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: $q_1, q_2, q_3$

starting states: $q_1$

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
  \[ q_1 \rightarrow q_2 \rightarrow q_2 \rightarrow q_3 \]
  \[ q_3 \] 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
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
- $\sigma$: 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,

\[ Q = \{ q_1, q_2, q_3 \} \]
\[ \Sigma = \{ 0, 1 \} \]
\[ q_0 = q_1 \]
\[ F = \{ q_2 \} \]

The \( \delta \) function:

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>( q_1 )</td>
<td>( q_1 )</td>
<td>( q_2 )</td>
</tr>
<tr>
<td>( q_2 )</td>
<td>( q_3 )</td>
<td>( q_2 )</td>
</tr>
<tr>
<td>( q_3 )</td>
<td>( q_2 )</td>
<td>( q_2 )</td>
</tr>
</tbody>
</table>
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

- 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
Example 1.11

- Fig 1.12
Example 1.11 III

- \( L(M) = ? \)

\[ a \ldots a, b \ldots b \]

\[ \ldots \text{can be any string of} \ a \ \text{and} \ b \]
Example 1.13

- Figure 1.14
Example 1.13 II
Example 1.13 III

- $\Sigma = \{\langle \text{reset} \rangle, 0, 1, 2\}$

\[
L(M) = \ldots \langle \text{reset} \rangle \ldots \langle \text{reset} \rangle \ldots \\
= \{\text{sum of the last segment mod 3} = 0\}
\]

- Example:

  $10\langle \text{reset} \rangle 22\langle \text{reset} \rangle 012$
Running this string

\[
\begin{align*}
q_0 &\xrightarrow{1} q_1 \xrightarrow{0} q_1 \xrightarrow{\langle \text{reset} \rangle} q_0 \xrightarrow{2} q_2 \xrightarrow{2} q_1 \\
\langle \text{reset} \rangle &\xrightarrow{} q_0 \xrightarrow{0} q_0 \xrightarrow{1} q_1 \xrightarrow{2} q_0
\end{align*}
\]

Accepted.