Laurence Stolzenberg (260657154)

Maxence Hull (260706895)

Milestone 3

Due March 26th 2017

Compiler Design (COMP520)
Presented to Alexander Krolik

McGill University, School of Computer Science

Decisions and details of our project

Python as a target language:

We chose Python 2.7 as the target language. It seems like a beneficial choice for multiple reasons:

- Like Go, Python natively includes type inference which makes our job easier for some statements like variables declarations. Moreover, Python syntax allows us to group some complex cases (like variable declaration, assignment and short variable declaration) into a single solution (since it does not make distinctions between those cases).
- Python's system library natively includes many of GoLite's operators and expressions (like append, print…).
- Many GoLite statements (if/else, for loops…) and expressions can be easily expressed in Python. In fact, Python and GoLite syntax are very similar.
- Debugging Python programs is easy: error messages are often very clear and no compilation is needed.

However, choosing Python as a target language also have some drawbacks:

- Explicit typing and type declarations are not allowed in Python. However, given the fact that many type declarations refer to basic types (string, integer…), some parts of this problem can be easily resolved.
- Structures do not have a strict equivalent in Python (but it might be simulated through objects).

- Arrays do not have a strict equivalent in Python (but can be simulated through lists).
- Since Python is an interpreted language, it is slow (compared to compiled languages such as C).
- Some statements (like switch) or operators (like binary bit-clear) are not present in Python 2.7

In conclusion, Python seems like a reasonable choice for 80% of GoLite. But some cases (like type declaration and structures) will be much harder to handle.

What have been done for Milestone 3:

- Operators translation from GoLite to Python
- Unary and binary expressions
- Literals (excepts runes, raw strings and octal)
- Many statements such as:
    - For statement
    - If statement
    - Variable and function declarations
    - Return statement
    - Switch statement
    - Increment, decrement statements
    - Short variable declaration statement
    - Assignment statement
    - Print, println statements

  - Break, continue and block statements
  - Op-assign statement

What must be done for Milestone 4:

- Type declaration
- Arrays, slices and structures
- Type casts
- Take into account complex types (comparison, binary and unary operations…)
- Resolve some type checking problems

Tests:

- Test_var_declaration.go
  - Test all implicit basic types inferences cases.
  - Test all ways to declare variables: classical, distributed, short declarations…
  - Test decimal, octal and hexadecimal integers (n.b.: since hexadecimal and octal do not seem to work with codegen, I decided to remove them from the test on GitHub)
  - Test runes, strings and raw strings
  - Test structure, array and slice declarations
  - Test variable declarations with complex type
- Type_declaration_test.go
  - Test type declarations derived from a basic type (int, float64…)

    - Test distributed type declaration
    - Test type declaration derived from a structure
    - Test type declaration derived from a newly declared type
    - Test type declaration derived from an array or a slice
    - Test type declaration inside a block
- Test_slices.go
    - Basic types (rune, bool, string, int and float64) slices
    - Complex types slices
    - Append new elements to all of them
    - Make an access to all of them
    - Tests multi-dimensional slices
- Test_equals.go
    - Basic type (int, float64, bool, rune and string) equality and inequality
    - Array (derived from basic types) equality and inequality
    - Structures (derived from basic types) equality and inequality
- Test_arrays.go
    - Basic types (rune, bool, string, int and float64) arrays
    - Complex types arrays
    - Add new elements to all of them
    - Replace some elements
    - Make an access to all of them
    - Tests multi-dimensional arrays