GoLite Compiler - Milestone 2

COMP 520, Compiler Design
McGill University

Anton Gladyr

April 2, 2020

1 Introduction

The goal of this milestone is to implement scoping and type checking systems. To test these two modules, 2 new modes were added to the compiler:

- *symbol*: Outputs the symbol table representing the declarations that are added in each scope as well as their type. If during scoping an error occurs, compiler generates an error message and exits with status code 1.
- *typecheck*: If the input program code is type correct, compiler exits with status code 0 and outputs "OK". Otherwise compiler prints an appropriate error message.

The scopes and type system for GoLite are nearly the same as for Go, except the restricted language subset. The typing and semantics rules for all GoLite constructs are defined in *GoLite Type Checking Specification* by Vincent Foley.[1] Therefore, all the rules in the compiler are implemented in accordance with the specification.

2 Symbol Table

Symbol table is used to describe and analyze declarations and uses of identifiers. The structure is implemented as a cactus stack of hash tables. Implemented symbol table provides 3 primary functions:

- Mapping identifiers to declarations/other information.
- Linking uses with their respective symbols.
- Scoping and unscoping.

2.1 Declarations

Declarations define new identifiers in the symbol table. GoLite makes a simplification that identifiers must be declared before they are used, so we avoid recursion calls. If an identifier is already declared in the current scope, the compiler generates an error. There is only exception for the special function *init* which may be declared multiple times, since it does not introduce a binding. If an identifier is already declared in the parent scope, the new mapping will shadow the previous mapping.

At the top-level scope identifiers *init* and *main* must be used only for function declarations with no parameters and return type. *main* function must be declared only once.

2.2 Types

In provided implementation, a symbol table stores pre-declared (base) types at the top-level scope. GoLite supports 5 base types: *int, float64, bool, rune, string.* top-level scope also stores boolean identifiers. If a given type of a declaration is not declared in the symbol table, an error is raised.

References

[1] Foley, V.: *GoLite Type Checking Specification* [https://www.cs.mcgill.ca/ cs520/2020/project_mini/ Milestone2_Specifications_Mini.pdf]. Montreal, Canada: McGill University, School of Computer Science.