Automata

Understanding Computational Models

What makes a computer... a computer?

Learning Objectives

By the end of this session, you will be able to:

- **Define** what an automaton is and its role in computer science
- **Identify** the key components of abstract machines
- **Explain** why we study theoretical models of computation

Part 1: What is a Computer?

Let's explore two perspectives...

Computers in Practice

The machines we use daily

- Laptops, smartphones, servers
- Physical devices with:
    - Processors (CPUs, GPUs)
    - Memory (RAM, storage)
    - Input/Output devices
    - Operating systems and software

These are complex, real-world implementations

Computers in Theory

Abstract Mathematical Models

> An **automaton** (plural: automata) is an abstract machine or computational model

Key Insight

- We deliberately **ignore** implementation details
- Focus on **fundamental capabilities**
- Study the **essence** of computation

🤔 Active Learning: Think-Pair-Share

Question: Why might computer scientists want to study simplified, abstract models instead of real computers?

1. **Think** (1 min): Write down your ideas
2. **Pair** (2 min): Discuss with a neighbor
3. **Share** (2 min): Class discussion

Why Study Abstract Machines?

Fundamental Questions in Computer Science

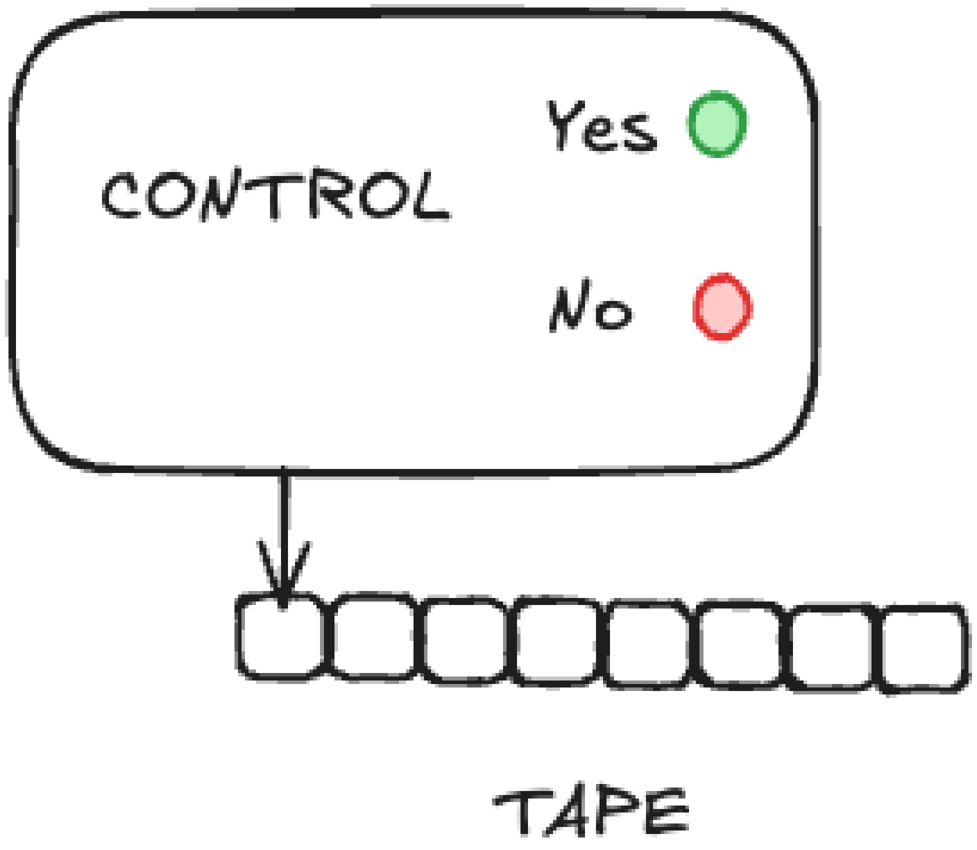
1. **Capabilities & Limitations**
    - What can computers do? What can't they do?
2. **Computational Power**
    - Are some models fundamentally more powerful?
3. **Undecidability**
    - Are there problems **no** computer can solve?
4. **Complexity**
    - Which problems are **impractical** to solve?

Real-World Impact

These aren't just theoretical curiosities!

- **Compiler Design**: Regular expressions → Finite automata
- **Security**: Undecidability → Perfect virus detection is impossible
- **Cryptography**: Complexity theory → RSA encryption
- **AI/ML**: Computational limits → What neural networks can learn
- **Database Systems**: Query optimization using automata

Anatomy of an Abstract Machine



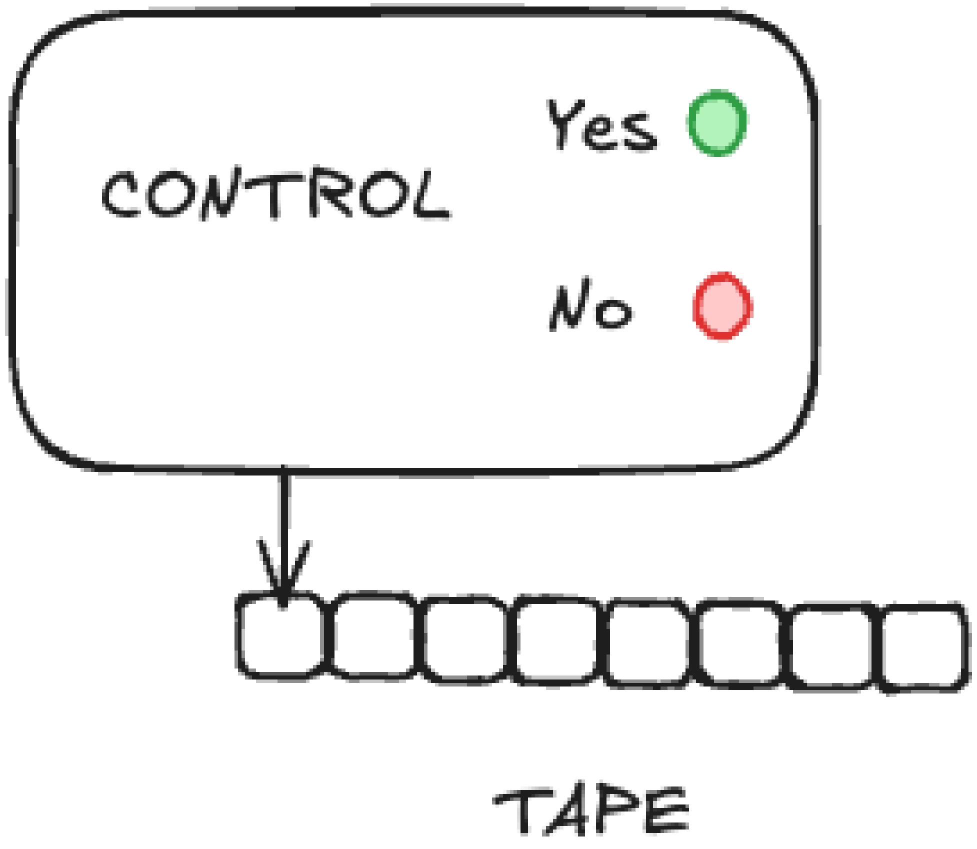
Component 1: The Tape

Input and Storage Medium

- Contains the **input string** for computation
- Can be viewed as:
  - Infinite or finite sequence of cells
  - Each cell contains a symbol from an alphabet

Example: Input string "0110" on tape:

| 0 | 1 | 1 | 0 | _ | _ | ...



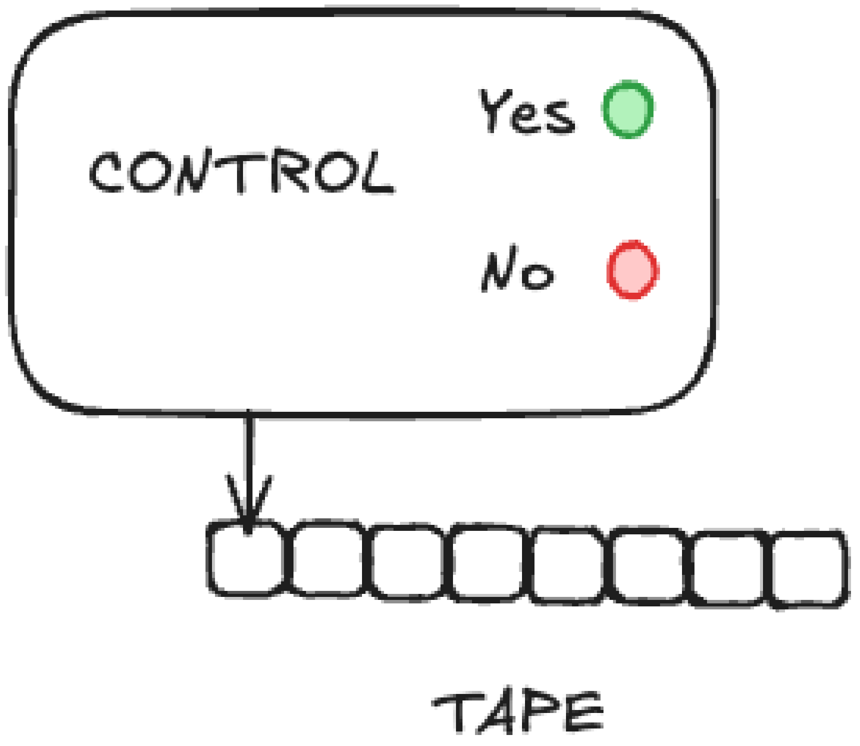
Component 2: The Control

The "Brain" of the Machine

Processes symbols from the tape **one at a time**

Varying Capabilities:

Capability	Description
Read-only	Can only examine symbols
Read-write	Can modify symbols on tape
Scan-right only	Moves in one direction
Bidirectional	Can move left or right



Component 3: The Output

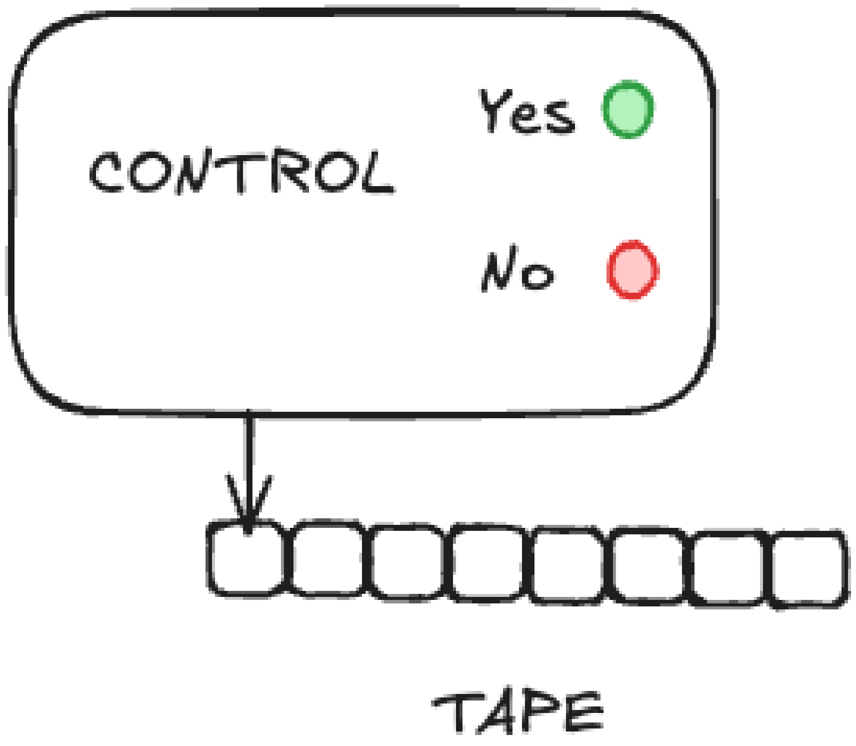
Two Types of Problems

1. Decision Problems

- Output: **Yes/No** (Accept/Reject)
- Example: "Is this string a palindrome?"

2. Function Problems

- Output: **Transformed string**
- Example: "Convert binary to decimal"
- Requires write capability on tape



Specific Automata

Of all conceivable automata, we'll focus on the following:

1. **Finite Automata**
    i. **Deterministic (DFA)**
    ii. **Nondeterministic (NFA)**
2. **Turing Machines (TM)**

There are good reasons to focus on the above atomata, and those will become clear when we examine them.