CS 839: Systems Verification

Lecture 11: Goose

Learning outcomes

1. Map from Go to its Goose translation, and back
2. Navigate the Goose codebase
3. Explain what is trusted in Goose

Verification, broadly

code + spec + proof

What is the code?

high-level view vs. executable image

Goose approach: import to a model

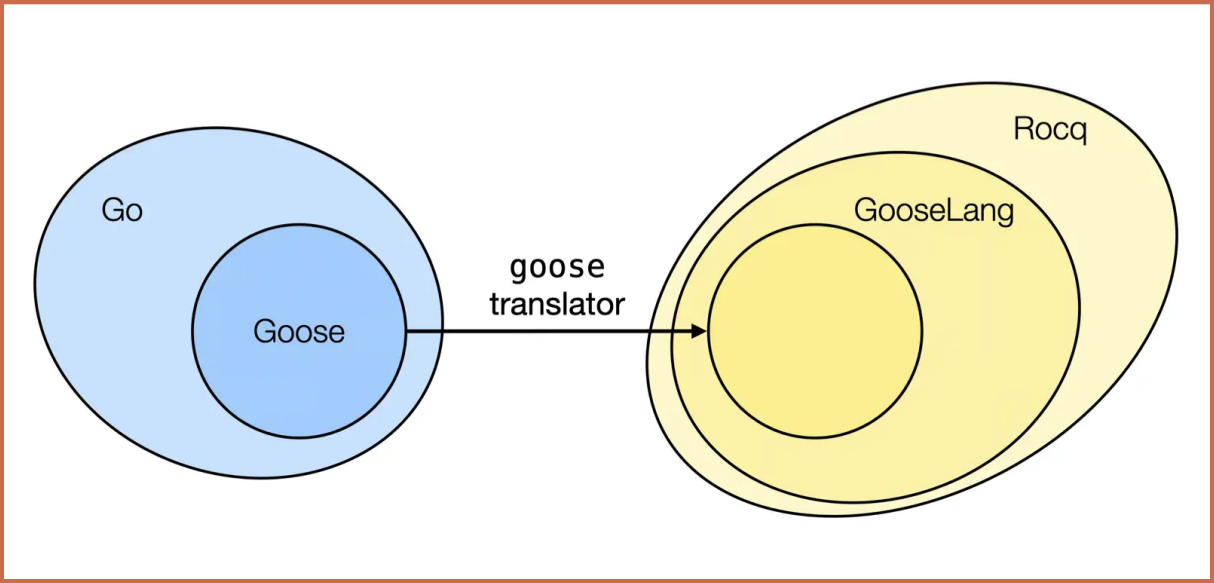
Pros

- Control what you're running
- Regular development tooling

Cons

- Proofs are separate from code

Overview



GooseLang (destination of translation)

Lambda calculus

Machine data (bytes, 64-bit integers, etc)

References, loads, stores

Concurrency

GoLang

Control flow

Structs

Slices

Maps

Packages

Translation

https://github.com/goose-lang/goose

~4,500 lines of Go

Use `go/ast` and `go/types`

(Go) packages become (Rocq) files

Functions (+ types, constants, methods) become definitions