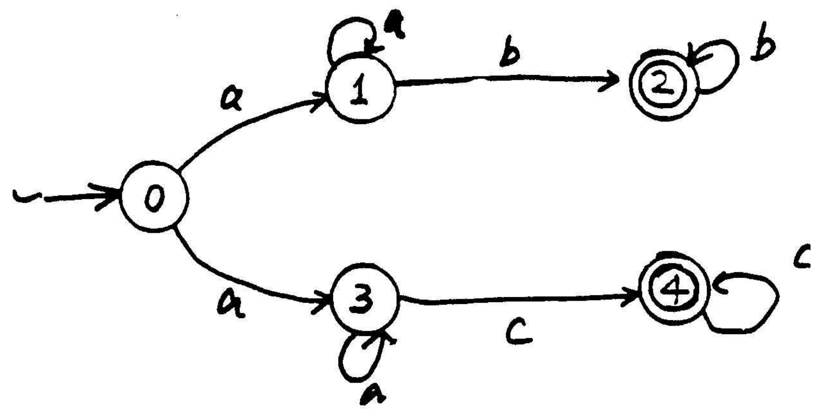
Transforming NFA to DFA

Consider the NFA M_1 again



There are only limited number of choices. For example:

$0\underline{a}ab \vdash 1ab \vdash 1b \vdash 2$

$0aab \vdash 3ab \vdash 3b$

$\{0\}aab \vdash \{1,3\}ab \vdash \{1,3\}b \vdash \{2\}$

Why <u>limited</u> number of choices?

The state set is finite.

We summarize the choices at each step by combining all configuration sequences into one "<u>super-conf. sequence</u>".

$\{0\}aab \vdash \{1,3\}ab \vdash \{1,3\}b \vdash \{2\}$.

We now have a set of all possible states at each step. From this point of view the computation of the NFA on an input word is <u>deterministic</u>.

A <u>super-configuration</u> has the form

$$Kx$$

where $K \subseteq Q$ and $x \in \Sigma^*$.

Note that $\underline{\emptyset x}$ is a super-conf., it means that the NFA cannot be in any state at that point, i.e., an abort has occured.

We say that

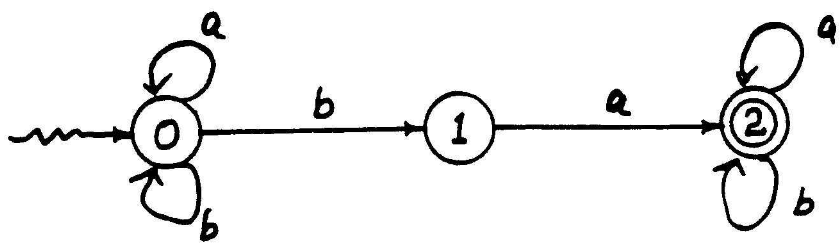
$$Kx \vdash Ny$$

if $x = ay$, for some $a \in \Sigma$, and

$N = \{q \mid (p, a, q) \in \delta$, **for some** $p \in K\}$

More examples on super-configurations

M: $L(M)$ **is the set of all words that have "ba" as a subword.**



The super-configuration sequence on input word "$abbaa$" is as follows:

$\{0\}abbaa \vdash \{0\}bbaa \vdash \{0,1\}baa \vdash \{0,1\}aa$
$\vdash \{0,2\}a \vdash \{0,2\}$

Notice that given a set $K \subseteq Q$ and an input symbol $a \in \Sigma$, the set $N \subseteq Q$ s.t. $Ka \vdash N$ is uniquely determined.

<u>Lemma</u> (2.3.1) (Determinism Lemma) Let $M = (Q, \Sigma, \delta, s, F)$ be an NFA. Then for all words $\underline{x}$ in Σ^* and for all $\underline{K} \subseteq Q$.

$\underline{Kx \vdash^* N \text{ and } Kx \vdash^* P}$

implies

$\underline{P = N}$.

<u>Lemma</u> (2.3.2) Let $M = (Q, \Sigma, \delta, s, F)$ be an NFA. Then for all words $\underline{x}$ in Σ^* and for all $\underline{q}$ in Q,

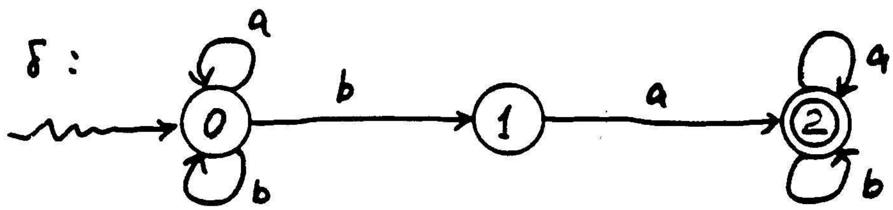
$\underline{qx \vdash^* p}$

iff $\underline{\{q\}x \vdash^* P}$, for some P with p in P.

Example (Transformation of an NFA to a DFA)

$M = (Q, \Sigma, \delta, s, F)$ **where**

$Q = 0, 1, 2, \qquad \Sigma = a, b$
$s = 0, \qquad F = \{2\}$



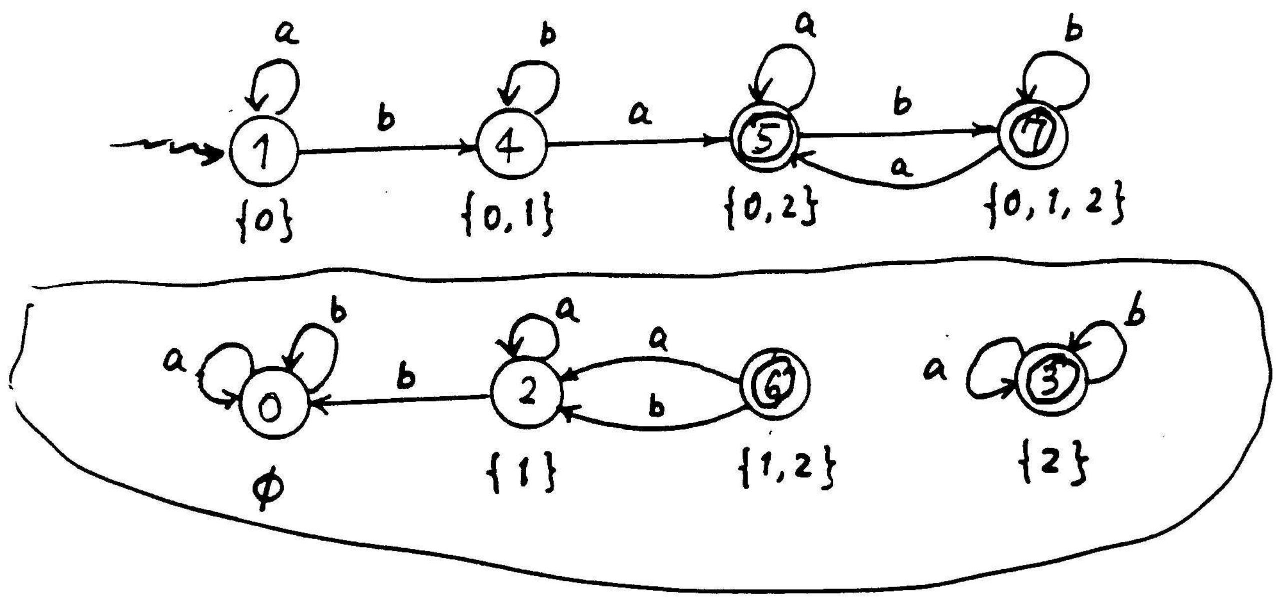
$M' = (Q', \Sigma, \delta', s', F')$ **where**

$$Q' = 2^Q = \{\emptyset, \{0\}, \{1\}, \{2\}, \{0,1\}, \{0,2\}, \{1,2\}, \{0,1,2\}\}$$

δ':

	current state \ input symbol	a	b
0	$\emptyset$	$\emptyset$	$\emptyset$
1	$\{0\}$	$\{0\}$	$\{0,1\}$
2	$\{1\}$	$\{2\}$	$\emptyset$
3	$\{2\}$	$\{2\}$	$\{2\}$
4	$\{0,1\}$	$\{0,2\}$	$\{0,1\}$
5	$\{0,2\}$	$\{0,2\}$	$\{0,1,2\}$
6	$\{1,2\}$	$\{2\}$	$\{2\}$
7	$\{0,1,2\}$	$\{0,2\}$	$\{0,1,2\}$

$\underline{\delta'(P, a) = \{q \mid (p, a, q) \in \delta \text{ and } p \in P\}}$



$s' = \{0\}$
$F' = \{$

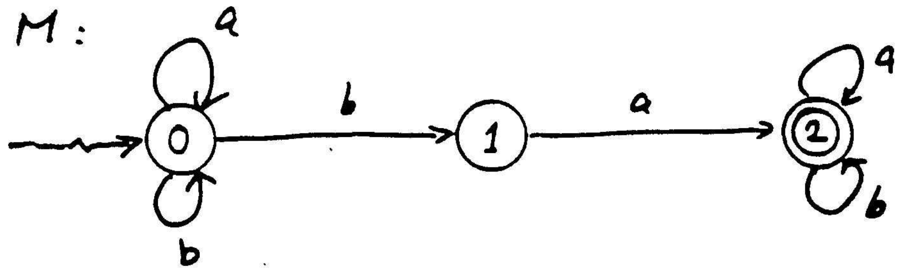
Algorithm NFA to DFA
—The Subset Construction

On entry: An NFA $M = (Q, \Sigma, \delta, s, F)$.

On exit: A DFA $M' = (Q', \Sigma, \delta', s', F')$
satisfying $L(M) = L(M')$.

begin Let $Q' = 2^Q, s' = \{s\}$ **and**
$F' = \{K \mid K \in Q',$ **and** $K \cap F \neq \emptyset\}$
We define $\delta' : Q' \times \Sigma \to Q'$ **by**
For all $K \in Q'$ **and for all** $a \in \Sigma$**,**
$\delta'(K, a) = N$**,** **if** $Ka \vdash N$ **in** M**.**
end of Algorithm

if $N = \{q \mid (p, a, q) \in \delta$ **and** $p \in K\}$

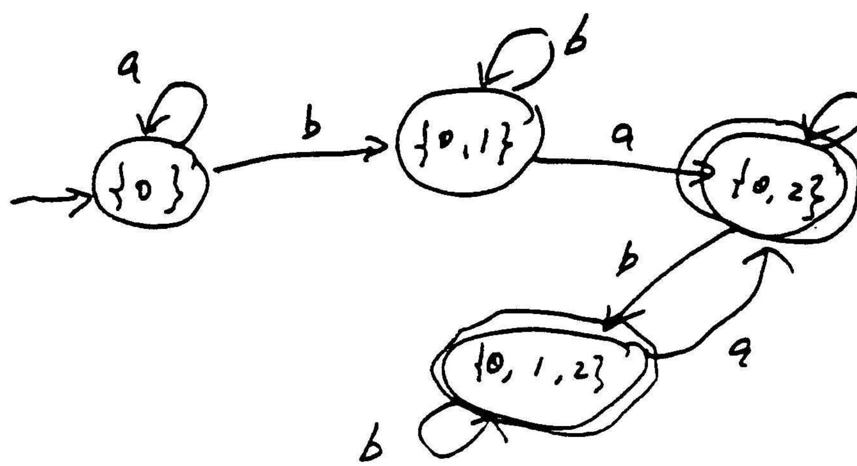


$s' = \{0\}$

current state \ input symbol	a	b
$\{0\}$	$\{0\}$	$\{0,1\}$
$\{0,1\}$	$\{0,2\}$	$\{0,1\}$
$\{0,2\}$	$\{0,2\}$	$\{0,1,2\}$
$\{0,1,2\}$	$\{0,2\}$	$\{0,1,2\}$

	a	b
0	$\{0\}$	$\{0,1\}$
1	$\{2\}$	ϕ
2	$\{2\}$	$\{2\}$

Algorithm NFA to DFA 2 —The Iterative Subset Construction



<u>Theorem</u> Given an NFA $M = (Q, \Sigma, \delta, s, F)$, then the DFA $M' = (Q', \Sigma', \delta', s', F')$ obtained by either subset construction satisfies $L(M') = L(M)$.

<u>Proof:</u>

By Lemma 2.3.2, for all $x \in \Sigma^*$ in M

$sx \vdash^* p,$ **iff** $\{s\}x \vdash^* P$ **for some** P **with** $p \in P$

By the construction of M',
$\{s\}x \vdash^* P$ **in** M **iff**
$\{s\}x \vdash^* P$ **in** M'**.**

$$\begin{aligned}
x \in L(M) &\Leftrightarrow sx \vdash^* f, \ \textbf{for some } f \in F \\
&\Leftrightarrow \{s\}x \vdash^* P, \ f \in P, \ \textbf{in } M \\
&\Leftrightarrow \{s\}x \vdash^* P, \ \textbf{in } M' \textbf{ and } P \cap F \neq \emptyset \\
&\Leftrightarrow s'x \vdash^* P, \ P \in F \\
&\Leftrightarrow x \in L(M')
\end{aligned}$$

Theorem

Every NFA Language is a DFA language and conversely.
$(\mathcal{L}_{NFA} = \mathcal{L}_{DFA})$

Example

Every finite language is accepted by a DFA.