Advanced Compiler Techniques
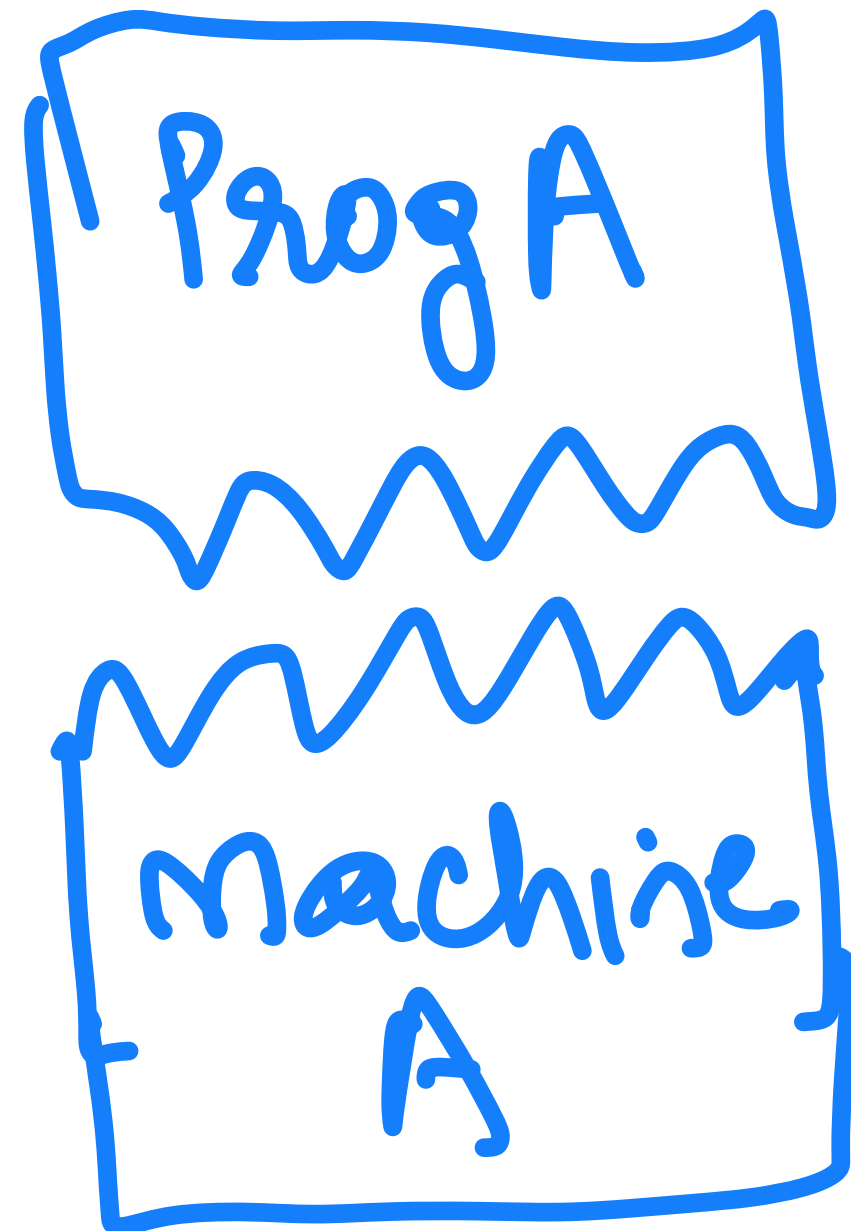
First Meeting

What is a compiler?

1. A compiler is a translator
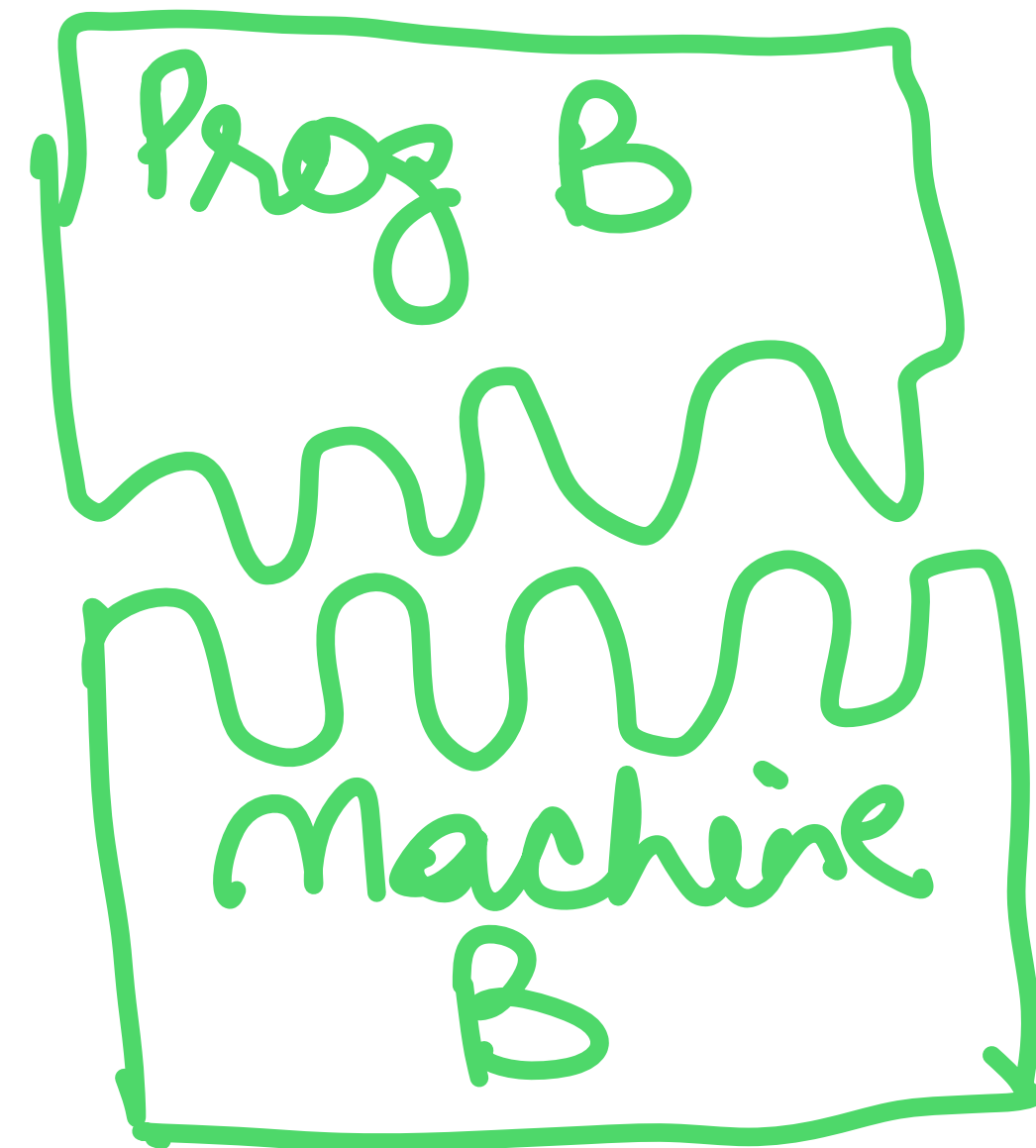
Find Prog B
such that

Prog A || Machine A
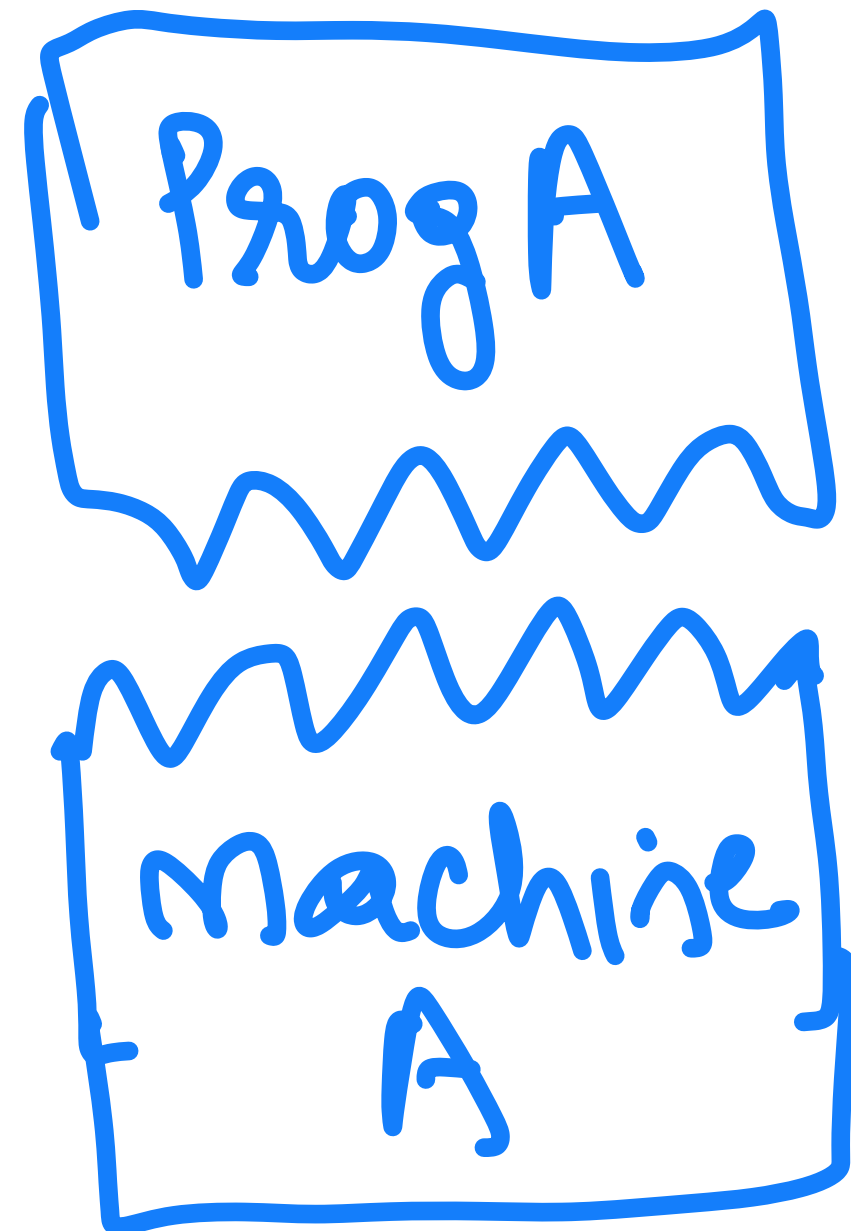$$\equiv$$
Prog B || Machine B

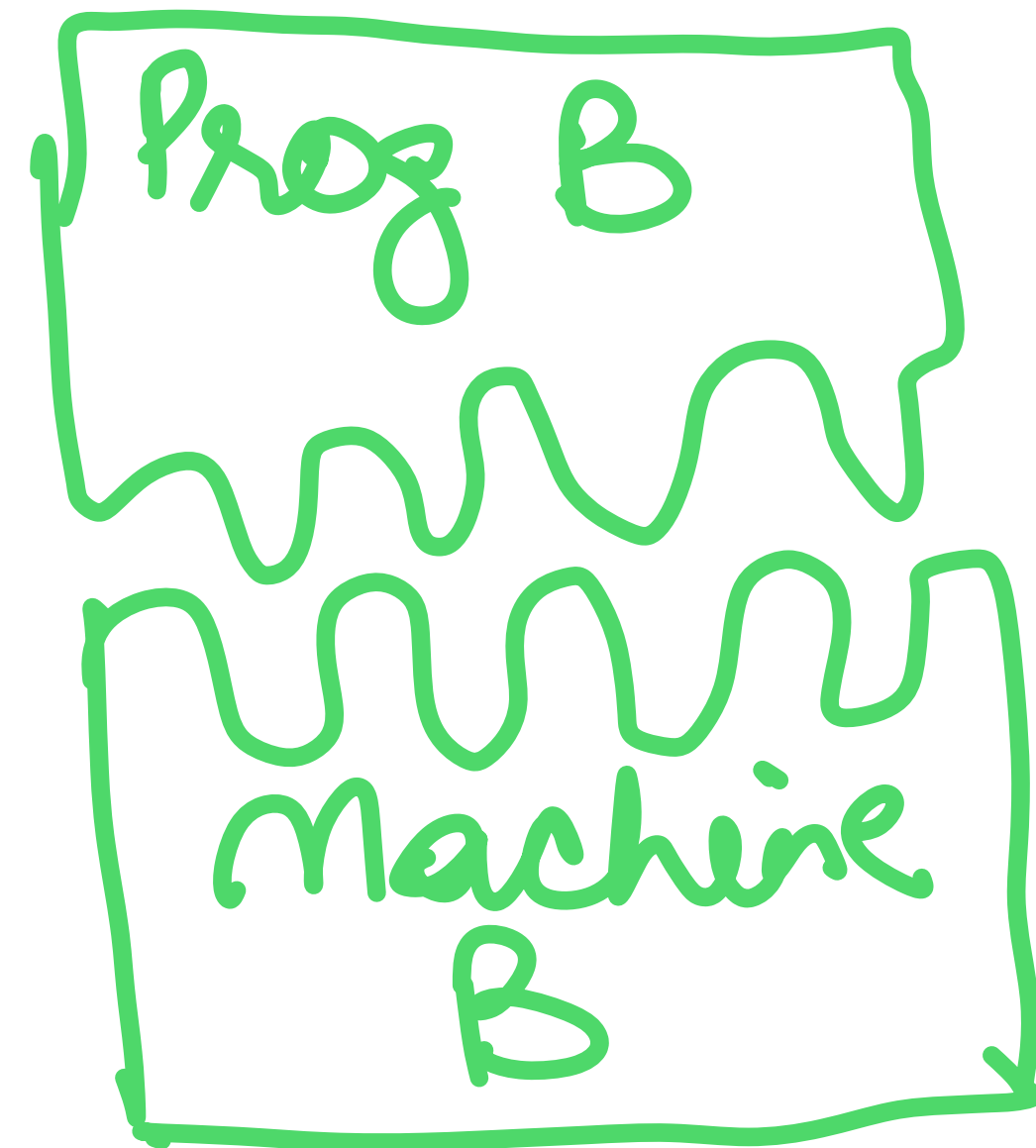# What is a compiler?

1. A compiler is a translator

Examples?

Find Prog B
such that

Prog A || Machine A
=
Prog B || Machine B

Prog A

machine A
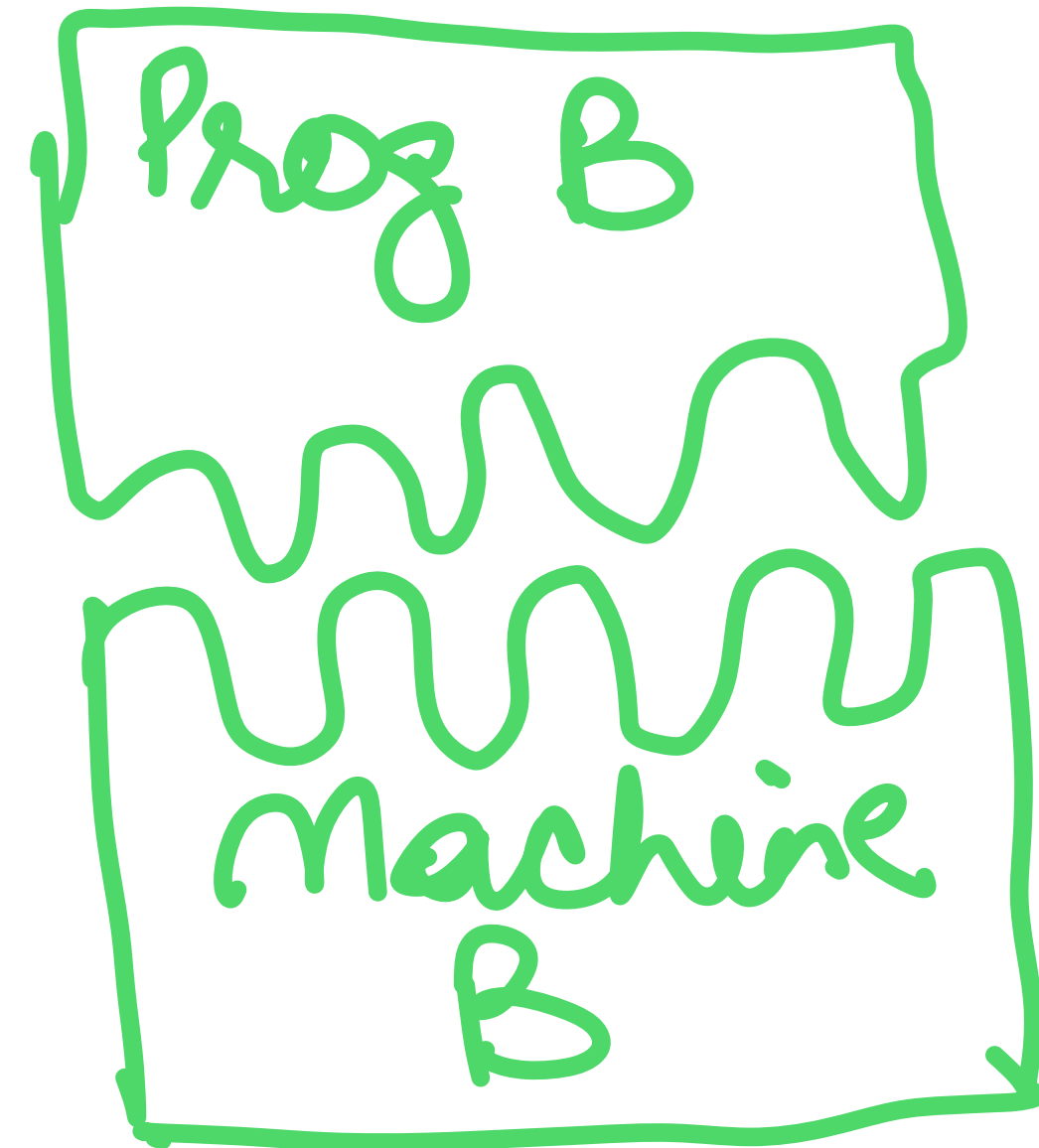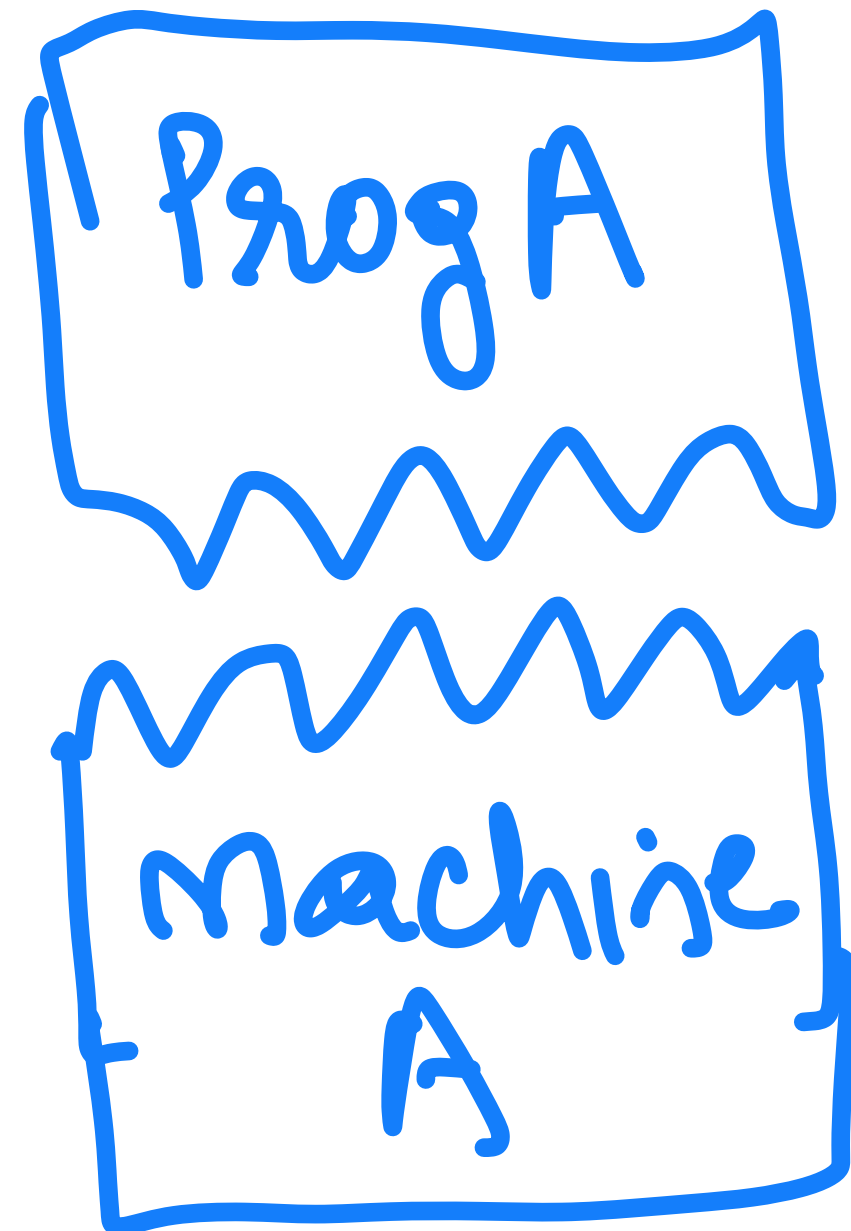
Prog B

machine B

What is a compiler?

1. A compiler is a translator

Always possible to find Prog B?

Counterexamples?

What is a compiler?

1. A compiler is a translator

set of progs on machine A

set of programs on machine B

many-to-one function

onto?

machine A

machine B
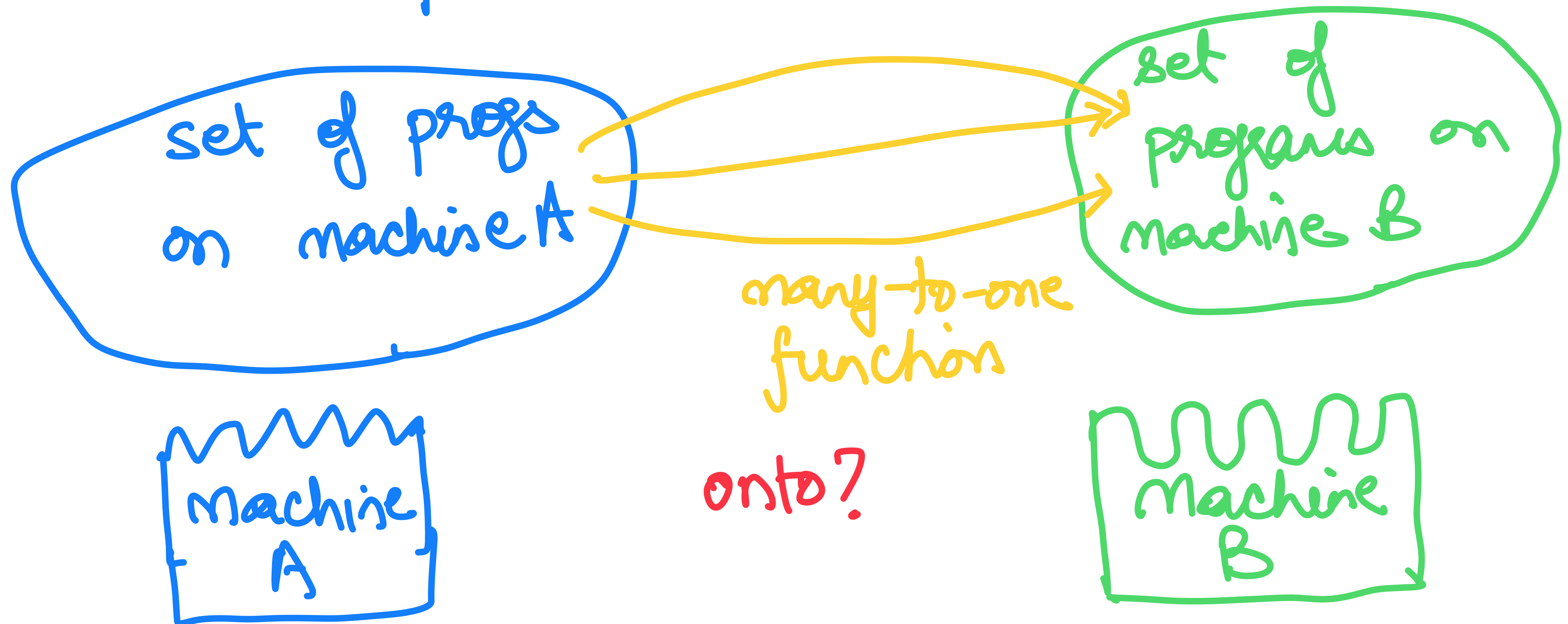
What is a compiler?

2. A compiler is an optimizer

set of
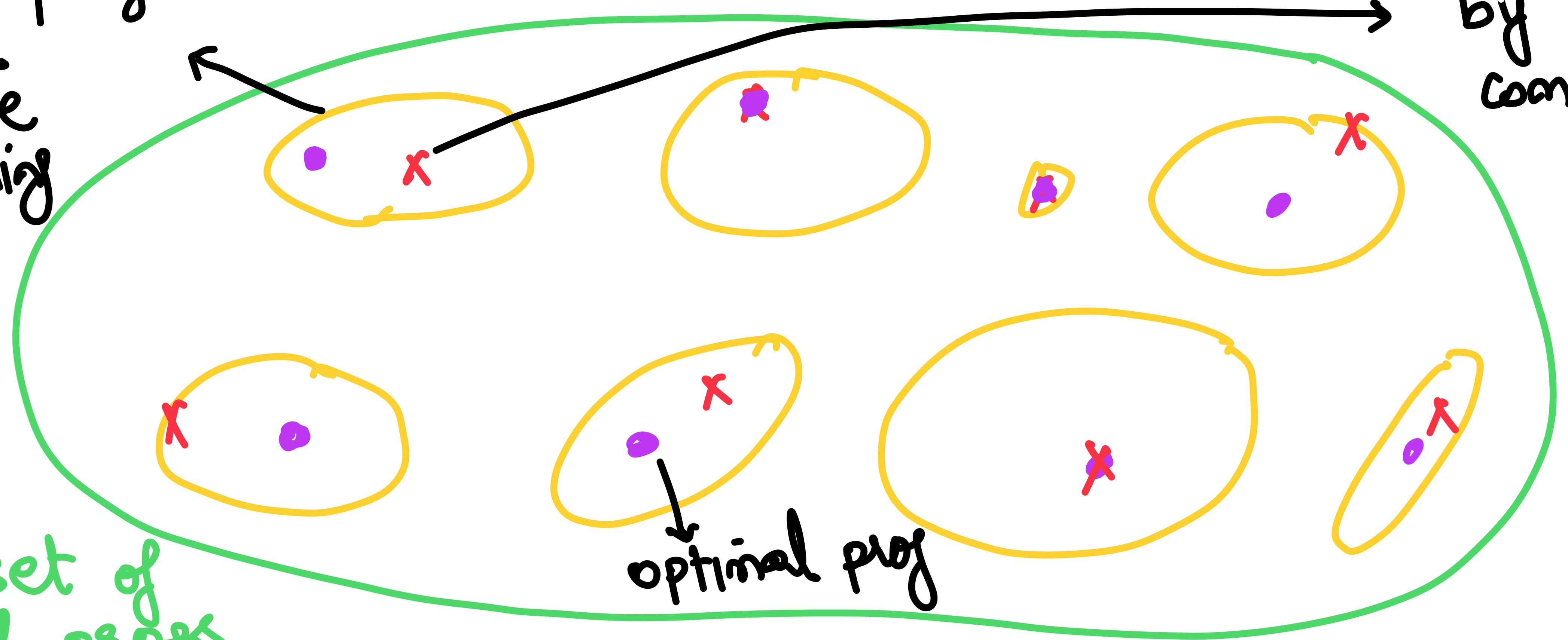all progs

What is a compiler?

2. A compiler is an optimizer

all progs have same meaning

chosen by the compiler

set of all progs

optimal prog

Is the compiler a solved problem?

1. Do we have good translators?

Is the compiler a solved problem?

1. Do we have good translators?
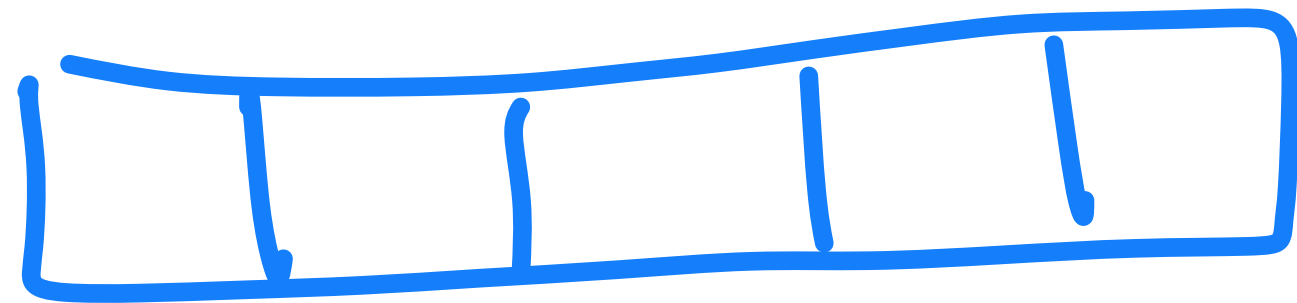
2. Do we have good optimizers?

So how have we been managing ?

A common approach:

1. Design Machine A so it is easy to program

2. Co-design Machines A and B so it is easy to translate + optimize

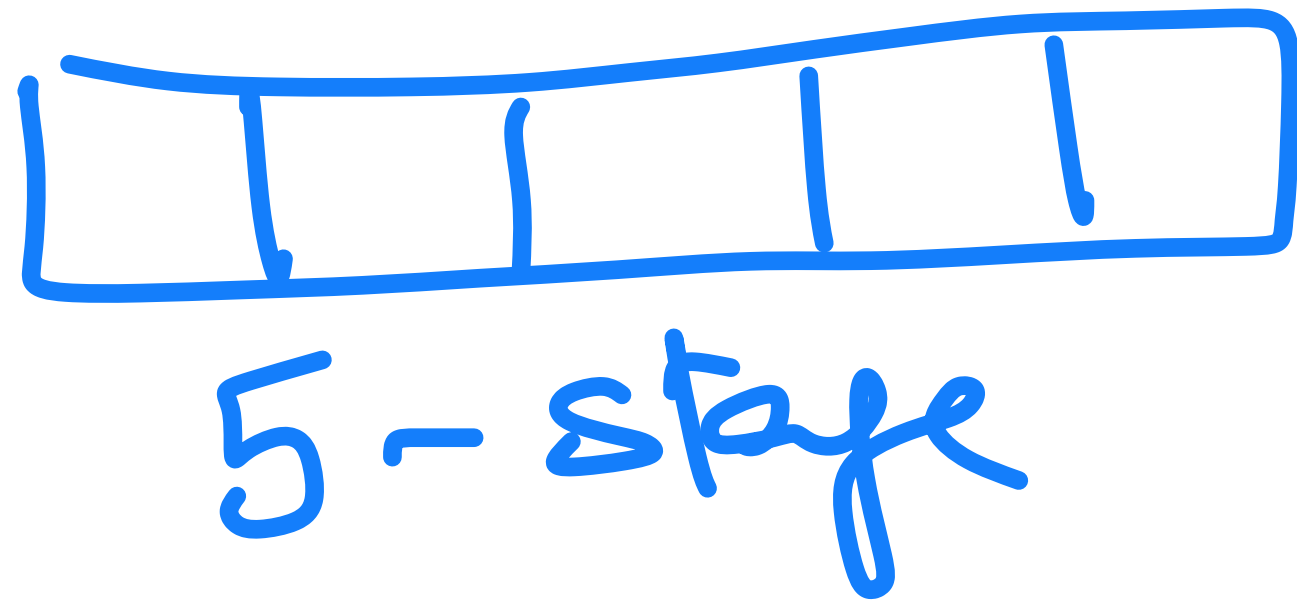# Can we keep going like this?
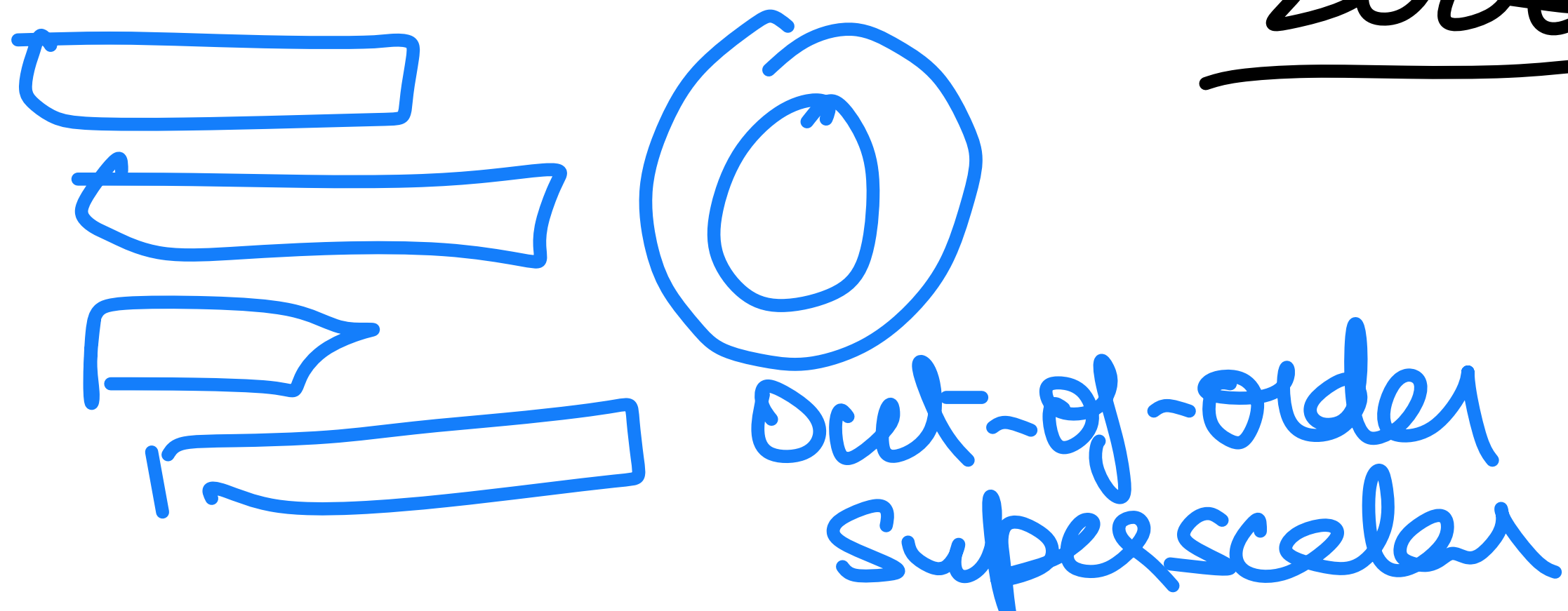
Can we keep going like this?

5-stage

80s-90s

Out-of-Order
Superscalar
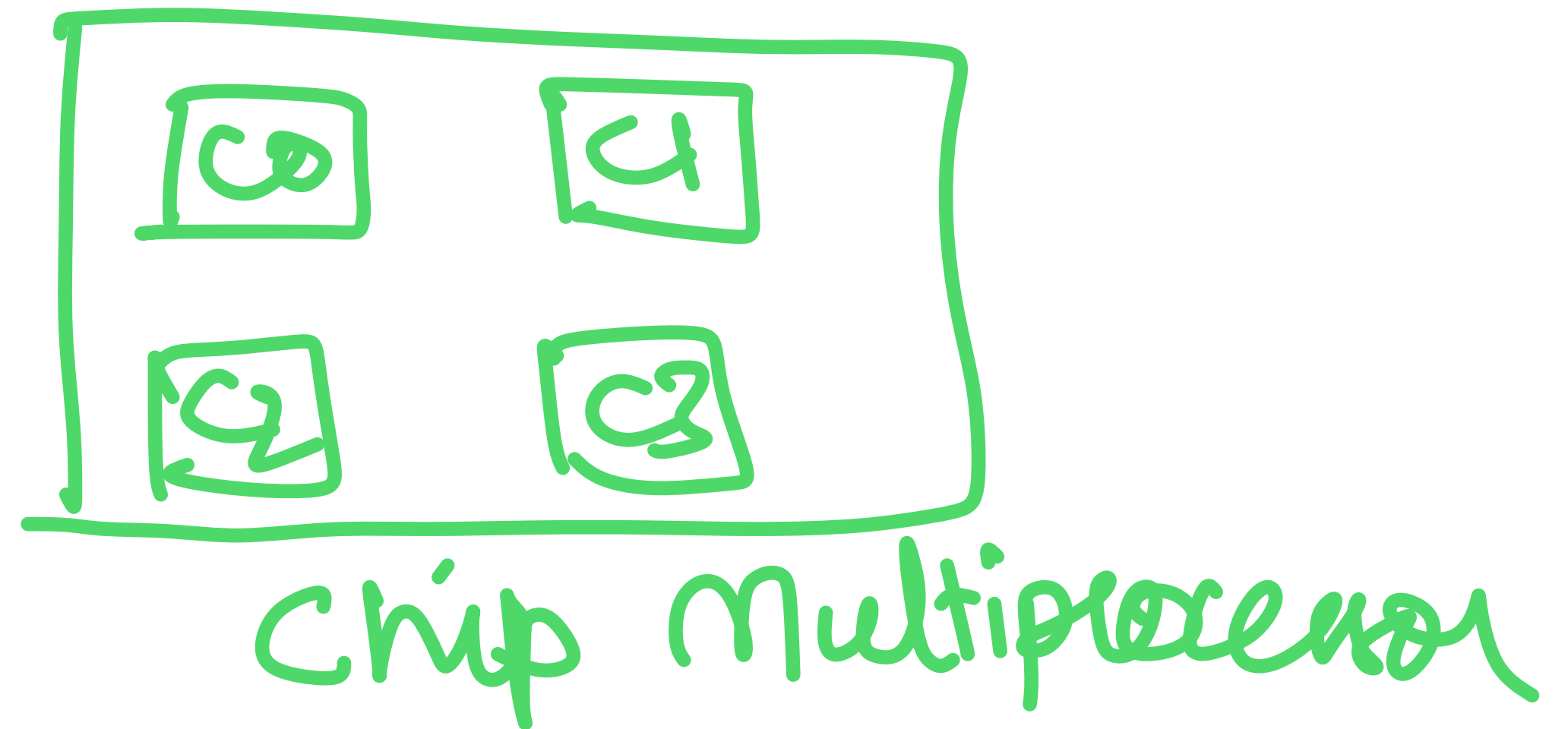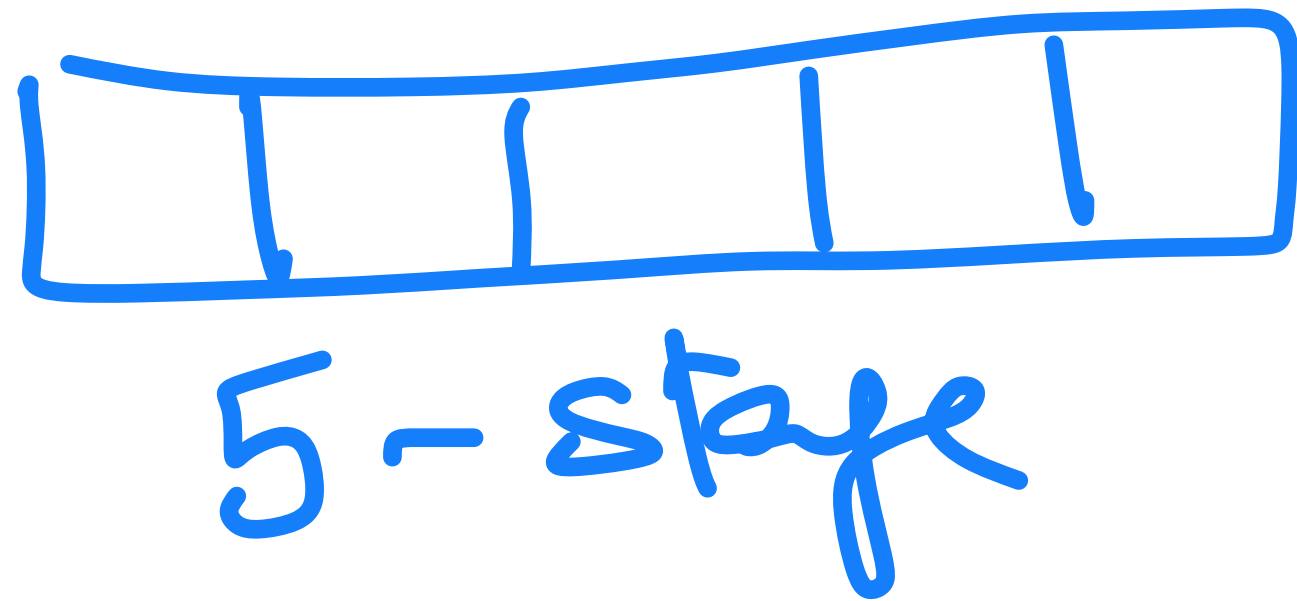
Can we keep going like this?

5-stage

80s-90s

Out-of-order
Superscalar

2000s →

Chip Multiprocessor

Can we keep going like this?

5-stage

80s-90s

Out-of-order
Superscalar

2000s

2010s

accelerator
based
designs

Chip Multiprocessor

Can we keep going like this?

5 - stage

80s - 90s

Out-of-order Superscalar

2000s

accelerator based designs

2010s

2020s

App. Specific
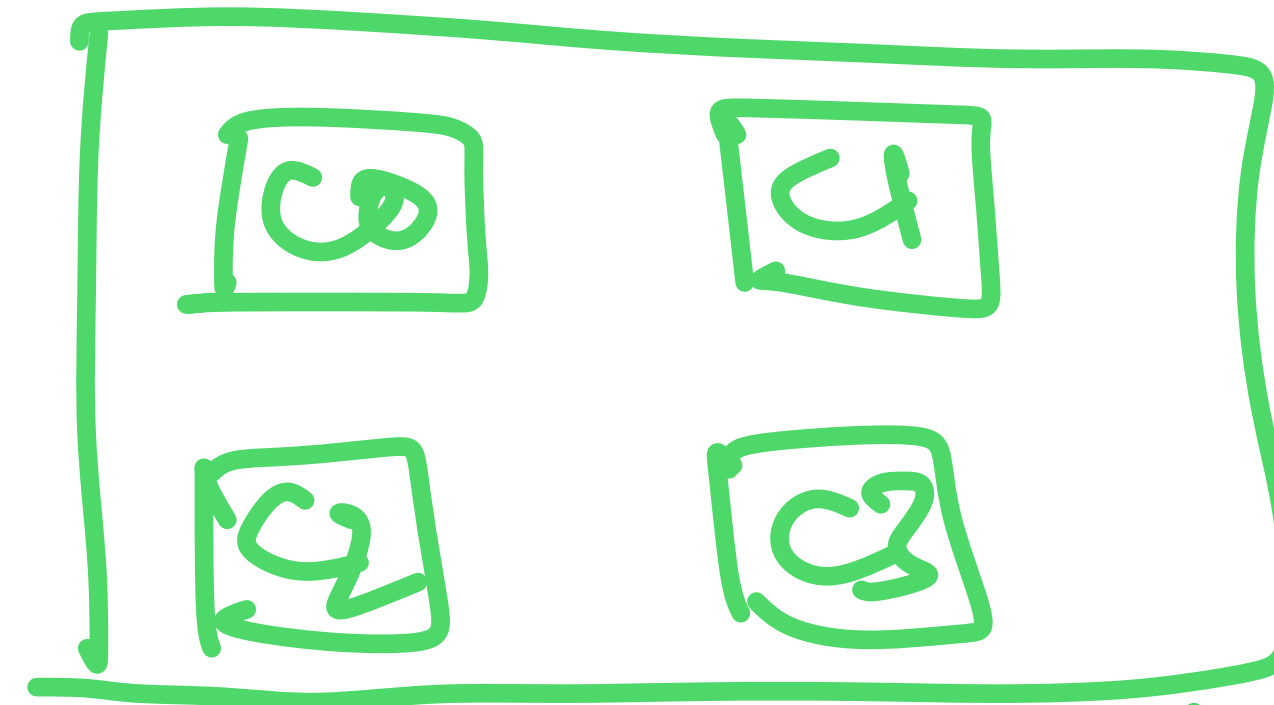
Chip Multiprocessor
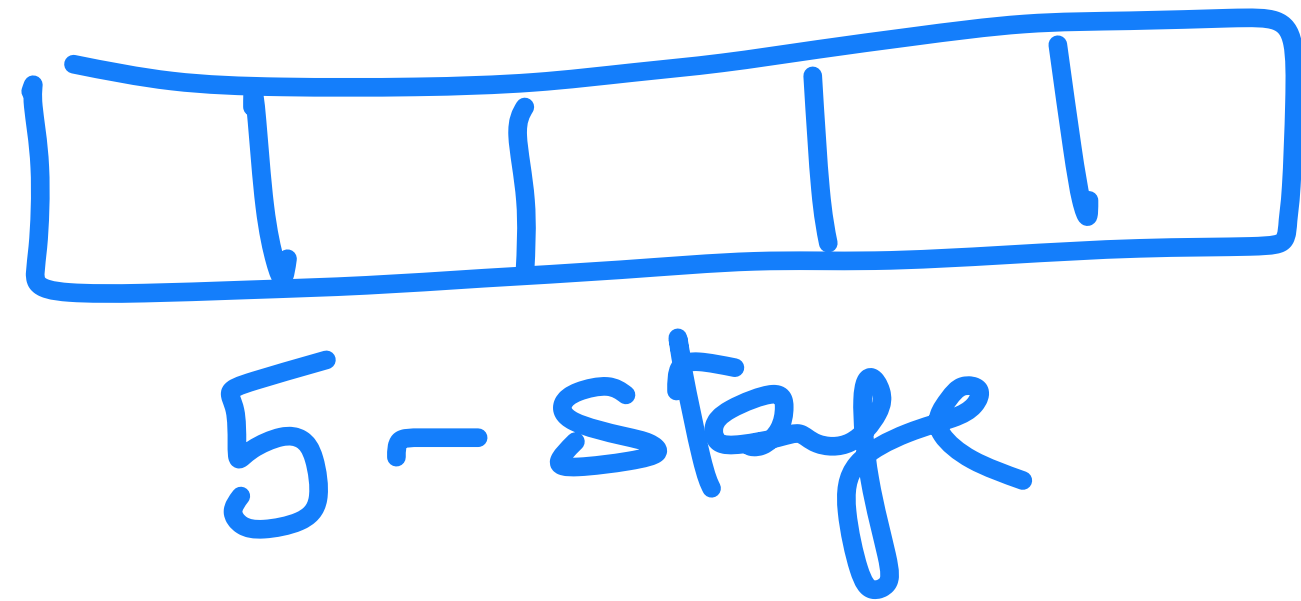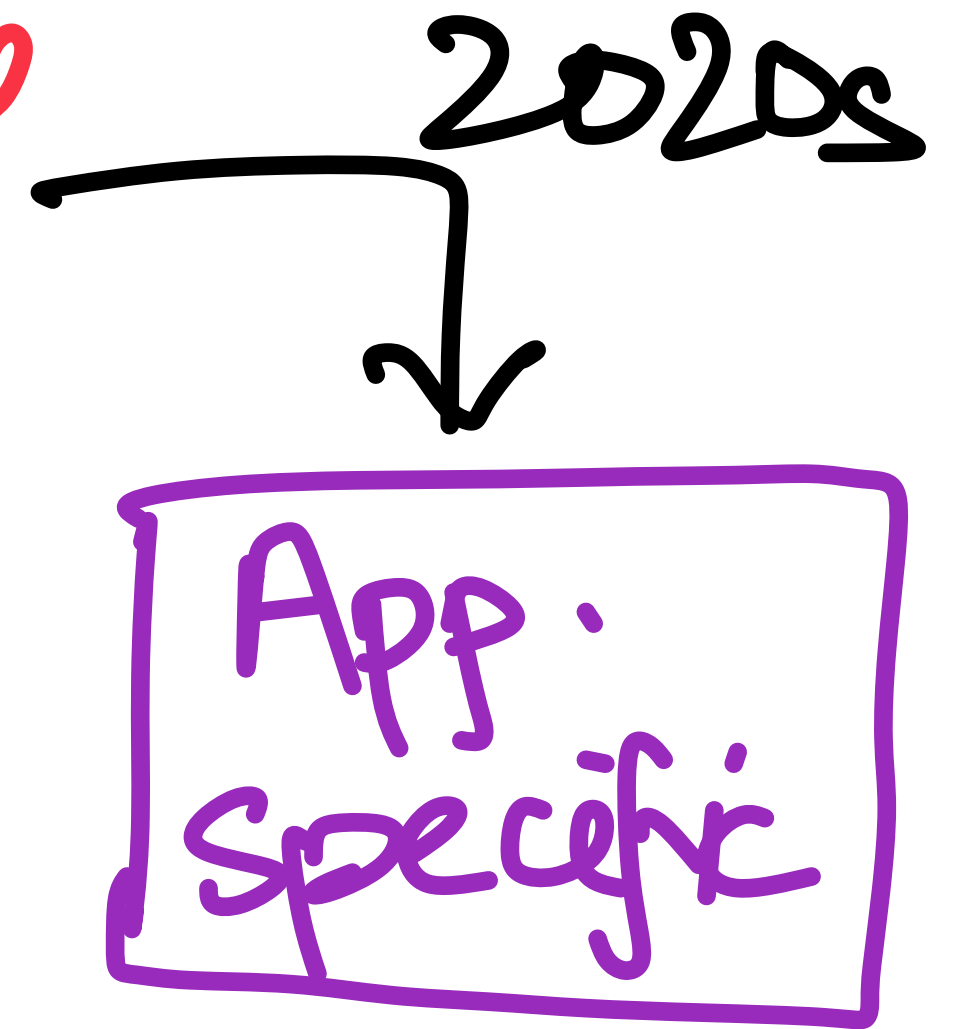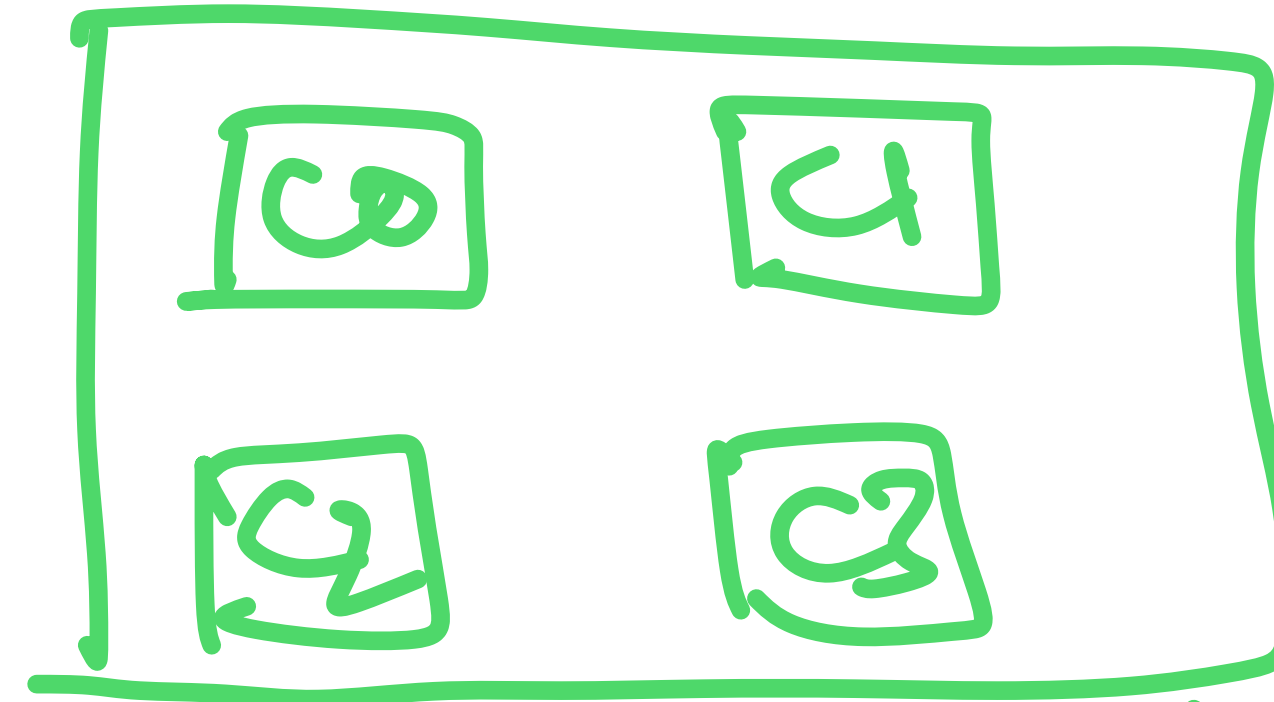
Programming Languages?

C/C++
Jave
Python
OCaml
Javascript

DSLs:

Tensorflow
Halide
Lustre
MapReduce
SQL
...

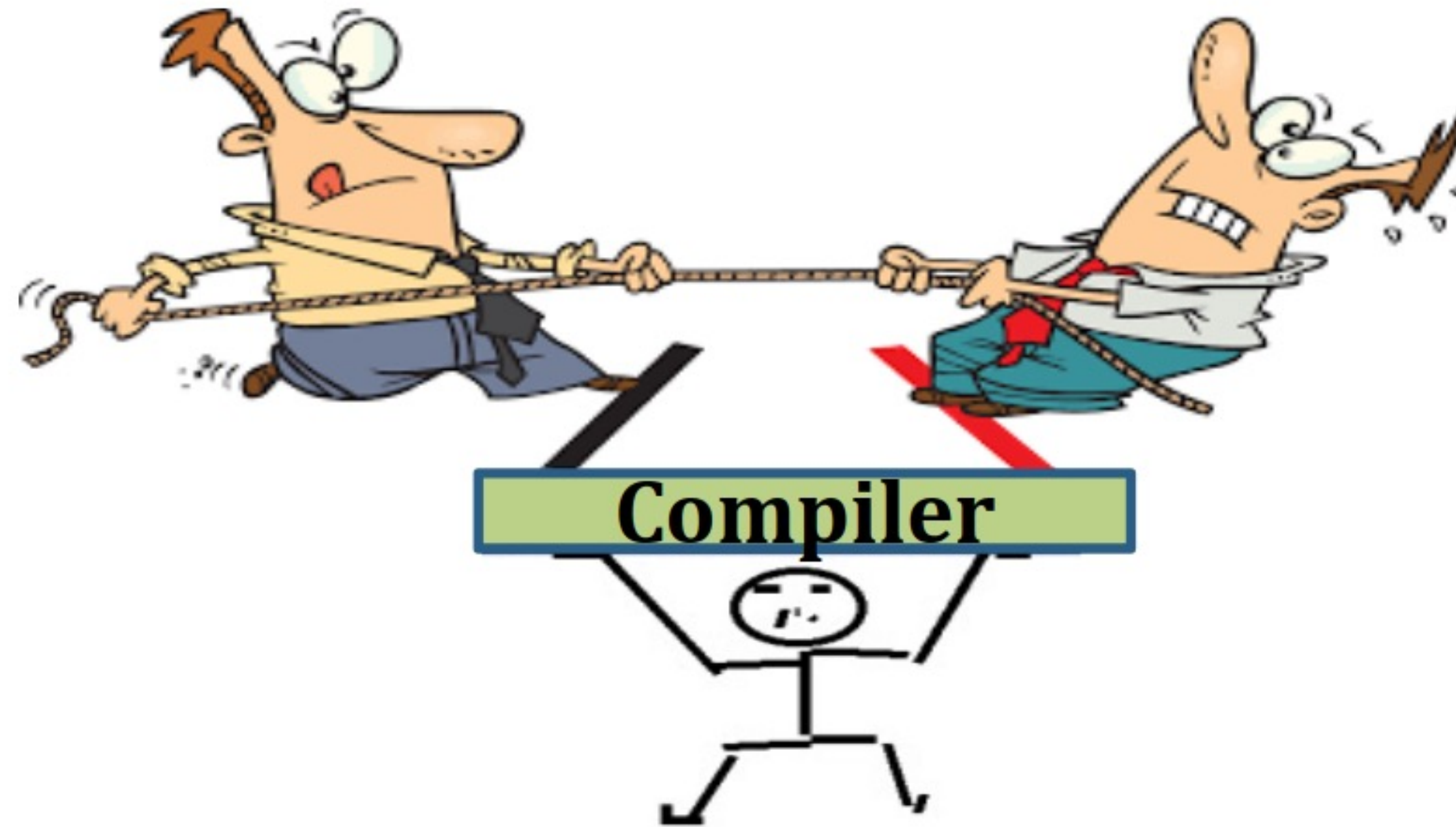# Architects vs. PL/Compiler Folks

Compilers

Architects

Compiler Problem

Advanced programming languages

Increasingly Complex ISA

Compiler

Symptoms of the Problem
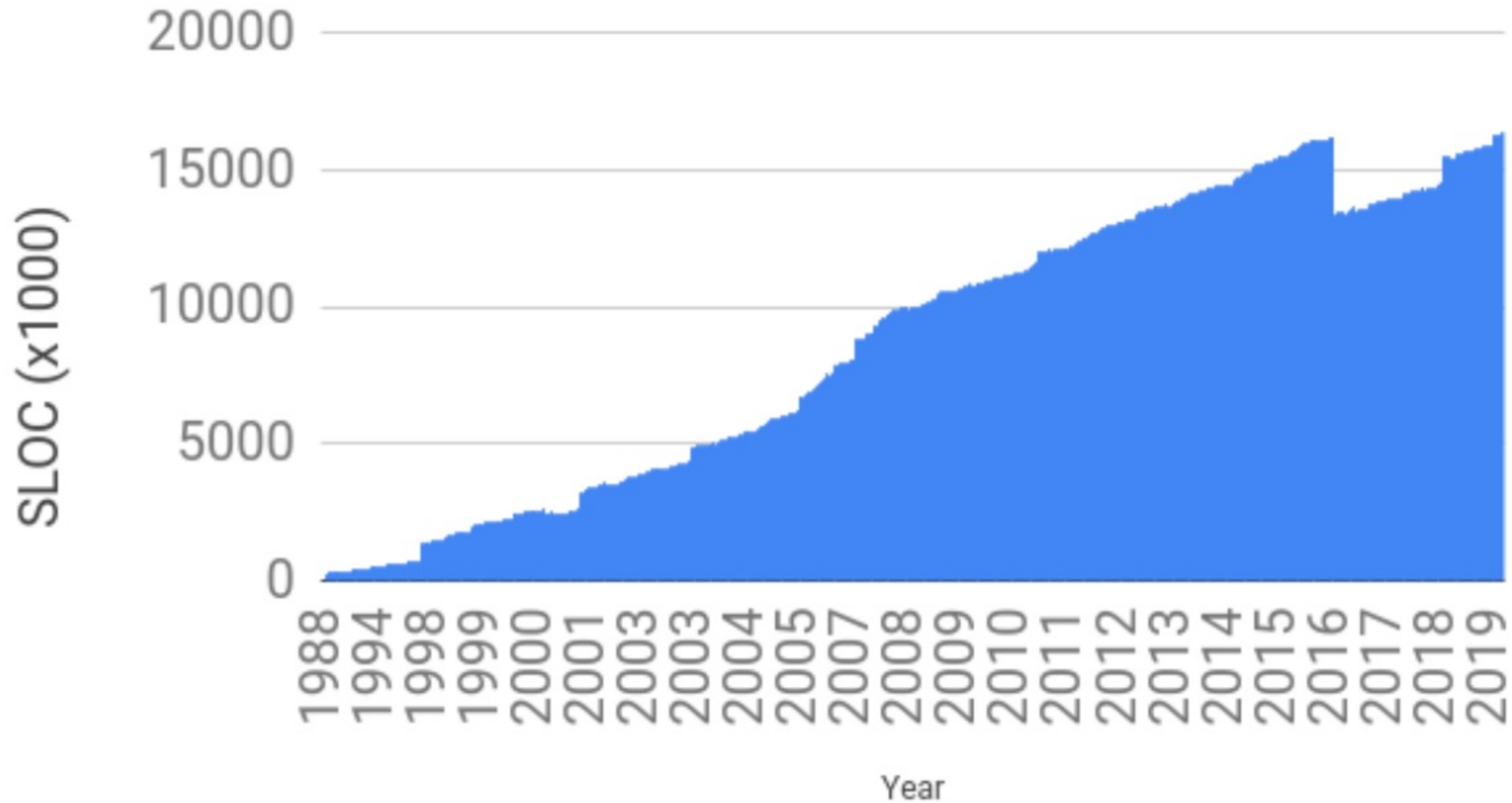
GCC

Symptoms of the Problem
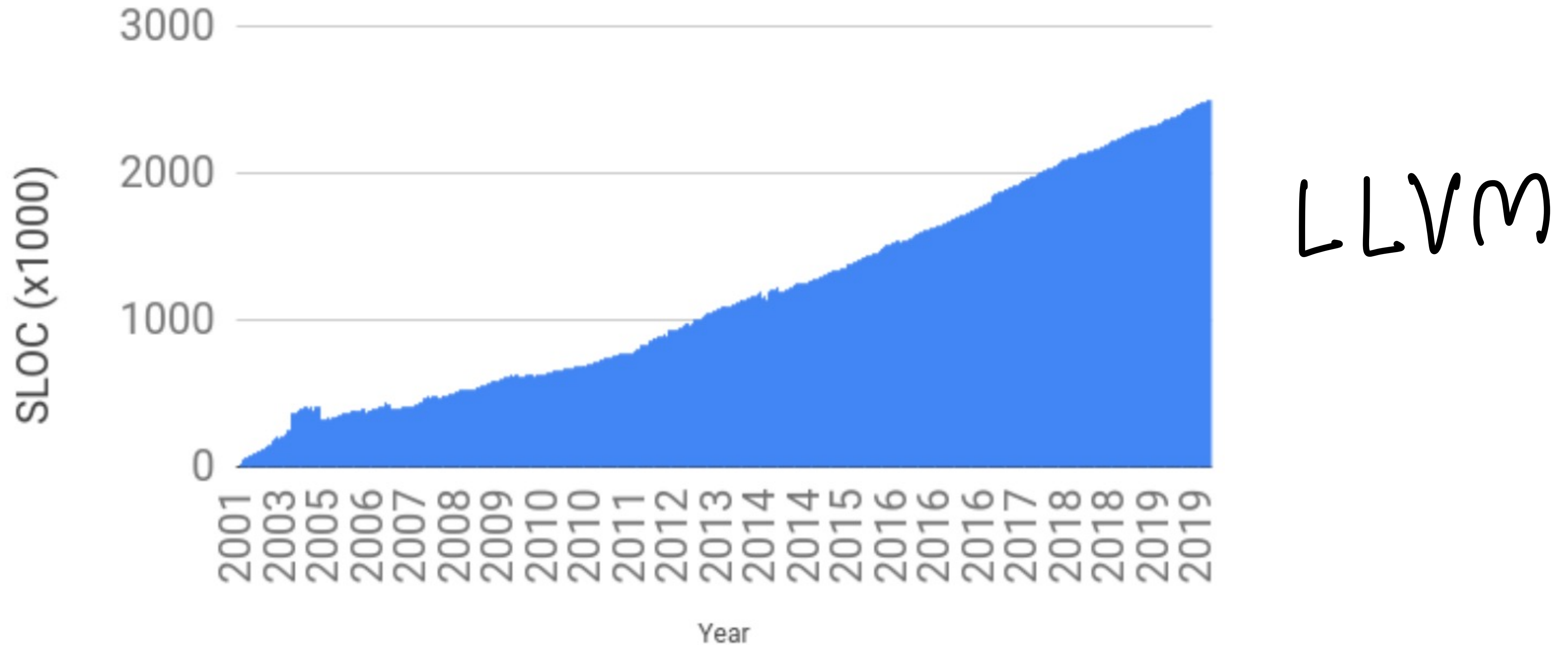
LLVM

Organization of a Compiler
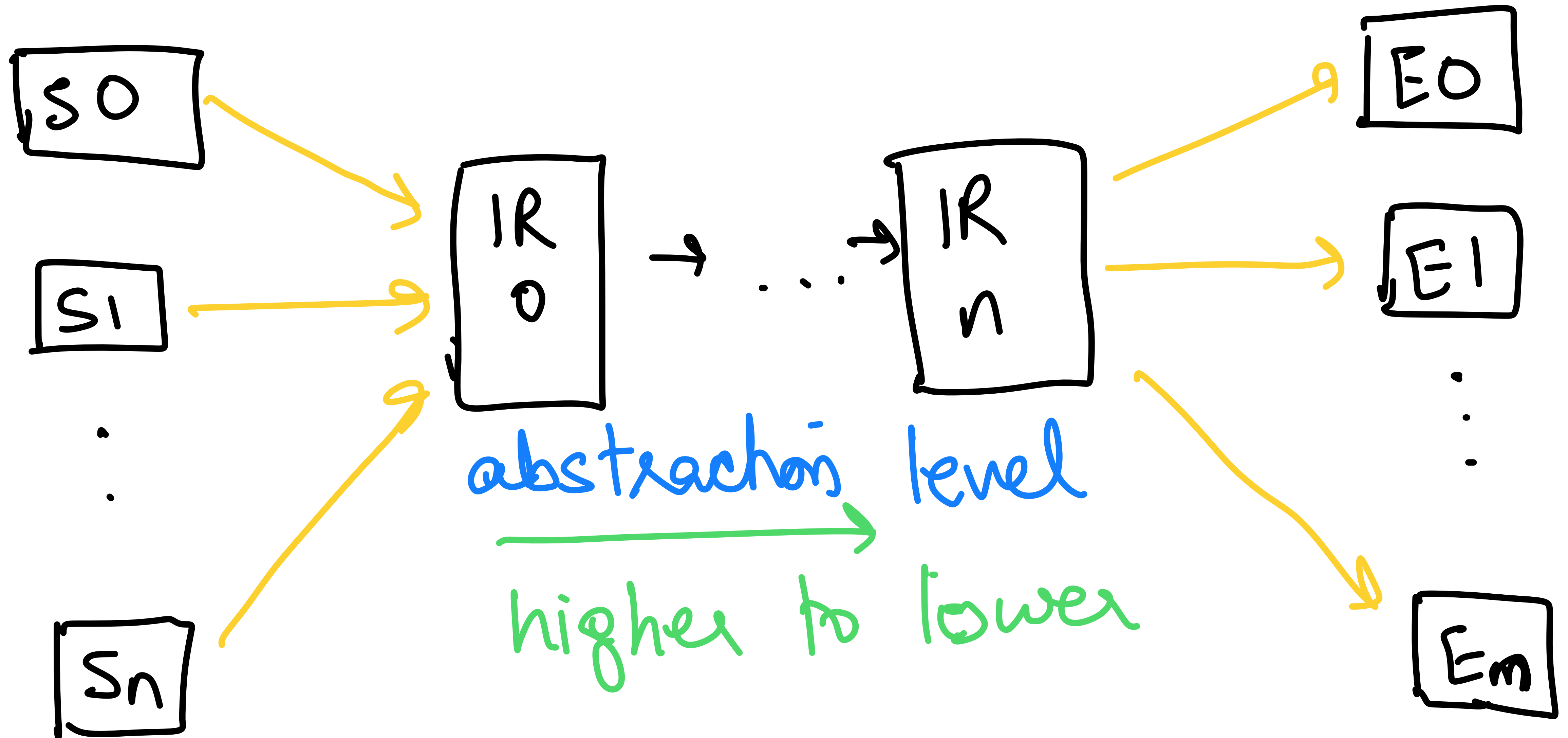
Some Common Abstraction Levels

Machine
ISA

x86

ARM

CUDA

. ..

Some Common Abstraction Levels

IRn

Machine
ISA

LLVM

x86

ARM

GIMPLE

CUDA

. . .

. . .

Some Common Abstraction Levels

IR_n

IR_{n-1}

Machine
ISA

x86
ARM
CUDA
. . .

LLVM

GIMPLE

. . .

MLIR

TVM

custom

Some Common Abstraction Levels

IRn

IRn-1

Source Lang

Machine ISA

x86
ARM
CUDA
...

LLVM
GIMPLE
...

MLIR
TVM
Custom

C/C++/Java/Python
TensorFlow
MapReduce
...

This Course : Polyhedral Analyses

Machine
ISA

x86
ARM
CUDA
. . .

IR_n

LLVM

GIMPLE

. . .

IR_{n-1}

MLIR

TVM

custom

array programs

Source
Lang

C/C++/Java/
Python

Tensorflow
MapReduce
. . .

This Course : Abstraction Design Subtleties

Machine ISA
x 86
ARM
CUDA
...

IRn
LLVM
GIMPLE
...

IRn-1
MLIR
TVM
Custom

Source Lang
C/C++/Java/Python
Tensorflow
MapReduce

non-determinism due to undefined/unspecified behaviour

This Course : Symbolic Analysis

IR_n IR_{n-1}

Machine
ISA

x86

ARM

CUDA

...

LLVM

GIMPLE

...

MLIR

TVM

Custom

Source
Lang

C/C++/Java/
Python

Tensorflow

MapReduce

...

safety
checkers

This Course : Compiler Validation
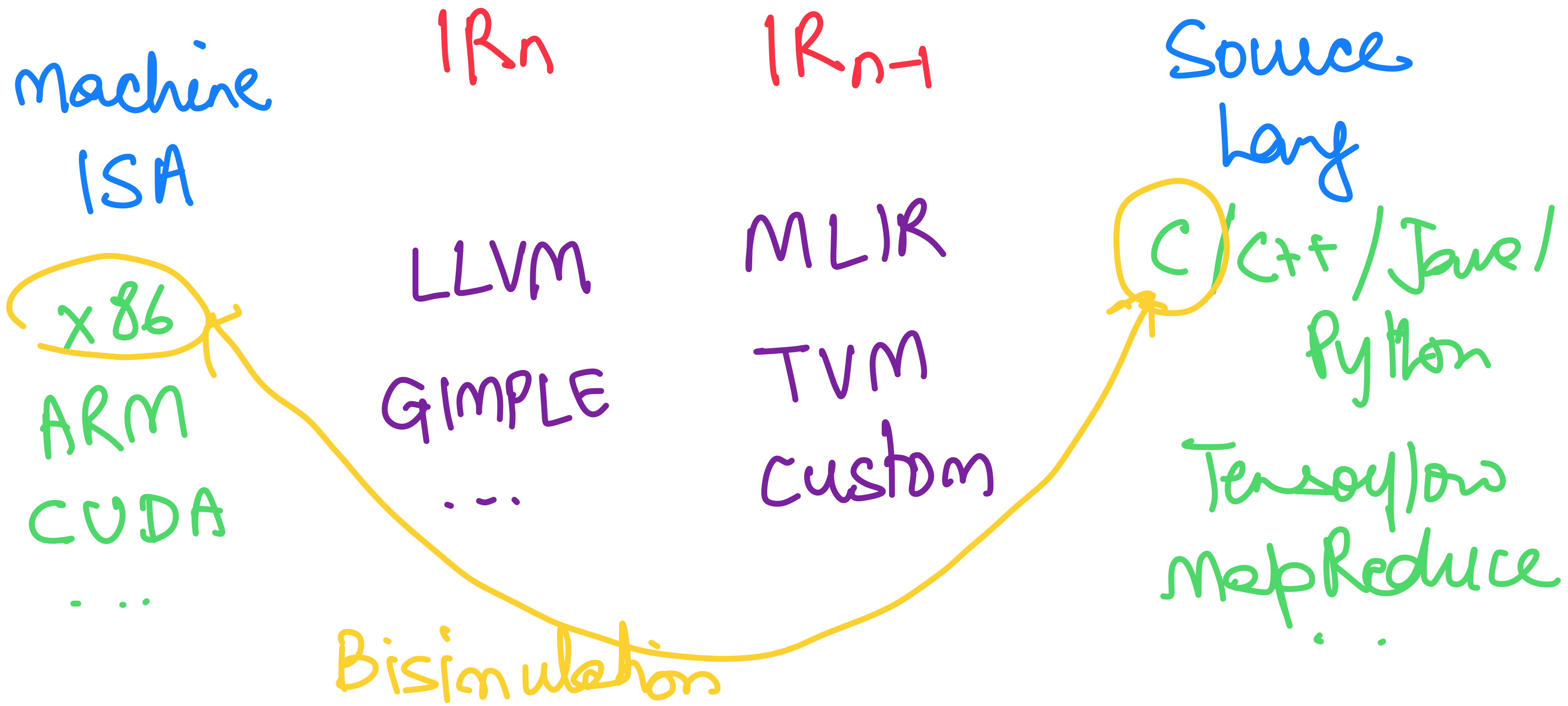
This Course : Push Button Verification

Machine ISA

x 86
ARM
CUDA
...

IRn

LLVM
GIMPLE
...

IRn-1

MLIR
TVM
Custom

Source Lang

C / C++ / Java / Python
Tensorflow
MapReduce

High-level functional language

equivalence