GoGo

A Go compiler written in Go (... and assembly)

Michael Lippautz Andreas Unterweger

Compiler Construction Course, Summer 2010

June 24, 2010

Responsibilities

Michael Lippautz	**Andreas Unterweger**
Scanner	I/O library
Parser	Memory/string management
Multiplication/Division	Addition/Subtraction
Conditionals	Assignments
Loops	Address/offset calculcations
Test suite	Symbol table

What is GoGo?

- A self-compiling Go compiler
- Input language: A subset of the Go language [1]
  - C-like syntax with additional keywords
  - Reduced feature set through EBNF
- Output language: Plan9 x64 assembly
  - Output in text form, not binary form
  - Requires Plan9 assembler for binary form
  - Requires Plan9 linker for ELF executables

[1] golang.org

What is so special about GoGo? (1/2)

- Advanced **string** management
  - More memory allocated than initially needed
  - "Spare" memory for future concatenations
  - Drastically reduces memory consumption
- Implementation of **pointers**
  - Implicit dereferring on structure access
  - No explicit dereferring possible (EBNF)
  - Address operator (&) complicates assignments

Example:

```
a = &b[i].c.d[3];
```

- Explanatory **comments** in assembly output
  - Source file and line included
  - Option to disable (debug level reduction)

What is so special about GoGo? (2/2)

- **Lazy evaluation** over multiple expression levels
  - Merging of positive and negative labels (if appropriate)

Example:

```
if (done!=1) && (((a<1) && (b<2)) || ((c<3) && (d<4))) { ...
```

- **Error handling**
  - Weak symbols (';', ...)
  - Syncing points (functions)
  - Code generation error stop compiling, but continues parsing

Building

Demo

- Recursive self-compilation
- Advanced Fibonacci example