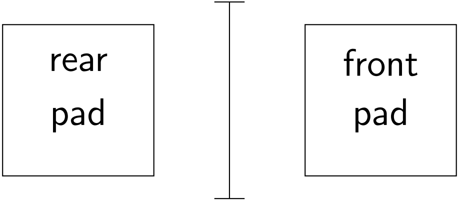
What is a computer I

- Computers are complicated, but we can construct idealized computational models to do analysis
- Finite automata are the idealized model that we will discuss
- Example: automatic door

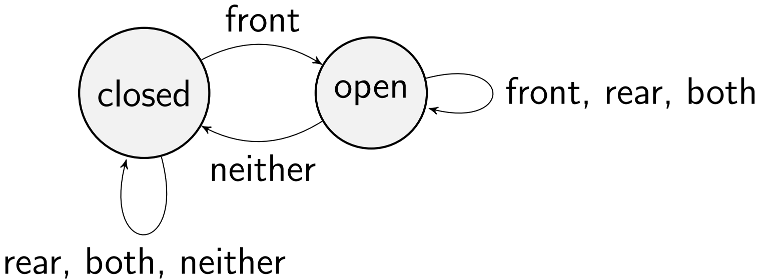
  Fig 1.1



What is a computer II

- Rules:

  When it moves, it cannot hit people
- We can use a simple graph to summarize all operations
- Fig 1.2



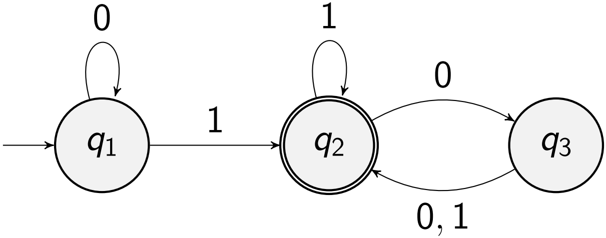
What is a computer III

- Single bit memory (open and closed)
- Automaton (single)

  automata (plural)

Examples of automata I

- Fig 1.4: a state diagram



states: q_1, q_2, q_3

start state: q_1

accept state q_2 (double circle)

Examples of automata II

- Example: running an input string 1101

$$q_1 \rightarrow q_2 \rightarrow q_2 \rightarrow q_3 \rightarrow q_2$$

  This string is accepted
- Example: running 10

$$q_1 \rightarrow q_2 \rightarrow q_3$$

  q_3 is not an accept state, so the string is rejected
- What are all strings accepted?
  We will say what this set is
  Unfortunately, it may not be always easy to know the set

Formal definition I

We formally define a state diagram as a 5-tuple $(Q, \Sigma, \delta, q_0, F)$

- Q: set of states. It is a finite set
- Σ: alphabet (i.e., set of characters in input string). It is a finite set
- $\delta : Q \times \Sigma \rightarrow Q$: transition function

  This is the most complicated part of the definition. We explain it by an example later
- $q_0 \in Q$: start state
- $F \subset Q$: set of accept states

Formal definition II

- For the example given above,

$$
\begin{aligned}
Q &= \{q_1, q_2, q_3\} \\
\Sigma &= \{0, 1\} \\
q_0 &= q_1 \\
F &= \{q_2\}
\end{aligned}
$$

- The δ function:

	0	1
q_1	q_1	q_2
q_2	q_3	q_2
q_3	q_2	q_2

Formal definition III

- Language of M: all strings accepted by M. Denoted as

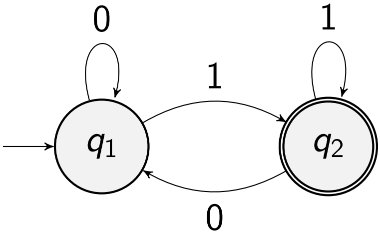
$$A = L(M)$$

- Figure 1.4:

$$A = \{w \mid w: \text{ at least one 1, even } \# \text{ 0 after the last 1}\}$$

Example 1.7 I

- Figure 1.8



- $M = (\{q_1, q_2\}, \{0, 1\}, \delta, q_1, \{q_2\})$
- What is $L(M)$? Anything ends with 1
- How to think about this ?

Example 1.7 II

- Before the last input character, we must be at q_1 or q_2. Then only if the last is 1 we can reach q_2 to get accepted