

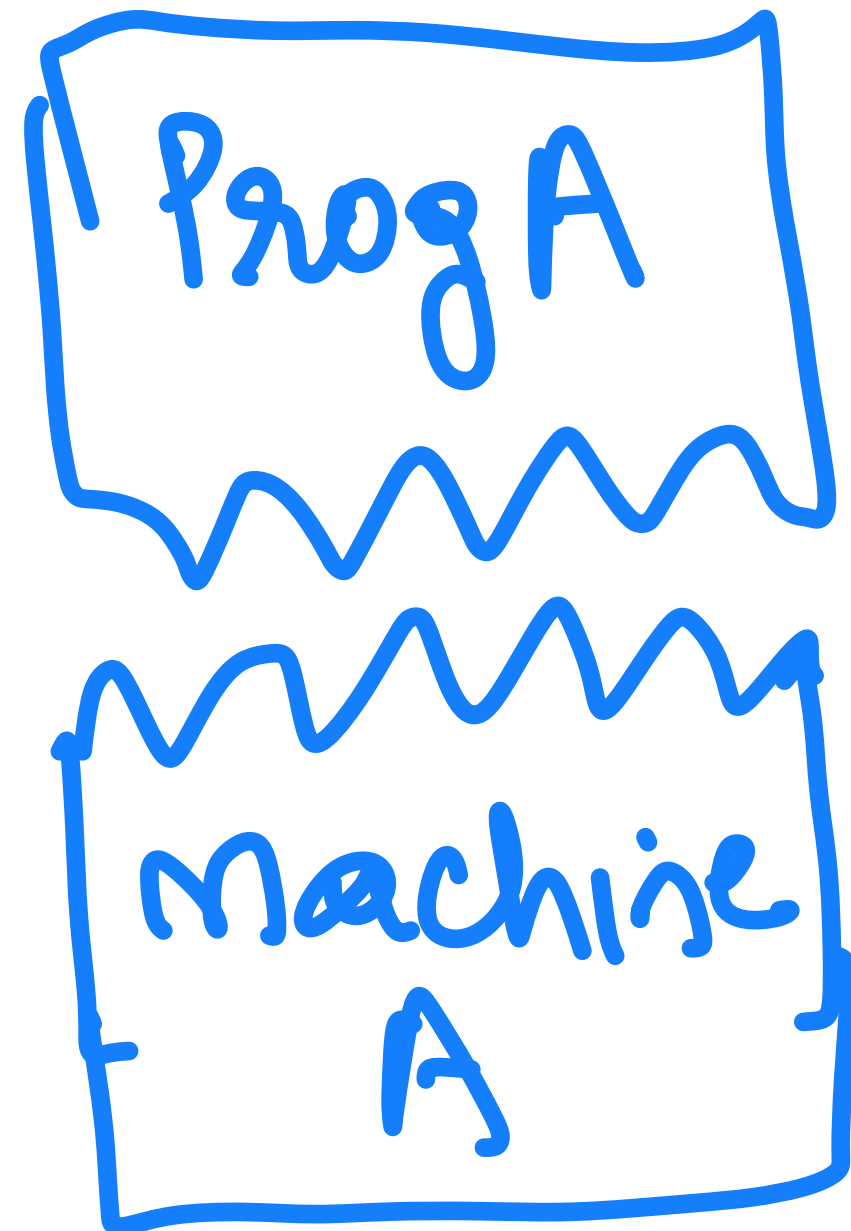
# Advanced Compiler Techniques

First Meeting

# What is a compiler?

1. A compiler is a translator

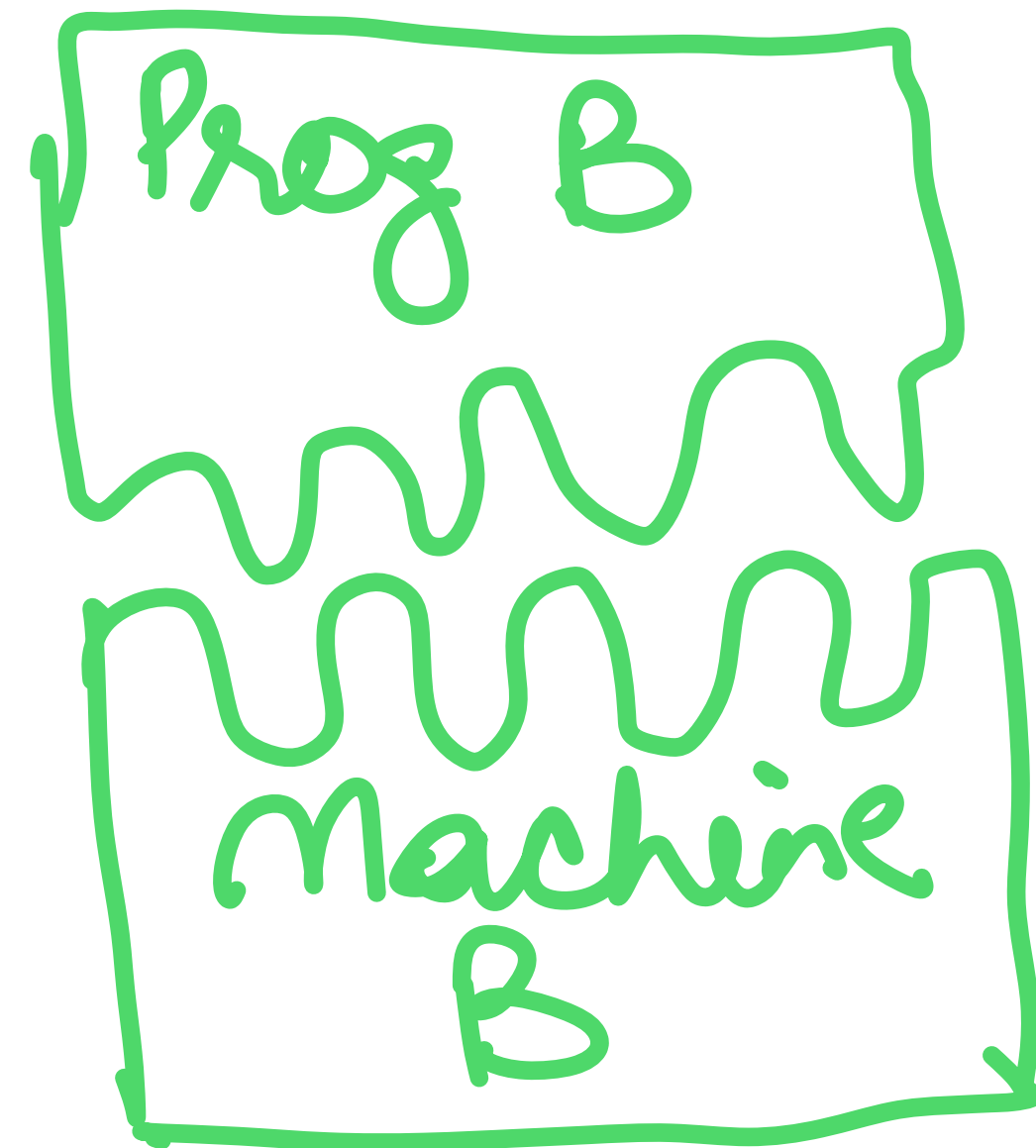
Find Prog B  
such that



Prog A || Machine A

≡

Prog B || Machine B

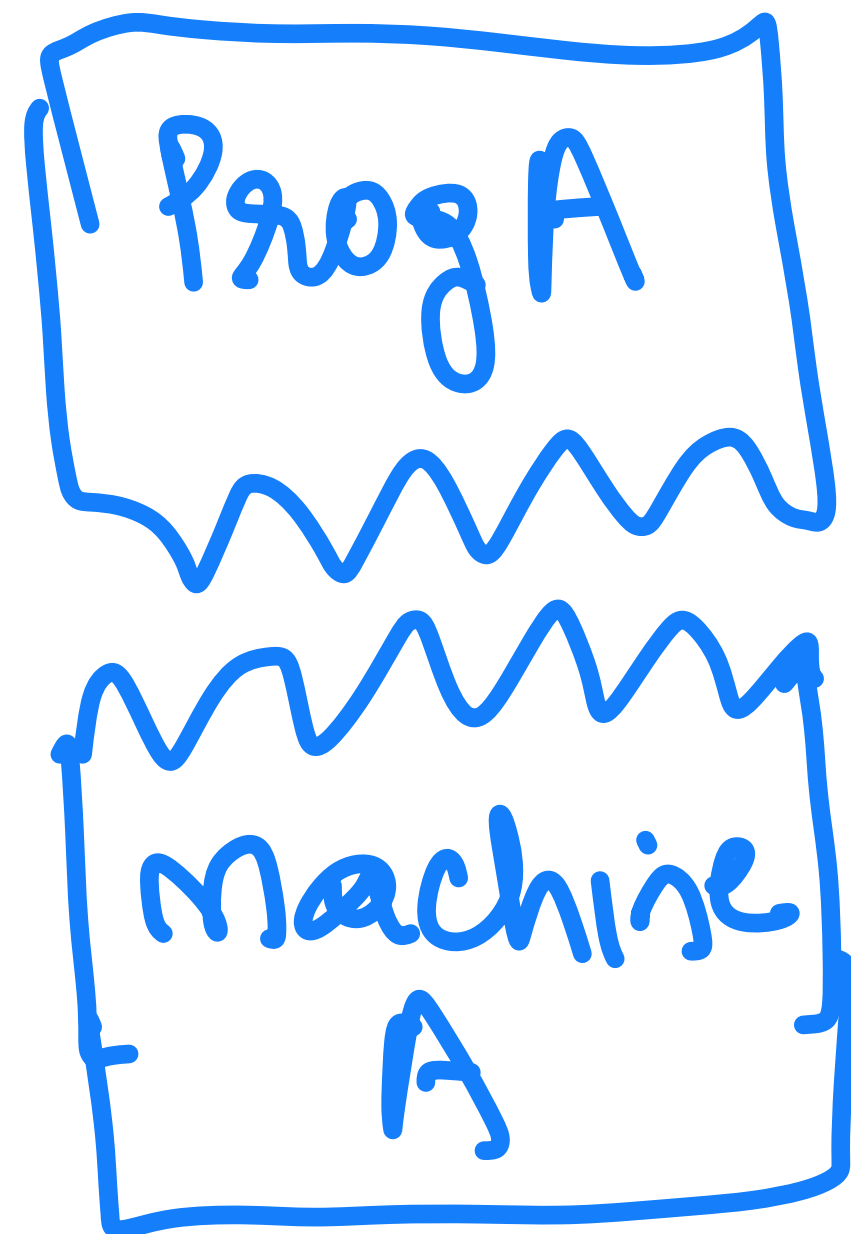


# What is a compiler?

1. A compiler is a translator

Examples?

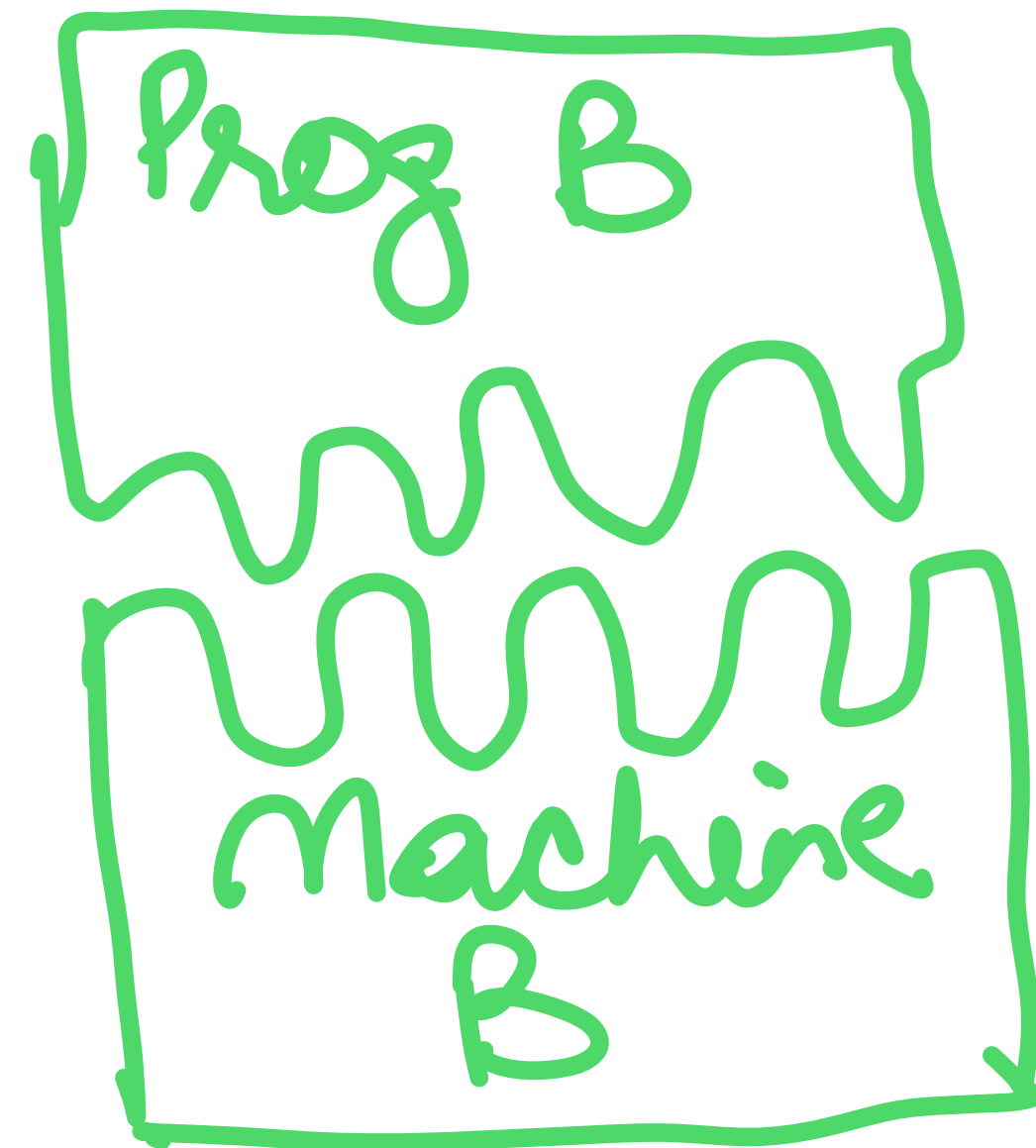
Find Prog B  
such that



Prog A || Machine A

≡

Prog B || Machine B

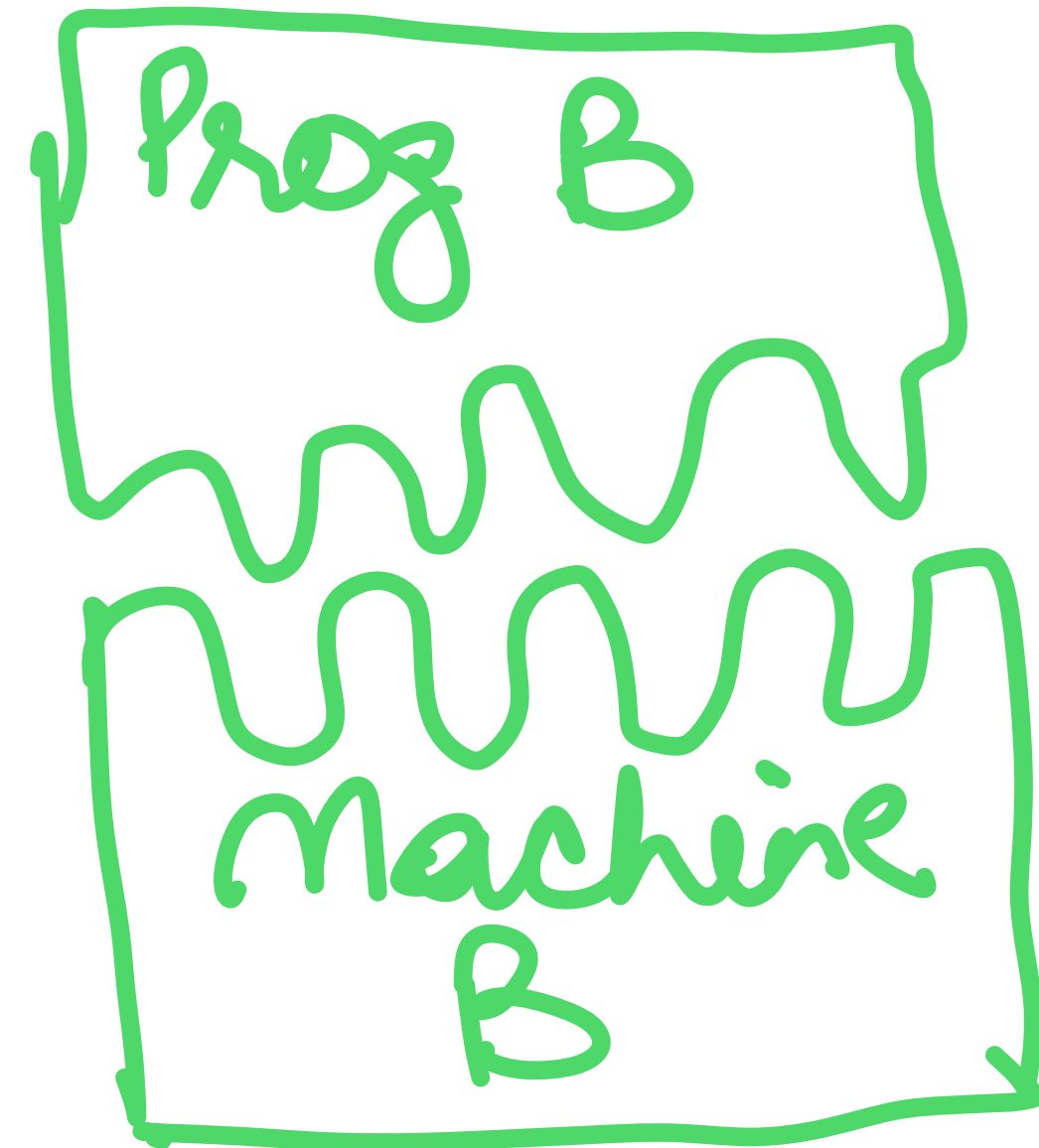
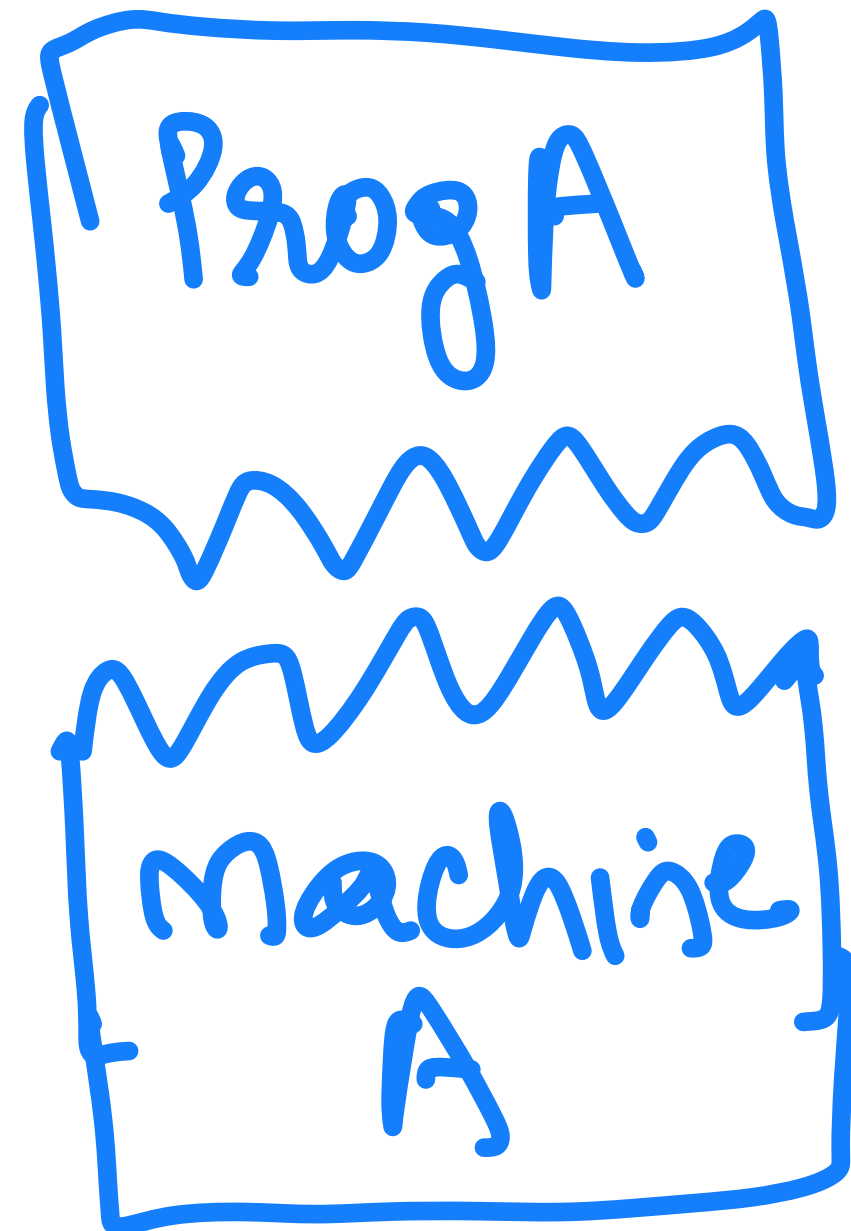


# What is a compiler?

1. A compiler is a translator

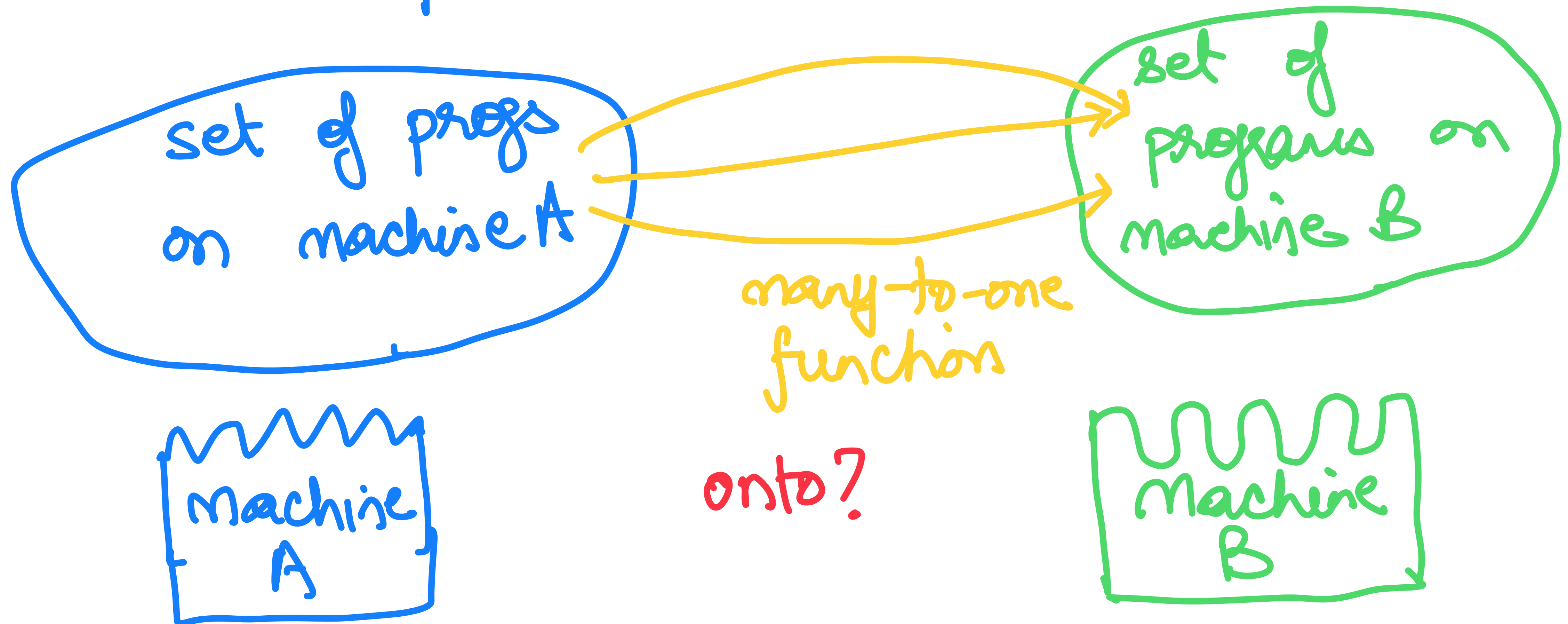
Always possible  
to find Prog B?

Counterexamples?



# What is a compiler?

1. A compiler is a translator



# What is a compiler?

2. A compiler is an optimizer

set of  
all progs

A large, hand-drawn green oval shape that encompasses most of the lower half of the page. It is drawn with a single, continuous green line.

# What is a compiler?

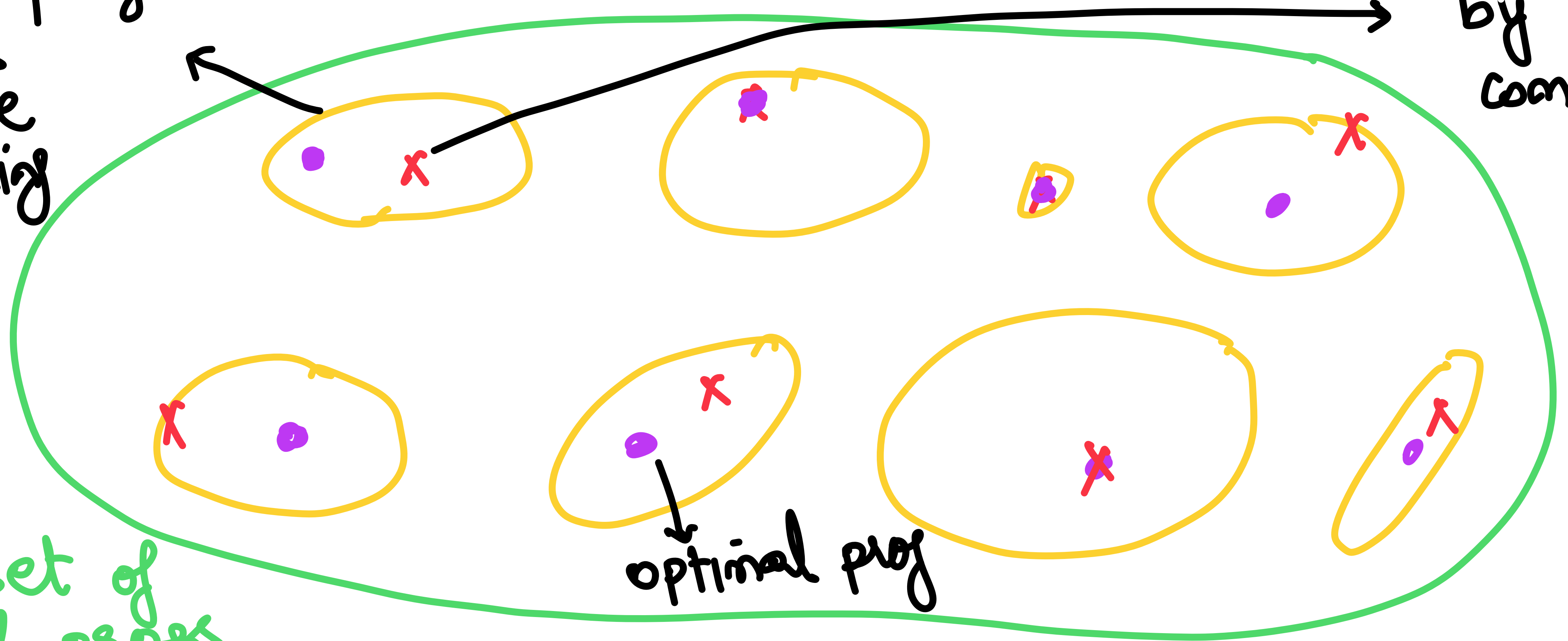
2. A compiler is an optimizer

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?

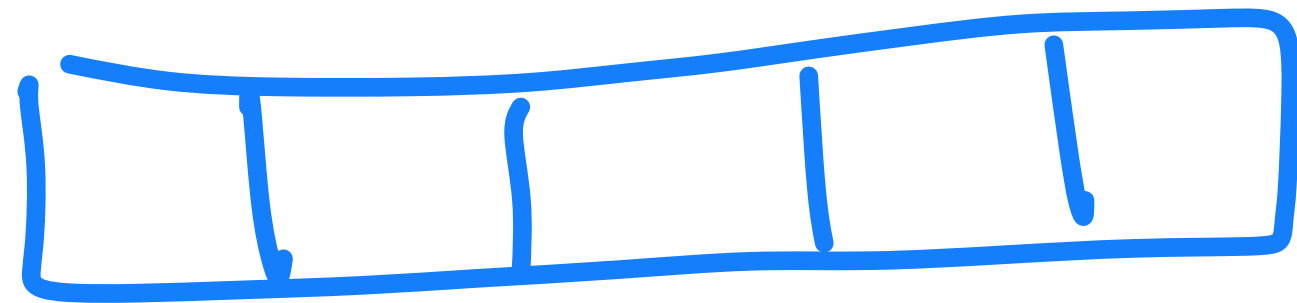
# So how have we been managing ?

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?



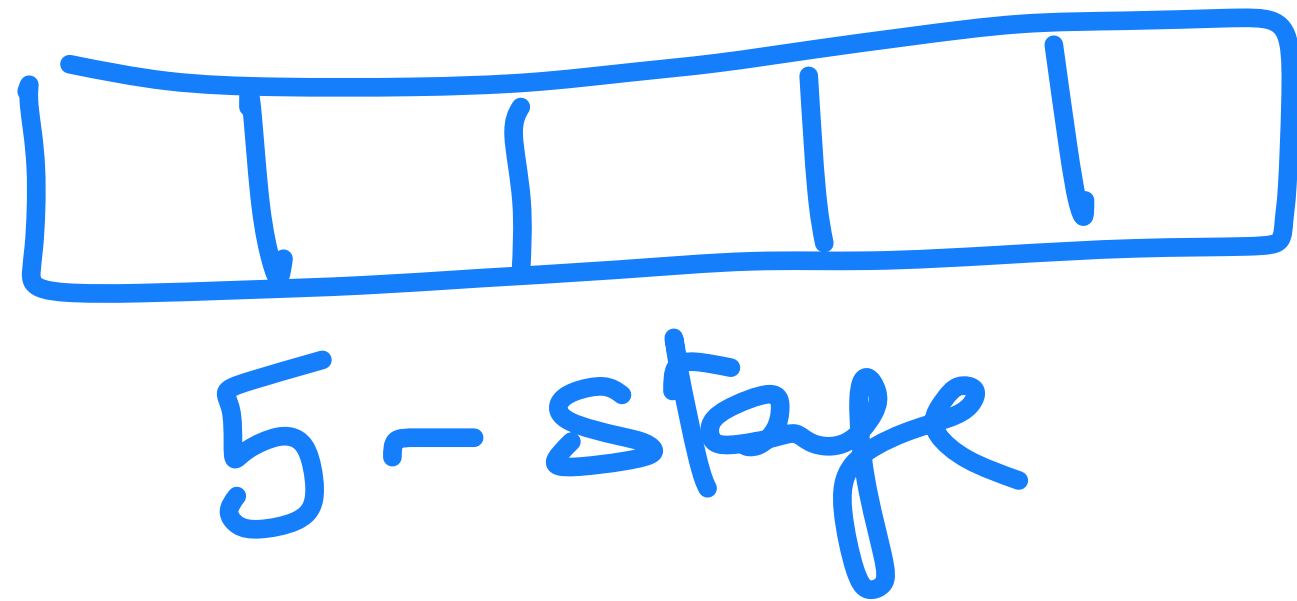
5-stage

↓ 80s-90s

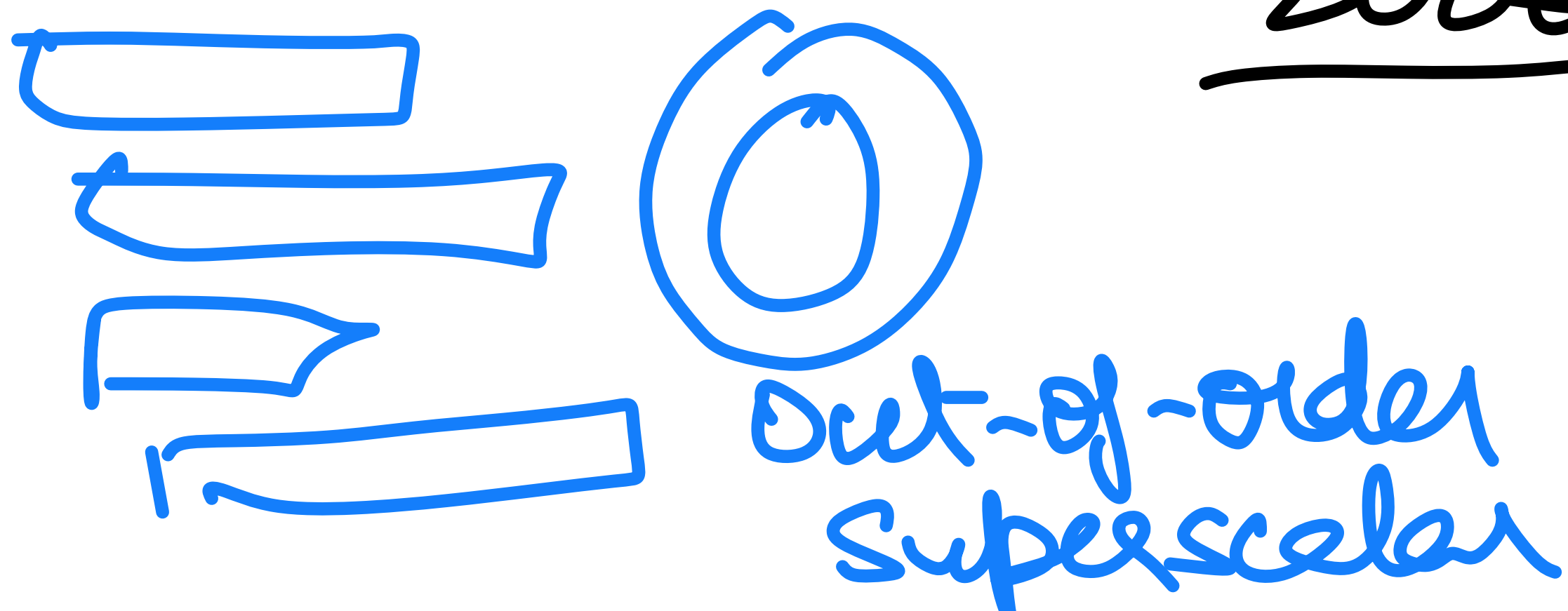


Out-of-Order  
Superscalar

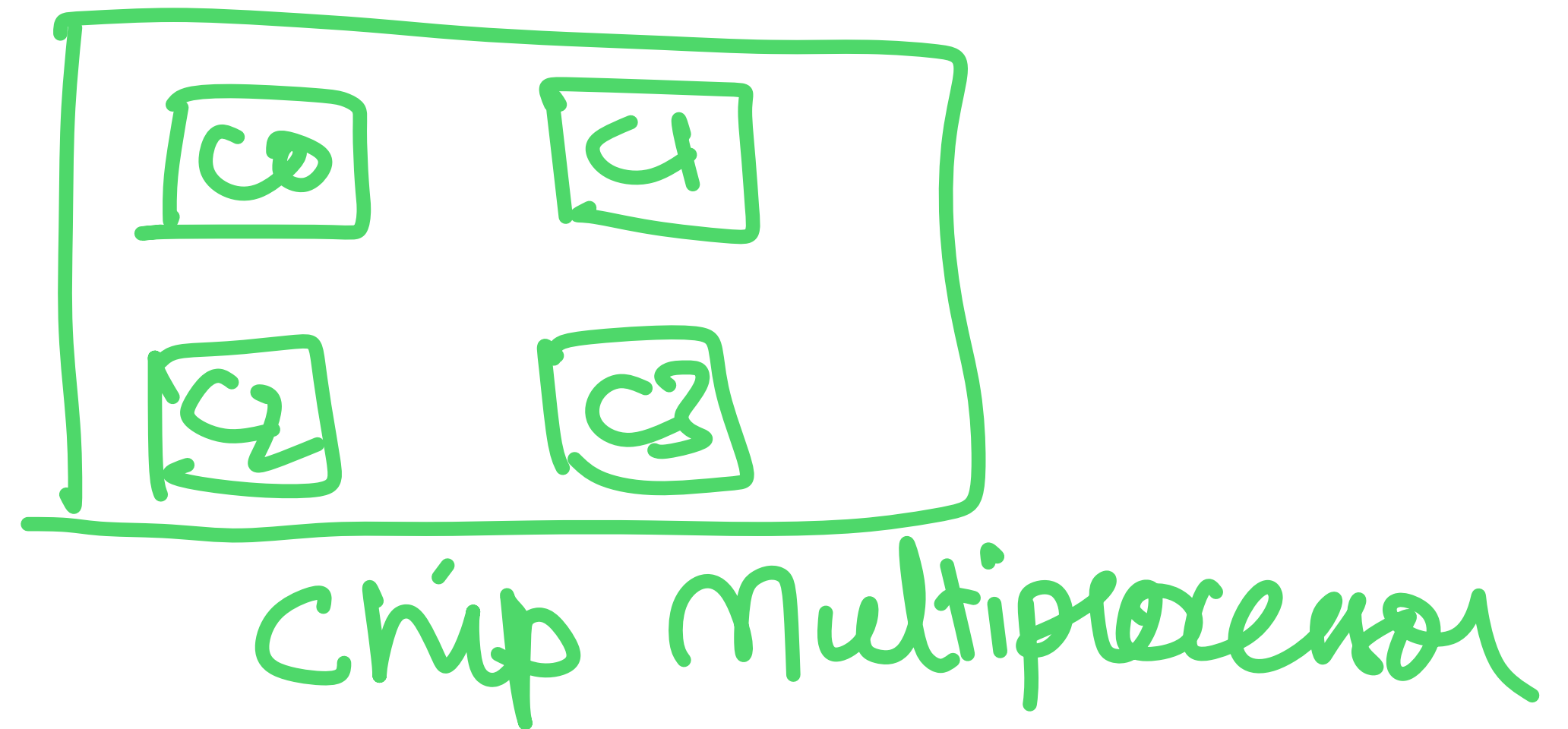
# Can we keep going like this?



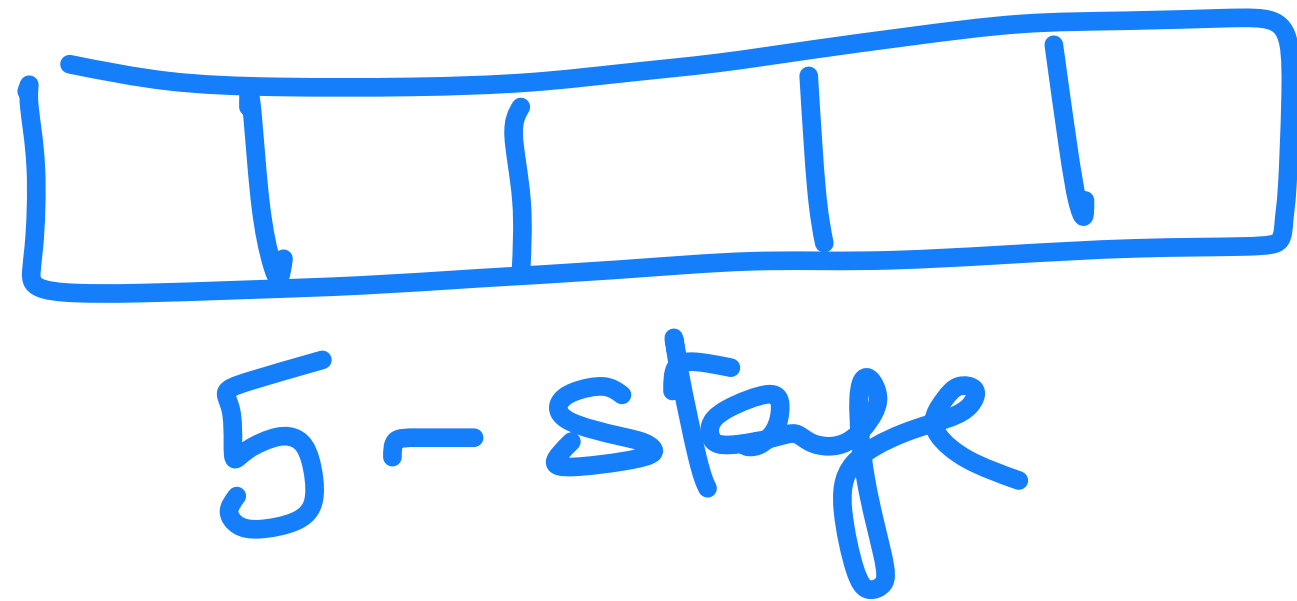
↓ 80s-90s



→ 2000s



# Can we keep going like this?



↓ 80s-90s



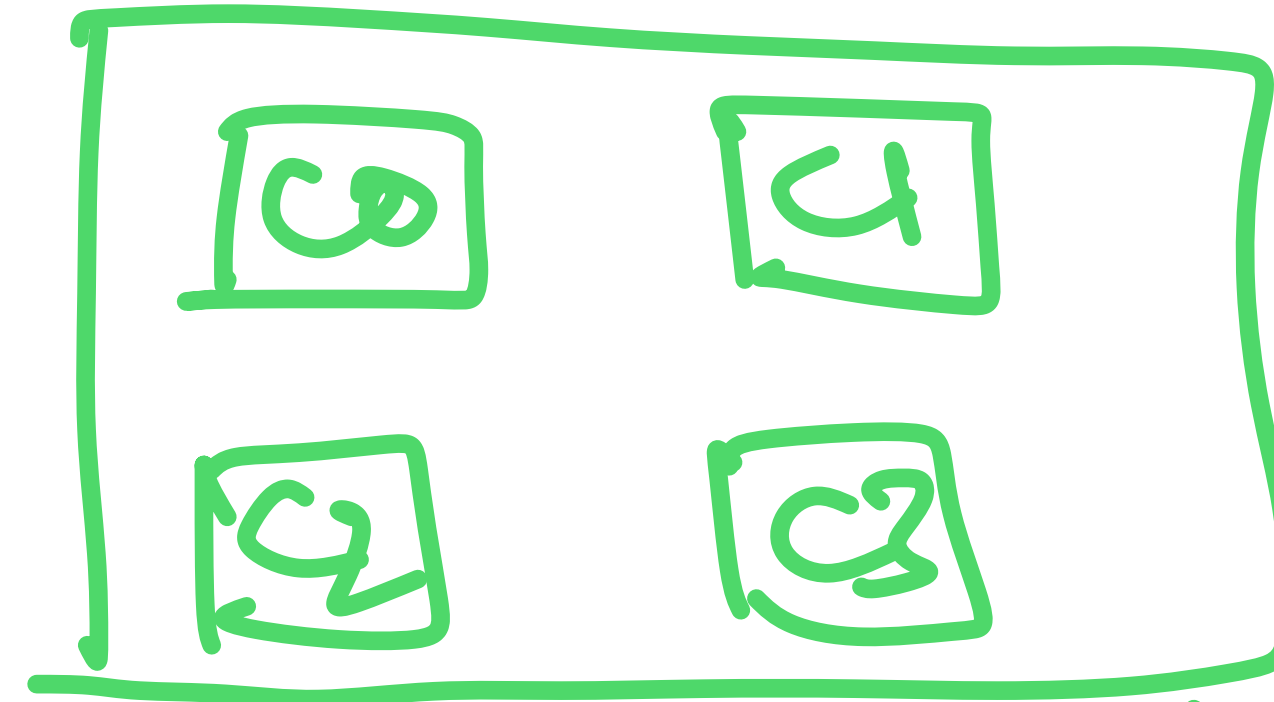
Out-of-order  
Superscalar



accelerator  
based  
designs

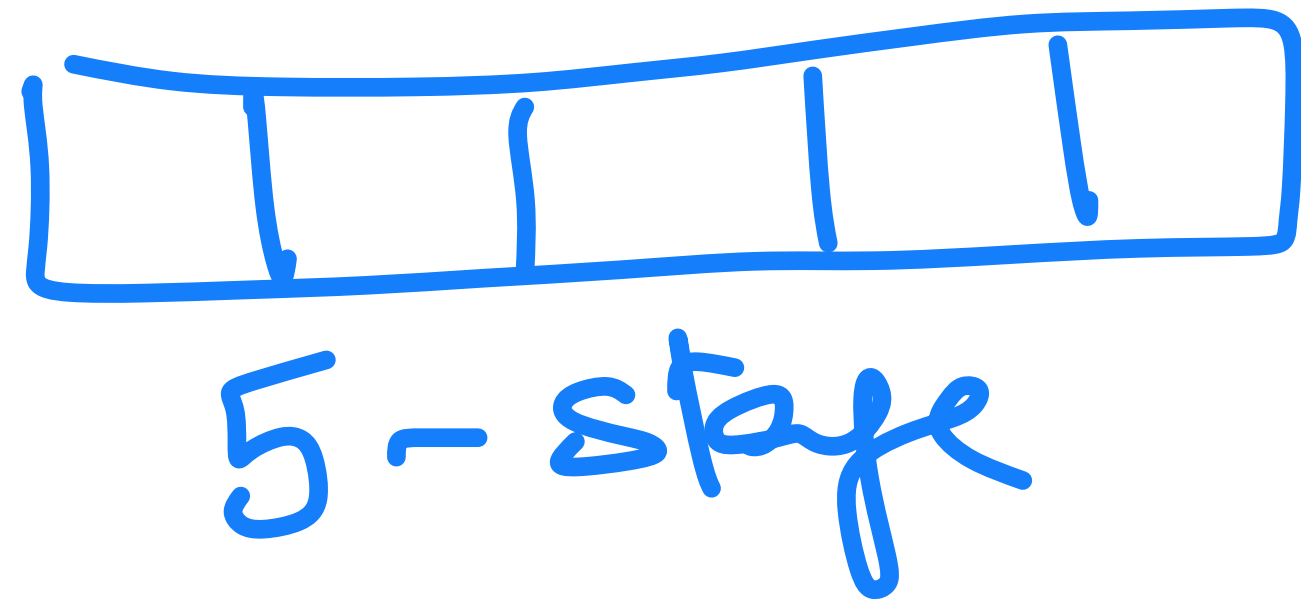
↖ 2010s

→ 2000s



Chip Multiprocessor

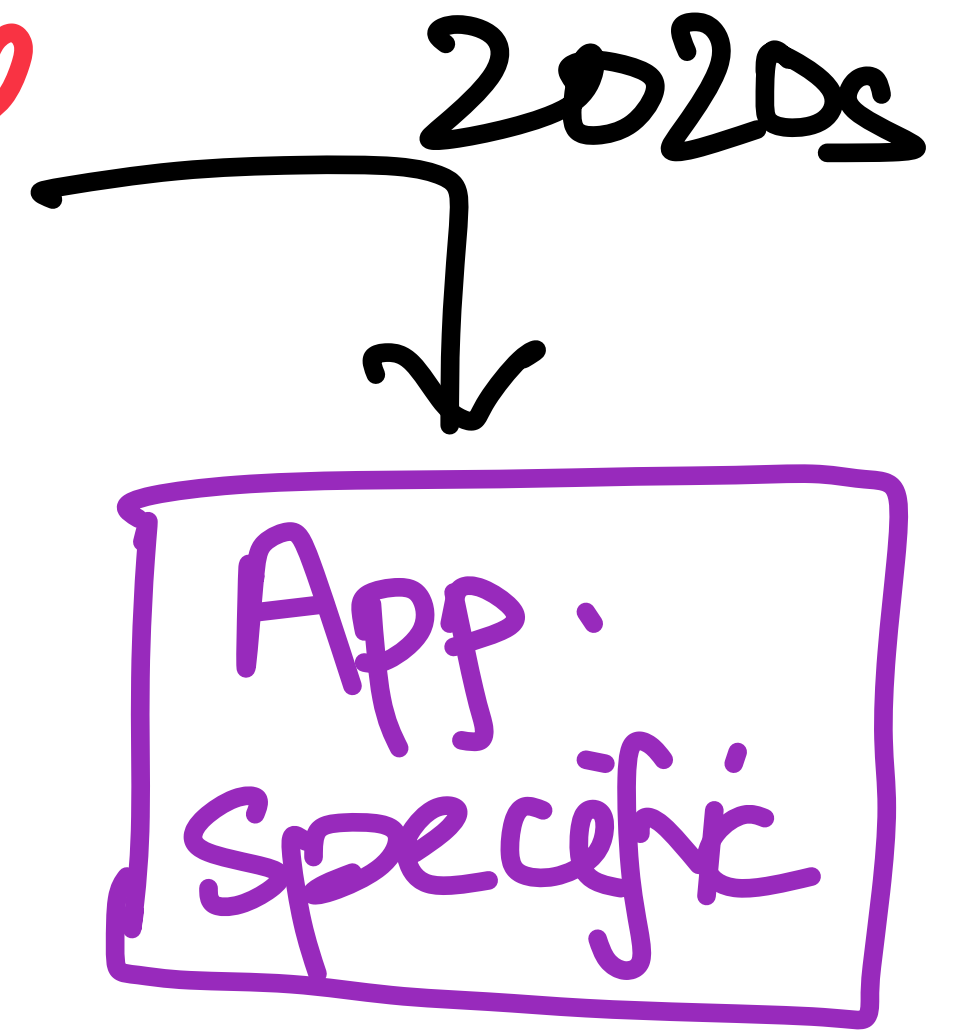
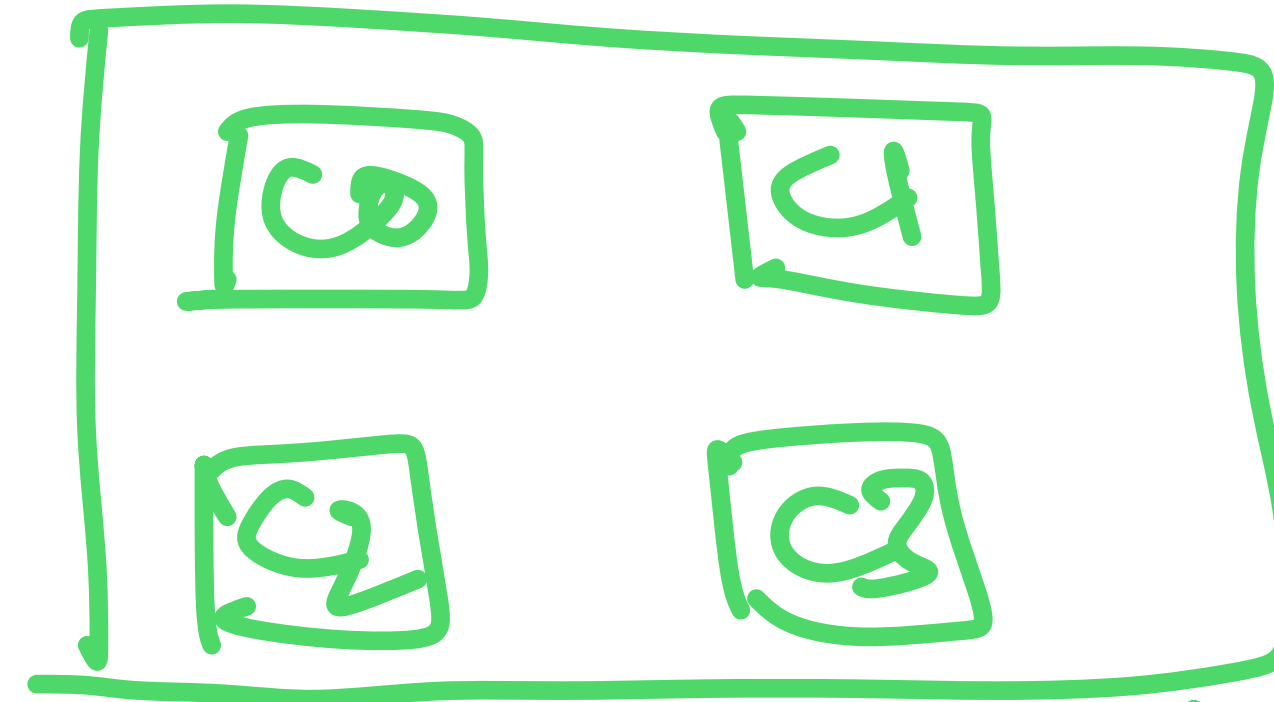
# Can we keep going like this?



80s-90s



2010s



# Programming Languages?

C/C++

Java

Python

OCaml

JavaScript

DSLs:

Tensorflow

Halide

Lustre

MapReduce

SQL

...



# Architects vs. PL/Compiler Folks



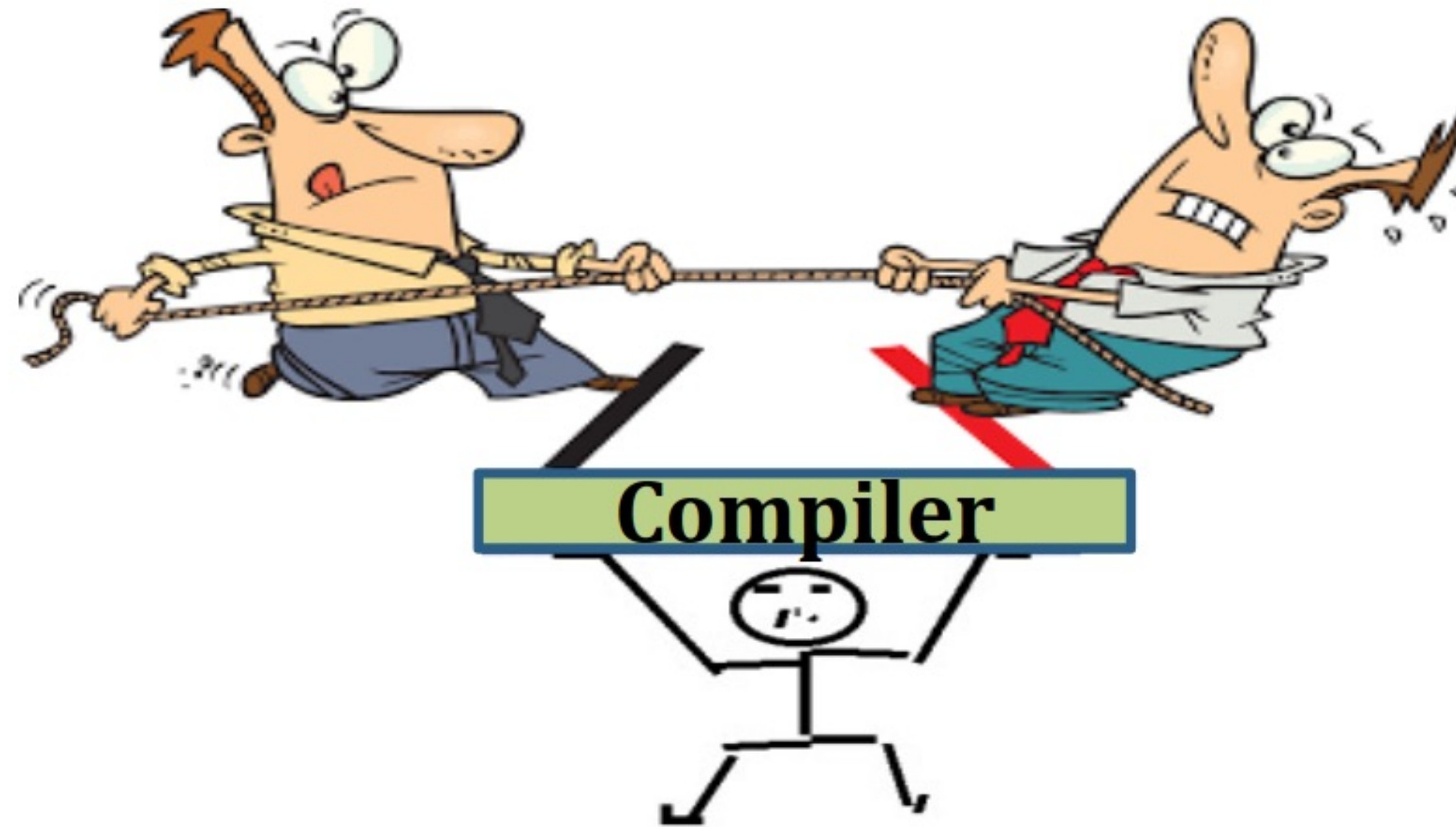
Compilers



Architects

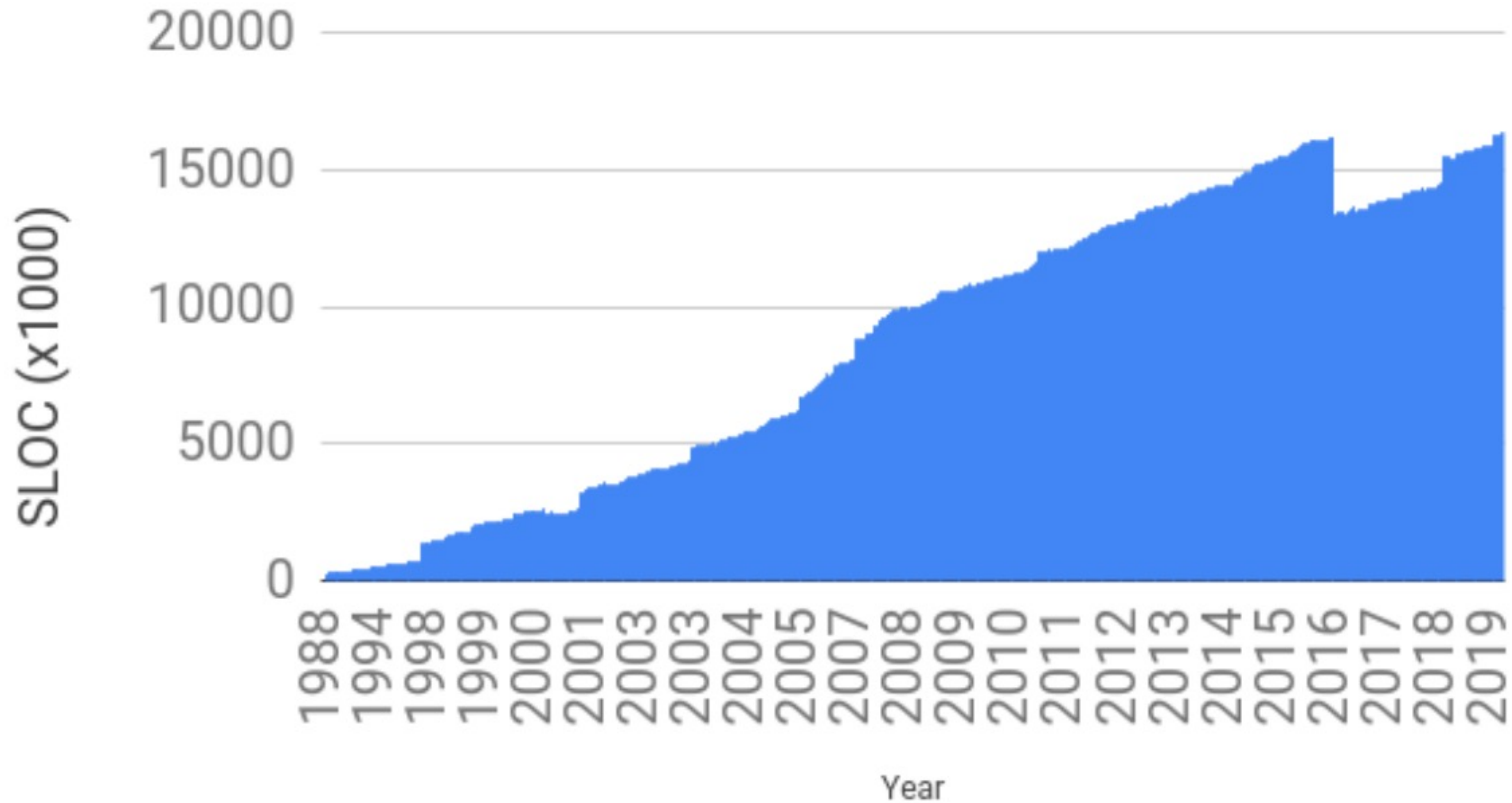
# Compiler Problem

**Advanced  
programming  
languages**



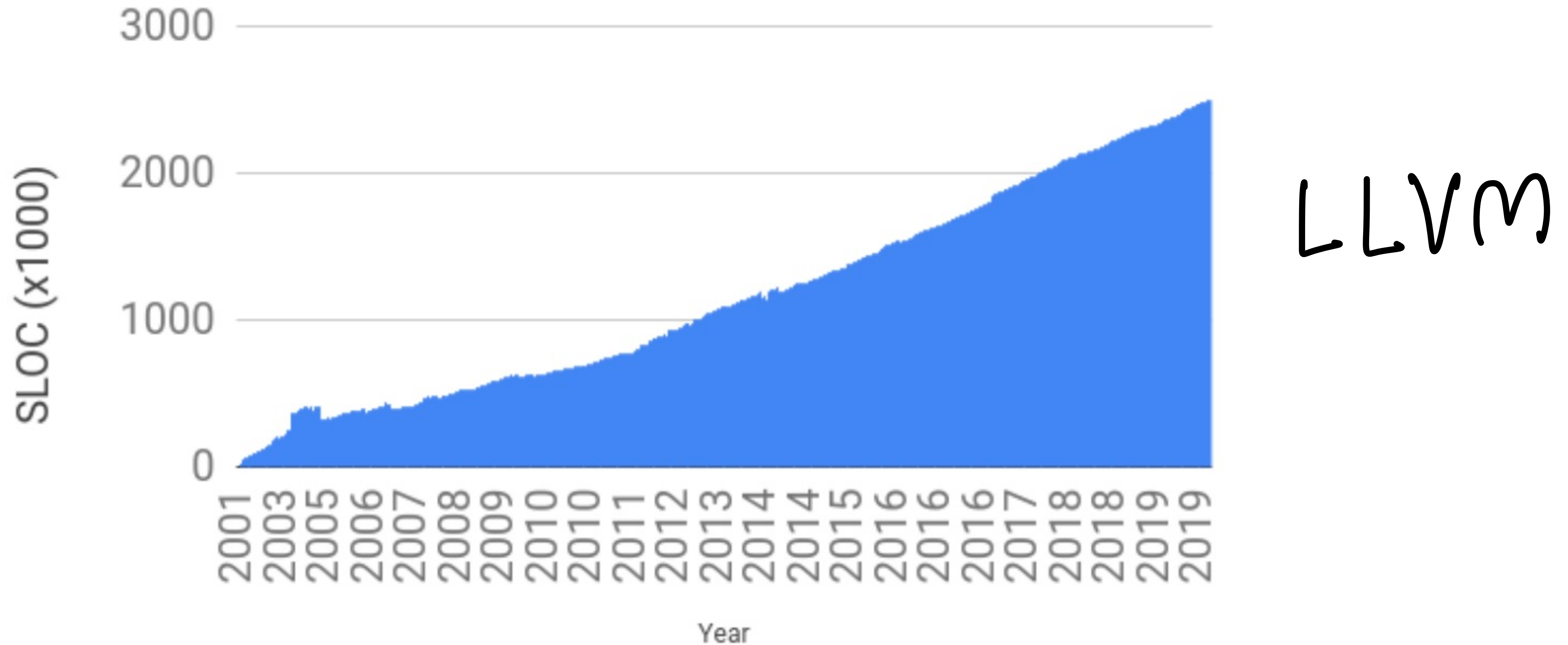
**Increasingly  
Complex ISA**

# Symptoms of the Problem

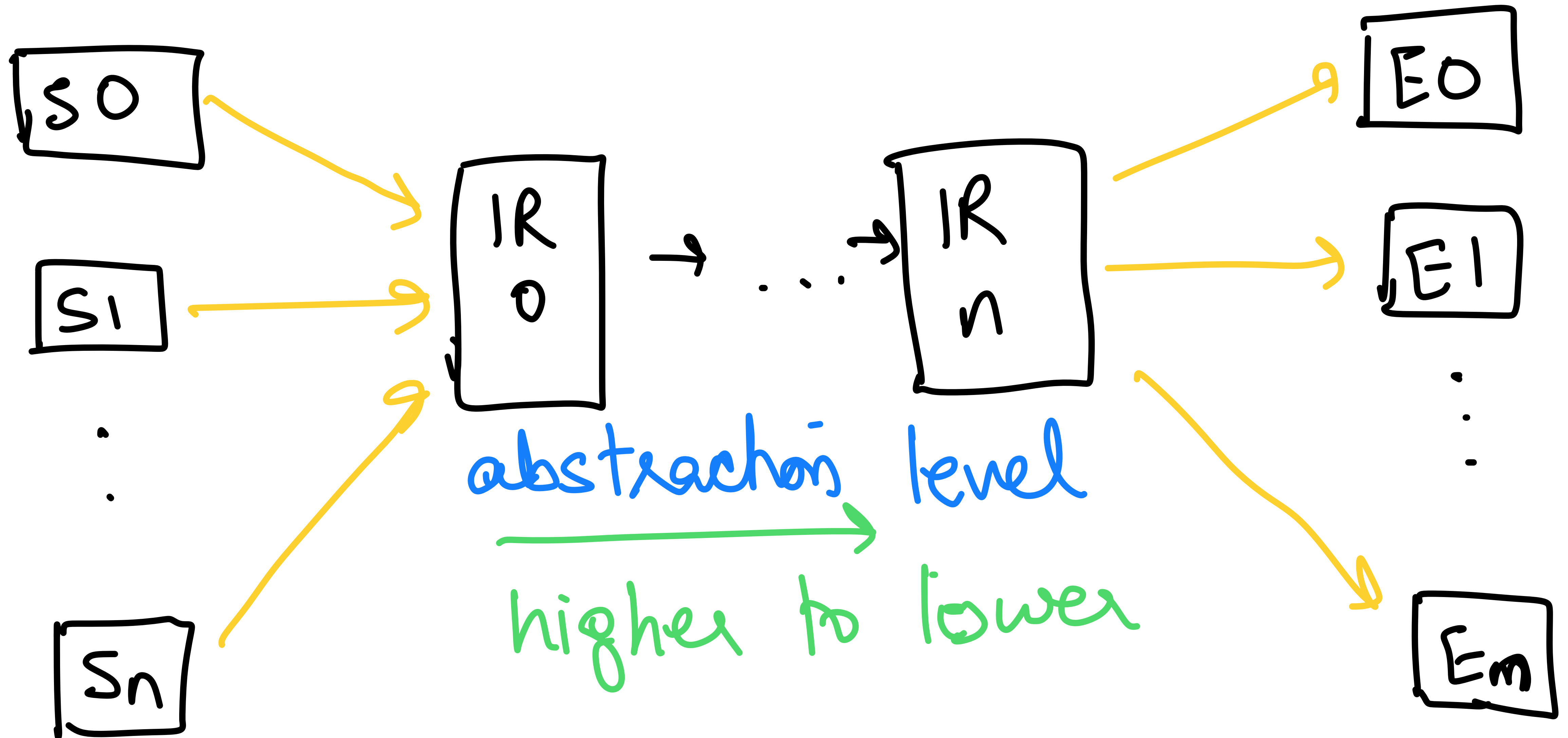


GCC

# Symptoms of the Problem



# Organization of a Compiler



# Some Common Abstraction Levels

machine

ISA

x86

ARM

CUDA

...

# Some Common Abstraction Levels

machine

ISA

x86

ARM

CUDA

...

IRn

LLVM

GIMPLE

...

# Some Common Abstraction Levels

machine

ISA

x86

ARM

CUDA

...

$IR_n$

LLVM

GIMPLE

...

$IR_{n-1}$

MLIR

TVM

custom



# Some Common Abstraction Levels

machine

ISA

x86

ARM

CUDA

...

IR<sub>n</sub>

LLVM

GIMPLE

...

IR<sub>n-1</sub>

MLIR

TVM

custom

Source

lang

C/C++/Java/

Python

TensorFlow

mapReduce

..

# This Course : Polyhedral Analyses

machine

ISA

x86

ARM

CUDA

...

IR<sub>n</sub>

LLVM

GIMPLE

...

IR<sub>n-1</sub>

MLIR

TVM

custom

array programs

Source

lang

C/C++/Java/

Python

Tensorflow

mapReduce

...

# This Course : Abstraction Design Subtleties

machine  
ISA  
x86  
ARM  
CUDA  
...

IR<sub>n</sub>

LLVM  
GIMPLE  
...

IR<sub>n-1</sub>

MLIR  
TVM  
custom

Source  
lang  
C/C++/Java/  
Python  
Tensorflow  
mapReduce  
behaviour

non-determinism  
due to undefined/unspecified

# This Course : Symbolic Analysis

machine

ISA

x86

ARM

CUDA

...

IR<sub>n</sub>

LLVM

GIMPLE

...

safety  
checkers

IR<sub>n-1</sub>

MLIR

TVM

custom

Source

lang

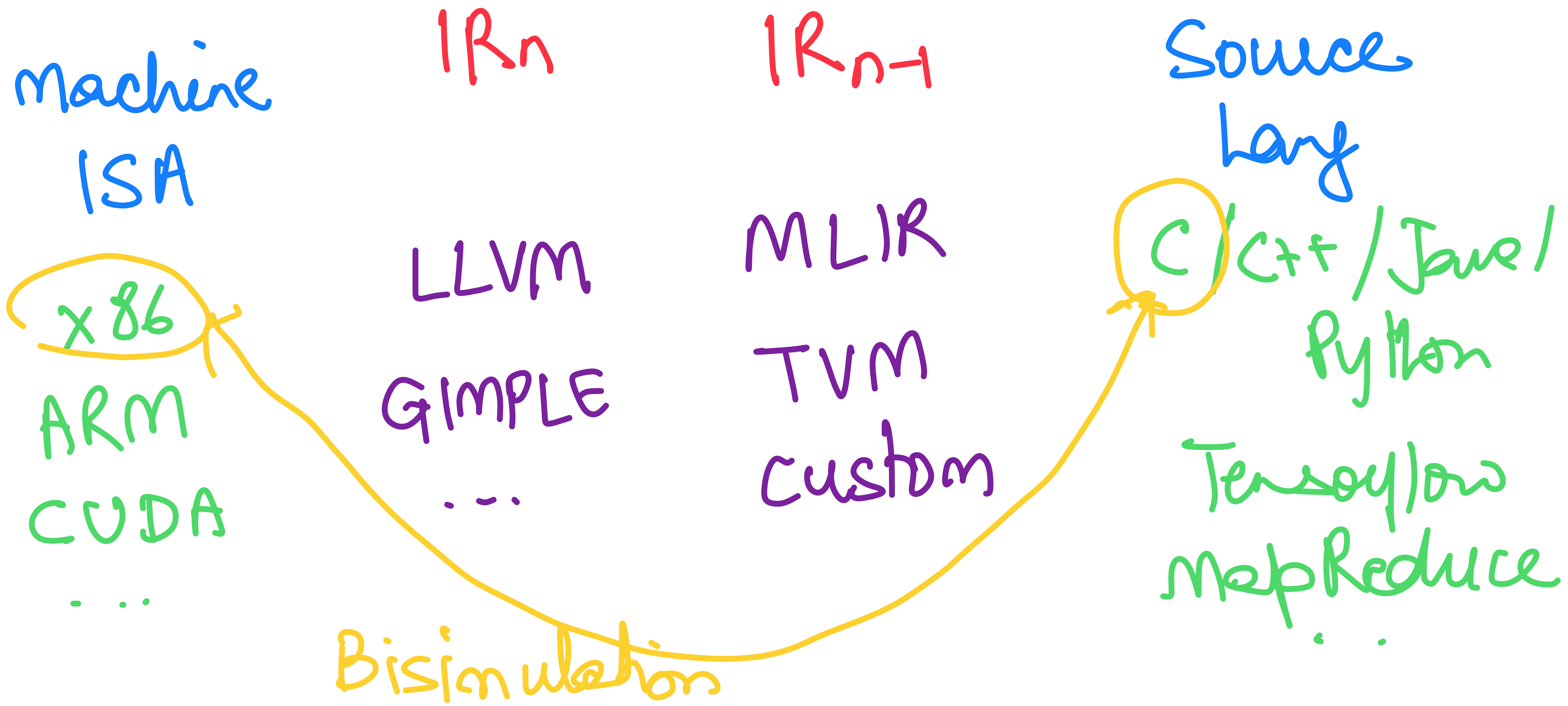
C/C++/Java/

Python

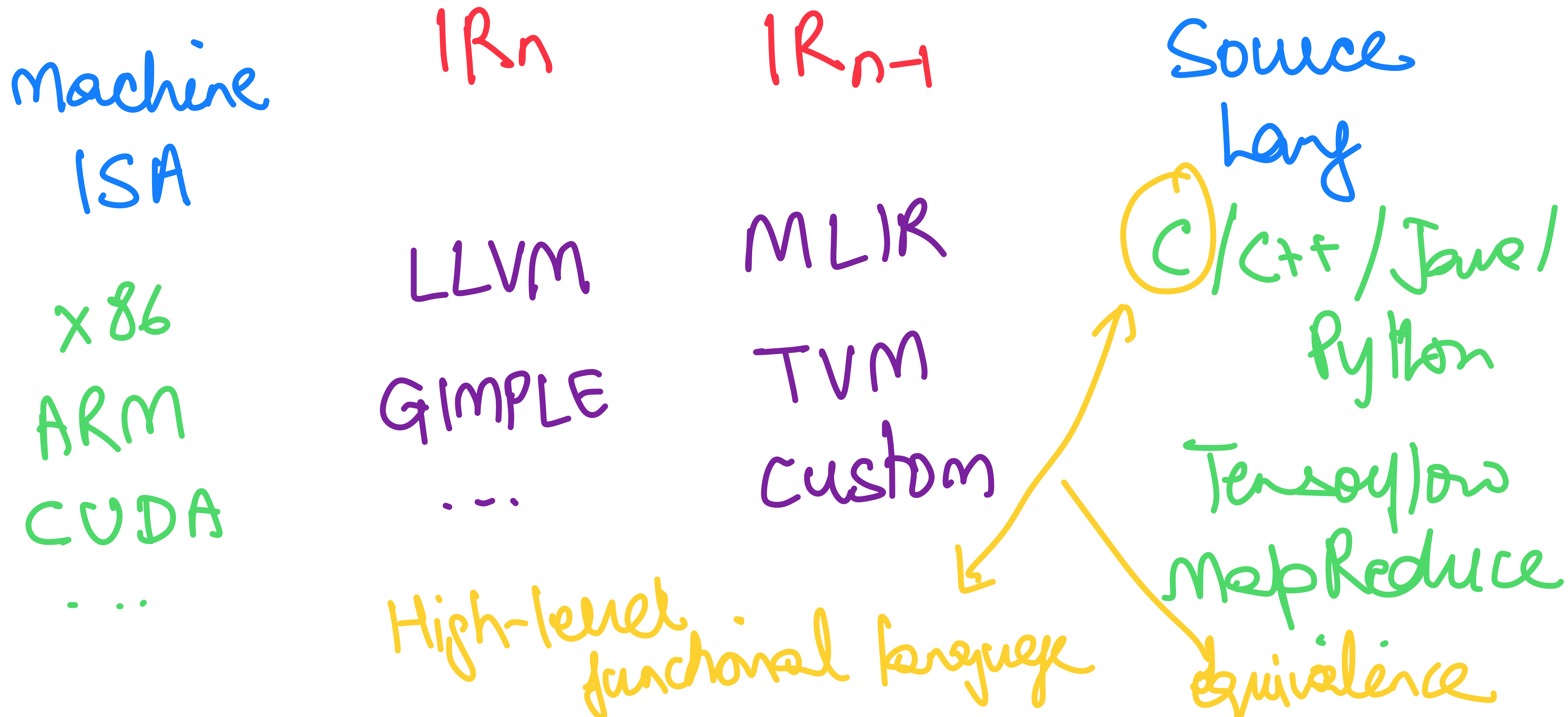
Tensorflow

mapReduce  
..

# This Course : Compiler Validation



# This Course : Push Button Verification



# Logistics

- One meeting every Thursday from 8-9.30am IST every Thursday
- Viewing material (YouTube videos) handed out every week
- Expected to view the material before the Thursday meeting
- Optional online exams for external students