Theory of Computing
Fall 2019

Dr. Corey Pennycuff
http://cscrunch.com:8890
What is Computer Science?

How is it different from Computer Programming?

What is a computer?

What does it mean to compute?

Is creativity a computation?

What resources are required for computation?

What is the relationship between a computer program and math?
“I didn’t say that!”
- - -
(it was actually Michael R. Fellows, 1991)

“Computer Science is no more about computers than Astronomy is about telescopes!”

-Edsger W. Dijkstra
Post’s Correspondence Problem - 1946


Game: https://www3.nd.edu/~dchiang/teaching/theory/2020/www/poco/
      Special thanks to Dr. Chiang for writing this implementation!

An “easy” Decision problem.

How can we write a program to solve this problem?
Computability and Decidability

Down the rabbit hole...

Given a computational problem:

- Is it possible to compute the answer?
- How do you know?
- Is it possible to know whether or not we can know?
- What type of machine could (or could not) perform the computation?
HILBERT’S ENTSCHEIDUNGSPROBLEM - 1928

Given a statement $\phi$ of first order logic and some axioms, is there an algorithm that decides whether $\phi$ is provable from the axioms?
In computability theory, the halting problem is the problem of determining, from a description of an arbitrary computer program and an input, whether the program will finish running (i.e., halt) or continue to run forever.
Alan Turing (1912-1954)

Influential in development of Theoretical Computer Science

Formalized concepts of Algorithms and Computation

Developed the idea of the Turing Machine

Portrayed by Benedict Cumberbatch.
Turing Machine

**Powerful** abstraction

Can compute **anything** that is computable

 Doesn’t need **electricity**

https://www.youtube.com/watch?v=vo8izCKHiF0

https://www.youtube.com/watch?v=E3keLeMwfHY&index=2&list=PLF87F259C658F6872&t=0s
Problem:

Turing machines are too powerful!!!
We Must Restrict Our Definitions!

Complex
- Turing Machines
- Deterministic Context-Free Languages
- Pushdown Automata
- Context-Free Grammars
- Regular Languages
- Finite Automata
- Simple

Begin with the simplest.

Build to the most powerful.

Understand the limitations.
But…

Can a Turing machine solve Post’s Correspondence Problem?

How can you know for sure?