull request #26 from Neitsch/better—regexp
| |
| |     Better regexp
| |
| * commit 84b03ee4733da1e7a89376cfce43bb76f0884999
| | Author: oracleofnj <jared.samet@aya.yale.edu>
| | Date:   Tue Oct 25 01:22:31 2016 —0400
| |
| |     Fix let order
| |
| * commit 91b40c52f9891693fe9cbb7d00418c9877fbae37
|/  Author: oracleofnj <jared.samet@aya.yale.edu>
|   Date:   Tue Oct 25 01:14:43 2016 —0400
|
|       Improve regexp
|
*   commit eb240362462693dca5918e90054fc421c65113fe
|\  Merge: ec7cc9c 991c918
| | Author: Jared Samet <jared.samet@aya.yale.edu>
| | Date:   Mon Oct 24 23:55:38 2016 —0400
| |
| |     Merge pull request #23 from Neitsch/file—io
| |
| |     File io
| |
| * commit 991c918de9ba07ab68722fe8042a1edc96b555a5
| | Author: oracleofnj <jared.samet@aya.yale.edu>
| | Date:   Mon Oct 24 23:20:12 2016 —0400
| |
| |     Replace fopen, fclose etc. with open, close etc.
| |
| * commit 338faa006d8d4c7858477028eb8074d69c24aee1
| | Author: oracleofnj <jared.samet@aya.yale.edu>
| | Date:   Mon Oct 24 23:14:30 2016 —0400
| |
| |     Fix file inclusion and rebuild PDF
| |
| *   commit b24edd3a58eed2763abd1fe1d4023138af05bbf7
| |\  Merge: 44a1cc5 2f09a64
| | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | Date:   Mon Oct 24 23:11:50 2016 —0400
| | |
| | |     Merge in expressions section
| | |
| | * commit 2f09a643221dec47af99592d0a53e15bcc7d67d7
| | | Author: Kevin <kevinye1113@gmail.com>
| | | Date:   Mon Oct 24 15:52:10 2016 -0400
| | |
| | |     Added the Expression Section 4 to LRM
| | |
| | *   commit 8cd39aca8a29d19fe986ad4e7a3540fa4347221f
| | |\  Merge: e5d2478 3609366
| | | | Author: Kevin <kevinye1113@gmail.com>
| | | | Date:   Mon Oct 24 11:05:33 2016 -0400
| | | |
| | | |     Added string literals to scanner
| | | |
| | * | commit e5d247809c610f428568dce665213019bc368047
| | | | Author: Kevin <kevinye1113@gmail.com>
| | | | Date:   Mon Oct 24 11:00:39 2016 -0400
| | | |
| | | |     Added string literals to scanner
| | | |
| * | |   commit 44a1cc53aee333cb2d9832f66c0750b8a7add7d8
| |\ \ \  Merge: 1ea3c28 3609366
| | | |/  Author: oracleofnj <jared.samet@aya.yale.edu>
| | |/|   Date:   Mon Oct 24 23:06:07 2016 -0400
| | | |
| | | |       Merge scanner changes and add regexp to properly escape strings
| | | |
| | * | commit 36093668180db647b45e208d1dfe0158e7281281
| | | | Author: kevinye1 <kevinye1@users.noreply.github.com>
| | | | Date:   Thu Oct 20 21:14:00 2016 -0400
| | | |
| | | |     Update scanner.mll
| | | |
| | * | commit 0d57652f53d8a6f4dbbb84d58a6807e5ef59a6fe
| | | | Author: Kevin <kevinye1113@gmail.com>
| | | | Date:   Thu Oct 20 21:10:27 2016 -0400
| | | |
| | | |     Fixed bug in scanner
| | | |
| | * | commit 1848813885a8c826151e62a27b4ef97b99004a96
| | | | Author: Kevin <kevinye1113@gmail.com>
| | | | Date:   Thu Oct 20 20:21:49 2016 -0400
| | | |
| | | |     Made scanner
| | | |
| * | |   commit 1ea3c283f511068ed0886e67e5e65e974b646015
| |\ \ \  Merge: a692466 ec7cc9c
| |/ / /  Author: oracleofnj <jared.samet@aya.yale.edu>
|/| | |   Date:   Mon Oct 24 15:26:16 2016 -0400
| | | |
| | | |       Merge branch ’master’ into file-io
| | | |
* | | | commit ec7cc9c42b88c8cab9bbfe748fb03109eb18bd9b
| |_|/  Author: oracleofnj <jared.samet@aya.yale.edu>
|/| |   Date:   Mon Oct 24 15:06:06 2016 -0400
| | |
| | |       Replace repetitive code with more idiomatic OCaml
| | |
| * | commit a69246646b48528da4e0f0ddf422cbc35522aeaf
| | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | Date:   Mon Oct 24 01:09:21 2016 -0400
| | |
| | |     Fix tests until strings ready
| | |
| * | commit 8553a50322c8741d29ed4f4b963a69f232f54750
| | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | Date:   Mon Oct 24 01:08:29 2016 -0400
| | |
| | |     Fix tests until string ready
| | |
| * |   commit 0ed4ad70e5f0db7388b1228df374a1f214f450e6
| |\ \  Merge: 71e0b1c 92ac506
| |/ /  Author: oracleofnj <jared.samet@aya.yale.edu>
|/| |   Date:   Mon Oct 24 00:55:08 2016 -0400
| | |
| | |       Add File IO, Entry point and Example to LRM
| | |
* | | commit 92ac50626674ad7d1ccd534f24d39892ca8cb3fa
| | | Author: ishaankolluri <ishaankolluri@gmail.com>
| | | Date:   Sun Oct 23 22:24:06 2016 -0400
| | |
| | |     Make small change to data type section
| | |
| * | commit 71e0b1cb20967ecf7698c343d3774588f361e71c
| | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | Date:   Sun Oct 23 22:58:21 2016 -0400
| | |
| | |     Fix section reference
| | |
| * | commit 6abb290b9aeaf16eaac6bc431862cb780337b4c6
|/ /  Author: oracleofnj <jared.samet@aya.yale.edu>
| |   Date:   Sun Oct 23 22:34:42 2016 -0400
| |
| |       Initial commit for File I/O section
| |
* | commit 67b4b65ab85c86f5369138e782c352143aa1cfec
| | Author: oracleofnj <jared.samet@aya.yale.edu>
| | Date:   Sun Oct 23 19:12:38 2016 -0400
| |
| |     Reduce eye pain
| |
* |   commit 2824ee9412ceb67f802e77c382b50dfbf98b57e3
|\ \  Merge: 13d0896 f8ae543
| | | Author: Nigel Schuster <Neitsch@users.noreply.github.com>
| | | Date:   Sun Oct 23 19:03:24 2016 -0400
| | |
| | |     Merge pull request #20 from Neitsch/samples
| | |
| | |     Samples from LRM
| | |
| * |   commit f8ae5439b7d1ffc27640e2e66cb657184f2cd1b8
| |\ \  Merge: e0c702d 13d0896
| |/ /  Author: Nigel Schuster <Neitsch@users.noreply.github.com>
|/| |   Date:   Sun Oct 23 18:23:11 2016 -0400
| | |
| | |       Merge branch ’master’ into samples
| | |
* | |   commit 13d089696d899b6c691b4c3af603e334fd462f04
|\ \ \  Merge: 9805753 3a2cd60
| | | | Author: Jared Samet <jared.samet@aya.yale.edu>
| | | | Date:   Sun Oct 23 18:20:03 2016 -0400
| | | |
| | | |     Merge pull request #19 from Neitsch/sequence-operator
| | | |
| | | |     Sequence operator
| | | |
| * | |   commit 3a2cd606ec6fd9bba26107ec9208b34f257d32d4
| |\ \ \  Merge: 57319c4 9805753
| |/ / /  Author: Jared Samet <jared.samet@aya.yale.edu>
|/| | |   Date:   Sun Oct 23 18:16:35 2016 -0400
| | | |
| | | |       Merge branch ’master’ into sequence-operator
| | | |
| * | | commit 57319c48db49a519a0f2a75f3abed55ece413df9
| | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | Date:   Sun Oct 23 17:11:13 2016 -0400
| | | |
| | | |     Remove intermediate files
| | | |
| * | | commit 53824eaef49ae831e43ab5685b9e1fcea5fcd678
| | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | Date:   Sun Oct 23 17:10:39 2016 -0400
| | | |
| | | |     Flip precedence of -> and ?: (?: is now lowest)
| | | |
| * | | commit 7dedf93f4ed468537317e5105eeb5a2b8f791aad
| | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | Date:   Sun Oct 23 17:05:23 2016 -0400
| | | |
| | | |     Add sequence operator to scanner/parser/AST
| | | |
| | * | commit e0c702d18ceb98ce5ec03ffe49ed53534540fb0d
| | | | Author: Neitsch <ns3158@columbia.edu>
| | | | Date:   Sun Oct 23 18:17:58 2016 -0400
| | | |
| | | |     Fixed .gitignore
| | | |
| | * | commit e42fe941b3c93b0cda424f01925754e2f12cbe43
| | | | Author: Neitsch <ns3158@columbia.edu>
| | | | Date:   Sun Oct 23 18:05:48 2016 -0400
| | | |
| | | |     Added code in LRM to test code samples
| | | |
| | * |   commit 9d2cd174bb0cddd3894d3e77ade8626a2700429c
| | |\ \  Merge: 167ddd2 9805753
| |_|/ /  Author: Neitsch <ns3158@columbia.edu>
|/| | |   Date:   Sun Oct 23 17:24:15 2016 -0400
| | | |
| | | |       Merge branch ’master’ into samples
| | | |
* | | |   commit 980575329d6ce70040a7911de284463f8c9ebf2f
```

```
|\ \ \ \  Merge: 3f015ee e0c7aed
| |/ / /  Author: Nigel Schuster <Neitsch@users.noreply.github.com>
|/| | |   Date:   Sun Oct 23 17:01:31 2016 —0400
| | | |
| | | |       Merge pull request #17 from Neitsch/make—correction
| | | |
| | | |       Added simple TCs, Moved Makefile to oasis config
| | | |
| * | | commit e0c7aed793ba6070ec94baa3bc52f009c5615c20
| | | | Author: Neitsch <ns3158@columbia.edu>
| | | | Date:   Sun Oct 23 16:59:33 2016 —0400
| | | |
| | | |     Fixed test
| | | |
| * | |   commit ec3d682eeeecb8b1c20f35a746b65e503b7cdeb4
| |\ \ \  Merge: ea05658 3f015ee
| |/ / /  Author: Nigel Schuster <Neitsch@users.noreply.github.com>
|/| | |   Date:   Sun Oct 23 16:41:00 2016 —0400
| | | |
| | | |       Merge branch ’master’ into make—correction
| | | |
* | | |   commit 3f015eedeeda8af3aac5cb728d9e6258d6b7acda
|\ \ \ \  Merge: d4961eb edf3dea
| | | | | Author: Jared Samet <jared.samet@aya.yale.edu>
| | | | | Date:   Sun Oct 23 15:52:01 2016 —0400
| | | | |
| | | | |     Merge pull request #18 from Neitsch/grammar—bug—fixes
| | | | |
| | | | |     Modify grammar to allow [m,n] foo, bar, baz;
| | | | |
| * | | |   commit edf3dead5be0840a9a80f407dcef8b63c898d14a
| |\ \ \ \  Merge: b45718d d4961eb
| |/ / / /  Author: Jared Samet <jared.samet@aya.yale.edu>
|/| | | |   Date:   Sun Oct 23 15:44:20 2016 —0400
| | | | |
| | | | |       Merge branch ’master’ into grammar—bug—fixes
| | | | |
| * | | | commit b45718decc1b9e3816105722eebb39afa45dbc5a
| | | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | | Date:   Sun Oct 23 02:27:36 2016 —0400
| | | | |
| | | | |     Modify grammar to allow [m,n] foo, bar, baz;
| | | | |
| | * | | commit ea056587ed93451542d5a5e67b9b993cba4a170f
| | | | | Author: Neitsch <ns3158@columbia.edu>
| | | | | Date:   Sun Oct 23 16:40:24 2016 —0400
| | | | |
| | | | |     Moved sequence file
| | | | |
| | * | |   commit 0ca56a0230a52c254d21724848c52ddc65350997
| | |\ \ \  Merge: 9d1094e 7e558c1
| | | | | | Author: Neitsch <ns3158@columbia.edu>
| | | | | | Date:   Sun Oct 23 16:10:14 2016 —0400
| | | | | |
| | | | | |     Merge
| | | | | |
| | | * | |   commit 7e558c19a9d85c1dea38f9745b2915dcc14418b7
| | | |\ \ \  Merge: 4652c67 d4961eb
| |_|_|/ / /  Author: Nigel Schuster <Neitsch@users.noreply.github.com>
|/| | | | |   Date:   Sun Oct 23 15:44:20 2016 −0400
| | | | | |
| | | | | |       Merge branch ’master’ into make−correction
| | | | | |
* | | | | |   commit d4961eb9497ffe2abacb91b8ab5d09ebf2b714a9
|\ \ \ \ \ \  Merge: 143fcba 0e0bda5
| |/ / / / /  Author: Nigel Schuster <Neitsch@users.noreply.github.com>
|/| | | | |   Date:   Sun Oct 23 15:43:16 2016 −0400
| | | | | |
| | | | | |       Merge pull request #15 from Neitsch/functions−doc
| | | | | |
| | | | | |       Functions doc
| | | | | |
| * | | | |   commit 0e0bda54deeafa312a3da9323d2ba2ff58dfdb79
| |\ \ \ \ \  Merge: cfe827d 143fcba
| |/ / / / /  Author: Nigel Schuster <Neitsch@users.noreply.github.com>
|/| | | | |   Date:   Sun Oct 23 15:05:42 2016 −0400
| | | | | |
| | | | | |       Merge branch ’master’ into functions−doc
| | | | | |
| * | | | | commit cfe827d4173deb1a3fff5719704eae3ff131a615
| | | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | | | | Date:   Fri Oct 21 20:50:51 2016 −0400
| | | | | |
| | | | | |     Completed initial functions section doc
| | | | | |
| * | | | |   commit 1b610ac614b03b24e829084a6d245b0d99fad01f
| |\ \ \ \ \  Merge: acb9b93 7542b5d
| | | | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | | | | | Date:   Thu Oct 20 13:50:22 2016 −0400
| | | | | | |
| | | | | | |     Merge
| | | | | | |
| | * | | | | commit 7542b5d1866554d70bd312ae649cd1d86f459f12
| | | | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | | | | | Date:   Thu Oct 20 11:16:35 2016 −0400
| | | | | | |
| | | | | | |     Added dimension section
| | | | | | |
| | * | | | | commit 995cf83b2b9a53b6312a0f3f468321c3d9d763c0
| | | |_|_|/  Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | |/| | |   Date:   Wed Oct 19 12:28:09 2016 −0400
| | | | | |
| | | | | |       Started working on Functions
| | | | | |
| * | | | | commit acb9b935dc9006e0ae878622402e28fad06907fa
| | | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | | | | Date:   Thu Oct 20 13:40:00 2016 −0400
| | | | | |
| | | | | |     Changed subsection header
| | | | | |
| * | | | | commit b95d039ea41668404b9005f1a7eb19e686abb071
| | | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | | | | Date:   Thu Oct 20 11:16:35 2016 -0400
| | | | | |
| | | | | |     Added dimension section
| | | | | |
| * | | | | commit 71b93bbd5cc99d7ebc3a8c5446b06d5b13ab96ac
| | | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | | | | Date:   Wed Oct 19 12:28:09 2016 -0400
| | | | | |
| | | | | |     Started working on Functions
| | | | | |
| | | * | | commit 9d1094e62f36b4170d996ffb07dcb5ee769050f2
| | | | | | Author: Neitsch <ns3158@columbia.edu>
| | | | | | Date:   Sun Oct 23 15:00:35 2016 -0400
| | | | | |
| | | | | |     Added simple TCs, Moved Makefile to oasis config
| | | | | |
| | | * | | commit 0a28413f410b7cf1f464b83ed99396ab988481d6
| | | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | | | | Date:   Fri Oct 21 20:50:51 2016 -0400
| | | | | |
| | | | | |     Completed initial functions section doc
| | | | | |
| | | * | | commit 0797f328f6f6f6e696a72b1222d4cab38f8e8507
| | | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | | | | Date:   Thu Oct 20 13:40:00 2016 -0400
| | | | | |
| | | | | |     Changed subsection header
| | | | | |
| | | * | | commit 9df31f7322f58cbc1b14ea9740b102750e0321ae
| | | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | | | | Date:   Thu Oct 20 11:16:35 2016 -0400
| | | | | |
| | | | | |     Added dimension section
| | | | | |
| | | * | | commit 8939903db79ed9de1a8e4b97fc02e79b4be54b61
| | | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | | | | Date:   Wed Oct 19 12:28:09 2016 -0400
| | | | | |
| | | | | |     Started working on Functions
| | | | | |
| | | * | | commit cae3b37031dc87451ff03790eb6321dce42c516b
| | | | | | Author: Nigel Schuster <nigel.schusters@googlemail.com>
| | | | | | Date:   Thu Oct 20 11:16:35 2016 -0400
| | | | | |
| | | | | |     Added dimension section
| | | | | |
| | | * | | commit 049c95d512ce6e6e8e48d4b2c838fd06df40b619
| |_|/ / /  Author: Nigel Schuster <nigel.schusters@googlemail.com>
|/| | | |   Date:   Wed Oct 19 12:28:09 2016 -0400
| | | | |
| | | | |       Started working on Functions
| | | | |
| | | | * commit 167ddd2addae79549eb6d887b98bee35cf2cc4cf
| | | | | Author: Neitsch <ns3158@columbia.edu>
| | | | | Date:   Sun Oct 23 17:18:35 2016 -0400
| | | | |
| | | | |     Removed test output
| | | | |
| | | | * commit 84d20b5448f24b98f308c1fb319f213de230d4f1
| | | |/  Author: Neitsch <ns3158@columbia.edu>
| | | |   Date:   Sun Oct 23 16:01:00 2016 -0400
| | | |
| | | |       Comparing sample code with correctly parsed code in samples_comp
| | | |
| | | * commit 4652c6727f848dd573a44509fb498729b28b3080
| |_|/  Author: Neitsch <ns3158@columbia.edu>
|/| |   Date:   Sun Oct 23 15:00:35 2016 -0400
| | |
| | |       Added simple TCs, Moved Makefile to oasis config
| | |
* | |   commit 143fcba457b49d2c4ab1495d17b830534554508e
|\ \ \  Merge: 660de8c a726236
| | | | Author: Jared Samet <jared.samet@aya.yale.edu>
| | | | Date:   Sat Oct 22 23:23:10 2016 -0400
| | | |
| | | |     Merge pull request #16 from Neitsch/more-AST
| | | |
| | | |     Hook up scanner and parser
| | | |
| * | | commit a7262368b391c2031c66cfcf1d9fd6224749be69
| | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | Date:   Sat Oct 22 20:51:27 2016 -0400
| | | |
| | | |     Add comments and sample program
| | | |
| | | |     * Added multiline comments using microC code
| | | |     * Added one-line comments but didn't test it yet
| | | |     * Added example file sequence.xtnd which is the program from proposal
| | | |
| | | |     To try:
| | | |
| | | |     $ make
| | | |     $ ./jsonify < sequence.xtnd
| | | |
| | | |     paste output into https://jqplay.org/
| | | |
| * | | commit 8db40989c4efe4ecefdb8409d45fe8347adc0c79
| | | | Author: oracleofnj <jared.samet@aya.yale.edu>
| | | | Date:   Sat Oct 22 19:44:48 2016 -0400
| | | |
| | | |     Fix minor grammar bug
| | | |
| | | |     * foo[:4,:5] was not being accepted
| | | |     * Added rules to lslice and rslice to allow this
| | | |
| * | | commit 80754c3f3072609502c902d6d26c2e6c33885a84
|/ / /  Author: oracleofnj <jared.samet@aya.yale.edu>
| | |   Date:   Sat Oct 22 18:19:27 2016 -0400
| | |
| | |       Hook up scanner and parser
| | |
| | |       * Added hand coded jsonification
| | |       * Didn’t look like Yojson would work with our types
| | |       * Currently not working on any input besides EOF
| | |       * Added temporary makefile
| | |
* | | commit 660de8c848406e8f5fffe099610b6147bc553cad
|/ /  Author: Jared Samet <jared.samet@aya.yale.edu>
| |   Date:   Sat Oct 22 13:54:32 2016 -0400
| |
| |       Add stuff to the grammar, minor corrections (#14)
| |
| |       * Started AST
| |
| |       * Made scanner
| |
| |       * Fixed bug in scanner
| |
| |       * Update scanner.mll
| |
| |       * Add stuff to the grammar, minor corrections
| |
| |       Operators, in order of precedence (lowest first)
| |       * LOGOR ||
| |       * LOGAND &&
| |       * EQ NOTEQ LT GT LTEQ GTEQ == != < > <= >=
| |       * PLUS MINUS BITOR BITXOR + - | ^
| |       * TIMES DIVIDE MOD LSHIFT RSHIFT BITAND * / % << >> &
| |       * POWER **
| |       * BITNOT LOGNOT NEG ~ ! -
| |       * I took the precedence from Golang
| |
| |       String and char literals
| |       New token ASN which is = (as opposed to EQ ==)
| |
| |       Range literals
| |       * {1, 2, 3; 4, 5, 6} is a 2x3 range
| |       * {foo(1); bar(2)} should work too
| |       * The semantic analysis phase will be responsible
| |       *   for ensuring that the dimensions make sense
| |       * Had to add mandatory parentheses to switch(cond) to
| |       *   avoid shift/reduce conflicts but C has that anyway
| |
| |       * Allow general expr slicing, imports and globals
| |       * New HASH token #
| |       * New IMPORT and GLOBAL tokens ("import", "global")
| |
| |       Expr slicing:
| |       * rls_sel no longer -> epsilon
| |       * replaced by expr -> expr rhs_sel, expr -> ID, expr -> HASH expr
| |
| |       Import:
| |       * import "othermodule";
| |
| |       Global variables:
| |       * global PI := 3.14159;
| |       * global [3,5] foo; is legal but has no effect
| |
| |       Program structure:
| |       * imports followed by globals followed by functions
| |       * This should theoretically be mix—and—matchable without much more work
| |
| |       Size inference:
| |       * foo := "Hello, world!" right now results in [1,1] foo == "H"
| |       * Semantic not grammar issue
| |
| |       * Fix error in col_list production
| |
| |       * Remove intermediate files
| |
| |       * Remove intermediate files
| |
| |       * Rework AST
| |
* |   commit a15772cba8b1fd16e7d21da07ec9b89d34b3062e
|\ \  Merge: d8794e9 dee63c7
| |/  Author: Ishaan Kolluri <ishaankolluri@users.noreply.github.com>
|/|   Date:   Thu Oct 20 13:38:08 2016 —0400
| |
| |       Merge pull request #10 from ishaankolluri/LRM
| |
| |       Initial LRM Commit
| |
| *   commit dee63c70c8c9e7396e255662c302f3d77a7bf7db
| |\  Merge: 4d763cb dc93dbf
| | | Author: Ishaan Kolluri <ishaankolluri@users.noreply.github.com>
| | | Date:   Thu Oct 20 13:26:28 2016 —0400
| | |
| | |     Merge pull request #1 from Neitsch/grammar—doc
| | |
| | |     Grammar import
| | |
| | * commit dc93dbfa68ff7ee65eb38ce8a26baf081da0fe30
| |/  Author: Nigel Schuster <nigel.schusters@googlemail.com>
| |   Date:   Thu Oct 20 13:18:29 2016 —0400
| |
| |       Grammar import
| |
| * commit 4d763cbd2505f38028dc9ba820c02aea2b66a380
| | Author: Ishaan Kolluri <ishaankolluri@gmail.com>
| | Date:   Thu Oct 20 12:21:21 2016 —0400
| |
| |     Made refactor and edits to intro section of LRM
| |
| *   commit e7443cc478f120ea4aff5db91f7dbfa55e94c4b4
| |\  Merge: 02a5c17 40c2a5a
| | | Author: Ishaan Kolluri <ishaankolluri@gmail.com>
| | | Date:   Thu Oct 20 11:46:54 2016 —0400
| | |
| | |     Merging
| | |
| | * commit 40c2a5aa75310086dbcfab58eaa1e3ce86ccbe72
| |/  Author: ishaankolluri <ishaankolluri@gmail.com>
|/|   Date:   Wed Oct 19 03:43:06 2016 —0400
```

```
| |
| |      Initial LRM Commit part 1
| |
| * commit 02a5c175a1032b34fbf0f1726596270d934aec11
|/  Author: Ishaan Kolluri <ishaankolluri@gmail.com>
|   Date:   Tue Oct 18 18:38:21 2016 -0400
|
|       Added LRM initial info
|
*   commit d8794e96f4544ed0af9678ecb55e83e1e4b46c25
|\  Merge: 5111202 70aa1b9
| | Author: Ishaan Kolluri <ishaankolluri@users.noreply.github.com>
| | Date:   Mon Oct 17 19:47:42 2016 -0400
| |
| |     Merge pull request #9 from Neitsch/documentation
| |
| |     Added PDF Latex template
| |
| * commit 70aa1b9846e135d1e8e3f2159148d1bc4e9ea0d8
|/  Author: Nigel Schuster <nigel.schusters@googlemail.com>
|   Date:   Sun Oct 16 13:36:23 2016 -0400
|
|       Added PDF Latex template
|
* commit 5111202a9f0992d7b5a3f934f4109552064df4e1
| Author: Jared Samet <jared.samet@aya.yale.edu>
| Date:   Fri Oct 14 19:59:45 2016 -0400
|
|     Added a bunch of stuff to the grammar: (#8)
|
|     * Cleaned up the existing token list
|     ** Removed a couple duplicates/unused that menhir flagged
|     ** Added PLUS MINUS TIMES DIVIDE MOD
|     ** Added SWITCH CASE DEFAULT
|     ** Added UNDERSCORE (for [_,2] in function declaration)
|     ** Separated LITERAL into LIT_INT and LIT_FLOAT (dimensions can
|     **      only be integers)
|     ** Added a couple associativity/precedence rules that
|     **      are undoubtedly incomplete
|
|     * Made hopefully all the optional / repetition cases consistent
|     ** Function declarations [] foo() {}    [] bar() {}   ....
|     ** Function parameters ()   ([] foo)  ([]foo, []bar), ...
|     ** Statements stmt1; stmt2; ...
|     ** Function arguments (), (foo), (foo, bar), ...
|     ** Switch cases (>=1 required) case foo1: bar1; case foo2: bar2; ...
|     ** Case conditions (>=1 required) case foo1, foo2, ...
|
|     * Added a few expression types
|     ** switch [expr] {case foo_1a, foo_1b: bar1; case foo_2: bar2}
|     *** intended to work as in Go: if condition is present, evaluates
|     *** to first case where foo == expr. if condition is absent, evaluates
|     *** to first case where foo is truthy.
|     ** function calls foo(x,y,z)
|     ** expression in parentheses
|     ** literals
```

```
|
|     * Slicing
|     ** Refactored slice_val into lslice_val and rslice_val
|     ** lslice_val can be any expr
|     ** rslice_val can be any expr or [expr]
|     ** both types of selections and slices are now consistent until
|     **      you get down to val
|
|     * Function return dimensions
|     ** Added _ to the allowed return dimension (for [_,2])
|
* commit da967e421b3f6887acd7c0d659f13d60b1898b6a
| Author: Nigel Schuster <Neitsch@users.noreply.github.com>
| Date:   Wed Oct 12 13:24:50 2016 –0400
|
|     CFG Grammar (#6)
|
|     Added simple parser that has most of our rules.
|
* commit fea4e4b262513b011775899f35d17e29cb7a1642
| Author: Nigel Schuster <Neitsch@users.noreply.github.com>
| Date:   Sat Oct 8 11:42:39 2016 –0400
|
|     There is no need to constantly build all branches. (#2)
|
|     Only building master.
|
* commit 7a5ccfc7ada9ddefba61d5da995a7949d7fd1e9d
| Author: Jared Samet <jared.samet@aya.yale.edu>
| Date:   Sat Oct 8 11:31:31 2016 –0400
|
|     Added greeting and newlines (#4)
|
* commit 10b17f7772e4f7521d248b9656a917df781d96a4
| Author: Nigel Schuster <Neitsch@users.noreply.github.com>
| Date:   Sat Oct 8 11:31:08 2016 –0400
|
|     Imported microc (#5)
|
|     * Imported microc
|     * Fixed sources
|
* commit 726456fcc25a785cc26d90845b8c0c6292fec61f
| Author: Ishaan Kolluri <ishaankolluri@users.noreply.github.com>
| Date:   Tue Sep 20 09:45:07 2016 –0400
|
|     [test] Add sample greeting to repo (#3)
|
|     * Added sample greeting
|
|     * Add double semicolon notation
|
* commit 9a2183d3439d67d696ec6babcb6c2be233c2f3bd
| Author: Nigel Schuster <nigel.schusters@googlemail.com>
| Date:   Thu Sep 15 18:44:00 2016 –0400
|
|     Added merlin config
|
* commit 163e1761ac0f8113e017511f456029282e665d8b
| Author: Nigel Schuster <nigel.schusters@googlemail.com>
| Date:   Wed Sep 14 18:51:53 2016 -0400
|
|     Moved whole build to script
|
* commit d401eeaab71e5e927932c403be099e24700cdc8f
| Author: Nigel Schuster <nigel.schusters@googlemail.com>
| Date:   Wed Sep 14 18:43:58 2016 -0400
|
|     Added oasis opam package
|
* commit ba7fd9c3ccaa617d0709b395da84b43cca2c1860
| Author: Nigel Schuster <nigel.schusters@googlemail.com>
| Date:   Wed Sep 14 18:38:58 2016 -0400
|
|     Added ocaml configure (maybe this helps travis)
|
* commit a461eaea15fdf84b3a580af6cfa46451ab490b31
| Author: Nigel Schuster <nigel.schusters@googlemail.com>
| Date:   Wed Sep 14 18:26:10 2016 -0400
|
|     Configuring opam environment for travis
|
* commit ba2df2f48c360588df163b22ad674297e3ae8b16
| Author: Nigel Schuster <nigel.schusters@googlemail.com>
| Date:   Wed Sep 14 18:19:26 2016 -0400
|
|     Added ocaml native compiler to apt package list
|
* commit a8e5958abf68181b0a7b4e7995a2cd7044de3bd4
| Author: Nigel Schuster <nigel.schusters@googlemail.com>
| Date:   Wed Sep 14 17:24:36 2016 -0400
|
|     Added some more (possibly necessary opam packages
|
* commit c54f5e367d0d985d54bf89630e51a59203f91065
| Author: Nigel Schuster <nigel.schusters@googlemail.com>
| Date:   Wed Sep 14 17:18:32 2016 -0400
|
|     Missed opam option
|
* commit b10adf0f2964eff28de9f33618806e7d614573fc
| Author: Nigel Schuster <nigel.schusters@googlemail.com>
| Date:   Wed Sep 14 17:13:57 2016 -0400
|
|     Fixed opam install
|
* commit 124f7f37083351df7beaed11cc77d503b1ff0d9b
| Author: Nigel Schuster <nigel.schusters@googlemail.com>
| Date:   Wed Sep 14 17:08:09 2016 -0400
|
|     Fixed YML error
|
* commit 4909fa82e010b27f9dc0f2489ebf5cadc790e4e3
| Author: Nigel Schuster <nigel.schusters@googlemail.com>
| Date:   Wed Sep 14 17:03:54 2016 -0400
|
|     Using avsm source
|
* commit 4b24046db1b50e97bac103170056c77be7273155
| Author: Nigel Schuster <nigel.schusters@googlemail.com>
| Date:   Wed Sep 14 16:58:33 2016 -0400
|
|     Allow sudo
|
* commit e7b50dbb8b2c463fa67f9ce07b6a8413f905b259
| Author: Nigel Schuster <nigel.schusters@googlemail.com>
| Date:   Wed Sep 14 16:56:57 2016 -0400
|
|     Fixed setup order
|
* commit f6d7ac43f5f27fc4c641667bfe8e24ea3b0d7117
| Author: Nigel Schuster <nigel.schusters@googlemail.com>
| Date:   Wed Sep 14 16:50:02 2016 -0400
|
|     Manually installing apt packages
|
* commit f4084ab464f4f3097d11e0582328f351c40d9516
| Author: Nigel Schuster <nigel.schusters@googlemail.com>
| Date:   Wed Sep 14 16:40:55 2016 -0400
|
|     Test commit
|
* commit d7c5e9a4dadd09c6d7ad97fc727f61aff1e88779
  Author: Nigel Schuster <Neitsch@users.noreply.github.com>
  Date:   Wed Sep 14 13:15:43 2016 -0400

      Initial commit
```