1. DFA as a recognizer.

2. Generator

000 1 1 0 0 0 1

3. A different kind of generator:

4. Q. What would it mean/ how could we define an equivalent recognizer?

A. Non-determinism
A finite state machine

\[ M = (Q, \Sigma, \delta, q_0, F) \]

where finite
- \( Q \) is a set (states)
- \( q_0 \in Q \) start state
- \( \Sigma \) is a finite set (alphabet)
- \( F \subseteq Q \) final states (accepting states)

\[ \delta: Q \times (\Sigma \cup \{\varepsilon\}) \to 2^Q \] transition function

E.g. for fig 2.7 M

\[ \delta(q_1, 0) = \{q_1, 3\} \]
\[ \delta(q_1, 1) = \{q_1, 8\} \]
\[ \delta(q_2, 1) = \emptyset \]
**Figure 1.31**

**Figure 1.32**
$L = \{ \text{w in } \{a,b\}^* \mid \text{3rd letter from the right end of w is } "a" \}$
Definition: \( M \) ends in state \( q_f \) after reading \( w \in \Sigma^* \) if

(1) \( w = w_1 w_2 \ldots w_n \)

where \( w_i \in \Sigma \cup \{ \$$, \$$ \} \)

(2) \( \exists \) state \( r_0, r_1, r_2 \ldots r_n \in Q \)

such that:

a) \( r_0 = q_0 \)

b) \( r_1 \leq i \leq n \)

\( r_i \leq s(r_{i-1}, w_i) = r_{i+1} \)

c) \( r_n = q_f \)

Fact: \( q_f \) is unique

\( \) because \( s \) is a function, basically.
**Defn**

M accepts $w \in \Sigma^*$ if the state, $q$, reached by M after reading $w$ is an accepting state, i.e., $q \in F$.

**Defn**

The language recognized by M, $L(M) = \{ w \in \Sigma^+ \mid M \text{ accepts } w \}$.

**Note**

Every M recognizes exactly one language. Implicitly, it "recognizes" both strings it must accept and those it must reject.

**??**

Very important: note that "might be in a non-final state" does not imply "reject".