Automata and Grammars (BIE-AAG)

3. Operations on finite automata

Jan Holub

Department of Theoretical Computer Science
Faculty of Information Technology
Czech Technical University in Prague

Relation between DFA and NFA

Definition

Finite automata M_1 and M_2 are called *equivalent* if they accept the same language, i.e. $L(M_1) = L(M_2)$.

Theorem

For every nondeterministic finite automaton M there exists an equivalent deterministic finite automaton M'.

Determinization of NFA

Algorithm Determinization of NFA (subset construction)

Input: NFA $M = (Q, \Sigma, \delta, q_0, F)$.

Output: DFA M' such that $L(M) = L(M')$.

$Q' \leftarrow \{\{q_0\}\}$
for $\forall q' \in Q'$ **do**
    $\delta'(q', a) \leftarrow \bigcup_{p \in q'} \delta(p, a)$, $\forall a \in \Sigma$
    $Q' \leftarrow Q' \cup \{\delta'(q', a) : a \in \Sigma\}$
end for
$q'_0 \leftarrow \{q_0\}$
$F' \leftarrow \{q' : q' \in Q', q' \cap F \neq \emptyset\}$
$M' \leftarrow (Q', \Sigma, \delta', q'_0, F')$
return M'

Determinization of NFA

Example

NFA M:

	δ_M	0	1
$\rightarrow$	q	$\{q, q_0\}$	$\{q, q_1\}$
	q_0	$\{q_0, q_f\}$	$\{q_0\}$
	q_1	$\{q_1\}$	$\{q_1, q_f\}$
$\leftarrow$	q_f	$\emptyset$	$\emptyset$

DFA M':

	$\delta_{M'}$	0	1
$\rightarrow$	$\{q\}$	$\{q, q_0\}$	$\{q, q_1\}$
	$\{q, q_0\}$	$\{q, q_0, q_f\}$	$\{q, q_0, q_1\}$
	$\{q, q_1\}$	$\{q, q_0, q_1\}$	$\{q, q_1, q_f\}$
$\leftarrow$	$\{q, q_0, q_f\}$	$\{q, q_0, q_f\}$	$\{q, q_0, q_1\}$
$\leftarrow$	$\{q, q_1, q_f\}$	$\{q, q_0, q_1\}$	$\{q, q_1, q_f\}$
	$\{q, q_0, q_1\}$	$\{q, q_0, q_1, q_f\}$	$\{q, q_0, q_1, q_f\}$
$\leftarrow$	$\{q, q_0, q_1, q_f\}$	$\{q, q_0, q_1, q_f\}$	$\{q, q_0, q_1, q_f\}$

Determinization of NFA

- How big can the resulting automaton be?

$|Q_{NFA}| = N$

$|Q_{DFA}| = 2^N$

Determinization of NFA

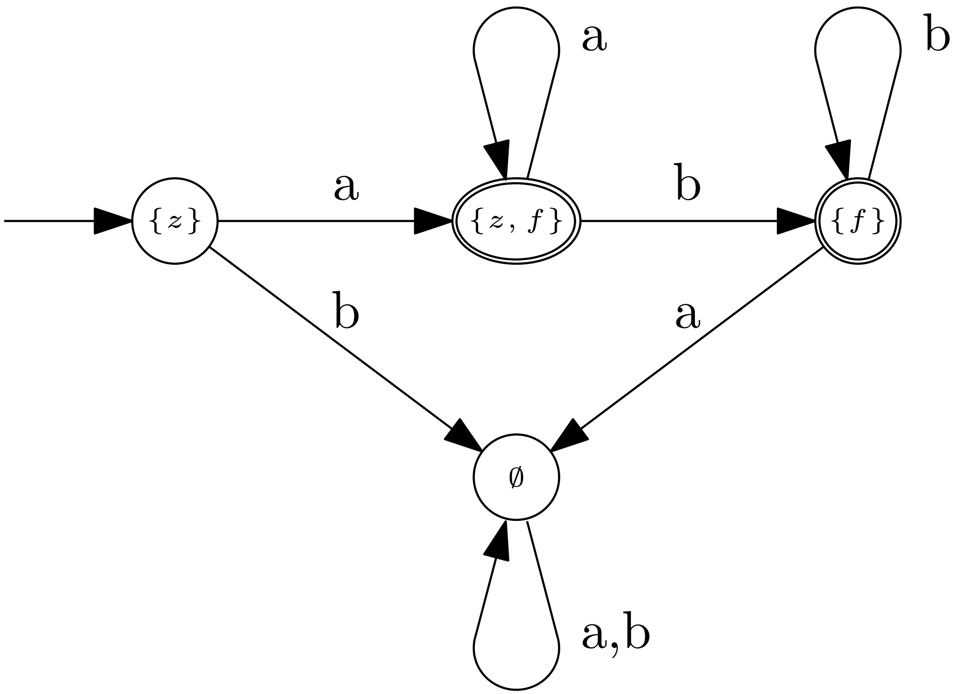
Example

NFA $M = (\{z, f\}, \{a, b\}, \delta, z, \{f\})$, where δ:

	δ	a	b
$\rightarrow$	z	$\{z, f\}$	$\emptyset$
$\leftarrow$	f	$\emptyset$	$\{f\}$

DFA $M' = (\{\{z\}, \{z, f\}, \{f\}, \emptyset\}, \{a, b\}, \delta', \{z\}, \{\{z, f\}, \{f\}\})$, where δ':

	δ'	a	b
$\rightarrow$	$\{z\}$	$\{z, f\}$	$\emptyset$
$\leftarrow$	$\{z, f\}$	$\{z, f\}$	$\{f\}$
$\leftarrow$	$\{f\}$	$\emptyset$	$\{f\}$
	$\emptyset$	$\emptyset$	$\emptyset$



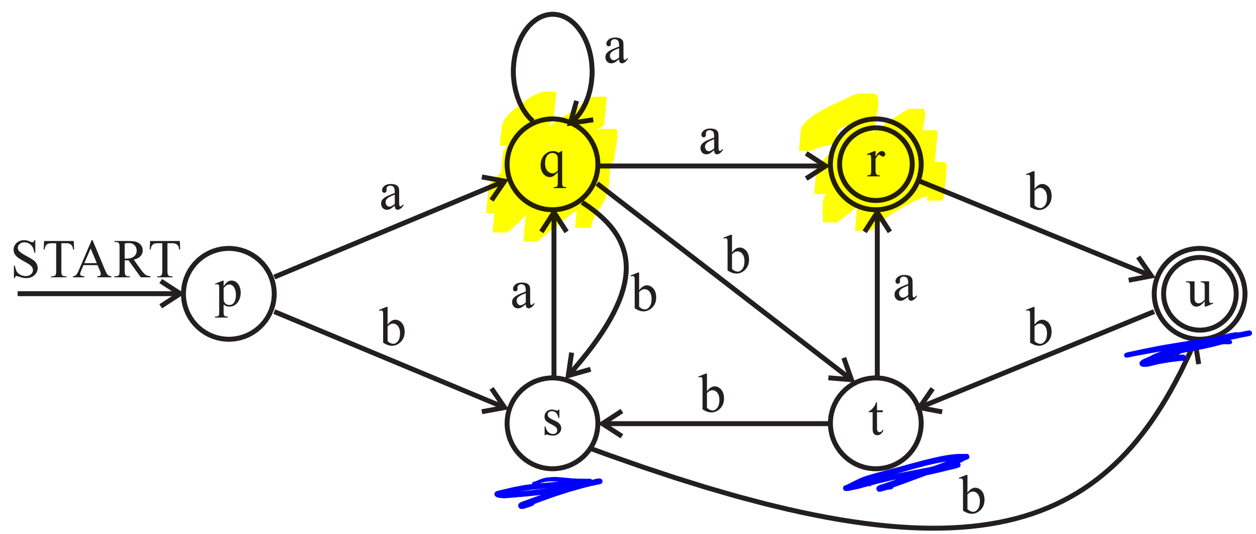
Homogeneous finite automaton

Group of STATes
All STATes / That Recieve a TRNsi7ion FROM A

Definition (Set of target states)
$M = (Q, \Sigma, \delta, q_0, F)$. For any $a \in \Sigma$ we define the set $Q(a) \subseteq Q$ of target states as: $Q(a) = \{q : q \in \delta(p, a), p, q \in Q\}$.

determinated action

Definition (Homogeneous finite automaton)
$M = (Q, \Sigma, \delta, q_0, F)$ and $Q(a)$ are the sets of target states $\forall a \in \Sigma$. If for all pairs of symbols $a, b \in \Sigma$, $a \neq b$ it holds that $Q(a) \cap Q(b) = \emptyset$, then the automaton M is called *homogeneous*.



$$\begin{aligned} Q(a) &= \{q, r\} \\ Q(b) &= \{s, t, u\} \end{aligned}$$

Homogeneous finite automaton

Theorem
The set of states of a homogeneous automaton $M = (Q, \Sigma, \delta, q_0, F)$ without unreachable states is partitioned as follows:

$$Q = \biguplus_{a \in \Sigma \cup \{\varepsilon\}} Q(a), \quad \text{where} \quad Q(\varepsilon) = \{q_0\} \setminus \bigcup_{a \in \Sigma} Q(a)$$

Union of $\bigcup$
mutually disjointed

Homogeneous finite automaton

Theorem

Let $M = (Q, \Sigma, \delta, q_0, F)$ be a homogeneous NFA. Then the number of states of the equivalent DFA $M' = (Q', \Sigma, \delta', q_0', F')$ gained by the standard determinization (subset construction) algorithm is bounded by the following equality:

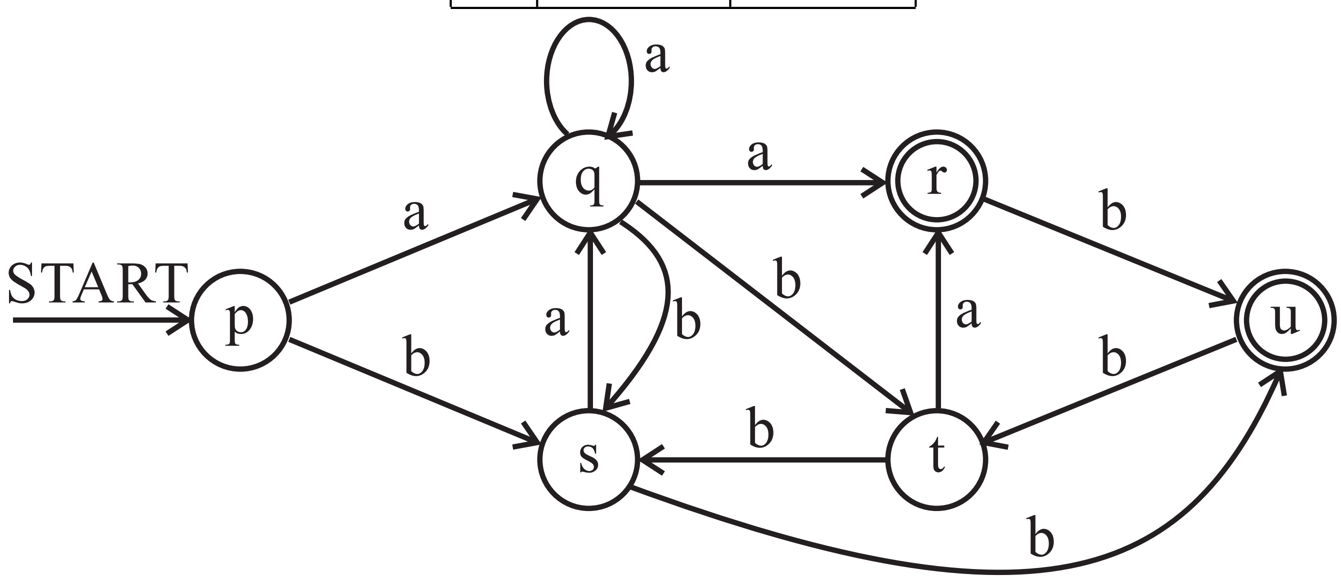
$$|Q'| \leq \sum_{a \in \Sigma} (2^{|Q(a)|}) - |\Sigma| + 1.$$

Homogeneous finite automaton

Example

Given homogeneous NFA $M = (\{p, q, r, s, t, u\}, \{a, b\}, \delta, p, \{r, u\})$, where δ:

	a	b
p	$\{q\}$	$\{s\}$
q	$\{q, r\}$	$\{s, t\}$
r		$\{u\}$
s	$\{q\}$	$\{u\}$
t	$\{r\}$	$\{s\}$
u		$\{t\}$



Homogeneous finite automaton

Example (continued)

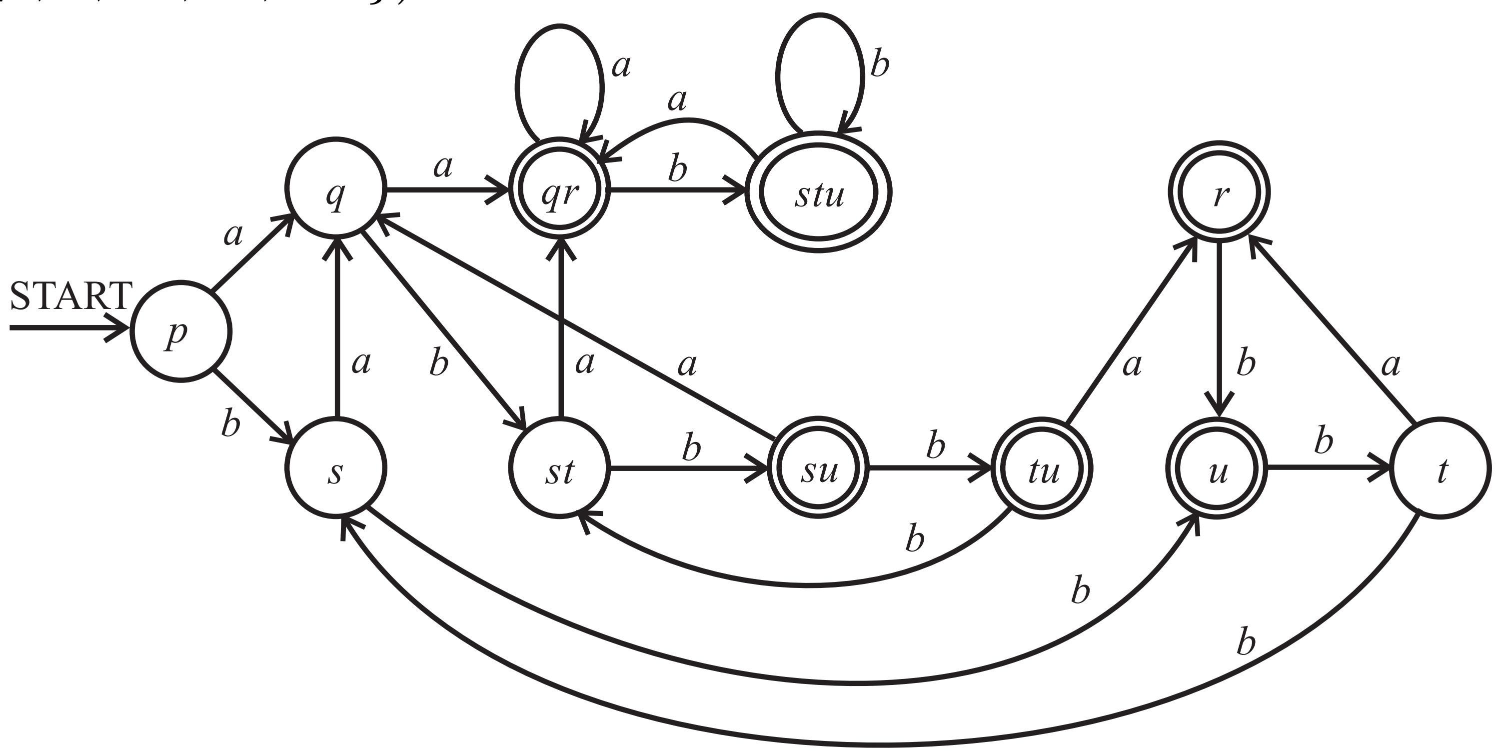
$Q(a) = \{q, r\},\ Q(b) = \{s, t, u\},\ Q(\varepsilon) = \{p\}$

$Q(a) \cap Q(b) = \emptyset$, therefore DFA for M is $M' = (Q', \Sigma, \delta', q_0, F')$, where

$|Q'| \leq 2^{|Q(a)|} + 2^{|Q(b)|} - |\Sigma| + 1 = 2^2 + 2^3 - 2 + 1 = 4 + 8 - 2 + 1 = 11$

Equivalent DFA is $M' = (\{p, q, qr, s, t, st, stu, u, su, t, tu\},$
$\{a, b\}, \delta, p, \{r, qr, u, su, tu, stu\})$, where δ:

	a	b
p	q	s
q	qr	st
s	q	u
qr	qr	stu
st	qr	su
u		t
stu	qr	stu
su	q	tu
t	r	s
tu	r	st
r		u



FA and union of languages

Algorithm NFA for a union of languages – ε-transitions

Input: NFA $M_1 = (Q_1, \Sigma, \delta_1, q_{01}, F_1)$, $M_2 = (Q_2, \Sigma, \delta_2, q_{02}, F_2)$,
$Q_1 \cap Q_2 = \emptyset$.

Output: NFA M, $L(M) = L(M_1) \cup L(M_2)$.

$Q \leftarrow Q_1 \cup Q_2 \cup \{q_0\},\ q_0 \notin Q_1 \cup Q_2$
$\delta(q_0, \varepsilon) \leftarrow \{q_{01}, q_{02}\}$
$\delta(q, a) \leftarrow \delta_1(q, a)$, $\forall q \in Q_1$, $\forall a \in \Sigma$
$\delta(q, a) \leftarrow \delta_2(q, a)$, $\forall q \in Q_2$, $\forall a \in \Sigma$
$F \leftarrow F_1 \cup F_2$
$M \leftarrow (Q, \Sigma, \delta, q_0, F)$
return M

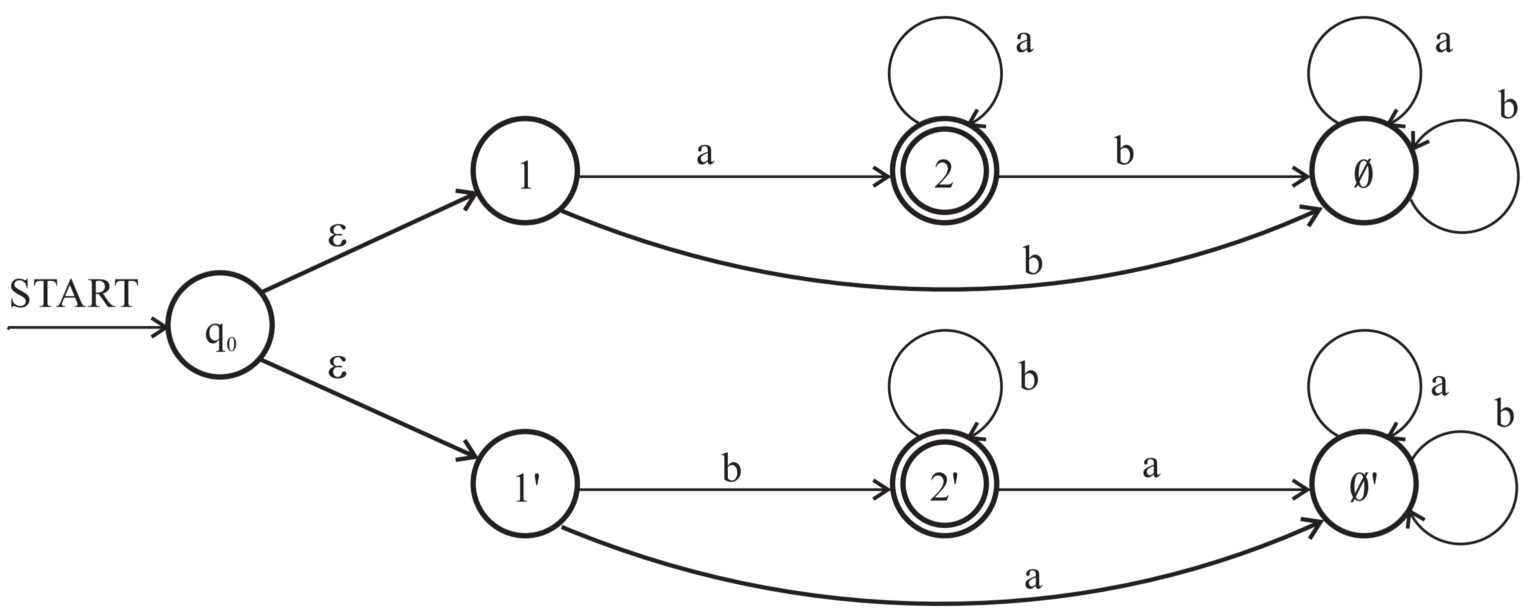
FA and union of languages

Example

$M_1 = (\{1, 2, \emptyset\}, \{a, b\}, \delta_1, 1, \{2\}),\ L(M_1) = \{a\}^+$
$M_2 = (\{1', 2', \emptyset'\}, \{a, b\}, \delta_2, 1', \{2'\}),\ L(M_2) = \{b\}^+$

	δ_1	a	b
$\rightarrow$	1	$\{2\}$	$\{\emptyset\}$
$\leftarrow$	2	$\{2\}$	$\{\emptyset\}$
	$\emptyset$	$\{\emptyset\}$	$\{\emptyset\}$

	δ_2	a	b
$\rightarrow$	$1'$	$\{\emptyset'\}$	$\{2'\}$
$\leftarrow$	$2'$	$\{\emptyset'\}$	$\{2'\}$
	$\emptyset'$	$\{\emptyset'\}$	$\{\emptyset'\}$



FA and union of languages

Definition (Total NFA)
NFA $M = (Q, \Sigma, \delta, q_0, F)$ is called *total* if the mapping $\delta(q, a) \neq \emptyset, \forall q \in Q, a \in \Sigma$.

Algorithm NFA for a union of languages – parallel run

Input: Total NFAs $M_1 = (Q_1, \Sigma, \delta_1, q_{01}, F_1)$ and $M_2 = (Q_2, \Sigma, \delta_2, q_{02}, F_2)$.
Output: NFA M accepting $L(M) = L(M_1) \cup L(M_2)$.

1: $M \leftarrow (Q_1 \times Q_2, \Sigma, \delta, (q_{01}, q_{02}), (F_1 \times Q_2) \cup (Q_1 \times F_2))$, where $\delta((q_1, q_2), a) \leftarrow \{(q_1', q_2') : q_1' \in \delta_1(q_1, a), q_2' \in \delta_2(q_2, a)\}, \forall (q_1, q_2) \in Q_1 \times Q_2, \forall a \in \Sigma$

FA and union of languages

Example

$M_1 = (\{1, 2, \emptyset\}, \{a, b\}, \delta_1, 1, \{2\}),\ L(M_1) = \{a\}^+$
$M_2 = (\{1', 2', \emptyset'\}, \{a, b\}, \delta_2, 1', \{2'\}),\ L(M_2) = \{b\}^+$

	δ_1	a	b
$\rightarrow$	1	$\{2\}$	$\{\emptyset\}$
$\leftarrow$	2	$\{2\}$	$\{\emptyset\}$
	$\emptyset$	$\{\emptyset\}$	$\{\emptyset\}$

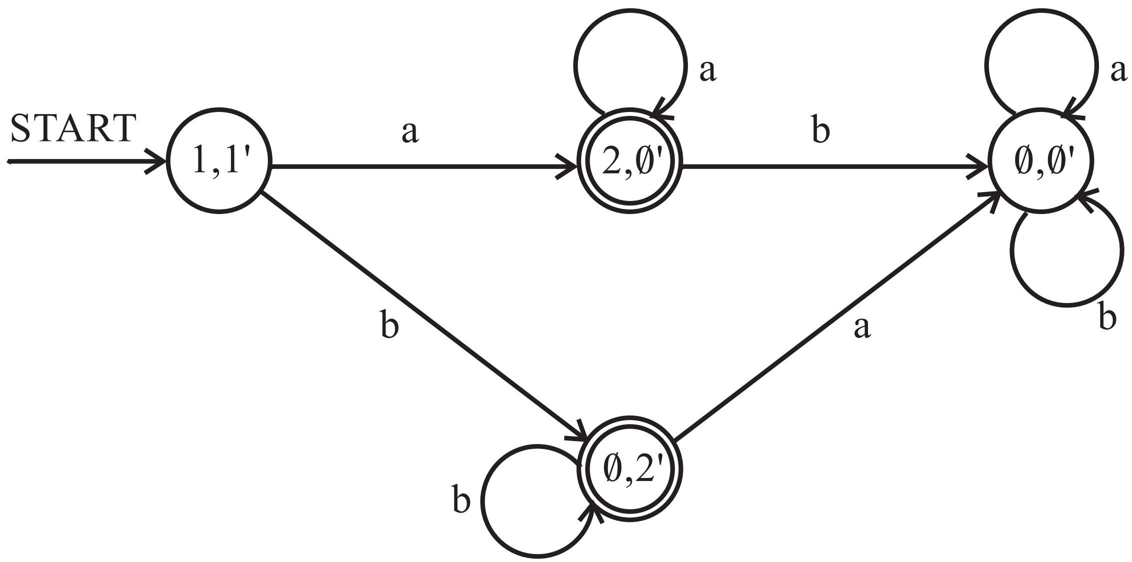
	δ_2	a	b
$\rightarrow$	$1'$	$\{\emptyset'\}$	$\{2'\}$
$\leftarrow$	$2'$	$\{\emptyset'\}$	$\{2'\}$
	$\emptyset'$	$\{\emptyset'\}$	$\{\emptyset'\}$

$L(M) = \{a\}^+ \cup \{b\}^+$
$M = (\{(1, 1'), (2, \emptyset'), (\emptyset, 2'), (\emptyset, \emptyset')\}, \{a, b\}, \delta, (1, 1'), \{(2, \emptyset'), (\emptyset, 2')\})$

	δ	a	b
$\rightarrow$	$(1, 1')$	$\{(2, \emptyset')\}$	$\{(\emptyset, 2')\}$
$\leftarrow$	$(2, \emptyset')$	$\{(2, \emptyset')\}$	$\{(\emptyset, \emptyset')\}$
$\leftarrow$	$(\emptyset, 2')$	$\{(\emptyset, \emptyset')\}$	$\{(\emptyset, 2')\}$
	$(\emptyset, \emptyset')$	$\{(\emptyset, \emptyset')\}$	$\{(\emptyset, \emptyset')\}$
	$\vdots$	$\vdots$	$\vdots$

FA and union of languages

Example (continued)



FA and intersection of languages

Algorithm NFA for the intersection of languages – parallel run

Input: NFA $M_1 = (Q_1, \Sigma, \delta_1, q_{01}, F_1), M_2 = (Q_2, \Sigma, \delta_2, q_{02}, F_2)$.
Output: NFA M accepting $L(M) = L(M_1) \cap L(M_2)$

1: $M \leftarrow (Q_1 \times Q_2, \Sigma, \delta, (q_{01}, q_{02}),\ F_1 \times F_2)$, where $\delta((q_1, q_2), a) \leftarrow \{(q'_1, q'_2) : q'_1 \in \delta_1(q_1, a), q'_2 \in \delta_2(q_2, a)\}, \forall (q_1, q_2) \in Q_1 \times Q_2, \forall a \in \Sigma$

FA and intersection of languages

Example

M: $L(M) = \{w : w \in \{a,b\}^*, aba$ is a prefix of w, bab is a suffix of $w\}$.
M_1 accepts strings that begin with prefix aba,
$M_1 = (\{1,2,3,4,\emptyset\}, \{a,b\}, \delta_1, 1, \{4\})$
M_2 accepts strings that end with suffix bab,
$M_2 = (\{1',2',3',4'\}, \{a,b\}, \delta_2, 1', \{4'\})$

	δ_1	a	b
$\rightarrow$	1	$\{2\}$	$\{\emptyset\}$
	2	$\{\emptyset\}$	$\{3\}$
	3	$\{4\}$	$\{\emptyset\}$
$\leftarrow$	4	$\{4\}$	$\{4\}$
	$\emptyset$	$\{\emptyset\}$	$\{\emptyset\}$

	δ_2	a	b
$\rightarrow$	$1'$	$\{1'\}$	$\{2'\}$
	$2'$	$\{3'\}$	$\{2'\}$
	$3'$	$\{1'\}$	$\{4'\}$
$\leftarrow$	$4'$	$\{3'\}$	$\{2'\}$

FA and intersection of languages

Example (continued)

$M = (\{(1,1'),(2,1'),(3,2'),(4,1'),(4,2'),(4,3'),(4,4'),(\emptyset,1'),$
$(\emptyset,2'),(\emptyset,3'),(\emptyset,4')\},\{a,b\},\delta,(1,1'),\{(4,4')\})$

	δ	a	b
$\rightarrow$	$(1,1')$	$\{(2,1')\}$	$\{(\emptyset,2')\}$
	$(2,1')$	$\{(\emptyset,1')\}$	$\{(3,2')\}$
	$(\emptyset,1')$	$\{(\emptyset,1')\}$	$\{(\emptyset,2')\}$
	$(\emptyset,2')$	$\{(\emptyset,3')\}$	$\{(\emptyset,2')\}$
	$(\emptyset,3')$	$\{(\emptyset,1')\}$	$\{(\emptyset,4')\}$
	$(\emptyset,4')$	$\{(\emptyset,3')\}$	$\{(\emptyset,2')\}$
	$(3,2')$	$\{(4,3')\}$	$\{(\emptyset,2')\}$
	$(4,3')$	$\{(4,1')\}$	$\{(4,4')\}$
	$(4,1')$	$\{(4,1')\}$	$\{(4,2')\}$
	$(4,2')$	$\{(4,3')\}$	$\{(4,2')\}$
$\leftarrow$	$(4,4')$	$\{(4,3')\}$	$\{(4,2')\}$
	$\vdots$	$\vdots$	$\vdots$

FA and intersection of languages

Algorithm NFA for intersection of languages – accessible states only

Input: NFA $M_1 = (Q_1, \Sigma, \delta_1, q_{01}, F_1)$, $M_2 = (Q_2, \Sigma, \delta_2, q_{02}, F_2)$.
Output: NFA M, $L(M) = L(M_1) \cap L(M_2)$.

$Q \leftarrow \{(q_{01}, q_{02})\}$
for $\forall q = (q_1, q_2) \in Q$ **do**
    $\delta((q_1, q_2), a) \leftarrow \{(q_1', q_2') : q_1' \in \delta_1(q_1, a), q_2' \in \delta_2(q_2, a)\}$, $\forall a \in \Sigma$
    $Q \leftarrow Q \cup \delta((q_1, q_2), a)$, $\forall a \in \Sigma$
end for
$q_0 \leftarrow (q_{01}, q_{02})$
$F \leftarrow Q \cap (F_1 \times F_2)$
$M \leftarrow (Q, \Sigma, \delta, q_0, F)$
return M

FA and complement of language

Algorithm DFA for complement of language

Input: Total DFA $M = (Q, \Sigma, \delta, q_0, F)$.
Output: DFA M', $L(M') = \Sigma^* \setminus L(M)$.

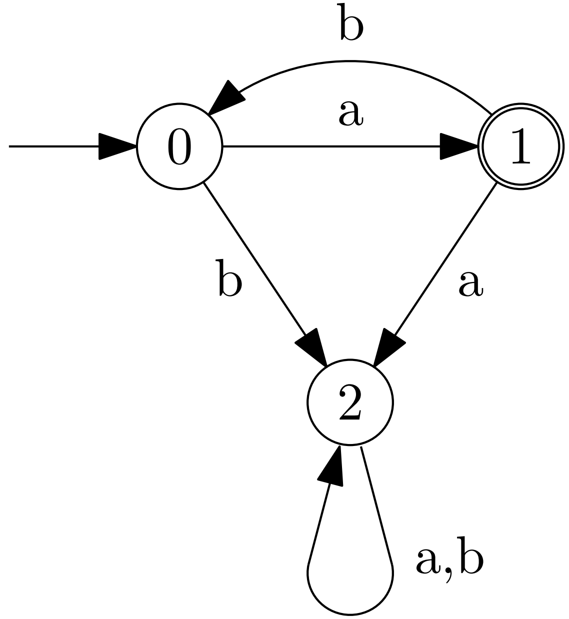
1: $M' \leftarrow (Q, \Sigma, \delta, q_0, Q \setminus F)$
2: **return** M

Only change Final to Non-final states and Vice Versa

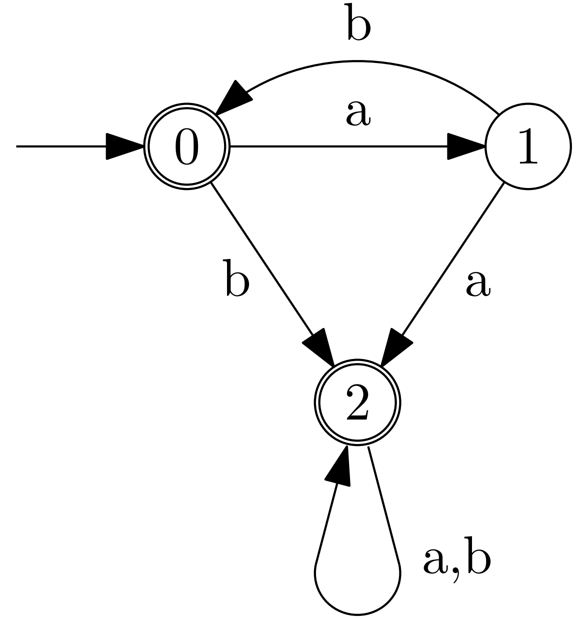
Example

DFA M that accepts all strings of the form $a(ba)^*$.

M:



M':



FA and product of languages

Algorithm NFA for the product of languages – ε-transitions

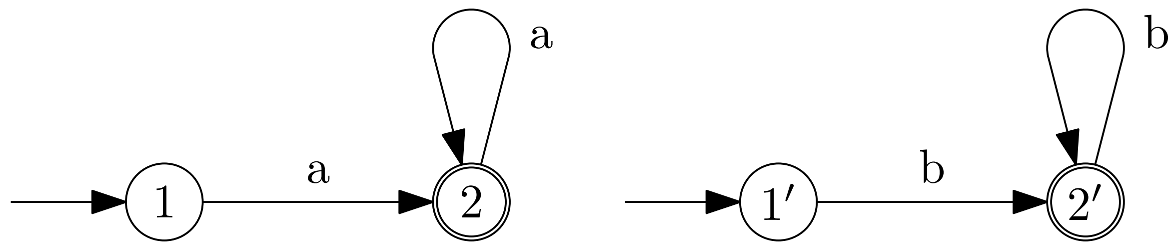
Input: NFA $M_1 = (Q_1, \Sigma, \delta_1, q_{01}, F_1)$, $M_2 = (Q_2, \Sigma, \delta_2, q_{02}, F_2)$, $Q_1 \cap Q_2 = \emptyset$.

Output: NFA M, $L(M) = L(M_1).L(M_2)$.

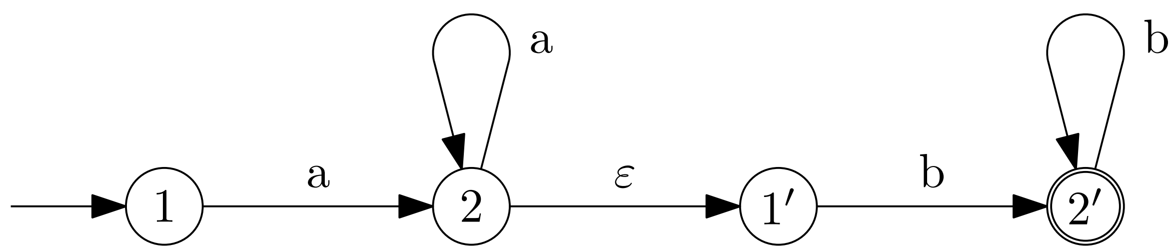
1: $Q \leftarrow Q_1 \cup Q_2$
2: $\delta(q, a) \leftarrow \delta_1(q, a)$, $\forall q \in Q_1, \forall a \in \Sigma$
3: $\delta(q, a) \leftarrow \delta_2(q, a)$, $\forall q \in Q_2, \forall a \in \Sigma$
4: $\delta(q, \varepsilon) \leftarrow \{q_{02}\}$, $\forall q \in F_1$
5: $M \leftarrow (Q, \Sigma, \delta, q_{01}, F_2)$
6: **return** M

FA and product of languages

Example
We construct a finite automaton for the product of languages a^+ and b^+.



$M = (\{1, 2, 1', 2'\}, \{a, b\}, \delta, 1, \{2'\})$



FA and product of languages

Algorithm NFA for a product of languages – without ε-transitions

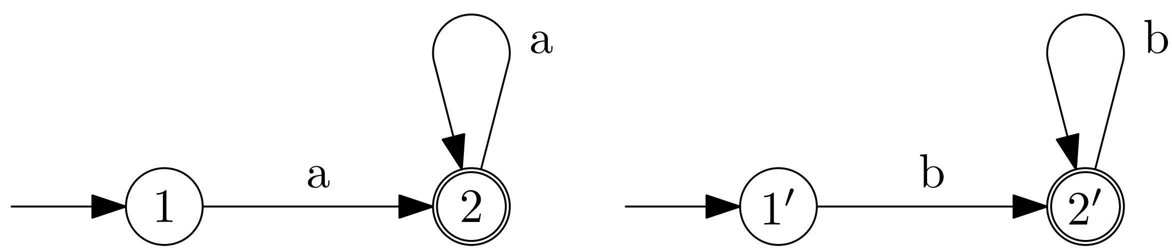
Input: NFA $M_1 = (Q_1, \Sigma, \delta_1, q_{01}, F_1)$, $M_2 = (Q_2, \Sigma, \delta_2, q_{02}, F_2)$.
Output: NFA automaton M, $L(M) = L(M_1).L(M_2)$.

1: $q_0 \leftarrow q_{01}$, if $q_{01} \notin F_1$
   $q_0 \leftarrow [q_{01}, q_{02}]$, if $q_{01} \in F_1$
2: $\delta(q, a) \leftarrow \delta_1(q, a)$, $\forall a \in \Sigma, \forall q \in Q_1$, if $\delta_1(q, a) \cap F_1 = \emptyset$,
   $\delta(q, a) \leftarrow \delta_1(q, a) \cup \{q_{02}\}$, $\forall a \in \Sigma, \forall q \in Q_1$, if $\delta_1(q, a) \cap F_1 \neq \emptyset$
3: $\delta(q, a) \leftarrow \delta_2(q, a)$, $\forall a \in \Sigma, \forall q \in Q_2$
4: $\delta(q_0, a) \leftarrow \delta_1(q_{01}, a) \cup \delta_2(q_{02}, a)$, $\forall a \in \Sigma$, if $q_0 = [q_{01}, q_{02}]$
5: $F \leftarrow F_2 \cup \{[q_{01}, q_{02}]\}$, if $q_{01} \in F_1 \wedge q_{02} \in F_2$
   $F \leftarrow F_2$, otherwise
6: $M \leftarrow (Q_1 \cup Q_2 \cup \{q_0\}, \Sigma, \delta, q_0, F)$
7: **return** M

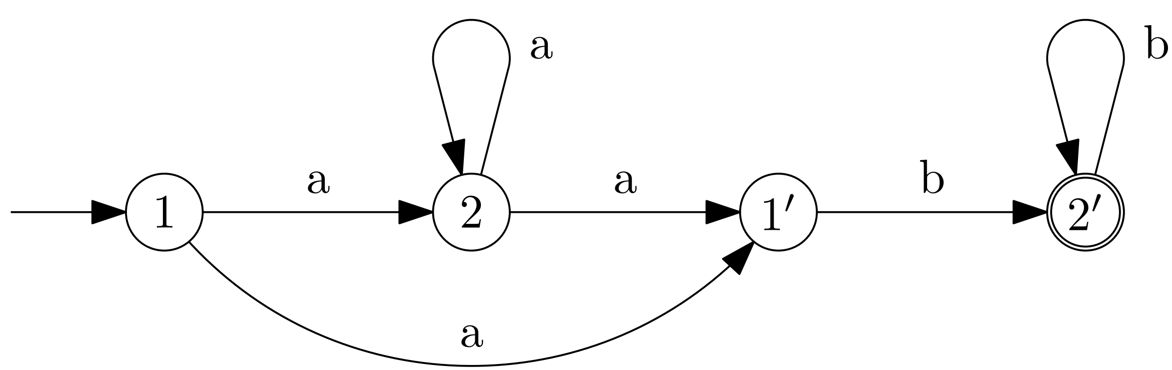
FA and product of languages

Example

We construct a finite automaton for the product of languages a^+ and b^+.



$$M = (\{1, 2, 1', 2'\}, \{a, b\}, \delta, 1, \{2'\})$$



FA and iteration of a language

Algorithm NFA for an iteration of a language – with ε-transitions

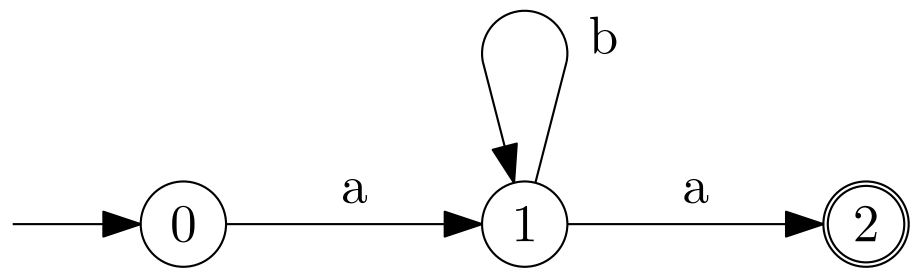
Input: NFA $M = (Q, \Sigma, \delta, q_0, F)$ accepting L.
Output: NFA M^* accepting L^*.

$\delta'(q, a) \leftarrow \delta(q, a)$, $\forall q \in Q$, $\forall a \in \Sigma$
$\delta'(q, \varepsilon) \leftarrow \{q_0\}$, $\forall q \in F$
$\delta'(q'_0, \varepsilon) \leftarrow \{q_0\}$
$M^* \leftarrow (Q \cup \{q'_0\}, \Sigma, \delta', q'_0, F \cup \{q'_0\})$
return M

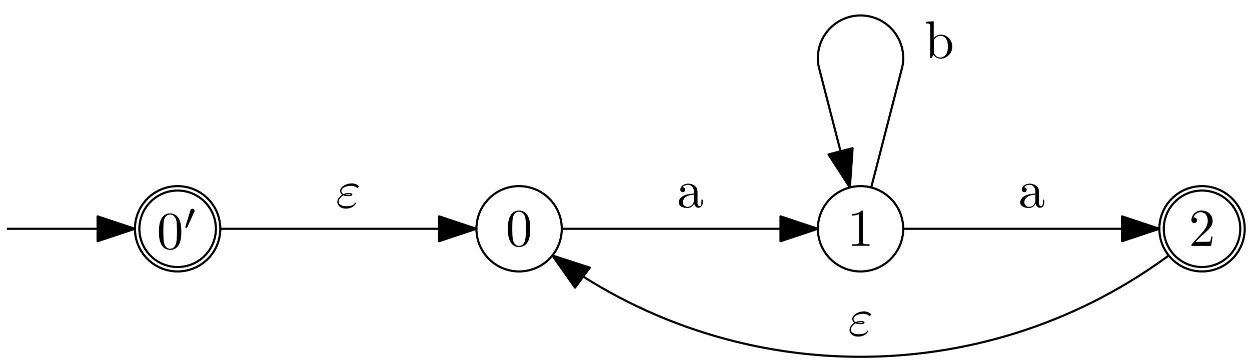
FA and iteration of a language

Example

We create NFA M^*accepting iteration of language ab^*a. An NFA M accepting all strings of the form ab^*a is given.



The resulting NFA is of the form $M^* = (\{0', 0, 1, 2\}, \{a, b\}, \delta, 0, \{0', 2\})$:



FA and iteration of a language

Algorithm NFA for an iteration of a language – without ε-transitions

Input: NFA $M = (Q, \Sigma, \delta, q_0, F)$ accepting L.

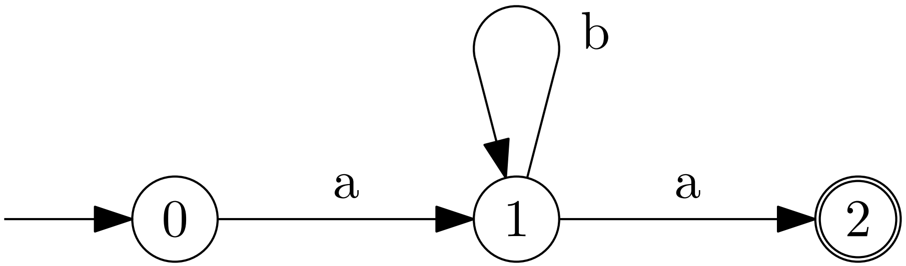
Output: NFA M^* accepting L^*.

1: $\delta'(q_0', a) \leftarrow \delta(q_0, a)$, $\forall a \in \Sigma$, if $\delta(q_0, a) \cap F = \emptyset$
2: $\delta'(q_0', a) \leftarrow \delta(q_0, a) \cup \{q_0\}$, $\forall a \in \Sigma$, if $\delta(q_0, a) \cap F \neq \emptyset$
3: $\delta'(q, a) \leftarrow \delta(q, a)$, $\forall q \in Q$, $\forall a \in \Sigma$, if $\delta(q, a) \cap F = \emptyset$
4: $\delta'(q, a) \leftarrow \delta(q, a) \cup \{q_0\}$, $\forall q \in Q$, $\forall a \in \Sigma$, if $\delta(q, a) \cap F \neq \emptyset$
5: $M^* \leftarrow (Q \cup \{q_0'\}, \Sigma, \delta', q_0', F \cup \{q_0'\})$
6: **return** M

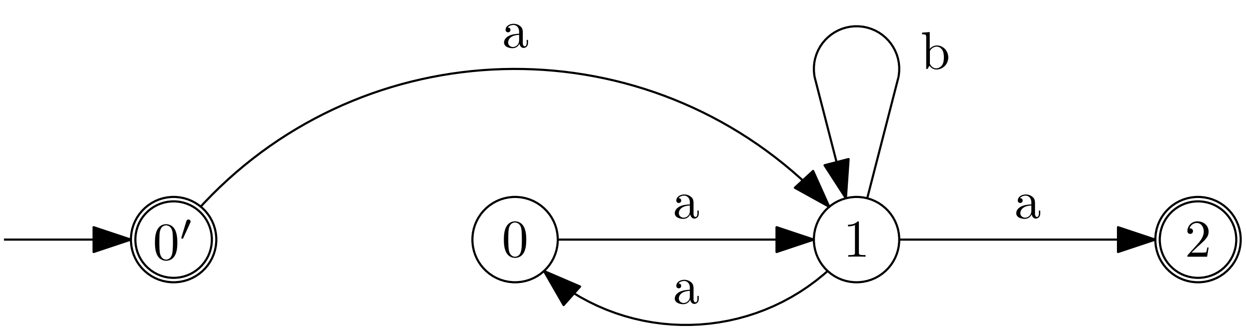
FA and iteration of a language

Example

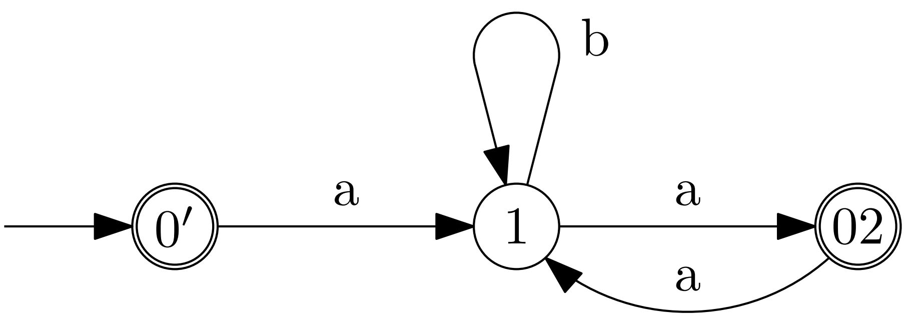
We have NFA M that accepts all strings of the form ab^*a.



NFA accepting the iteration of language ab^*a, i.e. language $(ab^*a)^*$:



DFA:



Minimal DFA

Definition (Minimal DFA)
Let $M = (Q, \Sigma, \delta, q_0, F)$ be a DFA. M is called *(state) minimal* DFA, if $\nexists M' = (Q', \Sigma, \delta', q_0', F')$ such that $L(M) = L(M')$ and $|Q| > |Q'|$.

Minimization of DFA

Algorithm Minimization of DFA

Input: DFA $M = (Q, \Sigma, \delta, q_0, F)$ without unreachable and redundant states.
Output: Minimal DFA $M' = (Q_m, \Sigma, \delta_m, q_{0m}, F_m)$, $L(M) = L(M')$.

1: Divide set Q into two subsets $Q_{\mathrm{I}} \leftarrow Q \setminus F$, $Q_{\mathrm{II}} \leftarrow F$.
2: **repeat**
3: Create a table δ', where for each state $q \in Q$ there is a row $\delta'(Q_i, a) = Q_j$, $q \in Q_i, \delta(q, a) \in Q_j, \forall a \in \Sigma$. (In the table, replace each state by the identificator of the subset it belongs to.)
4: If $\exists$ subset Q_i where all its rows are not identical, divide Q_i so that every new subset has its all rows identical.
5: **until** The subsets keep splitting
6: $Q_m \leftarrow$ the set of all resulting subsets
7: $\delta_m(Q_i, a) \leftarrow Q_j, \forall Q_i \in Q_m, \forall a \in \Sigma, \exists q \in Q_i, \delta(q, a) \in Q_j$
8: q_{0m} is the subset containing q_0
9: F_m are all the subsets of F
10: **return** M'

At the end of the algorithm it must hold that
$\delta_m(Q_i, a) = Q_j \Leftrightarrow \forall q \in Q_i, \delta(q, a) \in Q_j$.

Minimization of DFA

Example

Minimize the following DFA.

	state	input symbol	
	δ	a	b
$\rightleftarrows$	q_0	q_5	q_1
	q_1	q_4	q_3
	q_2	q_2	q_5
	q_3	q_3	q_0
	q_4	q_1	q_2
$\leftarrow$	q_5	q_0	q_4