CS 473: COMPILER DESIGN
Compilation in a Nutshell

Source Code (Character stream)
if (b == 0) { a = 1; }

Token stream:
if ( b == 0 ) { a = 0 ; }

Abstract Syntax Tree:
If
  Eq
  b
  0
  Assn
  a
  1
  None

Assembly Code
11:
cmpq %eax, $0
  jeq 12
  jmp 13
12:
  ...

Lexical Analysis
Parsing
Analysis & Transformation
Backend
Languages in a Compiler

Simple

Source code
if (b == 0) a = 0;

Lexical Analysis

Lex/Flex

Parsing

Yacc/Bison

Translation and Optimization

C

Code Generation

MIPS

Assembly code
CMP ECX, 0
SETBZ EAX

Intermediate code
l1:
%cnd = icmp eq i64 %b, 0
br i1 %cnd, label %l2, label %l3
Languages in a Compiler

Simple

Source code
if (b == 0) a = 0;

Lexical Analysis

Intermediate code
l1:
%cnd = icmp eq i64 %b, 0
br i1 %cnd, label %l2,
label %l3

JavaCC

Parsing

Translation and Optimization

Java

Code Generation

MIPS

Assembly code
CMP ECX, 0
SETBZ EAX
Languages in a Compiler

**Source**

Source code

```plaintext
if (b == 0) a = 0;
```

**Bootstrapping** is when a compiler is written in the (source) language it compiles.

**IRs**

Intermediate code

```plaintext
l1:
  %cnd = icmp eq i64 %b, 0
  br i1 %cnd, label %l2, label %l3
```

**Target**

Assembly code

```plaintext
CMP ECX, 0
SETBZ EAX
```
Bootstrapping: Overview

• *Bootstrapping* is when a compiler is written in the (source) language it compiles.

• Why would we want to do this?
  – Because we like the language we’re compiling!
  – A way to test our compiler’s correctness
  – If we come up with new optimizations, our compiler will run faster too
  – If the X compiler is written in X, then X programmers can work on the compiler without learning a new language

• How is this possible?
  – A compiler is just a program.
  – If our C compiler is written in C, how do we compile it?
  – We have to start somewhere!
Questions

Top
both in language X

Compiler Code (text) → ??? → X Compiler (executable) → Program (executable)

Program Code (text) → X Compiler (executable) → Program (executable)
If we already have access to C, we could write the other compiler in C, and compile it to an executable.

If we don’t, we might have to write the other compiler directly in assembly/machine code.
• The interpreter still needs to be written in another language, but doesn’t need to do so much translation and optimization.
• This might make compilation slow, but interpreters are a lot easier to get right!
• Maybe we don’t need all the features of language X to write a compiler!
• We can write a compiler for a small “core” in C, and then write the compiler for full X in core X
Getting Off the Ground

both in language X

Compiler Code (text) -> Compiler for a small part of X

Program Code (text) -> X Compiler (executable) -> Program (executable)

• What did we use to implement our compiler? Switch statements, structs, enums...
• If we want to write a compiler for our language in our language, we’d want to add those features to the language first!
Using the Bootstrapped Compiler

both in language X

Compiler Code (text) → Basic compiler or interpreter → X Compiler (executable) → Program (executable)

Program Code (text) → X Compiler (executable) → Program (executable)
Using the Bootstrapped Compiler

both in language X

Compiler Code (text) ➔ Basic compiler or interpreter ➔ X Compiler (executable) ➔ Program (executable)

Program Code (text)

• If we want to extend or modify the language...
If we want to extend or modify the language...
we can write all the new features in X, as long as the core language doesn’t change
• Now we have two different executables for the same compiler, based on the same source program!
• We can run them both through test programs to try to find bugs in the compiler
Bootstrapping: Summary

• We can write a compiler in the language it compiles, but we have to start with a basic compiler/interpreter written in another language, so we can execute the compiler!

• Why would we want to do this?
  – Because we like the language we’re compiling!
  – A way to test our compiler’s correctness
  – If we come up with new optimizations, our compiler will run faster too
  – If the X compiler is written in X, then X programmers can work on the compiler without learning a new language
  – We can write new features/extensions in the language we like, as long as we don’t change the features used in the compiler

• Exercise: take a few minutes to fill out the course feedback form! (don’t submit anything on Gradescope)