Compiler Design

01-Overview

Apr 24, 2025

Agenda

1. Administrative stuffs
2. Compiler Design
3. Labs

Administrative stuffs

The team

Prof. Dr. André Platzer

Dr. Enguerrand Prebet

HiWis:

- Hannes Greule
- Darius Schefer

Lectures

Lectures

Two lectures per week room 131 (here):

- Monday 11:30-13:00
- Thursday 9:45-11:15 (now)

What to expect

- Labs
- No graded theoretic homework
- Written exam

subject to change

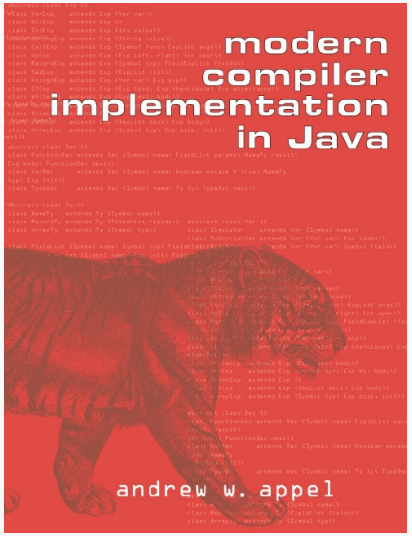
All info on `lfcps.org/course/compiler`

Registration on `moodle.vads.kastel.kit.edu`

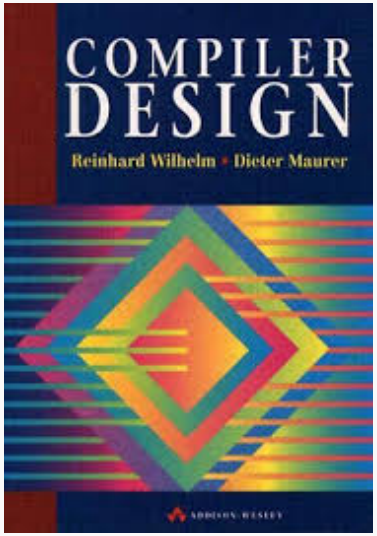
References

Lecture notes: `lfcps.org/course/compiler-schedule`

Andrew W. Appel, Modern Compiler Implementation in ML/Java/C



Reinhard Wilhelm et al, Compiler Design series



Do not replace the lecture

Compiler Design

Course Goals

- Understand the different phases of a compiler
- Perform analysis in the different layers
  - Optimising compiler and code
  - Managing memory
  - ...
- Implement a end-to-end compiler (Labs)

Why care about compilers?

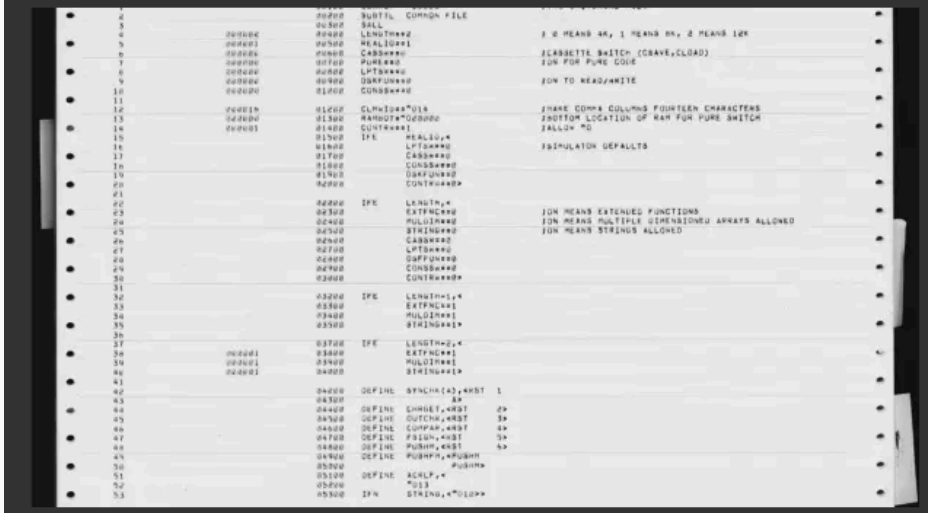
- Base of programming
  - What could go wrong?
- Building a compiler teaches lots of different reusable skills/concepts

Why care about compilers?

- Base of programming
  - What could go wrong?
- Building a compiler teaches lots of different reusable skills/concepts

"My coolest code": Bill Gates publishes source code from Altair Basic

Microsoft is celebrating its 50th birthday these days, and it all started with the Altair Basic program. Bill Gates has now published its source code.



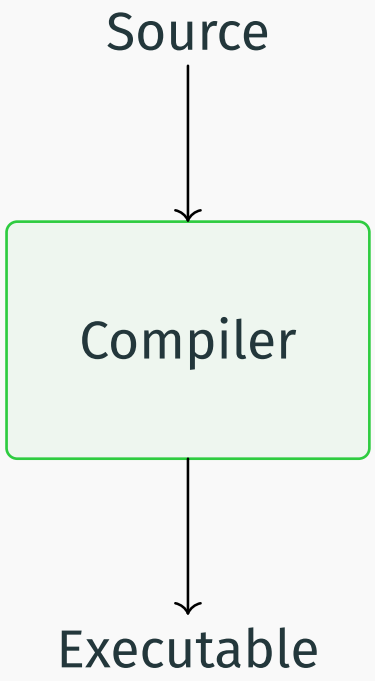
An original was scanned for publication. (Image: gatesnotes.com)

What is a compiler?

Translator for one language to another

5 principles:

- Correctness
- Efficiency
- Interoperability
- Usability
- Retargetability



Correctness

Correctness is essential for compiler.

→ need to give semantics to both languages

Correctness

Correctness is essential for compiler.

→ need to give semantics to both languages

“undefined behavior - ... the compiled program is not required to do anything meaningful.”

en.cppreference.com/w/c/language/behavior

Correctness

Correctness is essential for compiler.

→ need to give semantics to both languages

How to check for correctness?

- ~~Formally verify it~~[2]
- Test extensively

> “undefined behavior - ... the compiled program is not required to do anything meaningful.”
>
> en.cppreference.com/w/c/language/behavior

[2] compcert.org

Efficiency

Goals:

- The compiler is fast

Efficiency

Goals:

- The compiler is fast
- The code generated is also fast

Trade-off → spend compiler time optimising the code

Efficiency

Goals:

- The compiler is fast
- The code generated is also fast

Trade-off → spend compiler time optimising the code

But cannot forsake correctness!

> “But I got the wrong answer really fast!”

Interoperability & Retargetability

Interoperability

Programs do not work alone.

→ Need calling conventions/ABI

Retargetability

One compiler, multiple target (and source) languages (x86-64, ARM, LLVM...)

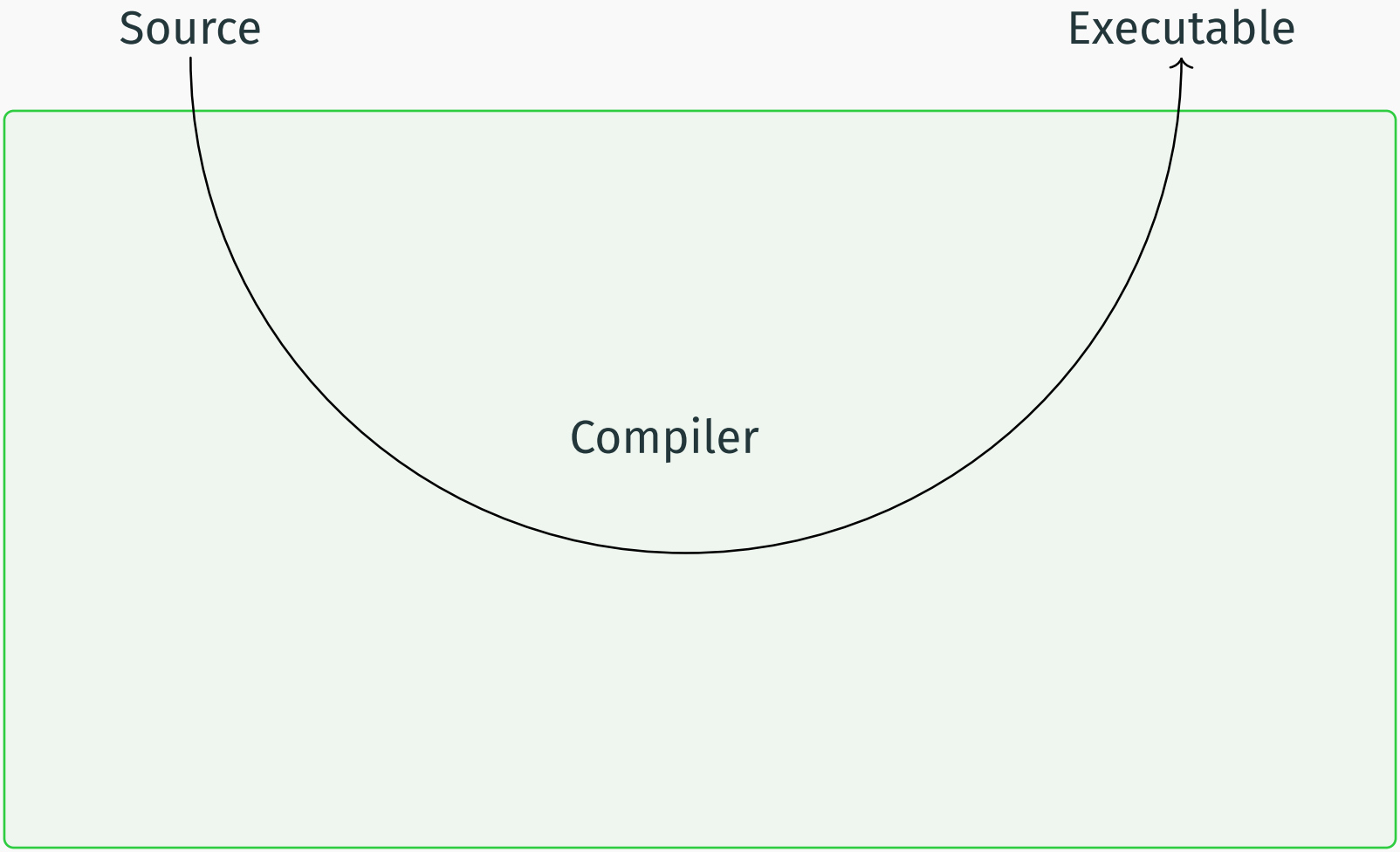
Usability

A compiler is meant to be used.

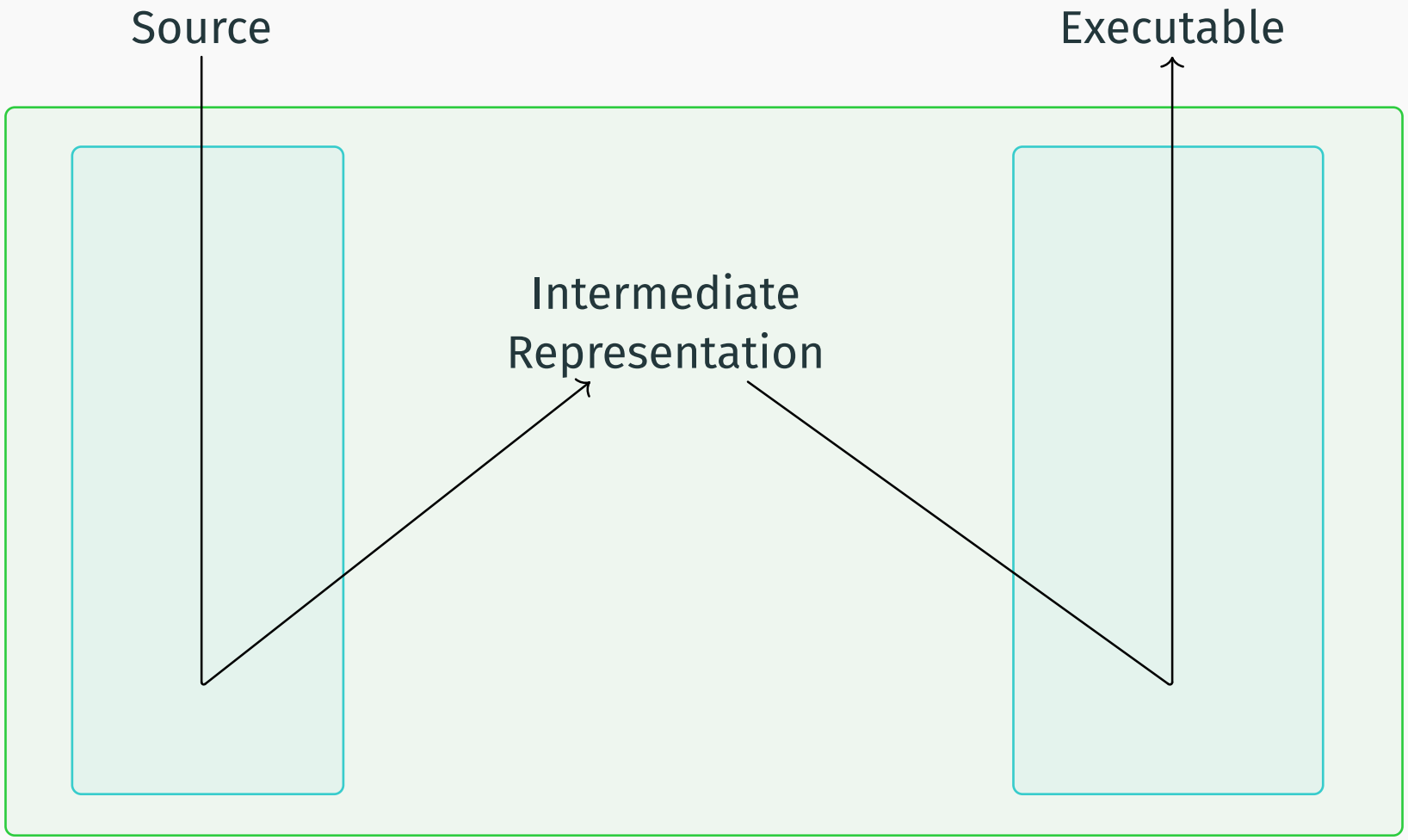
“Error.” compared to	“Type error”
	“Type error at foo.c”
	“Type error at foo.c:3:5, did you mean… ?”

Not tested but you will have to debug your compiler.

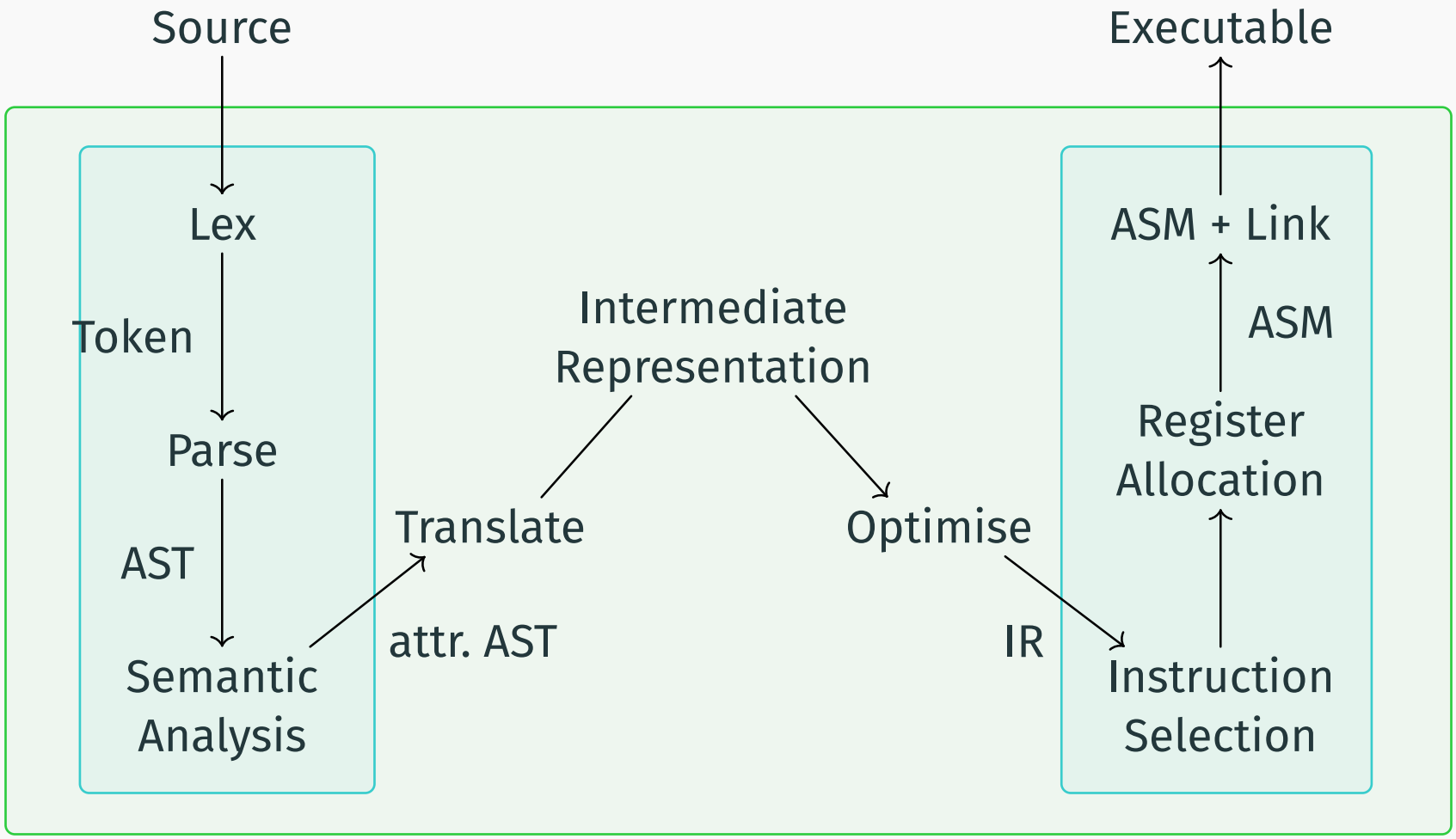
Compiler Structure



Compiler Structure



Compiler Structure



Labs

Overview

Build a compiler cumulatively (~5 labs)

- First compiler
- Control flow
- Functions
- Array

Each lab builds on the previous one → including errors...

Teams: solo or in pair (see `moodle`)

Languages

Source language: C0[2]

C but less weird

- Deterministic
- No cast

Target language: x86-64

Starter code in:

- Haskell
- Java

→ very strongly recommanded

[2] `c0.cs.cmu.edu`

Evaluation

Submission platform: `compiler.vads.kastel.kit.edu`

Submit tests

- positive
- negative
- be creative

Submit a compiler

Tested against **“everything”**

- all submitted tests
- previous labs tests
- our own tests

Good tests are purposely designed to find bugs

Bonus grade

Summary

- Register on moodle and form teams
- Don’t fall behind on labs

  → the next one depends on the current

Questions?