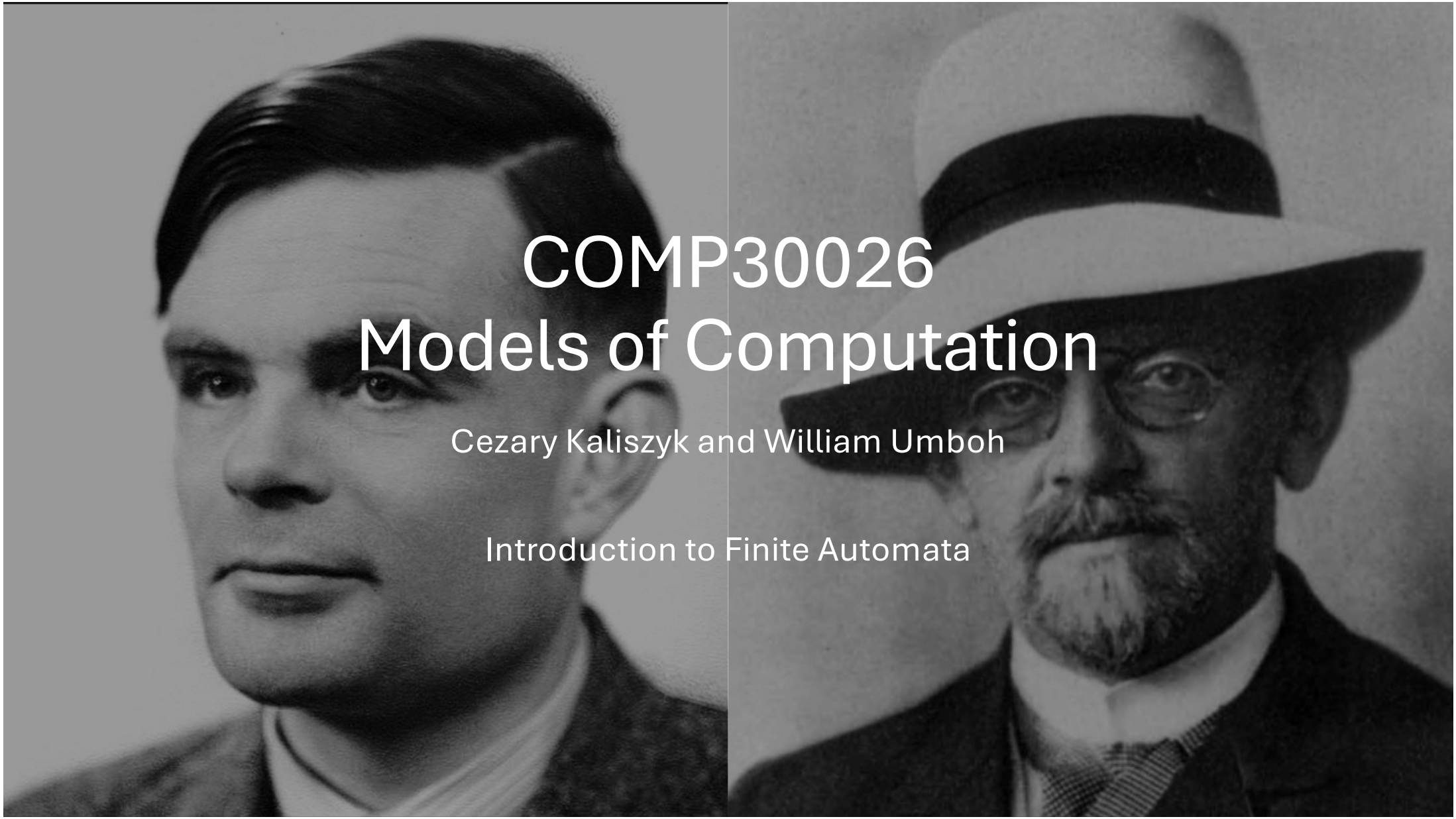

COMP30026
Models of Computation

Cezary Kaliszyk and William Umboh

Introduction to Finite Automata

Sparse slides ahead

- QR code for lecture notes
- Feel free to keep it open on your device
- Link in LMS Modules

Roadmap

- Motivation for studying models of computation
- Defining computation
- Introduction to finite automata

Some motivating questions

- Can LLMs do everything?
- How to design secure cryptographic protocols?
- How to process massive data streams?

All of these questions are about computational models!

Learning objectives

- Rigorously analyze and reason about computation
- Develop algorithms in restricted models
- Learn how to tackle challenging problems

What is computation?

Polleverywhere

Computation

- Computational problem = **specification**
  - for every input x, **what** is the correct output?
- Algorithm = **implementation**
  - for every input, **how** to obtain correct output

Want efficient algorithms

Algorithms use resources

- Time
- Space (aka memory)
- Randomness
- Quantum entanglement
- Input access
- And many others

computational model = resource constraints

Theoretical Computer Science

- Power and limits of computational models
- Tradeoff between resources

Space

Time

Quantum entanglement

Randomness

Impossibility results seem impossible?

- How do we rule out *every* possible algorithm?
- Let's start by simplifying
  - Inputs are bit strings (aka Boolean strings)
  - Outputs are YES/NO (i.e. single bit)

Simplify, simplify, simplify

- Inputs are bit strings (aka Boolean strings)
- Outputs are YES/NO (i.e. single bit)
  - Decision problems

- Computational problem = set of YES inputs = language

- Simple model called finite automata
- Key feature: memorylessness
  - Cannot allocate new memory ~ no new variables

What if we cannot make new memories?

Demo: Tally Counters