THEORY OF AUTOMATA & FORMAL LANGUAGES

Introduction to Automata

From everyday machines to mathematical models

Input → State → Rule → Decision

What is an automaton?

An automaton is an abstract machine that processes input according to precise rules.

Everyday intuition

A door can be open or closed.

A sensor detects a person.

A rule tells the door to open.

Another rule closes the door.

No human-like thinking is needed.

Computational intuition

Input: information received

State: current situation

Rule: how the state changes

Transition: a change of state

Result: action or decision

A machine can behave systematically using a small set of rules.

States: what must a machine remember?

A state summarizes the information needed to determine the next behaviour.

Example: a light switch

State OFF: light is not glowing.

State ON: light is glowing.

Input: press the switch.

OFF + press → ON

ON + press → OFF

Follow three presses

Initial state: OFF

Press 1 → ON

Press 2 → OFF

Press 3 → ON

Final state: ON

The state remembers whether an odd or even number of presses has occurred.

Transitions: rules drawn as a machine

Circles represent states; labelled arrows represent rules for moving between states.

Reading the diagram

Start in OFF.

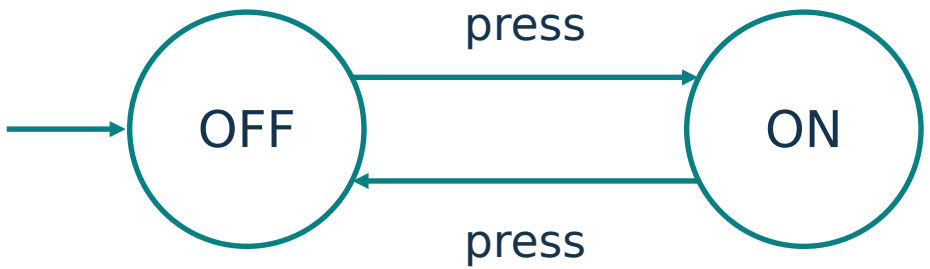
Follow an arrow labelled press.

Each press changes the state.

After two presses: OFF.

The same rules work repeatedly.

State diagram



Start at OFF • press to toggle

An incoming start arrow shows where processing begins.

From physical inputs to strings

A machine may receive symbols one at a time rather than physical events.

Binary input example

Available symbols: 0 and 1

A sequence: 1010

Read from left to right.

First symbol: 1

Last symbol: 0

What should the machine decide?

Rule: accept an even count of 1s.

1010 contains two 1s.

1011 contains three 1s.

1010 is accepted.

1011 is rejected.

A formal language is a set of strings; a recognizer tests membership.

A first finite automaton

Two states are enough to remember whether the number of 1s is even or odd.

Rules of the machine

Start at E (even count: zero).

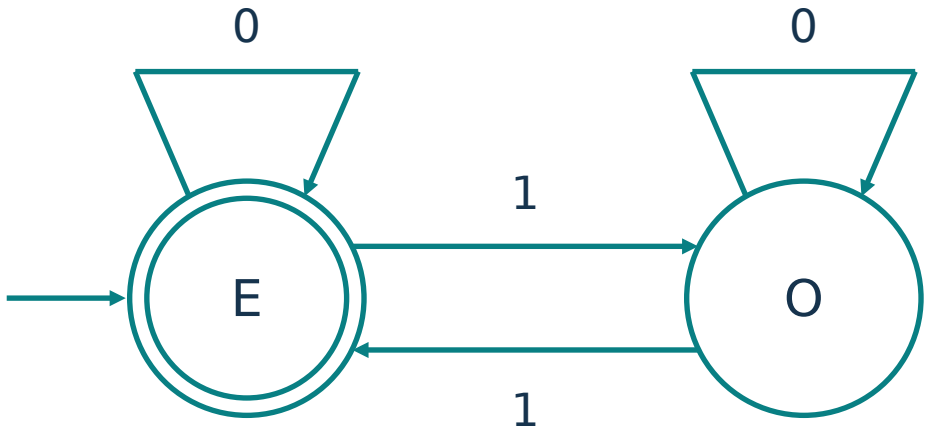
Reading 1 switches E ↔ O.

Reading 0 leaves the state unchanged.

E is accepting (double circle).

O is not accepting.

State diagram



Start at E • E accepts

Every state has one outgoing rule for each input symbol.

Trace a string step by step

Process the entire input before deciding whether the string is accepted.

Input 1010 — accepted

Start: E

Read 1 → O

Read 0 → O

Read 1 → E

Read 0 → E ✓

Input 1011 — rejected

Start: E

Read 1 → O

Read 0 → O

Read 1 → E

Read 1 → O ✗

Acceptance depends on the final state after all symbols are read.

What makes this a mathematical model?

The machine is specified by a finite set of states, symbols and exact transition rules.

Name the components

States: E and O

Input symbols: 0 and 1

Initial state: E

Accepting state: E

Transition rules: see table.

Transition table

Current state	0	1
E	E	O
O	O	E

Every state/input has one rule.

We call this a deterministic machine.

We will introduce the full formal definition of a DFA after language foundations.

Check your understanding

Use the same even-1s machine; do not count the zeros.

Practice

1. Trace 00.

2. Trace 111.

3. Trace 1001.

4. What if there is no input?

5. Does visiting E guarantee acceptance?

Answers and reasoning

1. E → E → E; accept.

2. E → O → E → O; reject.

3. E → O → O → O → E; accept.

4. Start at E; accept.

5. No. Finish in E to accept.

Next: symbols, alphabets and strings — the building blocks of formal languages.