

Question: What is a computer?

- A pocket calculator?
- A graphing calculator?
- A machine that can compile and run C code?
- A machine that just has a buggy version of Eclipse that can only check if Java code is syntactically correct (and nothing else).
- A machine that can just render LaTeX documents?
- A machine that can only run Minecraft?
- A machine that can only run Super Mario Brothers?

Main Loop For This Course

1. Identify a characteristic we think is important for computation. (e.g. memory)
2. Write down models that do/don't have that characteristic.
3. Ask what problems those models can or cannot solve.

Final goal: arrive at a model that resembles the computers we know.

Two ways of thinking about the course

1. Mathematics (definitions, formal proofs)
2. Programming strange machines with weird restrictions

State machine example

State machines have

1. a set of states and
2. a set of actions that cause the machine to change state.

At “runtime,” the machine keeps track of its current state and each action that forms its input causes it to update that state

Example: Water dispenser in a fridge: Water is dispensed only when the door is closed and the button is pressed. Closing the door while the button is pressed does *not* turn on the water.

States: $\{\text{open, closed}\} \times \{\text{pressed, not pressed}\} \times \{\text{on, off}\}$

Actions: Open/close door, press/release button

8 potential states, some, like (open, pressed, on) are invalid