

LLVM



The LLVM Compiler Infrastructure

Contents

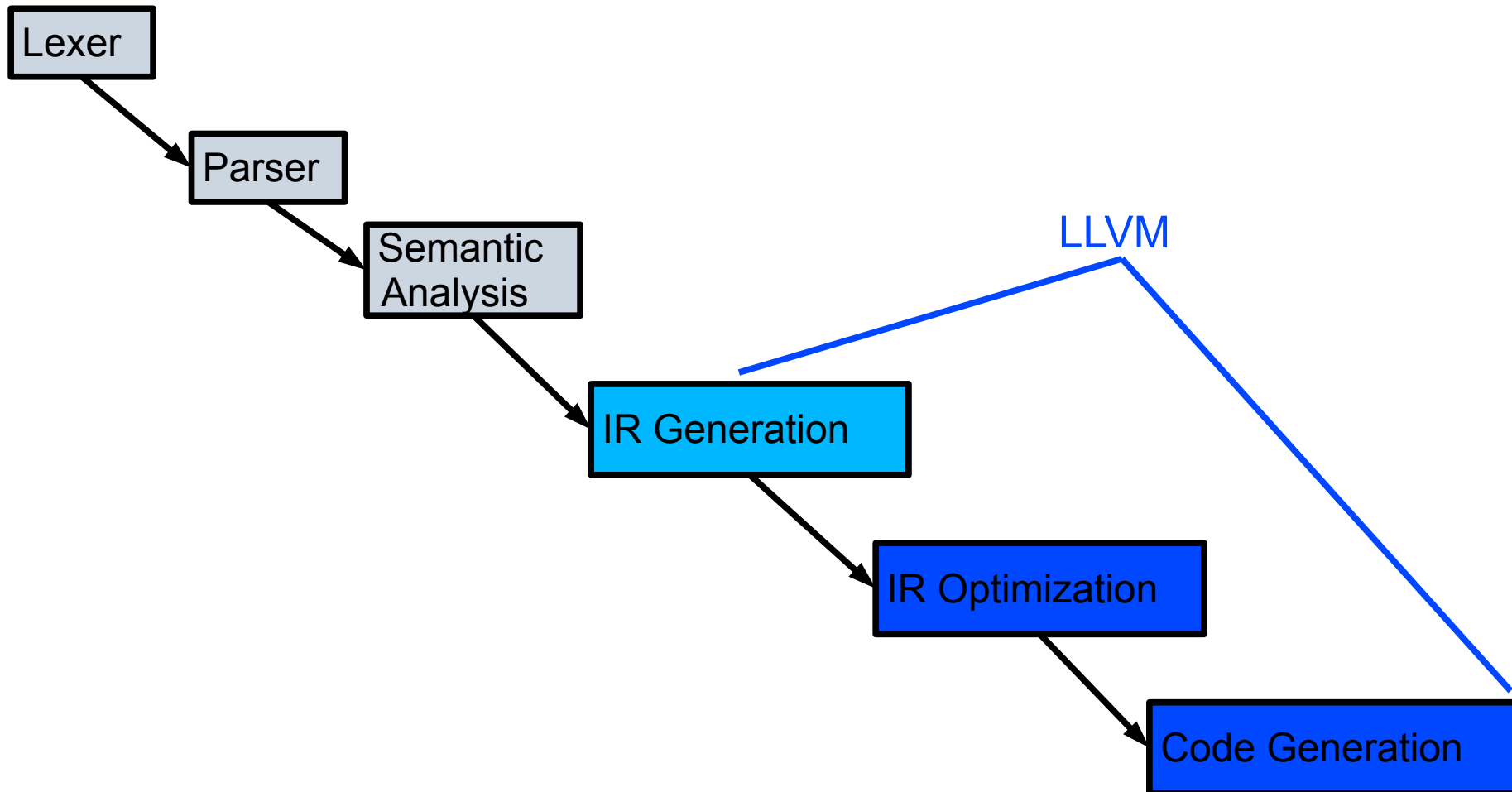
- What is LLVM
- Compiler Pipeline
- LLVM Components in Detail
- LLVM Intermediate Representation
- LLVM Features

What is LLVM

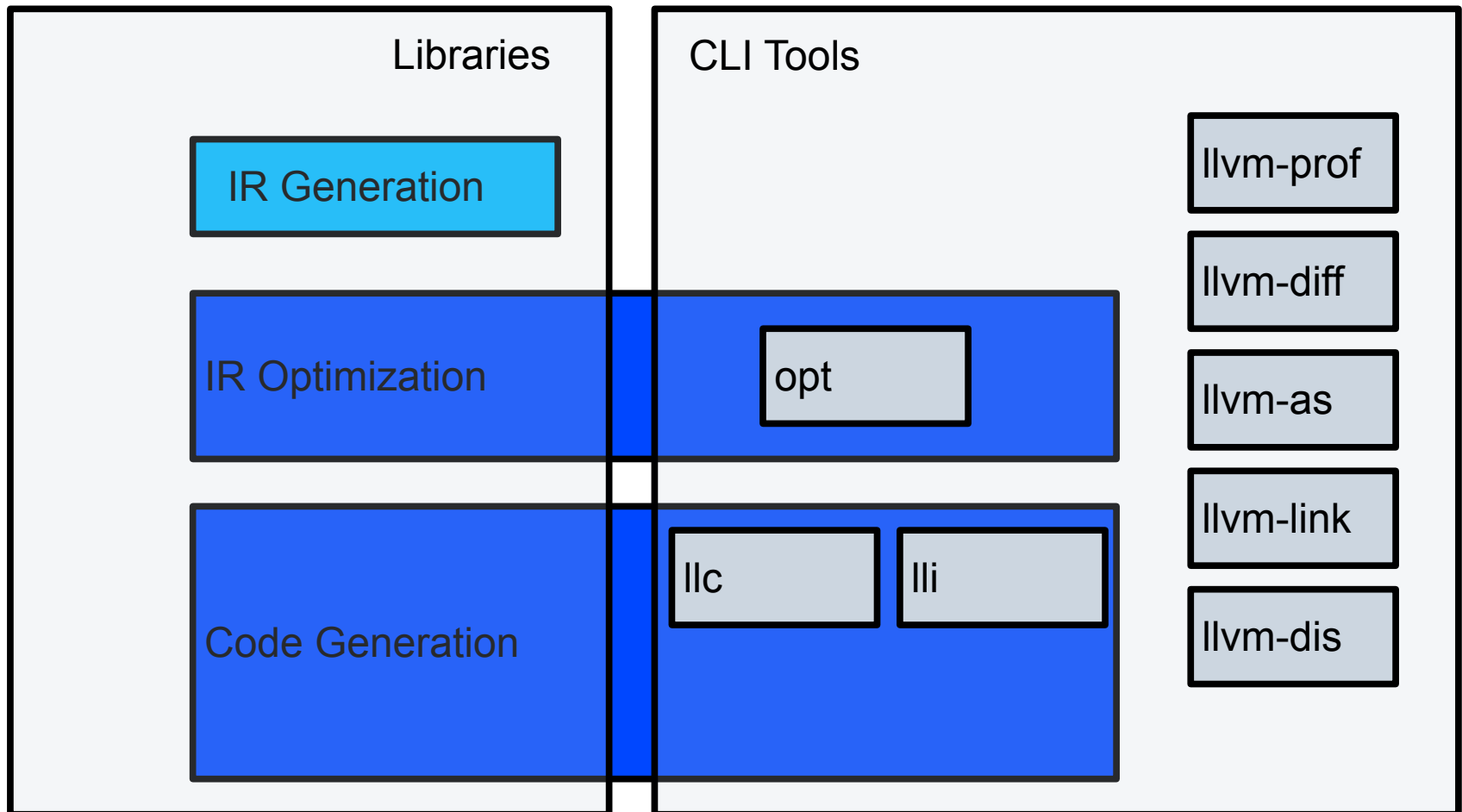


- „(...) is a collection of modular and reusable compiler and toolchain technologies.“
- Consists of:
 - C++ Libraries (API)
 - Command Line Tools

Compiler Pipeline



LLVM Components in Detail



LLVM Intermediate Representation

Sample.ll

```

; Declare the string constant as a global constant.
@.str = private unnamed_addr constant [13 x i8] c"hello world\0A\00"

; External declaration of the puts function
declare i32 @puts(i8* nocapture) nounwind

; Definition of main function
define i32 @main() { ; i32()*
  ; Convert [13 x i8]* to i8 *...
  %cast210 = getelementptr [13 x i8]* @.str, i64 0, i64 0

  ; Call puts function to write out the string to stdout.
  call i32 @puts(i8* %cast210)
  ret i32 0
}

```

- Three Representations of LLVM-IR

- In Memory (not specified)

- ASCII IR (see left)

- Bitcode format

- Bitcode and ASCII IR are equivalent

- llvm-as and llvm-dis convert between ASCII and bitcode format

Features of LLVM

- Large set of optimization passes, easy to write new passes
- Well defined, self contained intermediate representation
- Many targets, easier to write new targets than, i.e. for GCC
- Abstraction level supports easier llvm emission
(derived types/functions exist)
- Modular architecture allows reuse of individual features
- Interfacing with C possible (but not always platform independent)
- Continuous development over the recent years
- Permissive license

The End...

...thanks for Listening!