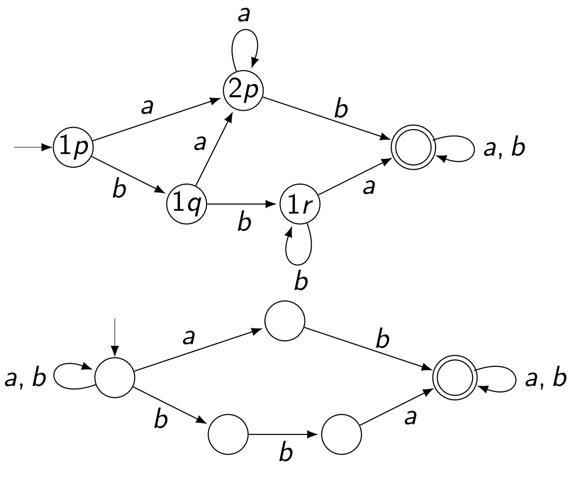
Section 3

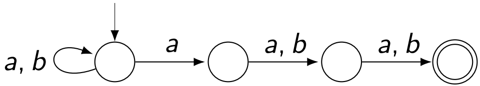
Non-Determinism, Regular Expressions, and Kleene's Theorem

3. Non-Determinism, Regular Expressions, and Kleene's Theorem
    - Examples
    - Allowing Λ-transitions
    - Definitions
    - Making the automaton deterministic

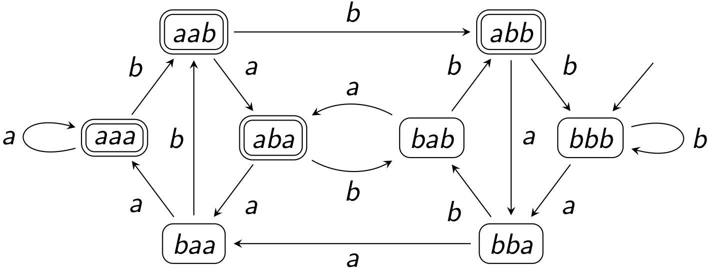
Non-determinism:
possibly many computations on given input
accept input when at least one of these computations accepts.



[M] see ↪E.2.18 (product construction)

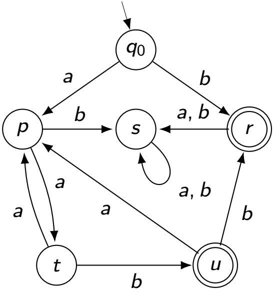


Also ↪deterministic

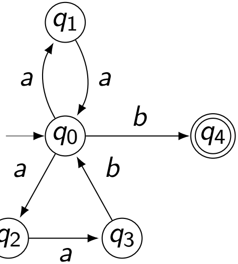


$n + 1$ versus 2^n states.

$L = \{aa, aab\}^*\{b\}$

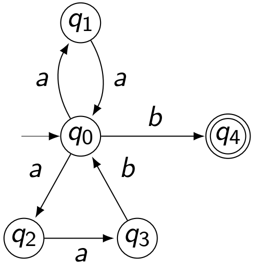


[M] E 2.22

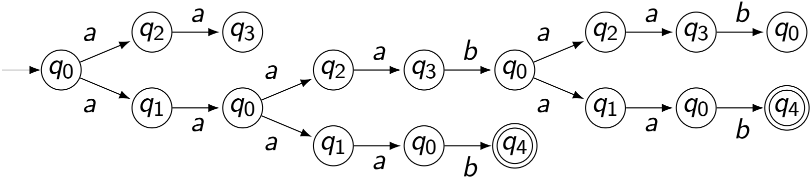


$x = aaaabaab$

[M] E 3.6. also ↪E 2.22

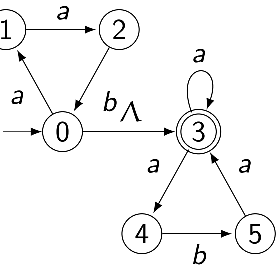


$x = aaaabaab$

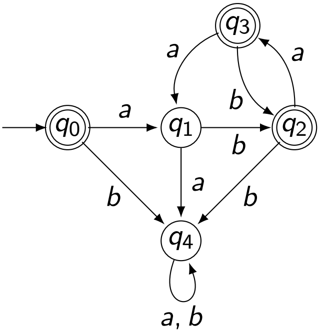


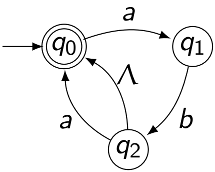
[M] E 3.6. also ↪E 2.22

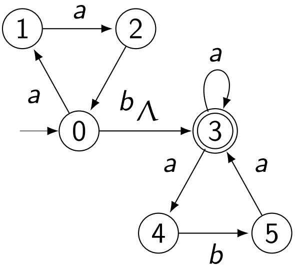
$\{aab\}^*\{a, aba\}^*$

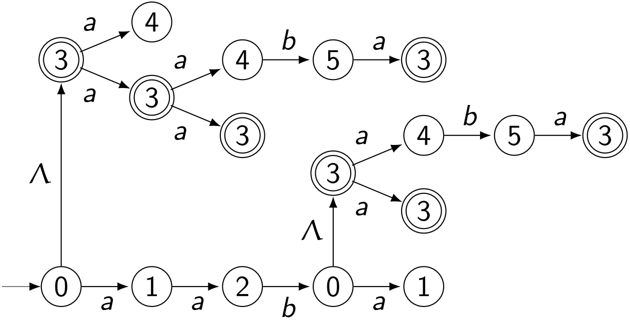


NFA


q3
a
a
b
a
q0
q1
b
q2
b
a
b
b
q4
a, b


a
q0
q1
Λ
a
b
q2





[M] E 3.9

From lecture 1:

Definition (FA)

[deterministic] finite automaton 5-tuple $M = (Q, \Sigma, q_0, A, \delta)$,
– Q finite set *states*;
– Σ finite *input alphabet*;
– $q_0 \in Q$ *initial* state;
– $A \subseteq Q$ *accepting* states;
– $\delta : Q \times \Sigma \to Q$ *transition function*.

[M] D 2.11 Finite automaton

[L] D 2.1 Deterministic finite accepter, has 'final' states

$\delta : Q \times \Sigma \to Q$

5-tuple $M = (Q, \Sigma, q_0, A, \delta)$

Definition ($\hookrightarrow$FA)

[deterministic] finite automaton

– $\delta : Q \times \Sigma \to Q$ *transition function*;

Definition (NFA)

nondeterministic finite automaton (with Λ-transitions)

– $\delta : \ldots$

5-tuple $M = (Q, \Sigma, q_0, A, \delta)$

Definition ($\hookrightarrow$FA)

[deterministic] finite automaton

– $\delta : Q \times \Sigma \to Q$ *transition function*;

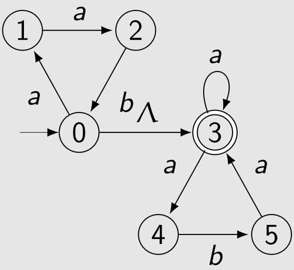
Definition (NFA)

nondeterministic finite automaton (with Λ-transitions)

– $\delta : Q \times (\Sigma \cup \{\Lambda\}) \to \dots$

Example

$\{aab\}^*\{a, aba\}^*$



NFA

δ	Λ	a	b
0	$\{3\}$	$\{1\}$	$\varnothing$
3	$\varnothing$	$\{3, 4\}$	$\varnothing$

5-tuple $M = (Q, \Sigma, q_0, A, \delta)$

Definition (↪FA)

[deterministic] finite automaton
– $\delta : Q \times \Sigma \to Q$ *transition function*;

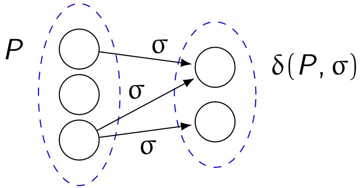
Definition (NFA)

nondeterministic finite automaton (with Λ-transitions)
– $\delta : Q \times (\Sigma \cup \{\Lambda\}) \to 2^Q$

Extended transfer function, without Λ-transitions

Extend δ to subsets P:

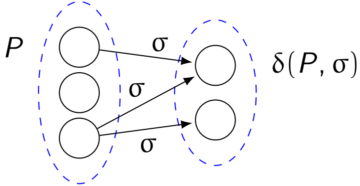
$\delta(P, \sigma) = \bigcup_{p \in P} \delta(p, \sigma) = \{q \in Q \mid q \in \delta(p, \sigma) \text{ for some } p \in P\}$.



$\delta^*(q, x)$ ...

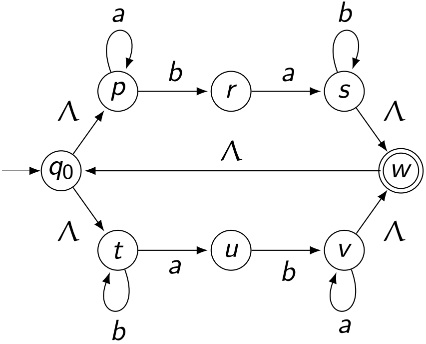
Extend δ to subsets P:

$\delta(P,\sigma) = \bigcup_{p \in P} \delta(p,\sigma) = \{q \in Q \mid q \in \delta(p,\sigma) \text{ for some } p \in P\}$.



$\delta^*(q,\Lambda) = \{q\}$
$\delta^*(q,y\sigma) = \delta(\delta^*(q,y),\sigma)$

Now, with Λ-transitions: $\delta^*(q,x)$ ...



Λ

$\delta^*(q_0, \Lambda) = \ldots$

$\Lambda \cdot a$

$\delta^*(q_0, a) = \ldots$

$a \cdot b$

$\delta^*(q_0, ab) = \ldots$

$ab \cdot a$

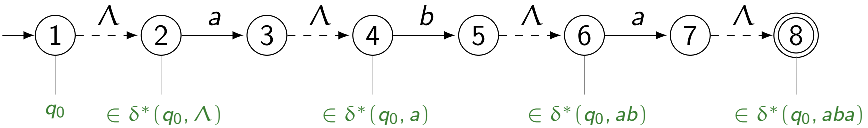
$\delta^*(q_0, aba) = \ldots$

[M] E 3.15

NFA $M = (Q, \Sigma, q_0, A, \delta)$ $\quad S \subseteq Q$

Definition

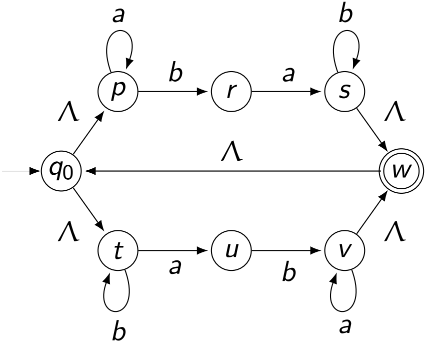
– $S \subseteq \Lambda(S)$
– $q \in \Lambda(S)$, then $\delta(q, \Lambda) \subseteq \Lambda(S)$



Definition

– $\delta^*(q, \Lambda) = \Lambda(\{q\}) \quad q \in Q$
– $\delta^*(q, y\sigma) = \Lambda(\ \delta(\ \delta^*(q, y), \sigma\ )\ ) \quad q \in Q,\ y \in \Sigma^*,\ \sigma \in \Sigma$

[M] D 3.13 & 3.14



Λ

$\Lambda(\{q_0\}) = \{q_0, p, t\}$

$\delta^*(q_0, \Lambda) = \{q_0, p, t\}$

$\Lambda \cdot a$

$\delta(\{q_0, p, t\}, a) = \{p, u\}$

$\delta^*(q_0, a) = \Lambda(\{p, u\}) = \{p, u\}$

$a \cdot b$

$\delta(\{p, u\}, b) = \{r, v\}$

$\delta^*(q_0, ab) = \Lambda(\{r, v\}) =$
$\{r, v, w, q_0, p, t\}$

$ab \cdot a$

$\delta(\{r, v, w, q_0, p, t\}, a) = \{s, v, p, u\}$

$\delta^*(q_0, aba) = \Lambda(\{s, v, p, u\}) =$
$\{s, w, q_0, p, t, v, u\}$

[M] E 3.15

NFA $M = (Q, \Sigma, q_0, A, \delta)$

Theorem

$q \in \delta^*(p, x)$ iff there is a path in [the transition graph of] M from p to q with label x (possibly including Λ-transitions).

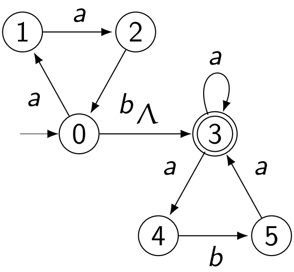
$\delta^*(q_0, x) = \varnothing$ no path for x from initial state

Definition

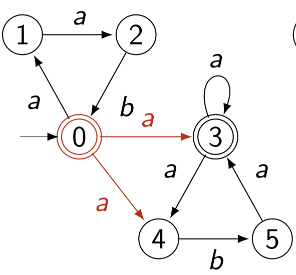
A string $x \in \Sigma^*$ is accepted by $M = (Q, \Sigma, q_0, A, \delta)$ if $\delta^*(q_0, x) \cap A \neq \varnothing$.
The *language $L(M)$ accepted* by M is the set of all strings accepted by M.

[M] D 3.14 [L] D 2.2

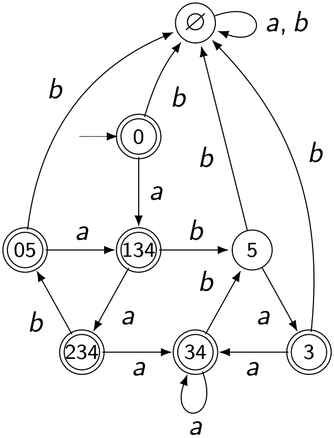
$\{aab\}^*\{a, aba\}^*$



NFA-Λ



NFA



FA

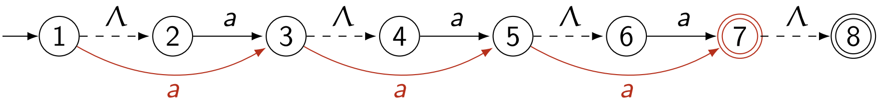
Theorem

For every language $L \subseteq \Sigma^*$ accepted by an NFA $M = (Q, \Sigma, q_0, A, \delta)$, there is an NFA M_1 with no Λ-transitions that also accepts L.

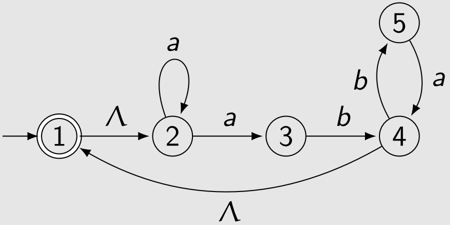
[M] T 3.17

The precise inductive proof of this result does not have to be known for the exam. However, the construction in the next slides has to be known.

Different from book!

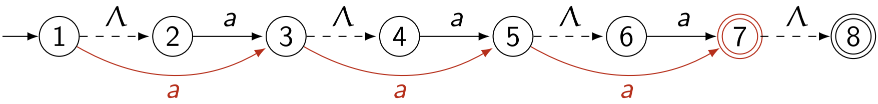


Example

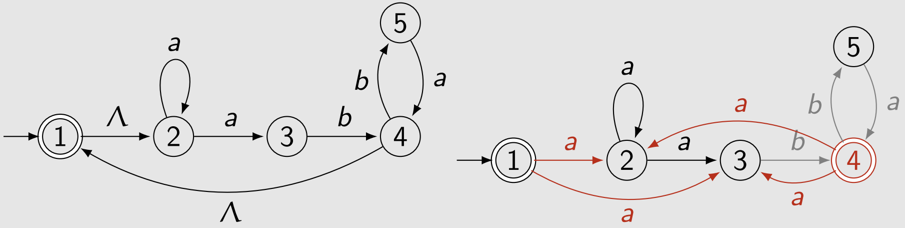


[M] E 3.19 but fewer edges!

Different from book!

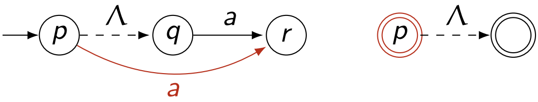


Example



[M] E 3.19 but fewer edges!

Different from book!

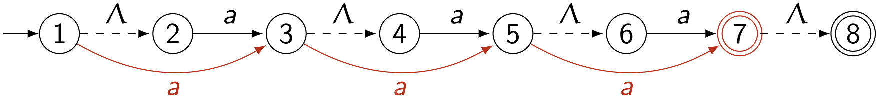


Λ-removal

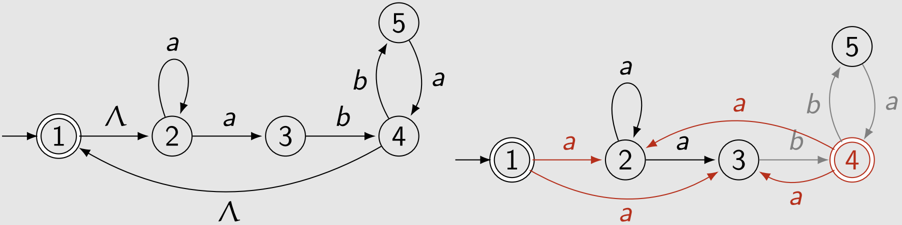
NFA $M = (Q, \Sigma, q_0, A, \delta)$
construct NFA $M_1 = (Q, \Sigma, q_0, A_1, \delta_1)$ without Λ-transitions
– whenever $q \in \Lambda_M(\{p\})$ and $r \in \delta(q, a)$, add r to $\delta_1(p, a)$
– whenever $\Lambda_M(\{p\}) \cap A \neq \varnothing$, add p to A_1.

In particular,
– non-Λ-transitions are maintained
– $A \subseteq A_1$

Different from book!



Example



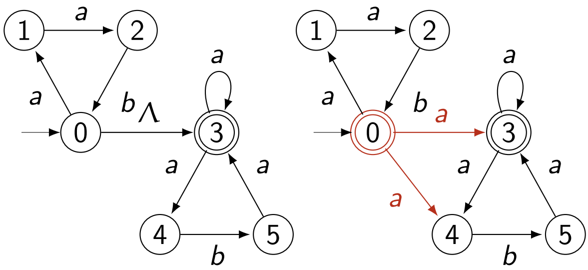
Construction book: $\delta_1(p, \sigma) = \delta^*(p, \sigma)\ldots$

Accepting states. . .

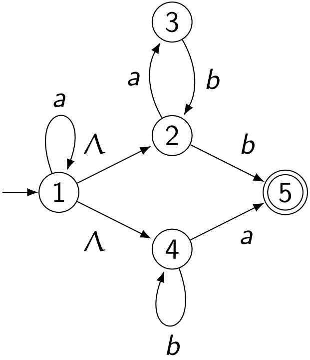
[M] E 3.19

Example

$L = \{aab\}^*\{a, aba\}^*$



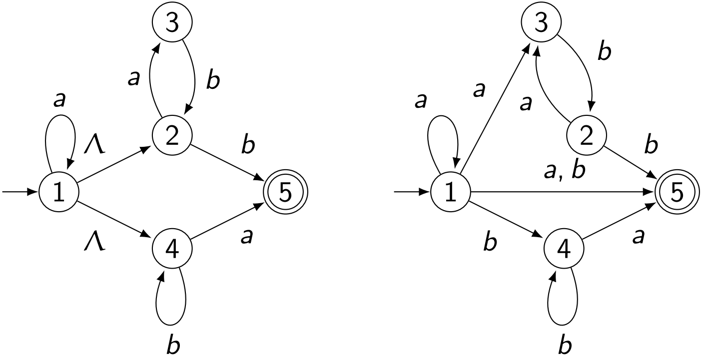
$\{a\}^*[\,\{ab\}^*\{b\} \cup \{b\}^*\{a\}\,]$



q	$\delta(q,a)$	$\delta(q,b)$	$\delta(q,\Lambda)$	$\Lambda(\{q\})$
1	1	—	2, 4	1, 2, 4
2	3	5	—	2
3	—	2	—	3
4	5	4	—	4
5	—	—	—	5

[M] E 3.23

$\{a\}^*[\,\{ab\}^*\{b\} \cup \{b\}^*\{a\}\,]$



[M] E 3.23 but fewer edges!

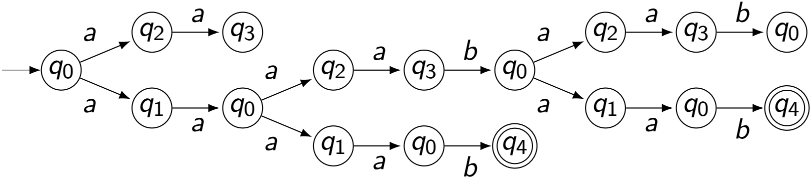
Theorem

For every language $L \subseteq \Sigma^*$ accepted by an NFA $M = (Q, \Sigma, q_0, A, \delta)$ without Λ-transitions, there is an FA M_1 that also accepts L.

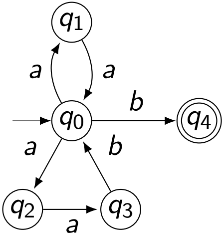
[M] T 3.18

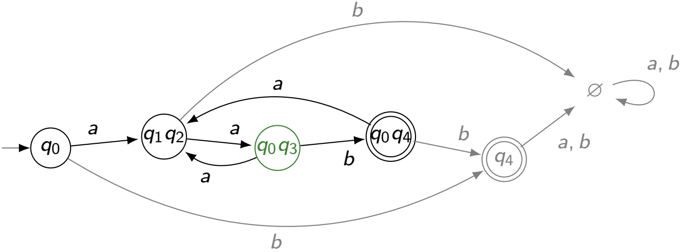
The precise inductive proof of this result does not have to be known for the exam. However, the construction in the next slides has to be known.

Folding the computation tree

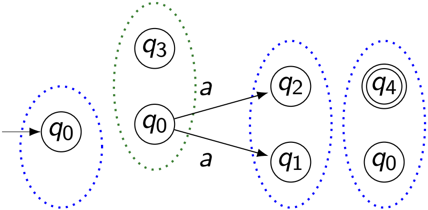


$\{q_0\}$ $\{q_1, q_2\}$ $\{q_0, q_3\}$ $\{q_1, q_2\}$ $\{q_0, q_3\}$ $\{q_0, q_4\}$ $\{q_1, q_2\}$ $\{q_0, q_3\}$ $\{q_0, q_4\}$





[M] E 3.6 ánd E 3.21

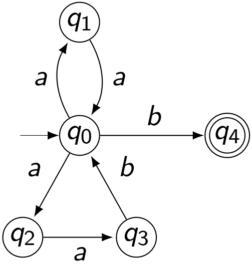


Subset construction

NFA $M = (Q, \Sigma, q_0, A, \delta)$ without Λ-transitions
construct FA $M_1 = (Q_1, \Sigma, \delta_1, q_1, A_1)$

- $Q_1 = 2^Q$
- $q_1 = \{q_0\}$
- $A_1 = \{\, q \in Q_1 \mid q \cap A \neq \varnothing \,\}$
- $\delta_1(q, \sigma) = \bigcup_{p \in q} \delta(p, \sigma)$

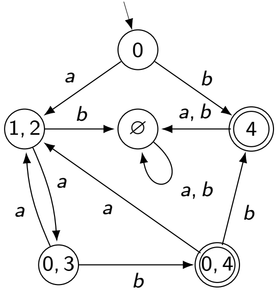
[M] Th 3.18



q	$\delta(q, a)$	$\delta(q, b)$
0	1, 2	4
1	0	—
2	3	—
3	—	0
4	—	—

[M] E 3.21. also ↪E 3.6

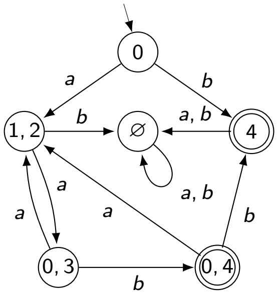
$L = \{aa, aab\}^*\{b\}$

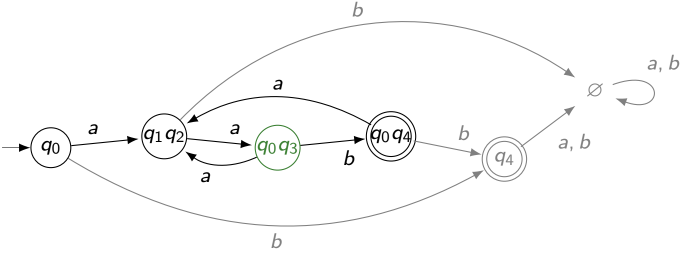


Minimal (this time)

[M] E 3.21. also ↪E 3.6

Once more $\{aa, aba\}^*\{b\}$





[M] E 3.6 ánd E 3.21

above

The subset construction (or powerset construction) can be used to transform a non-deterministic finite state automaton (without Λ) into an equivalent deterministic automaton. The states of the new automaton consist of sets of states of the original automaton (hence powerset). The set collects all possible states that the original automaton could have ended in with the same input.

Note that the constructed automaton may be exponential in size compared to the nondetereministic one.

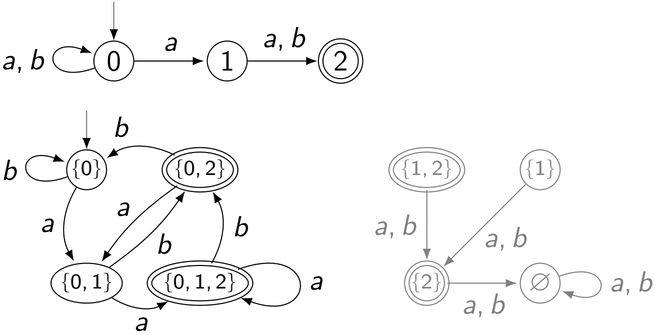
Reference

M.O. Rabin, D. Scott. Finite automata and their decision problems. IBM Journal of Research and Development. 3 (2): 114125, 1959. doi:10.1147/rd.32.0114

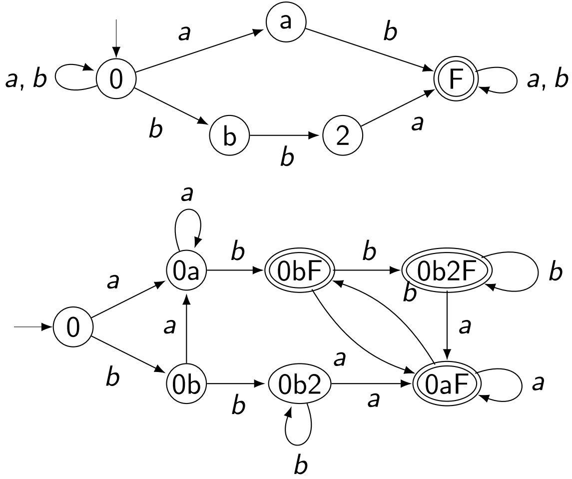
below

Unreachable states can be omitted.

Reachable states



also ↪3rd from the end

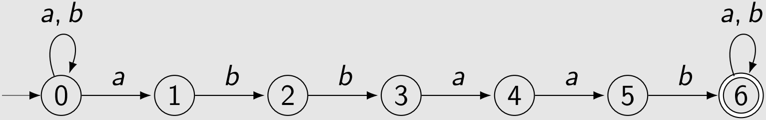


[M] language from ↪E 2.18

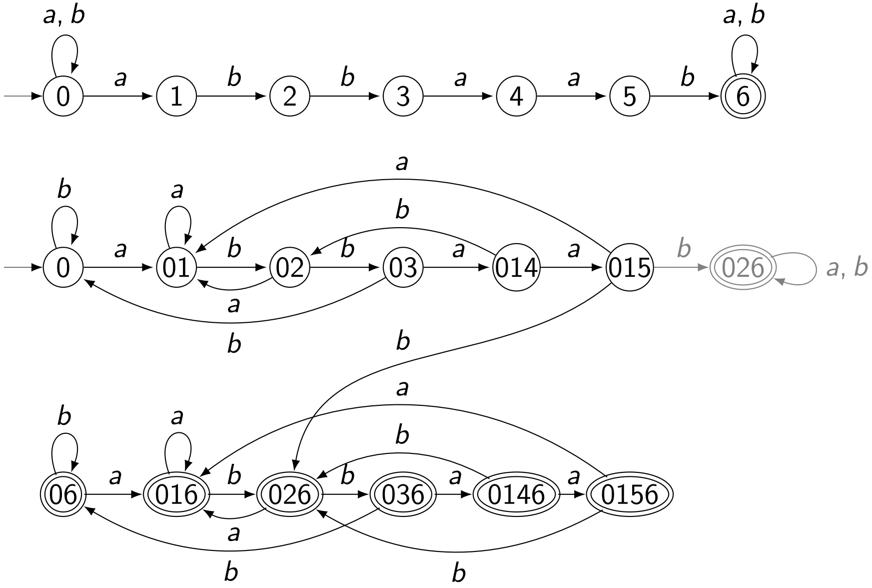
What about this one ...

Example

$L_3 = \{ x \in \{a, b\}^* \mid x \text{ contains the substring } abbaab \}$



[M] ↪E. 2.5 (deterministic)

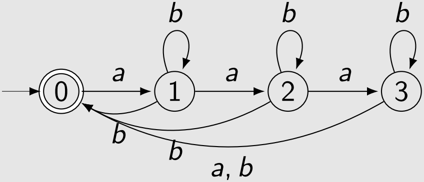

a, b
a
b
b
a
a
b
a, b
0
1
2
3
4
5
6
a
b
a
b
a
b
b
a
a
b
0
01
02
03
014
015
026
a, b
a
b
b
a
b
a
b
a
a
a
06
016
026
036
0146
0156
b
b

ABOVE

Illustration.

The determinization algorithm for the nondeterministic automaton for "has substring x" will always generate two copies of x. In the last copy all nodes are accepting, and they can be reduced to one node.

Example ($n = 4$)



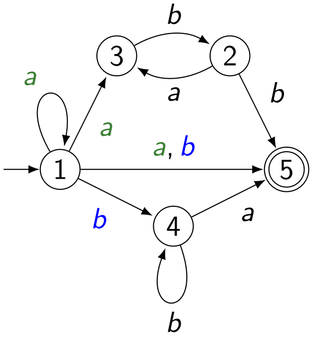
all 2^n subsets are reachable, nonequivalent, states.

ABOVE

Theoretically, the subset construction used on a set Q with n nodes constructs an automaton with state set 2^Q with 2^n nodes. In practice however, not all nodes are really necessary.
Usually not all nodes are reachable, and we omit those from the construction.
Sometimes nodes can be joined because they are equivalent.

This worst-case example however needs all nodes. So the determinization algorithm applied to a finite state automaton in the worst case will blow-up the original nondeterministic automaton exponentially in size.

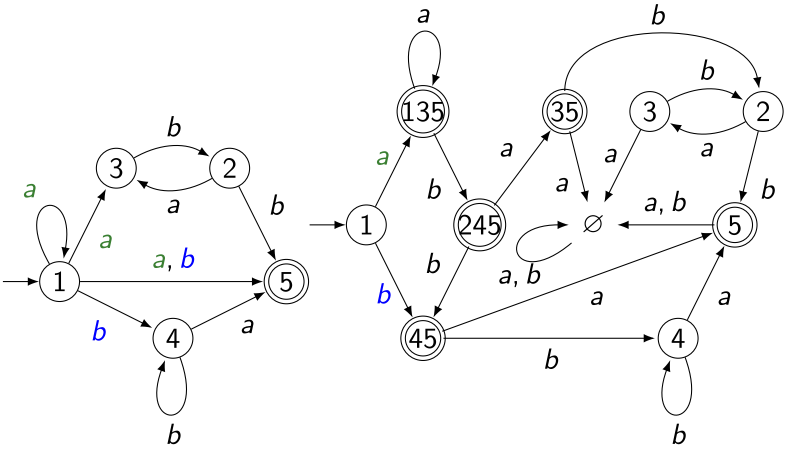
$\{a\}^*[\, \{ab\}^*\{b\} \cup \{b\}^*\{a\}\,]$



q	$\delta(q, a)$	$\delta(q, b)$
1	1, 3, 5	4, 5
2	3	5
3	—	2
4	5	4
5	—	—

[M] E 3.23 ctd.

$\{a\}^*[\,\{ab\}^*\{b\} \cup \{b\}^*\{a\}\,]$



[M] E 3.23 ctd.

Construct an equivalent FA, applying the appropriate algorithms.