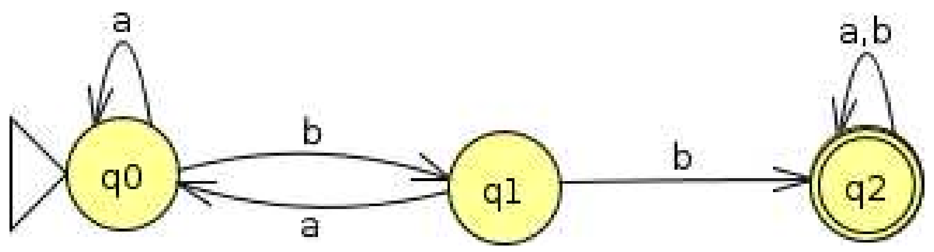
Finite Automata



The elements of Σ are the "moves" in the graph
A word in Σ^* is a sequence of moves
Start state is indicated by '$>$'; accepting state by double circle
There can be several accepting states

The word $abba$ is accepted, but $baab$ is not accepted (rejected)

$M = \langle Q, \Sigma, \delta, q_0, F \rangle$ with

$Q = \{q_0, q_1, q_2\}$, $\Sigma = \{a, b\}$, $F = \{q_2\}$ and δ given by

δ	q_0	q_1	$\boldsymbol{q}_2$
a	q_0	q_0	$\boldsymbol{q}_2$
b	q_1	$\boldsymbol{q}_2$	$\boldsymbol{q}_2$

M is a DFA over Σ if $M = (Q, \Sigma, q_0, \delta, F)$ with

Q	is a finite set of states
Σ	is a finite alphabet
$q_0 \in Q$	is the initial state
$F \subseteq Q$	is a finite set of final states
$\delta : Q \times \Sigma \rightarrow Q$	is the transition function

Reading function $\hat{\delta} : Q \times \Sigma^* \rightarrow Q$ (multi-step transition)

$$\begin{aligned} \hat{\delta}(q, \lambda) &= q \\ [\hat{\delta}(q, a) &= \delta(q, a)] \\ \hat{\delta}(q, wa) &= \delta(\hat{\delta}(q, w), a) \end{aligned}$$

The language accepted by M, notation $L(M)$, is:

$$L(M) = \{w \in \Sigma^* \mid \hat{\delta}(q_0, w) \in F\}$$

Computation for $\hat{\delta}(q, w)$ in the example $w = abba$:

$$
\begin{array}{llll}
[q, abba] & \vdash & [\delta(q,a), bba] & \hat{\delta}(q,a) = \delta(q,a) \\
& \vdash & [\delta(\delta(q,a),b), ba] & \hat{\delta}(q,ab) = \delta(\delta(q,a),b) \\
& \vdash & [\delta(\delta(\delta(q,a),b),b), a] & \hat{\delta}(q,abb) = \delta(\delta(\delta(q,a),b),b) \\
& \vdash & [\delta(\delta(\delta(\delta(q,a),b),b),a), \lambda] & \hat{\delta}(q,abba) = \delta(\delta(\delta(\delta(q,a),b),b),a)
\end{array}
$$

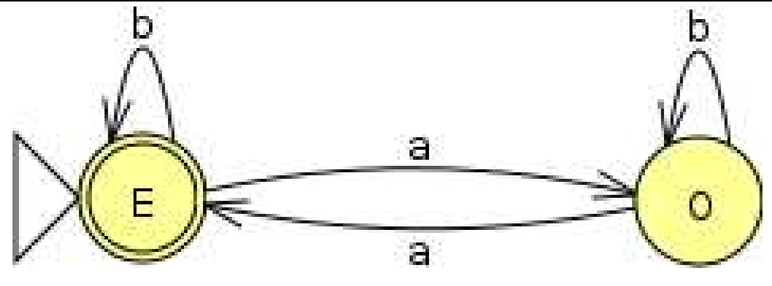
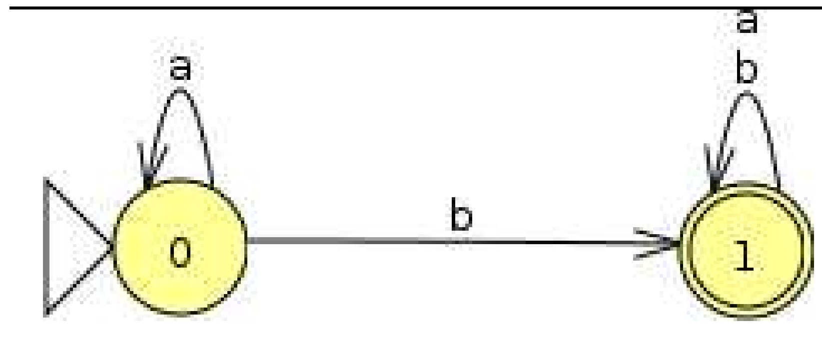
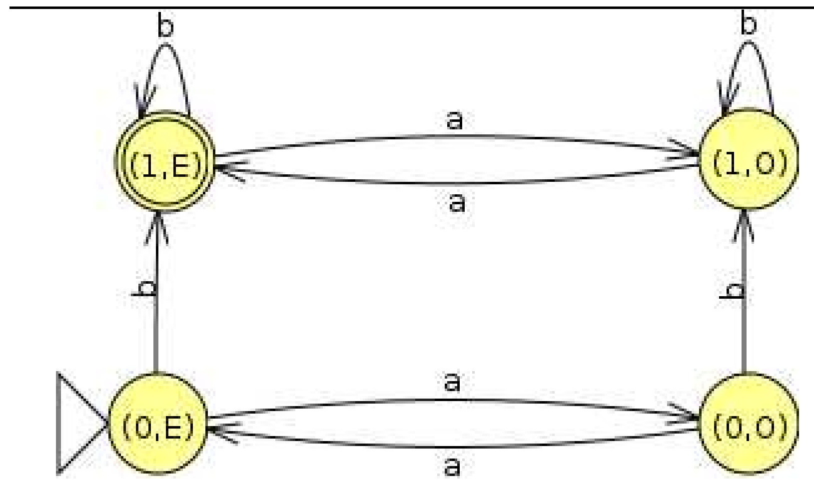
This computation corresponds to an equivalent definition of $\hat{\delta}$:

$$
\begin{array}{rcl}
\hat{\delta}(q,\lambda) & = & q \\
\hat{\delta}(q,a) & = & \delta(q,a) \\
\hat{\delta}(q,aw) & = & \hat{\delta}(\delta(q,a),w)
\end{array}
$$

Example transition table for δ with $Q = \{0,1,2,3,4\}$, $\Sigma = \{a,b\}$, $q_0 = 0$, and $F = \{4\}$

δ	a	b
0	1	0
1	1	2
2	1	3
3	4	0
4	4	4

We have $\hat{\delta}(0, abba) = 4 \in F$ and $[0, abba] \vdash^* [4, \lambda]$, hence $abba \in L(M)$
Similarly $\hat{\delta}(0, baba) = 1 \notin F$; so even if $[0, baba] \vdash^* [1, \lambda]$ we have $baba \notin L(M)$.
Even if $\hat{\delta}(1, bba) = 4 \in F$ and $[1, bba] \vdash^* [4, \lambda]$ we have $bba \notin L(M)$.

M	$L(M)$
	$L_1 = \{w \mid \#_a(w) \text{ is even}\}$
	$L_2 = \{w \mid \#_b(w) \geq 1\}$
	$L_1 \cap L_2 =$ $\{s \mid \#_a(w) \text{ is even and } \#_b(w) \geq 1\}$

Given two DFAs over the same Σ

$$\begin{aligned} M_1 &= \langle Q_1, \Sigma, \delta_1, q_{01}, F_1 \rangle \\ M_2 &= \langle Q_2, \Sigma, \delta_2, q_{02}, F_2 \rangle \end{aligned}$$

Define

$$M_1 \times M_2 = \langle Q_1 \times Q_2, \Sigma, \delta, q_0, F \rangle$$

with

$$\begin{aligned} q_0 &:= \langle q_{01}, q_{02} \rangle \\ \delta(\langle q_1, q_2 \rangle, a) &:= \langle \delta_1(q_1, a), \delta_2(q_2, a) \rangle \end{aligned}$$

Then with

$$F := F_1 \times F_2 := \{\langle q_1, q_2 \rangle \mid q_1 \in F_1 \text{ and } q_2 \in F_2\}$$

we have

$$L(M_1 \times M_2) = L(M_1) \cap L(M_2)$$

Proposition Closure under complement

If L is accepted by some DFA, then so is $\overline{L} = \Sigma^* - L$.

Proof. Suppose that L is accepted by $M = \langle Q, \Sigma, \delta, q_0, F \rangle$.

Then $\overline{L}$ is accepted by $M = \langle Q, \Sigma, \delta, q_0, \overline{F} \rangle$. ■

Proposition Closure under intersection and union

If L_1, and L_2 are accepted by some DFA, then so are $L_1 \cap L_2$ and $L_1 \cup L_2$.

Proof. For the intersection, this follows from the product construction on the previous slide.

For the union, this can be seen by the product construction, taking a different F (which one?) or by noticing that $L_1 \cup L_2 = \overline{\overline{L_1} \cap \overline{L_2}}$. ■

Theorem The languages accepted by DFAs are exactly the regular languages

We will prove this in this and teh next lecture by

1. If $L = L(M)$, for some DFA M, then there is a regular expression e such that $L = L(e)$ (this lecture)
2. If $L = L(e)$, for some regular expression e, then there is a non-deterministic finite automaton (NFA) M such that $L = L(M)$. (next lecture)
3. For every NFA M, there is a DFA M' such that $L(M) = L(M')$ (next lecture)

Given the DFA $M = \langle Q, \Sigma, \delta, q_0, F\rangle$, we construct for every $q_i \in Q$ a regular expression r_i such that r_i describes the "language accepted from state q_i":

$$L(r_i) = \{w \in \Sigma^* \mid \hat{\delta}(q_i, w) \in F\}$$

Let $\Sigma = \{a, b\}$, $Q = \{q_0, q_1, q_2, q_3\}$. Write, for $i \in \{0, 1, 2, 3\}$

$$\begin{array}{rcl} r_i & = & ar_j \cup br_k \ (\cup \lambda) \qquad \text{with } \cup \lambda \text{ only if } q_i \in F \\ & & \text{if } \delta(q_i, a) = q_j, \delta(q_j, b) = q_k \end{array}$$

This gives a set of equations, which can be solved by starting from the last as follows:

- write the last equation as $r_i = wr_i + v$
- substitute in all other equations $r_i := w^*v$
- remove the last equation

This ends with a regulare expression $r_0 := e$.

Proposition $L(M) = L(e)$