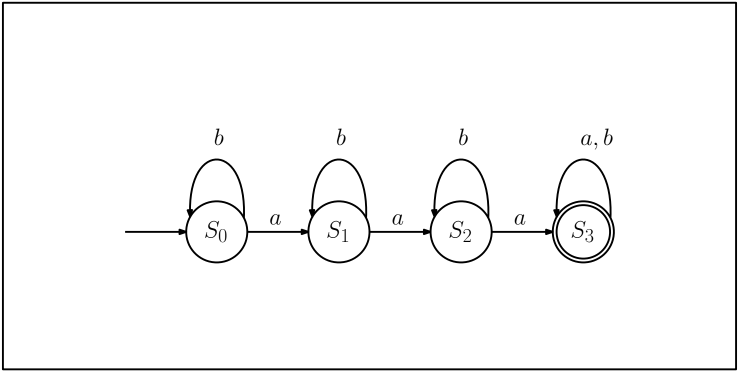
Deterministic Finite Automata

- A DFA looks somewhat like a flowchart.
- It reads input strings from left to right and accepts some strings, while rejecting others.
- It reads the characters of the input string one at a time, and has a state for each state.
- According to its structure, the DFA may change its state or remain in the same state after a particular step.
- It accepts a string when it finishes reading the string and reaches an accept state.

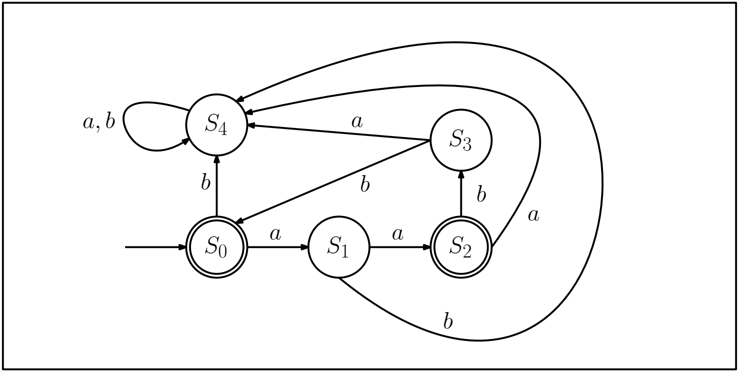
Deterministic Finite Automata

A DFA that accepts a string if and only if 'a' occurs at least 3 times, where input letters are from $\{a, b\}$.



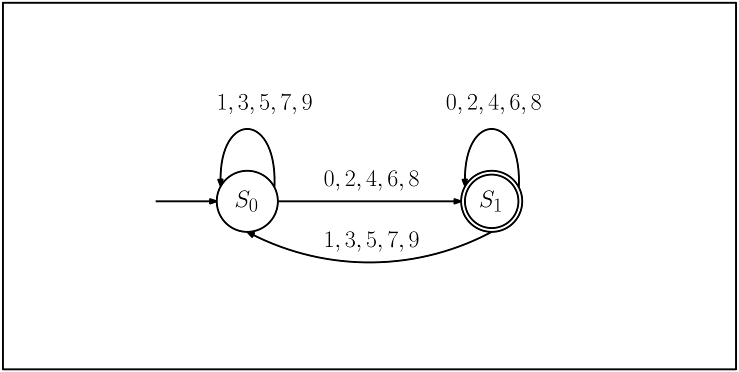
Deterministic Finite Automata

A DFA that accepts strings only of the form aa, aabb, aabbaa...
where input letters are from $\{a, b\}$.



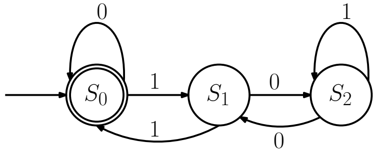
Deterministic Finite Automata

A DFA that accepts even numbers.



Deterministic Finite Automata

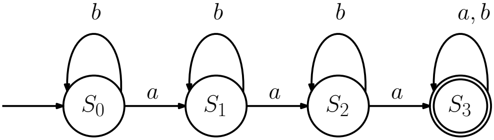
A DFA that accepts binary numbers divisible by 3.



Representing a DFA

A DFA can be represented by a table or a transition diagram.

	a	b
S_0	S_1	S_0
S_1	S_2	S_1
S_2	S_3	S_2
S_3F	S_3	S_3



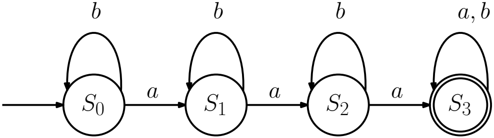
Representing a DFA

The table representing a DFA basically shows a function. This function takes the DFA from one state to another after reading a particular character. So we can call this function δ, where δ takes a state and a character, and outputs a state. In other words, $\delta : Q \times \Sigma \to Q$.

Representing a DFA

Here we have: $\delta(S_0, a) = S_1$, $\delta(S_0, b) = S_0$, $\delta(S_1, a) = S_2$, $\delta(S_1, b) = S_1$, $\delta(S_2, a) = S_3$, $\delta(S_2, b) = S_2$, $\delta(S_3, a) = S_3$, $\delta(S_3, b) = S_3$.

	a	b
S_0	S_1	S_0
S_1	S_2	S_1
S_2	S_3	S_2
S_3F	S_3	S_3



Definition

A *deterministic finite automaton* or DFA is a structure $M = (Q, \Sigma, \delta, s, F)$, where

- Q is a finite set, elements of Q are called *states*
- Σ is a finite set, the *input alphabet*
- $\delta : Q \times \Sigma \rightarrow Q$ is the *transition function*
- $s \in Q$ is the *start state*
- $F \subseteq Q$, the elements of F are called *accept states* or *final states*.

More on the transition function

A transition function is only a one step operation. However, we can define $\hat{\delta} : Q \times \Sigma^* \to Q$ from δ, as follows:

- $\hat{\delta}(q, \epsilon) = q$
- $\hat{\delta}(q, xa) = \delta(\hat{\delta}(q, x), a)$

Formally, a string x is said to be *accepted* by the DFA M if $\hat{\delta}(s, x) \in F$. Similarly, x is said to be *rejected* by M if $\hat{\delta}(s, x) \notin F$.

- The set or *language* accepted by M is the set of all strings accepted by M and is denoted as $L(M) = \{x \in \Sigma^* \mid \hat{\delta}(s, x) \in F\}$.
- A subset $A \subseteq \Sigma^*$ is said to be *regular* if $A = L(M)$ for some finite automaton M.