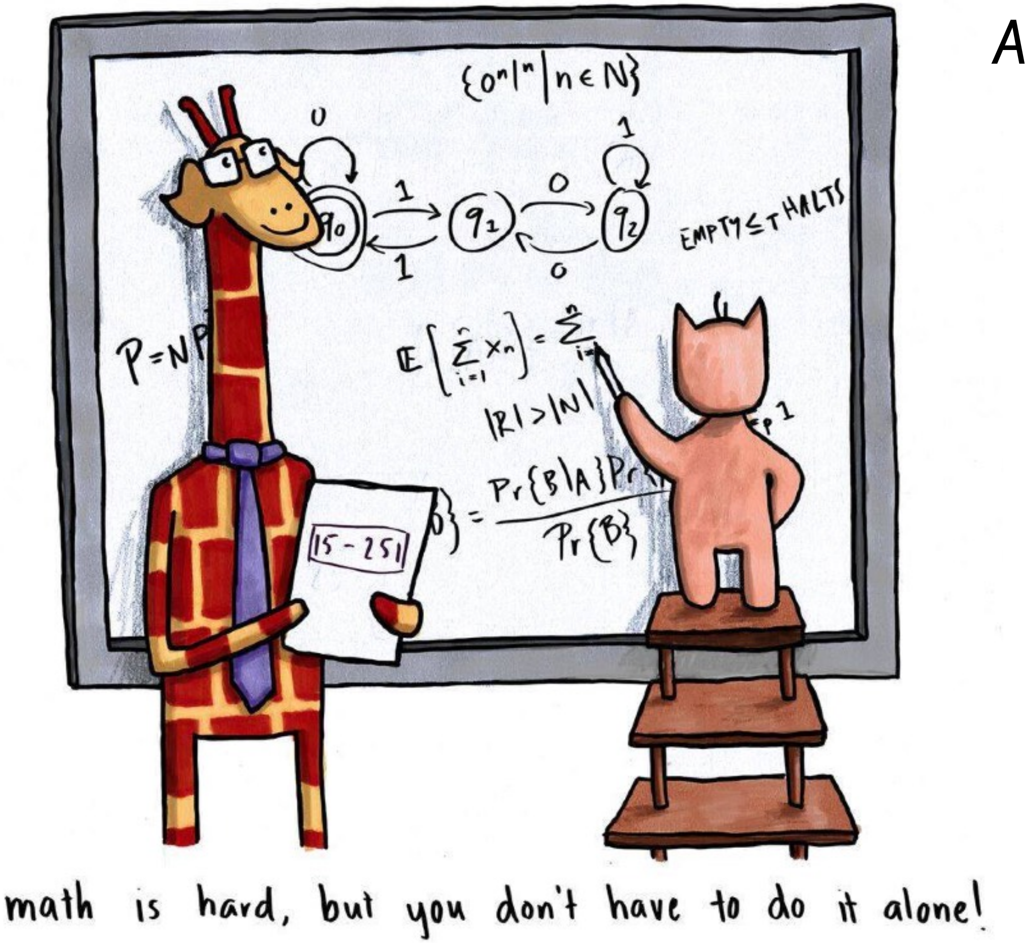
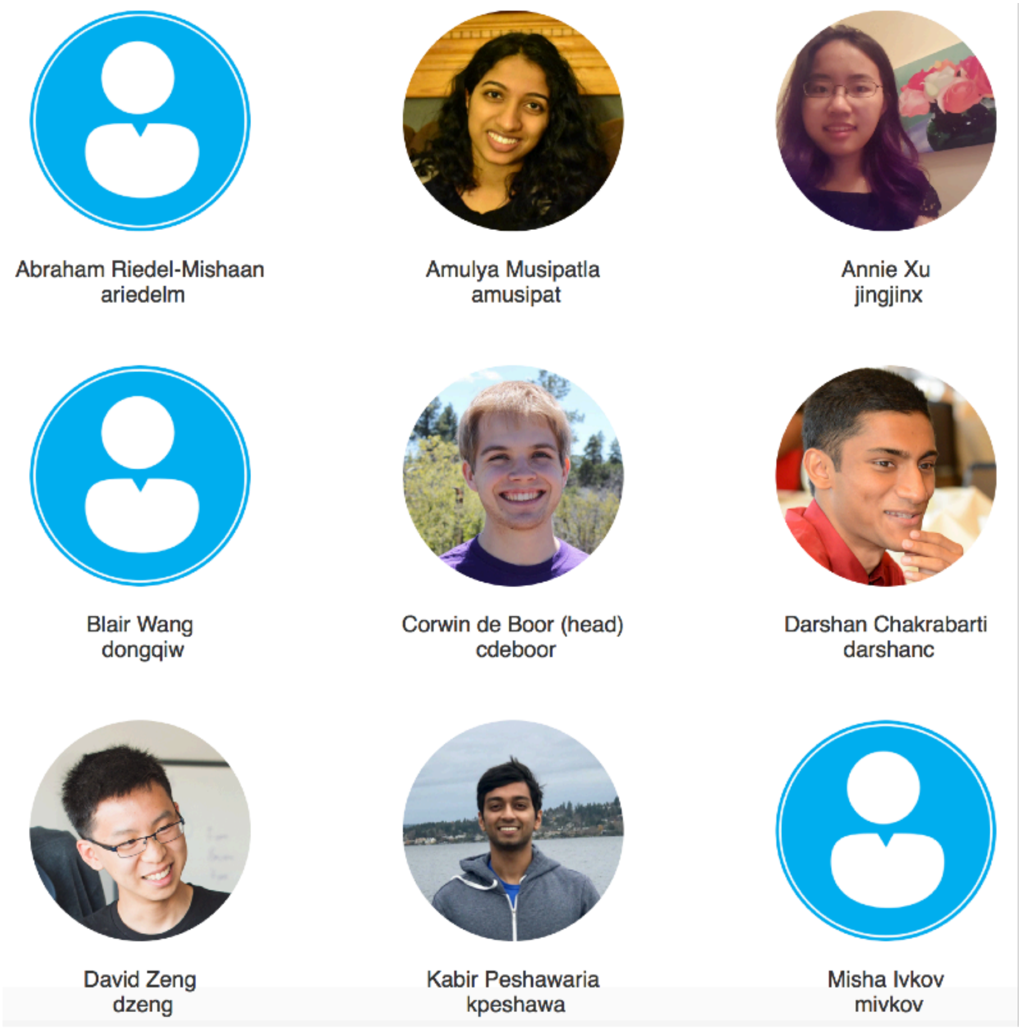
15-251: Great Ideas in Theoretical Computer Science

Lecture 1: Introduction to the course



August 28th, 2018

Teaching Assistants



Abraham Riedel-Mishaan
ariedelm

Amulya Musipatla
amusipat

Annie Xu
jingjinx

Blair Wang
dongqiw

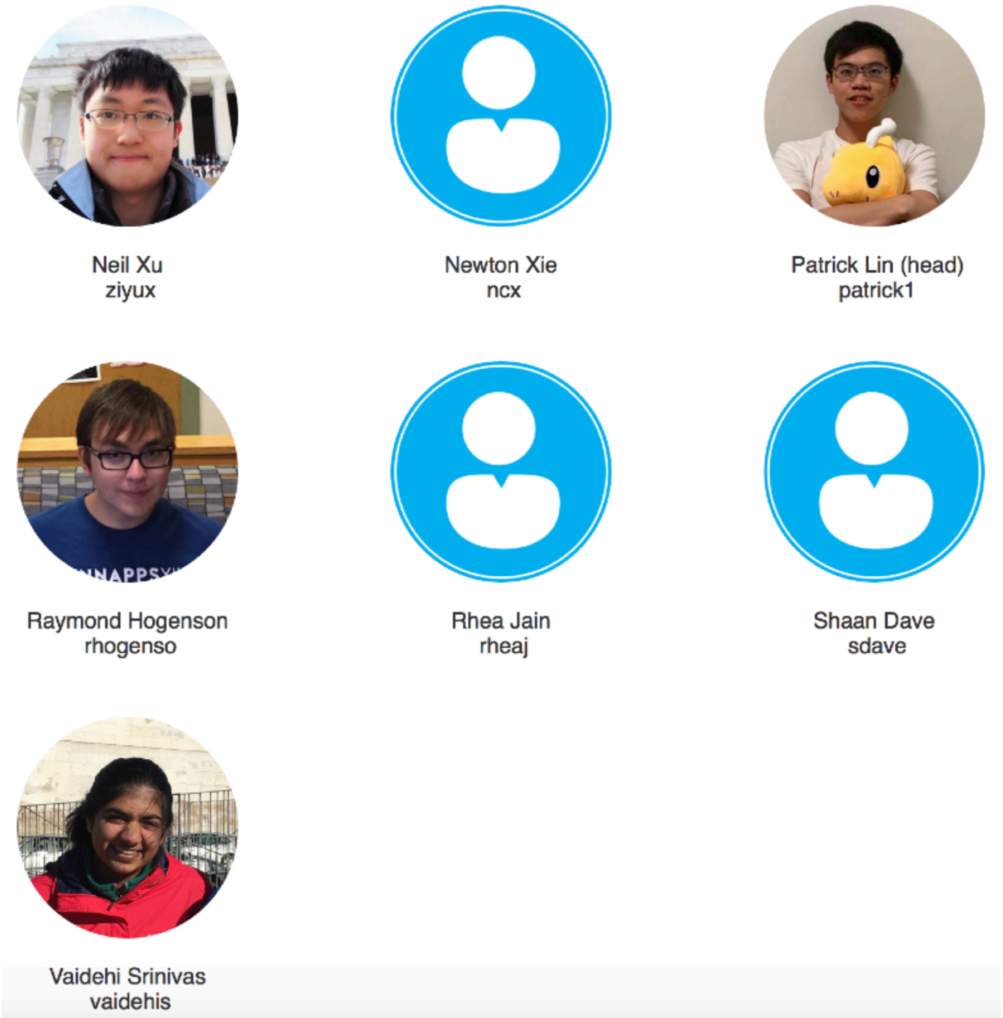
Corwin de Boor (head)
cdeboor

Darshan Chakrabarti
darshanc

David Zeng
dzeng

Kabir Peshawaria
kpeshawa

Misha Ivkov
mivkov



Neil Xu
ziyux

Newton Xie
ncx

Patrick Lin (head)
patrick1

Raymond Hogenson
rhogenso

Rhea Jain
rheaj

Shaan Dave
sdave

Vaidehi Srinivas
vaidehis

Instructors

Anil Ada
aada

Bernhard Haeupler
haeupler

Course webpage: www.cs.cmu.edu/~15251

Lecture tomorrow: DH 2315 6:30 - 7:50pm

What is **computer science**?

What is ***theoretical*** **computer science**?

What is computer science?

Is it a branch of:

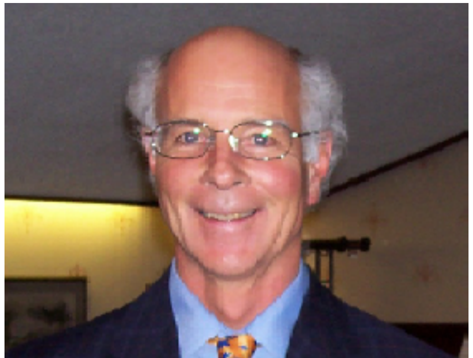
- science?
- engineering?
- math?
- philosophy?
- sports?

Motivational Quote of the Course

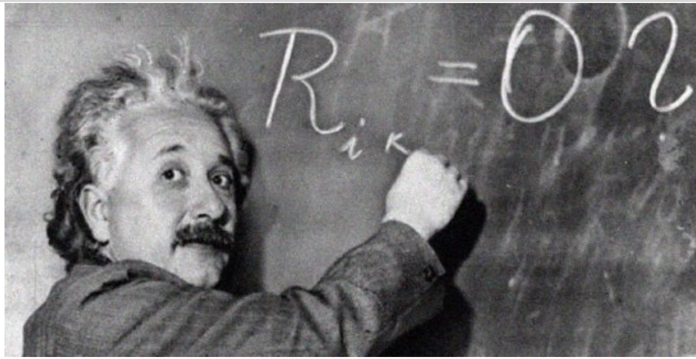
"Computer Science is no more about computers than astronomy is about telescopes."

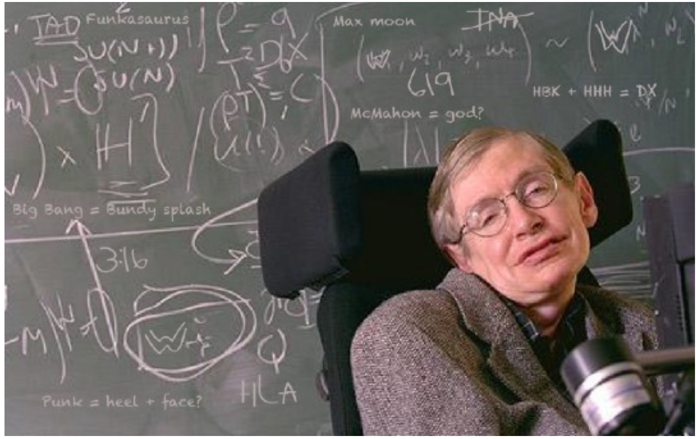
~~Edsger Dijkstra~~

- *Michael Fellows*



Physics





Theoretical physics

- come up with mathematical models

  Nature’s language is mathematics

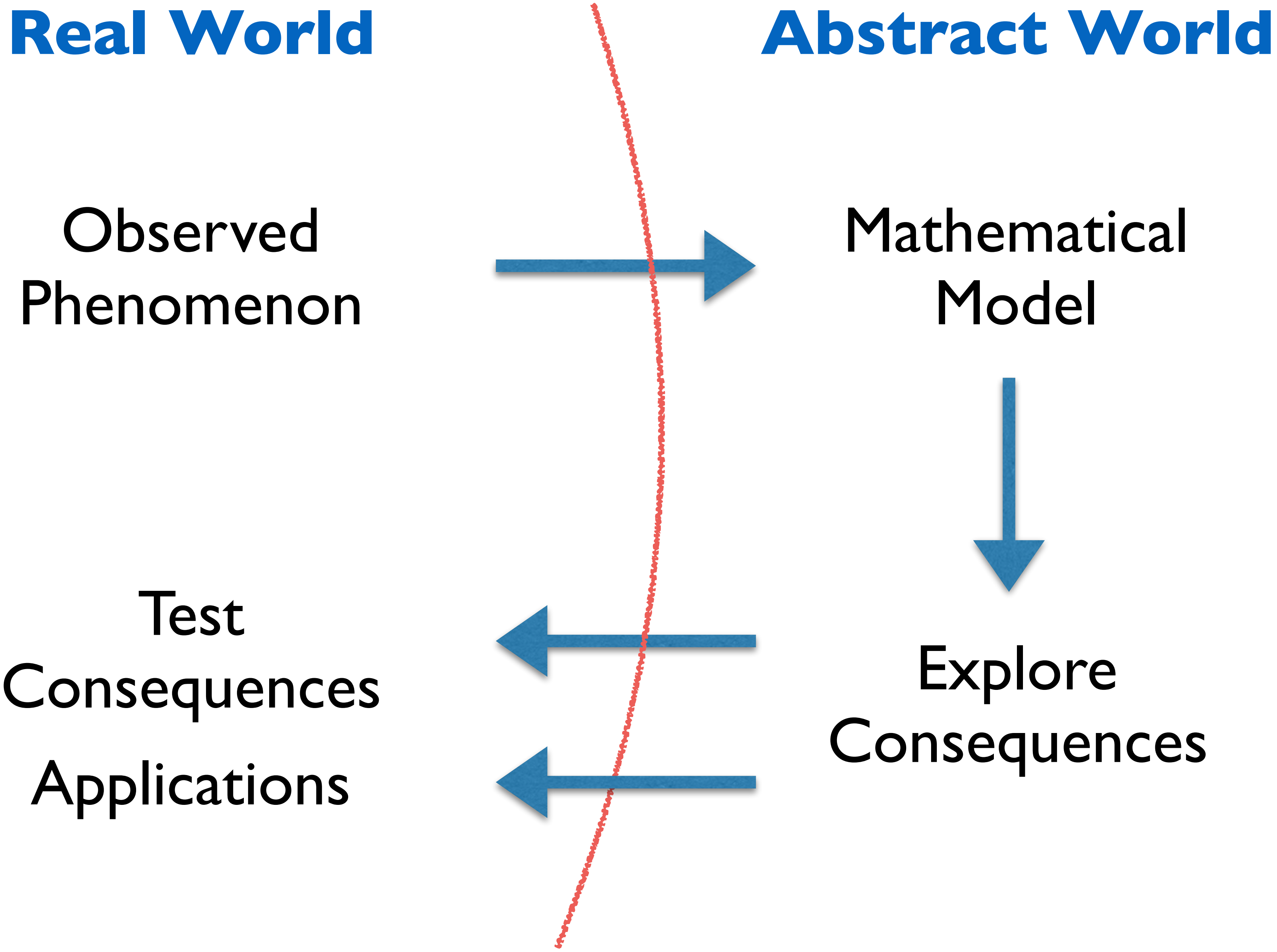
- derive the logical consequences

Experimental physics

- make observations about the universe
- test mathematical models with experiments

Applications/Engineering

The role of theoretical physics



Real World

Abstract World

Observed Phenomenon

Mathematical Model

Explore Consequences

Test Consequences

Applications

Physics

- science?
- engineering?
- math?
- philosophy?
- sports?

Computer Science

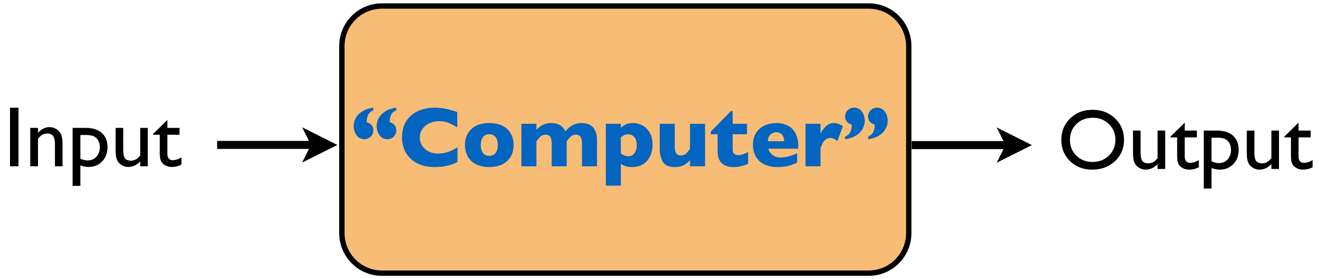
The science that studies computation.

Computation: manipulation of information/data.

Algorithm: description of how the data is manipulated.

Computational problem: the input-output pairs.

Usually



Computer Science

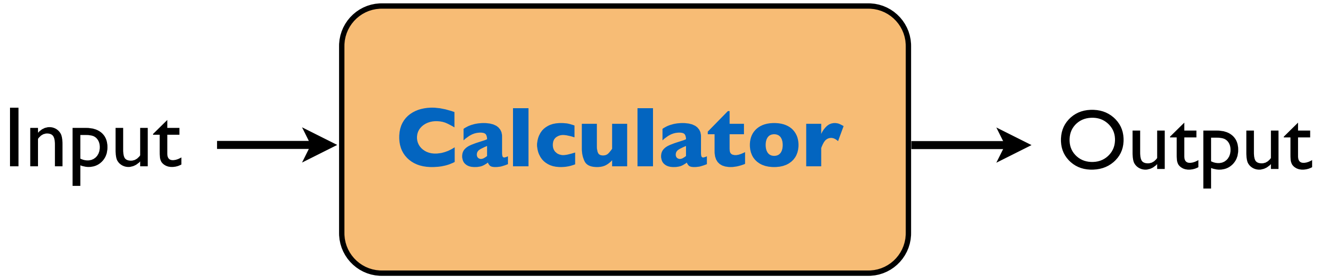
The science that studies computation.

Computation: manipulation of information/data.

Algorithm: description of how the data is manipulated.

Computational problem: the input-output pairs.

Usually



Computer Science

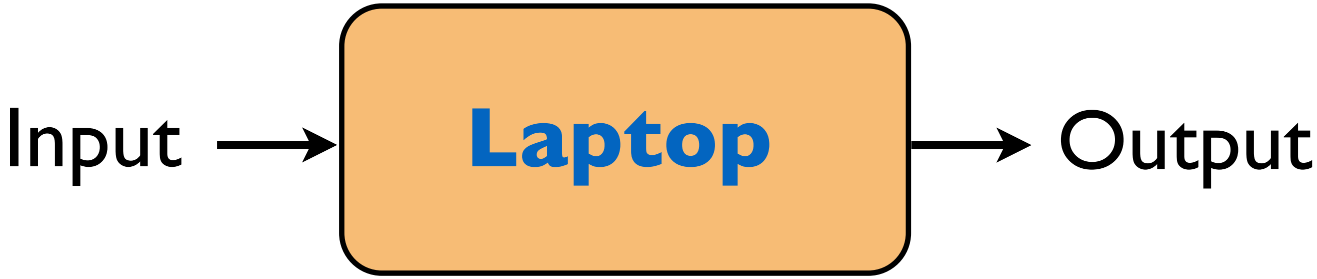
The science that studies computation.

Computation: manipulation of information/data.

Algorithm: description of how the data is manipulated.

Computational problem: the input-output pairs.

Usually



Computer Science

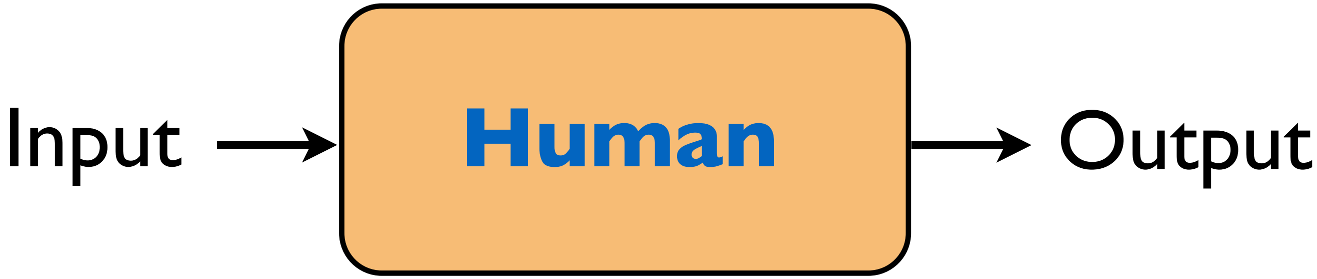
The science that studies computation.

Computation: manipulation of information/data.

Algorithm: description of how the data is manipulated.

Computational problem: the input-output pairs.

Usually



Input → **Human** → Output

“Computers” in early 20th century

Computer Science

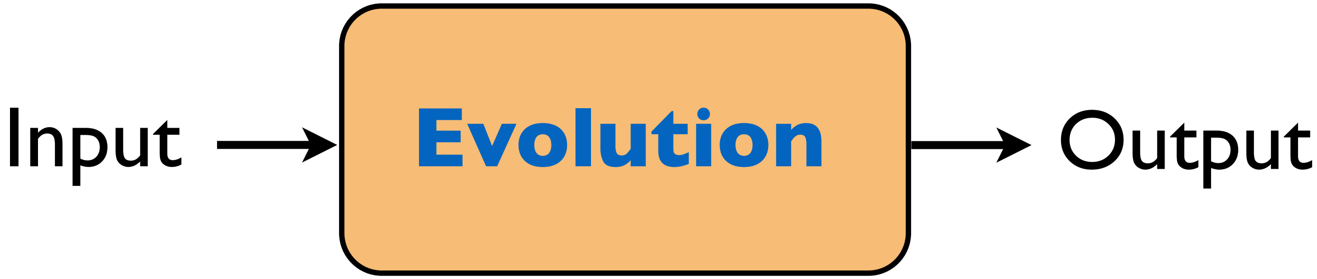
The science that studies computation.

Computation: manipulation of information/data.

Algorithm: description of how the data is manipulated.

Computational problem: the input-output pairs.

Usually



The computational lens

Computational physics

Computational biology

Computational chemistry

Computational neuroscience

Computational economics

Computational finance

Computational linguistics

Computational statistics

Computational social choice

…

The role of theoretical computer science

Build a mathematical model for computation.

Explore the logical consequences.
Gain insight about computation.

http://youtu.be/pTeZP-XfuKI

https://goo.gl/gGkpMv

http://youtu.be/J4TkHuTmHsg

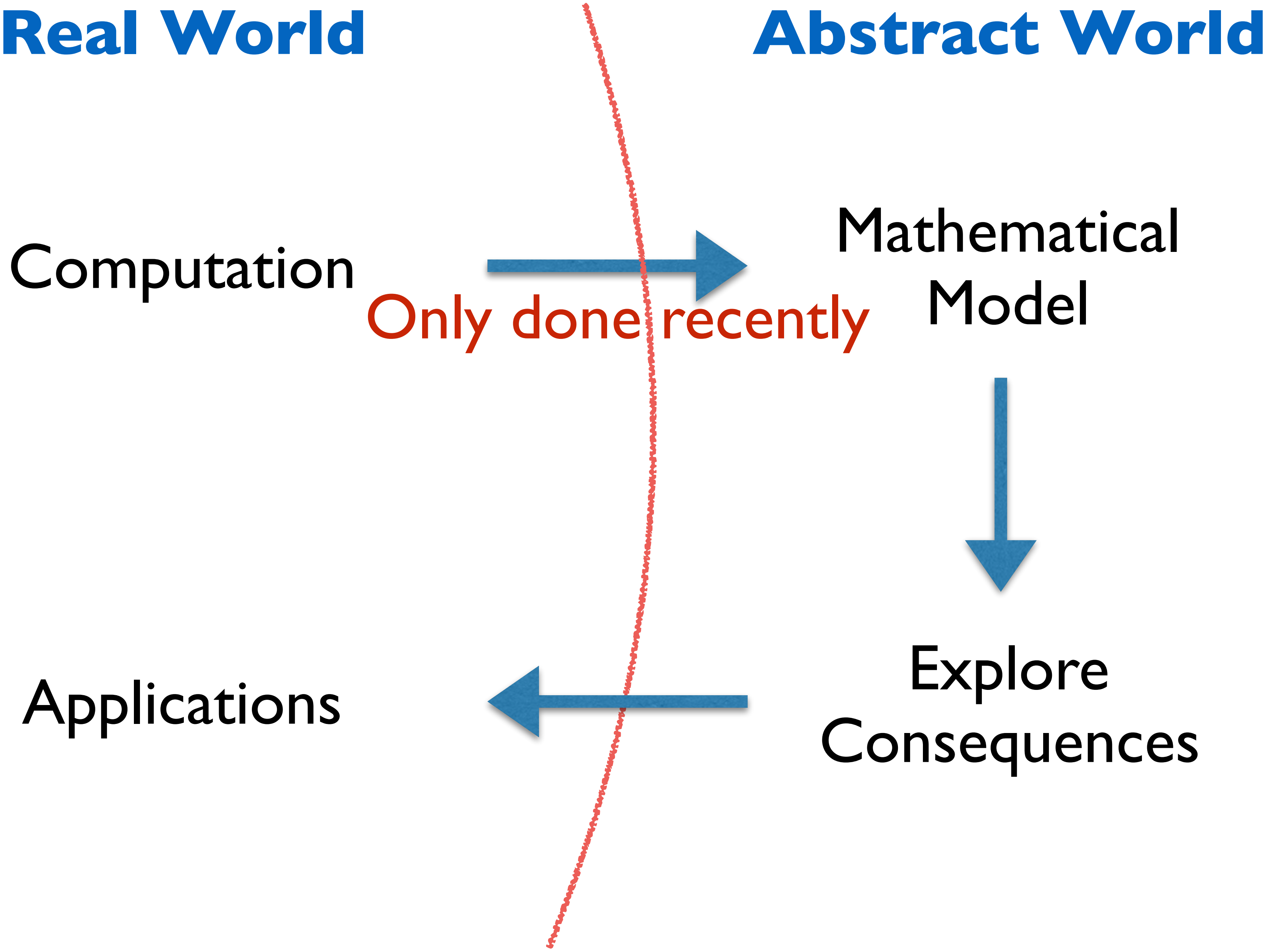
Look for interesting applications.

CMU undergrad

CMU Prof.

OK, we don't have everybody

The role of theoretical computer science



Real World

Abstract World

Computation

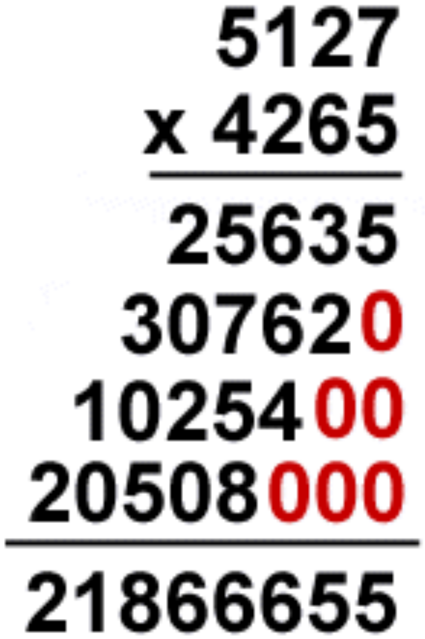
Mathematical Model

Only done recently

Explore Consequences

Applications

We have been using algorithms for thousands of years.



```
    5127
  x 4265
  ------
   25635
  307620
 1025400
20508000
--------
21866655
```

We have been using algorithms for thousands of years.

Euclid’s algorithm (~ 300BC):

```
def gcd(a, b):
    while (a != b):
        if (a > b):
            a = a - b
        else:
            b = b - a
    return a
```

Formalizing computation

Algorithm/Computation was only formalized in the 20th century!

Someone had to ask the right **question**.

David Hilbert, 1900

The Problems of Mathematics

"Who among us would not be happy to lift the veil behind which is hidden the future; to gaze at the coming developments of our science and at the secrets of its development in the centuries to come? What will be the ends toward which the spirit of future generations of mathematicians will tend? What methods, what new facts will the new century reveal in the vast and rich field of mathematical thought?"

2 of Hilbert's Problems

Hilbert's 10th problem (1900)

Is there a finitary procedure to determine if a given multivariate polynomial with integral coefficients has an integral solution?

e.g. $$5x^2yz^3 + 2xy + y - 99xyz^4 = 0$$

Entscheidungsproblem (1928)

Is there a finitary procedure to determine the validity of a given logical expression?

e.g. $$\neg\exists x, y, z, n \in \mathbb{N} : (n \geq 3) \wedge (x^n + y^n = z^n)$$

(Mechanization of mathematics)

2 of Hilbert's Problems

Fortunately, the answer turned out to be NO.

2 of Hilbert's Problems

Gödel (1934):

Discusses some ideas for mathematical definitions of computation. But not confident what is a good definition.

Church (1936):

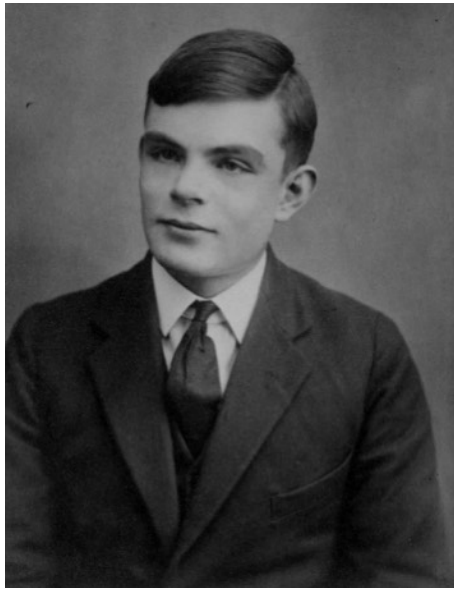
Invents lambda calculus.
Claims it should be the definition of an "algorithm".

Gödel, Post (1936):

Arguments that Church's claim is not justified.

Meanwhile… in New Jersey… a certain British grad student, unaware of all these debates…

2 of Hilbert’s Problems



Alan Turing (1936, age 22):

Describes a new model for computation, now known as the Turing Machine.™

Gödel, Kleene, and even Church:

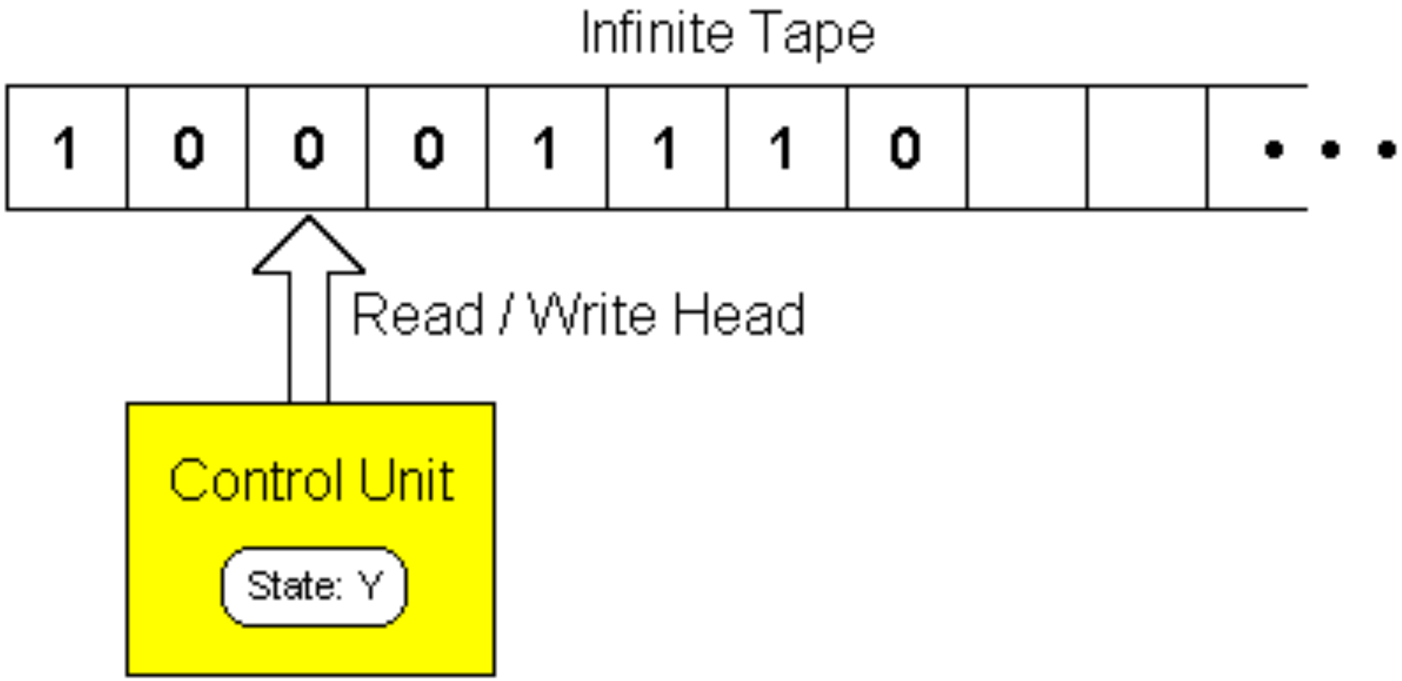
“Umm. Yeah. He nailed it. Game over. ‘Algorithm’ defined.”

Turing (1937):

TMs ≡ lambda calculus

Formalization of computation: Turing Machine

Turing Machine:



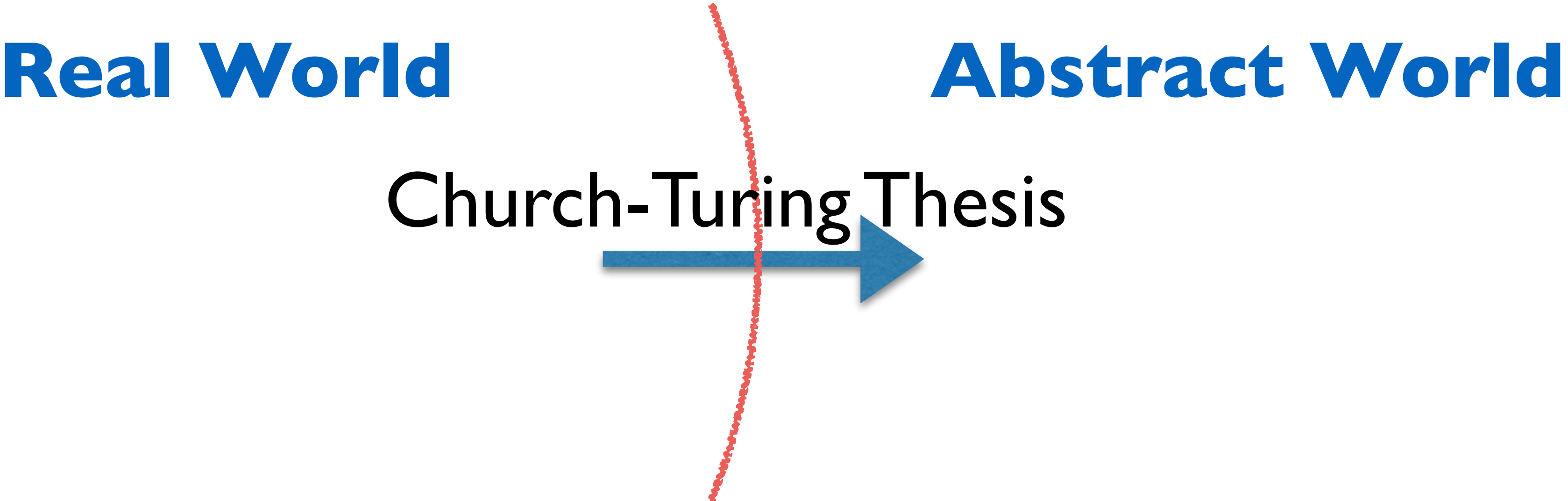
Church-Turing Thesis

Church-Turing Thesis:

The intuitive notion of "computable" is captured by functions computable by a Turing Machine.

(Physical) Church-Turing Thesis

Any computational problem that can be solved by a physical device, can be solved by a Turing Machine.



Real World

Abstract World

Church-Turing Thesis

Back to Hilbert's Problems

Hilbert's 10th problem (1900)

Is there an **algorithm** (a TM) to determine if a given multivariate polynomial with integral coefficients has an integral solution?

e.g. $$5x^2yz^3 + 2xy + y - 99xyz^4 = 0$$

Entscheidungsproblem (1928)

Is there an **algorithm** (a TM) to determine the validity of a given logical expression?

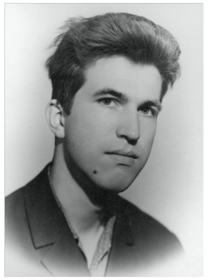
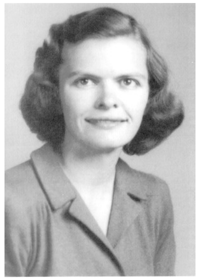
e.g. $$\neg \exists x, y, z, n \in \mathbb{N} : (n \geq 3) \wedge (x^n + y^n = z^n)$$

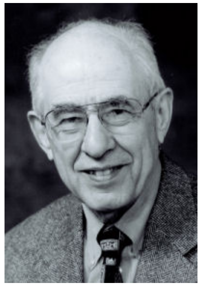
(Mechanization of mathematics)

Back to Hilbert's Problems

Hilbert's 10th problem (1900)

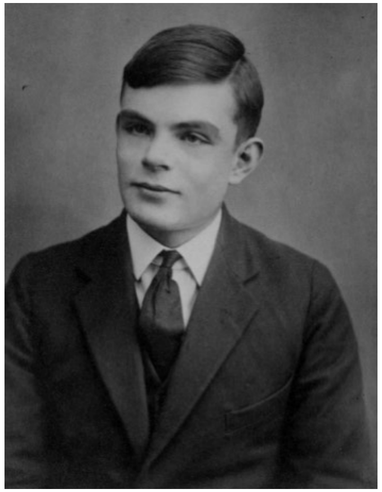
Matiyasevich-Robinson-Davis-Putnam (1970):



There is no algorithm to solve this problem.

Entscheidungsproblem (1928)



Turing (1936):

There is no algorithm to solve this problem.

Computer science

- science?
- engineering?
- math?
- philosophy?
- sports?

More on Theoretical Computer Science (TCS)

2 Main Questions in TCS

Computability of a problem:

Is there an algorithm to solve it?

Complexity of a problem:

Is there an efficient algorithm to solve it?

- time
- space (memory)
- randomness
- quantum resources

Computational Complexity

Complexity of a problem:

Is there an efficient algorithm to solve it?

- time
- space (memory)
- randomness
- quantum resources

2 camps:

- trying to come up with efficient algorithms (algorithm designers)
- trying to show no efficient algorithm exists (complexity theorists)

Computational Complexity

2 camps:

- trying to come up with efficient algorithms (algorithm designers)
- trying to show no efficient algorithm exists (complexity theorists)

multiplying two integers

factoring integers

detecting communities in social networks

protein structure prediction

simulation of quantum systems

computing Nash Equilibria of games

Some other interesting questions

P vs **NP**

Some other interesting questions

Time vs **Space**

Some other interesting questions

Deterministic vs **Randomized**

Some other interesting questions

Cryptography and **Security**

Some other interesting questions

Socioeconomics

(e.g. privacy, fairness)

Some other interesting questions

Learning Theory

Some other interesting questions

Quantum Computation

Some other interesting questions

…

Learning Objectives

Perspective 1

Overview of Topics

Part 1: Formalizing the notions of
problems, algorithms, and computability.

Part 2: Computational complexity:
theory and applications.

Part 3: Randomness in CS and
some highlights of theoretical CS.

BIG-PICTURE COURSE

Finite automata

Turing machines

Uncountability and Undecidability

Graph theory

Time complexity

P vs NP

Approximation algorithms

Probability

Randomized algorithms

Basic number theory

Cryptography

Quantum computation

Error-Correcting Codes

Communication Complexity

...

Perspective 1

Goals

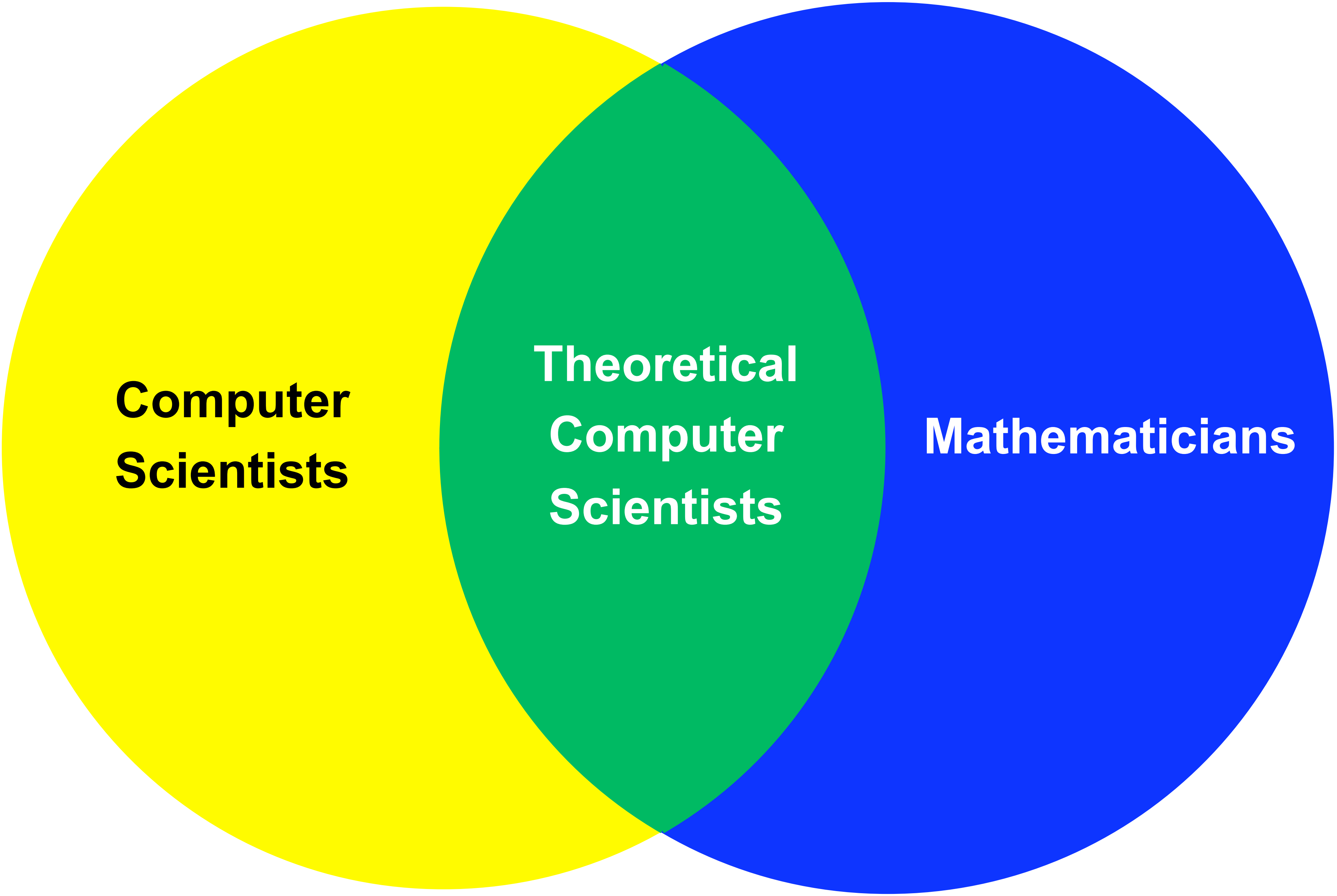
- Provide a formal introduction to the foundations of computer science.

- Improve your rigorous, logical and abstract thinking skills.

- Prepare you to be innovators in computer science.

- Push you to strive for clarity of thought and clarity in expression of thought.

Perspective 2

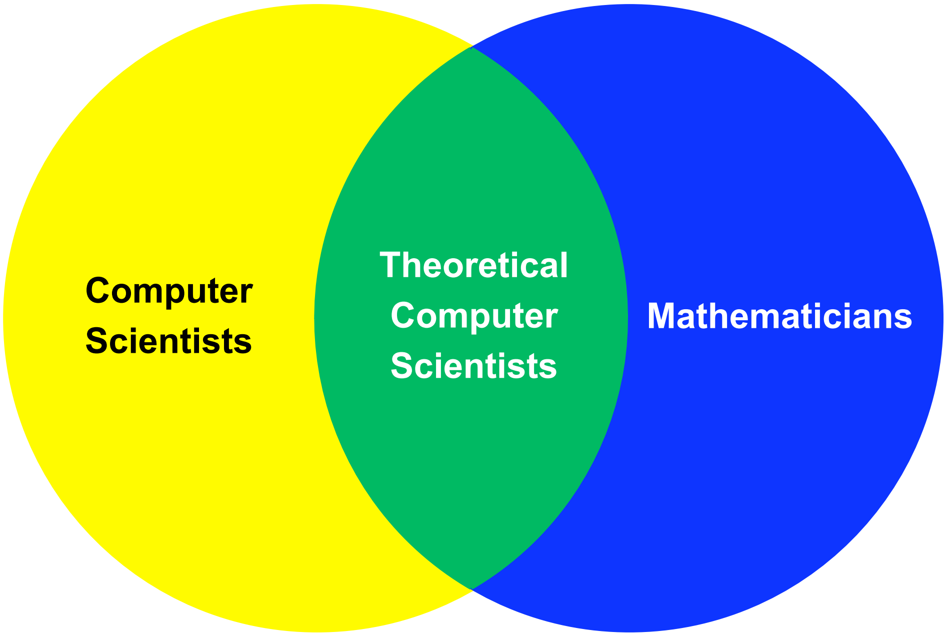


Computer Scientists

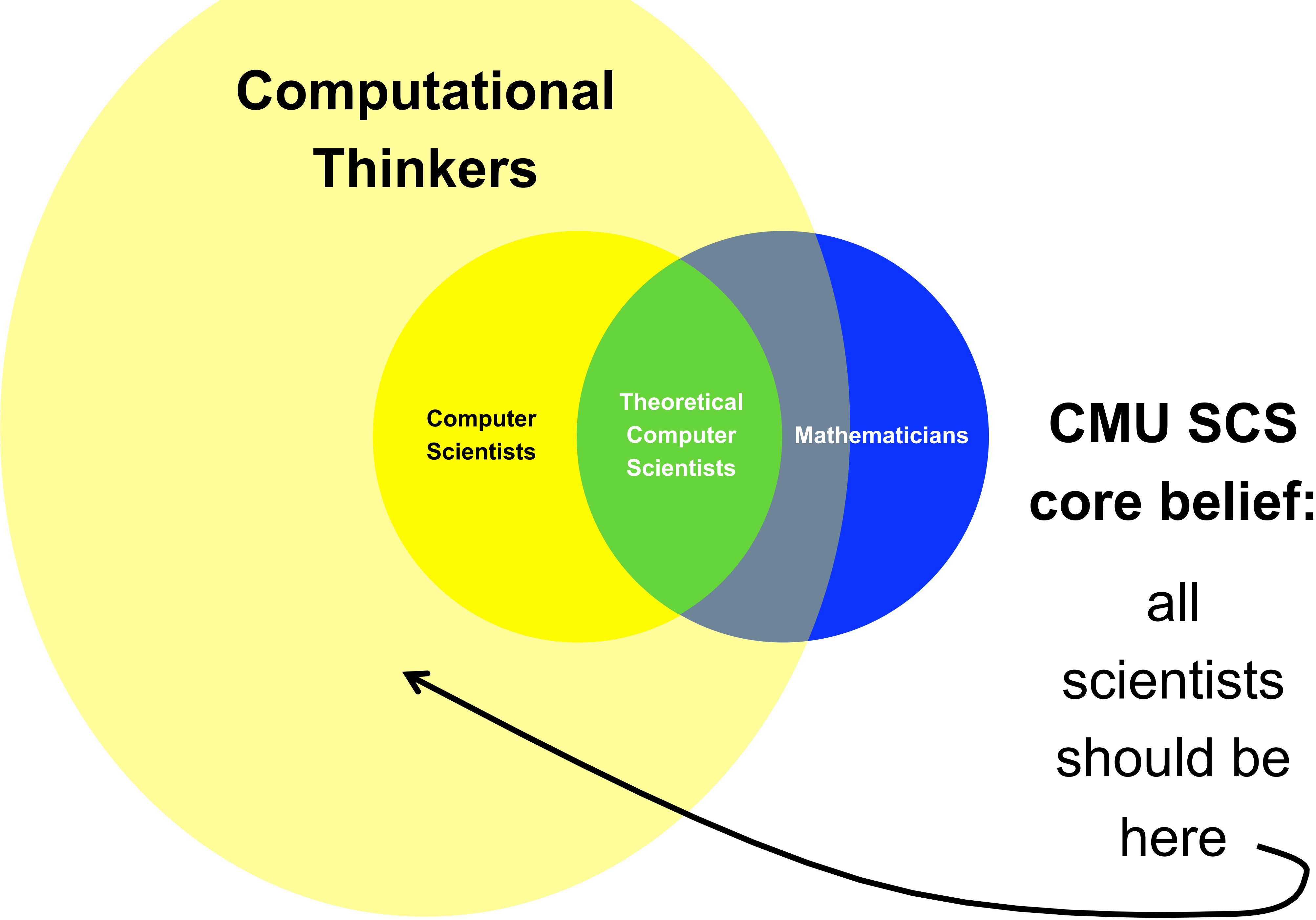
Theoretical Computer Scientists

Mathematicians

Perspective 2



Perspective 2



Computational Thinkers

Computer Scientists

Theoretical Computer Scientists

Mathematicians

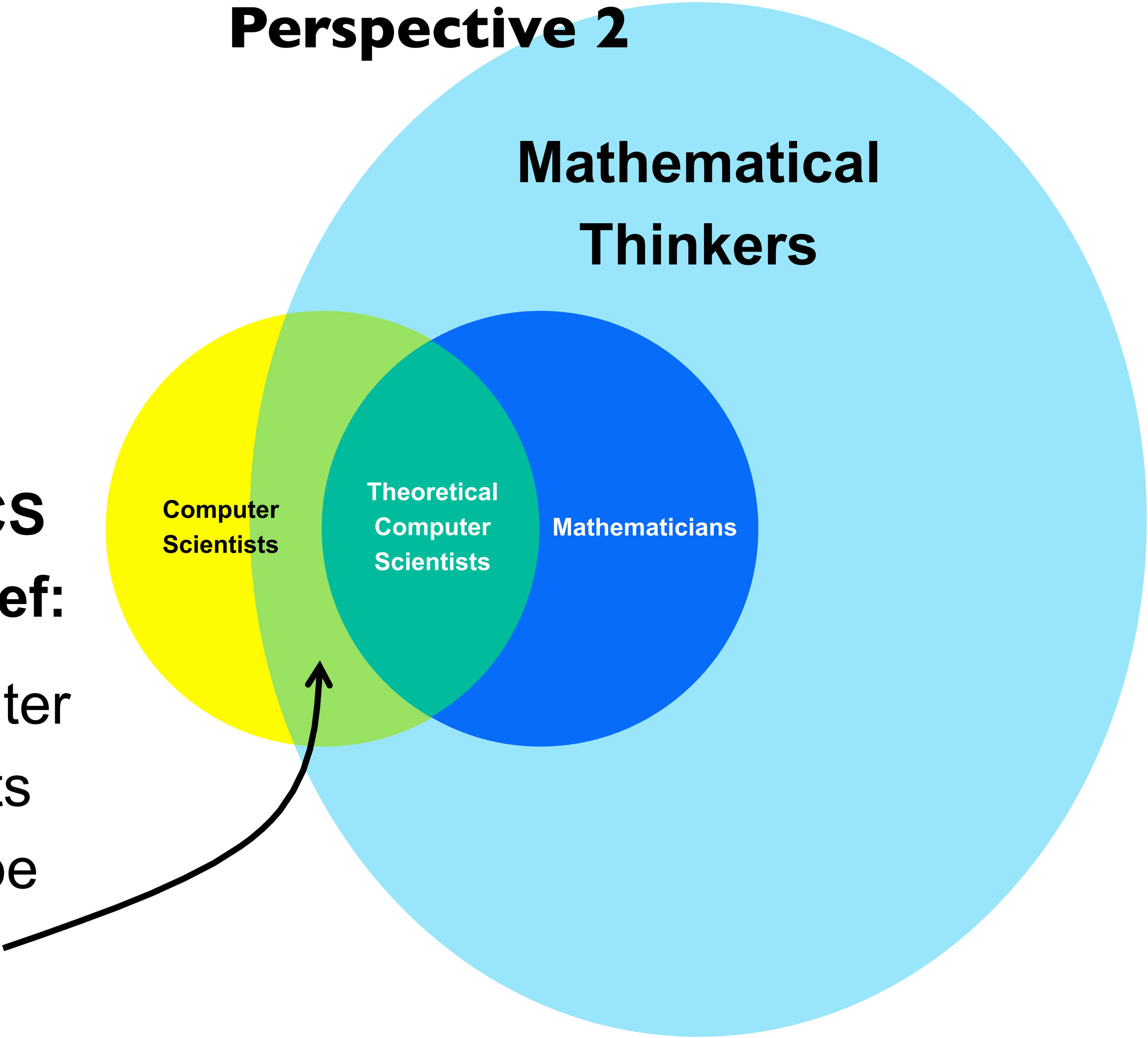
CMU SCS core belief:

all scientists should be here

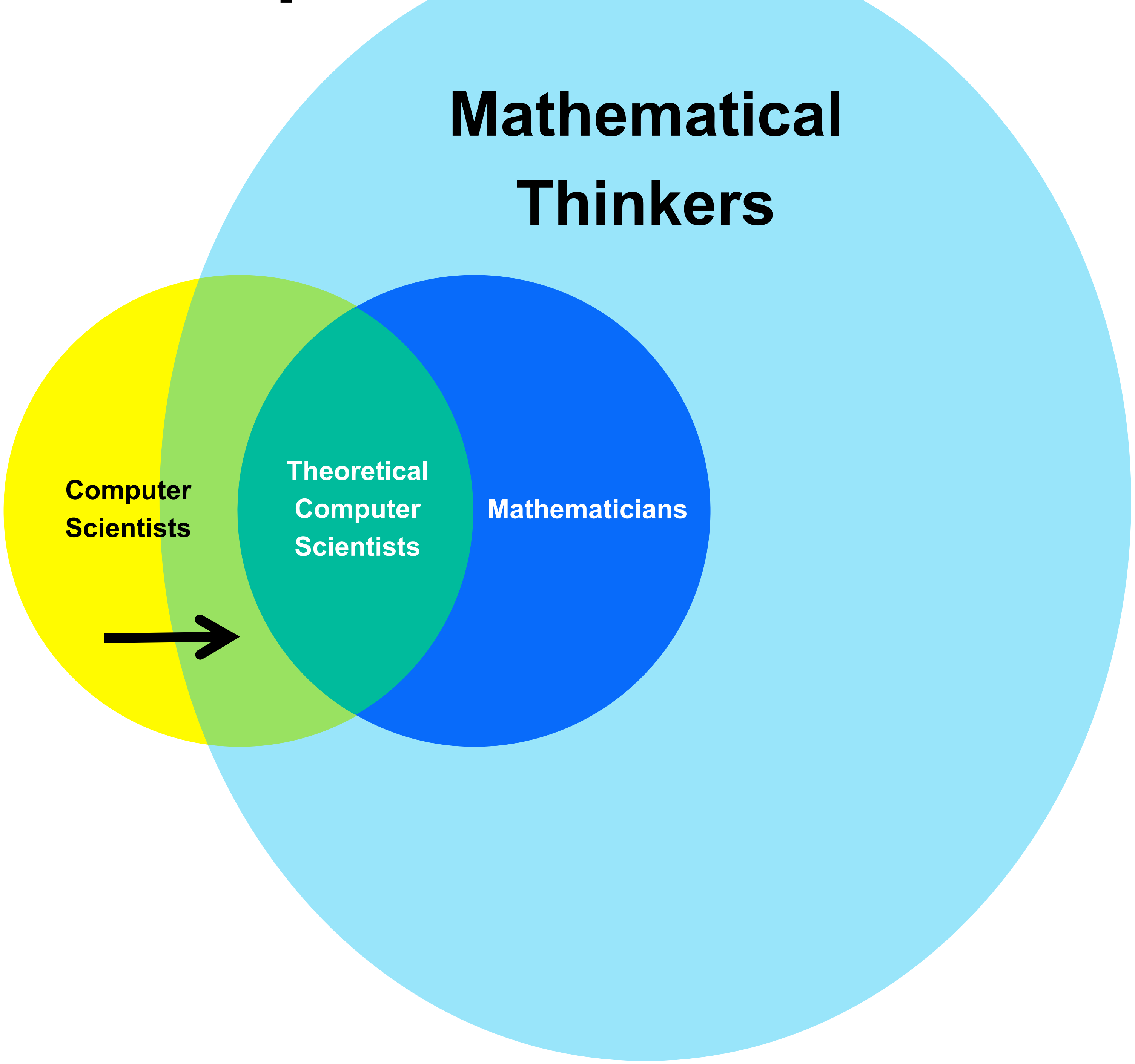
Perspective 2

CMU SCS core belief:

all computer scientists should be here



Perspective 2



Mathematical Thinkers

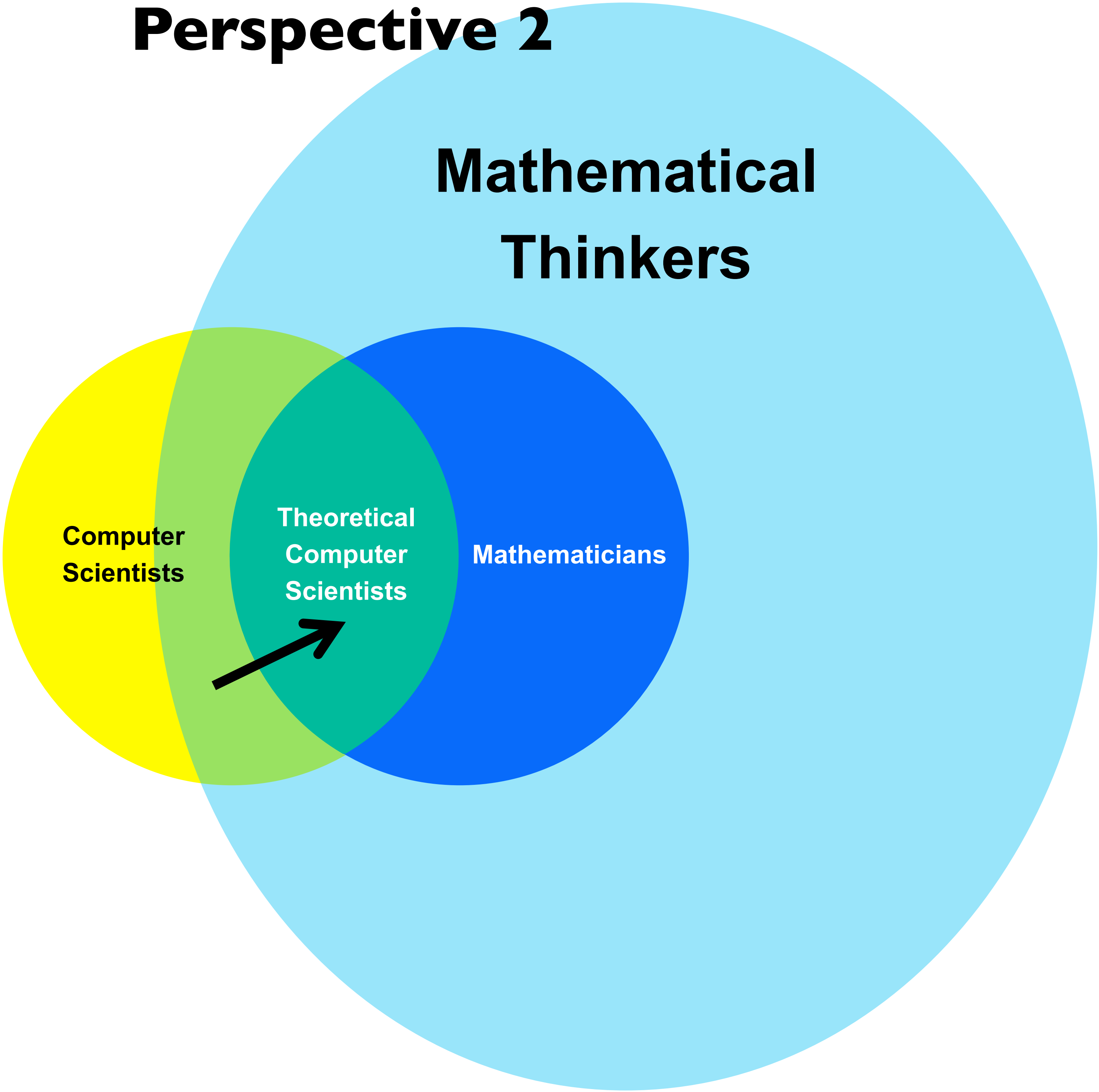
Computer Scientists

Theoretical Computer Scientists

Mathematicians

Goal #1 of 15-251

Perspective 2



Goal #2 of 15-251 (maybe)

Perspective 3

Mathematics is like… cilantro.

There are **5** kinds of people when it comes to cilantro.

Perspective 3

1. People who do not know what cilantro is.

Perspective 3

1. People who do not know what cilantro is.

Perspective 3

1. People who do not know what cilantro is.

Coriandrum sativum

Coriander (leaves)

香菜

धनिया

고수

گشنیز

الكزبرة

ngò

φύλλα κόλιανδρου

கொத்தமல்லி

ধনে

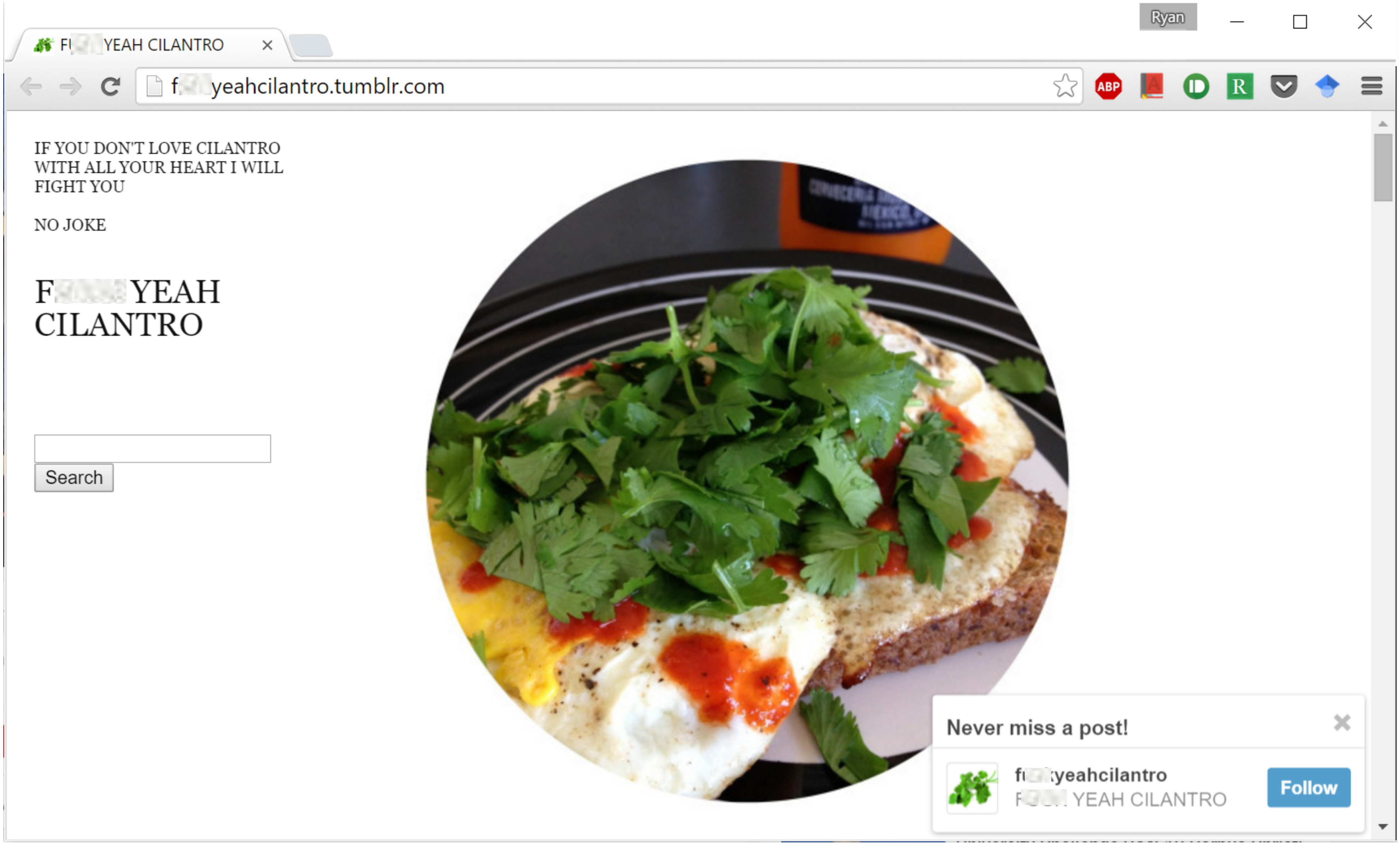
kişniş

кинза

כוסברה

Perspective 3

2. People who love cilantro.



F[redacted] YEAH CILANTRO

f[redacted]yeahcilantro.tumblr.com

IF YOU DON'T LOVE CILANTRO
WITH ALL YOUR HEART I WILL
FIGHT YOU

NO JOKE

F[redacted] YEAH
CILANTRO

Never miss a post!

fu[redacted]yeahcilantro
F[redacted] YEAH CILANTRO

Perspective 3

3. People who think cilantro is fine.

Goal: have everyone at least in this category
by the end of the course.

Perspective 3

4. People who don’t like cilantro.

Still gotta try it.

Hope to move you to previous category.

If not, hope you can eat cilantro if necessary.

Perspective 3

5. People with a genetic condition that makes cilantro taste like soap.

Is this true?

Video