Deterministic Finite Automata (DFA)

- DFAs are easiest to present pictorially:

They are directed graphs whose nodes are *states* and whose arcs are labeled by one or more symbols from some alphabet $\Sigma$. Here $\Sigma$ is \{0,1\}. 
• One state is *initial* (denoted by a short incoming arrow), and several are *final/accepting* (denoted by a double circle). For every symbol $a \in \Sigma$ there is an arc labeled $a$ emanating from every state.

• Automata are string processing devices. The arc from $q_1$ to $q_2$ labeled 0 shows that when the automaton is in the state $q_1$ and receives the input symbol 0, its next state will be $q_2$. 
• Every path in the graph spells out a string over $S$. Moreover, for every string $w \in \Sigma^*$ there is a unique path in the graph labelled $w$. (Every string can be processed.) The set of all strings whose corresponding paths end in a final state is the *language of the automaton*.

• In our example, the language of the automaton consists of strings over $\{0, 1\}$ containing at least two occurrences of 0.
• Modify the automaton so that its language consists of strings containing \textit{exactly two} occurrences of 0.
Formal Definition

- A DFA is a quintuple $A = (Q, \Sigma, s, F, \delta)$, where
  - $Q$ is a set of states
  - $\Sigma$ is the alphabet of input symbols
  - $s$ is an element of $Q$ --- the initial state
  - $F$ is a subset of $Q$ --- the set of final states
  - $\delta: Q \times \Sigma \rightarrow Q$ is the transition function
Example

- In our example,
  - $Q=\{q_0, q_1, q_2\}$,
  - $\Sigma=\{0, 1\}$,
  - $s=q_0$,
  - $F=\{q_2\}$,
  - and
  - $\delta$ is given by 6 equalities
- $\delta(q_0, 0)=q_1$,
- $\delta(q_0, 1)=q_0$,
- $\delta(q_2, 1)=q_2$
- ...

Diagram:

- $q_0$ to $q_1$ with an arrow labeled 0
- $q_1$ to $q_0$ with an arrow labeled 1
- $q_1$ to $q_2$ with an arrow labeled 0
- $q_2$ with a double circle

0, 1
• All the information presenting a DFA can be given by a single thing -- its *transition table*:

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Q_0$</td>
<td>$Q_1$</td>
<td>$Q_0$</td>
</tr>
<tr>
<td>$Q_1$</td>
<td>$Q_2$</td>
<td>$Q_1$</td>
</tr>
<tr>
<td>$*Q_2$</td>
<td>$Q_2$</td>
<td>$Q_2$</td>
</tr>
</tbody>
</table>

• The initial and final states are denoted by $\rightarrow$ and * respectively.
Extension of $\delta$ to Strings

• Given a state $q$ and a string $w$, there is a unique path labeled $w$ that starts at $q$ (why?). The endpoint of that path is denoted $\delta(q, w)$

• Formally, the function $\delta : Q \times \Sigma^* \rightarrow Q$
• is defined recursively:

\[
\begin{align*}
- \delta(q, \varepsilon) &= q \\
- \delta(q, ua) &= \delta(\delta(q, u), a)
\end{align*}
\]

• Note that $\delta(q, a) = \delta(q, a)$ for every $a \in \Sigma$;
• so $\delta$ does extend $\delta$. 
Example trace

• Diagrams (when available) make it very easy to compute $\delta(q, w)$ --- just trace the path labeled $w$ starting at $q$.

• E.g. trace 101 on the diagram below starting at $q_1$.

![Diagram](image-url)
• Implementation and precise arguments need the formal definition.

\[ \delta(q_1, 101) = \delta(\delta(q_1, 10), 1) \]
\[ = \delta(\delta(\delta(q_1, 1), 0), 1) \]
\[ = \delta(\delta(q_1, 0), 1) \]
\[ = \delta(q_2, 1) \]
\[ = q_2 \]
Language of accepted strings

A DFA \( = (Q, \Sigma, s, F, \delta) \), accepts a string \( w \) iff \( \delta(s, w) \in F \)

The language of the automaton A is

\[ L(A) = \{ w \mid A \text{ accepts } w \} . \]

More formally

\[ L(A) = \{ w \mid \delta(\text{Start}(A), w) \in \text{Final}(A) \} \]

Example:

Find a DFA whose language is the set of all strings over \( \{a, b, c\} \) that contain \( aaaa \) as a substring.
DFA’s as Programs

data DFA q s = DFA { states :: [q],
symbols :: [s],
delta :: q -> s -> q,
start :: q,
final :: [q]}
Transition function

\[\text{trans} :: (q \to s \to q) \to q \to [s] \to q\]
\[\text{trans } d \text{ q } [] = q\]
\[\text{trans } d \text{ q } (s:ss) = \text{trans } d \text{ (d q s) ss}\]

\[\text{accept} :: (Eq q) \Rightarrow \text{DFA } q \text{ s } \to [s] \to \text{Bool}\]
\[\text{accept}\]
\[\quad m@(\text{DFA}\{\delta = d, \text{start} = q0, \text{final} = f\})\ w\]
\[\quad = \text{elem } (\text{trans } d \ q0 \ w) \ f\]
An Example

ma = DFA { states = [0,1,2],
          symbols = [0,1],
          delta = \p a ->
                   (2*p+a) `mod` 3,
          start = 0,
          final = [2] }