Justin Domingue – 260588454

Daniel Pham – 260526252

Seguei Nevarko – 260583807

Milestone 2

COMP 520

Work presented to Prof. Laurie Hendren

McGil University

Tuesday, March 22nd 2016

# Symbol Table

Instead of using a stack of hash tables, we used a stack of red-black trees.

# Symbol Pass

Symbols are typed as: variables, type aliases, functions or inferred where inferred has its type determined at typechecking. Variable and type alias pointers point directly to the type with with they are associated.

# Scoping Rules

These follow the GoLite specifications.

# Type Check

Again, these follow the GoLite specifications.

Some Rules:

* **Binary operations (particular to our implementation)**:

Types must be the same on both sides of the operator. We do not allow any constant coercion, meaning the integer 1 is never automatically converted to a float64 in `1.2 + 1`. This has a side effect that increments and decrements are only supported on integers.

bin\_op\_nocoercion.go

* Function types and returns:

The type of the return statement has to be assignable to the return type of the enclosing function.

return\_mismatch.go

* Variable assignments:

The expression has to be assignable to the lvalue on the left hand side.

assign\_string.go

* Type casts:

Expressions can only be converted to “int, float64, bool, rune, or a type alias that maps to one of those four”

conversion\_nonbasic.go

* Append:

First arguments must be a slice.

append\_array.go

* Structs:

Are considered equivalent if they have the same fields of the same type in the same order.

struct\_mismatch.go

* Printing:

We only allow printing of basic types (no arrays or slices)

print\_nonbasic.go

# Team Organization

Dan:

* Type checker.
* Symbol table printing.

Justin:

* Symbol pass.
* Expression type printing.

Sergei:

* Symbol table.