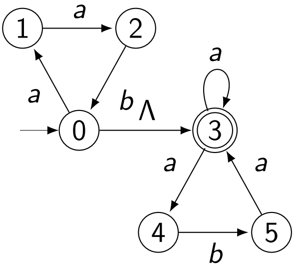
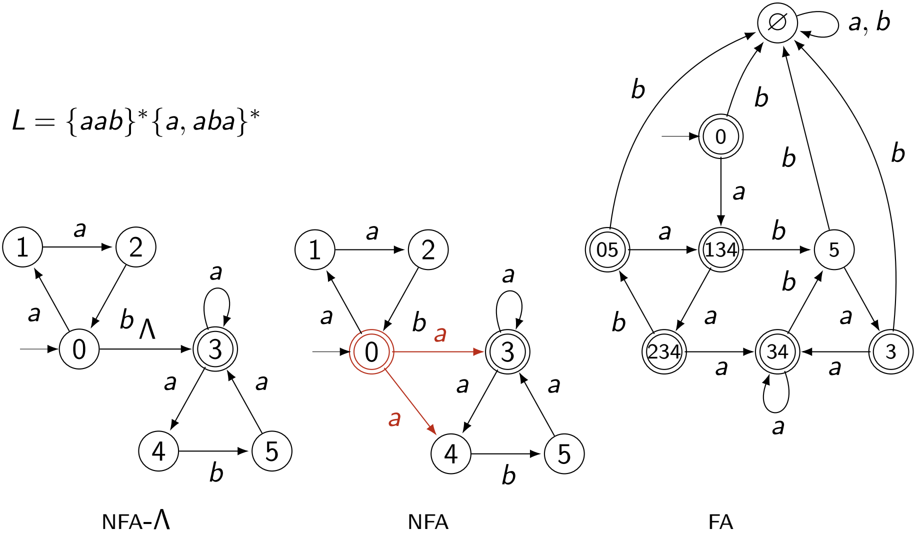
$L = \{aab\}^*\{a, aba\}^*$



NFA-Λ

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



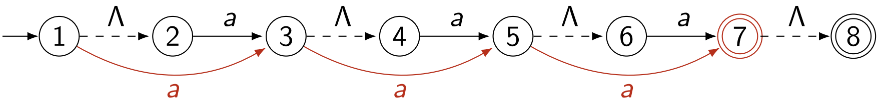
Theorem

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.

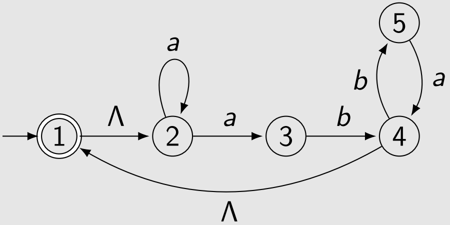
[M] T 3.17

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!



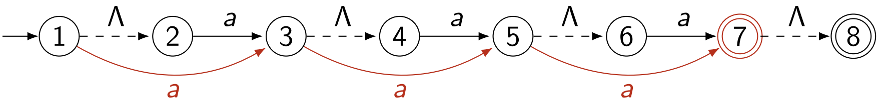
Example



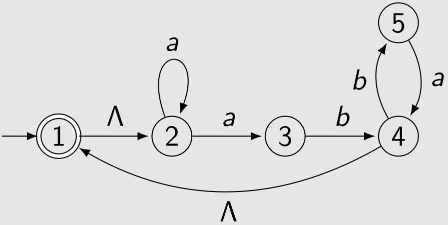
[M] E 3.19 but fewer edges!

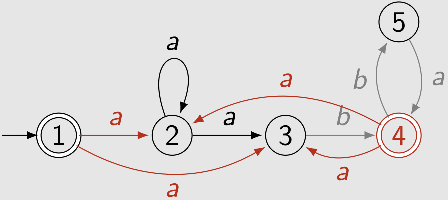
Construction: removing Λ-transitions

Different from book!



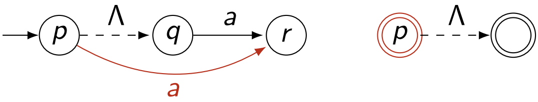
Example





[M] E 3.19 but fewer edges!

Different from book!

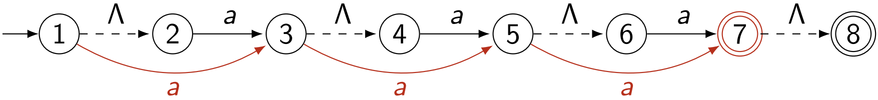


Λ-removal

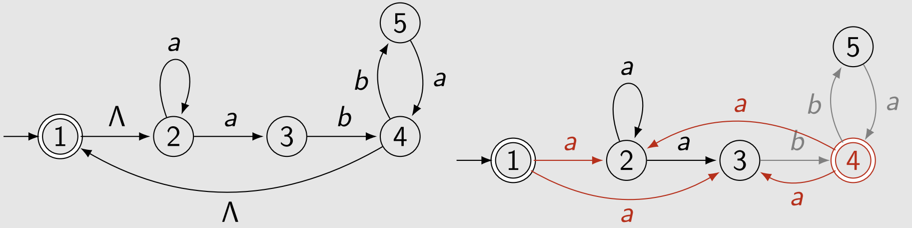
NFA $M = (Q, \Sigma, q_0, A, \delta)$
construct NFA $M_1 = (Q, \Sigma, q_0, A_1, \delta_1)$ without Λ-transitions
– whenever $q \in \Lambda_M(\{p\})$ and $r \in \delta(q, a)$, add r to $\delta_1(p, a)$
– whenever $\Lambda_M(\{p\}) \cap A \neq \varnothing$, add p to A_1.

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

Different from book!



Example

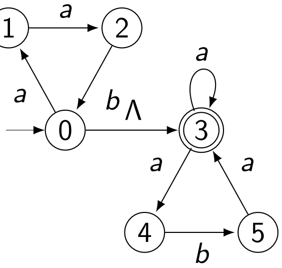


Construction book: $\delta_1(p,\sigma) = \delta^*(p,\sigma) \qquad \forall p \in Q, \forall \sigma \in \Sigma$

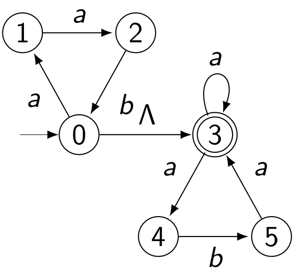
Accepting states: $A \cup \{q_0\}$ if $\Lambda \in L$, A otherwise

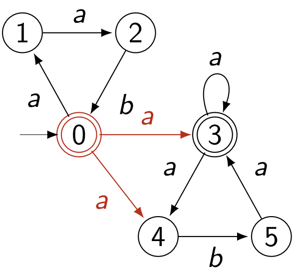
[M] E 3.19

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

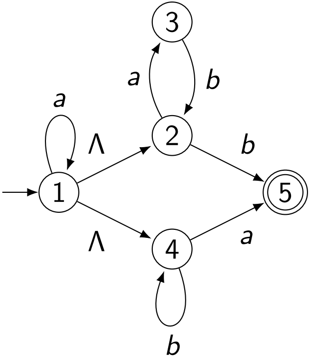


$L = \{aab\}^*\{a, aba\}^*$





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

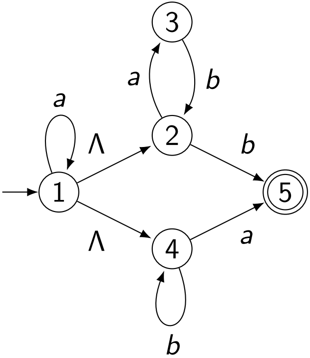


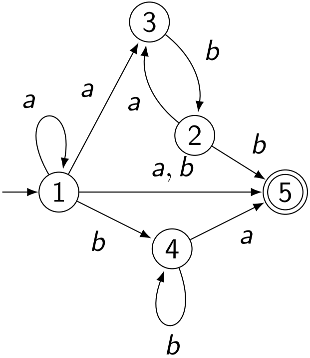
q	$\delta(q,a)$	$\delta(q,b)$	$\delta(q,\Lambda)$	$\Lambda(\{q\})$
1	1	—	2, 4	1, 2, 4
2	3	5	—	2
3	—	2	—	3
4	5	4	—	4
5	—	—	—	5

[M] E 3.23

Example 3.23

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





[M] E 3.23 but fewer edges!

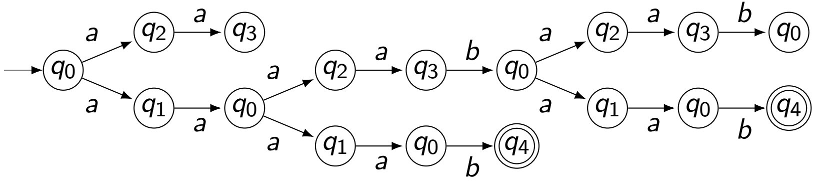
Theorem

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.*

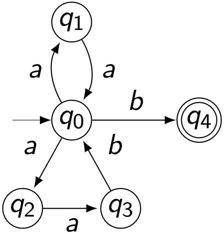
[M] T 3.18

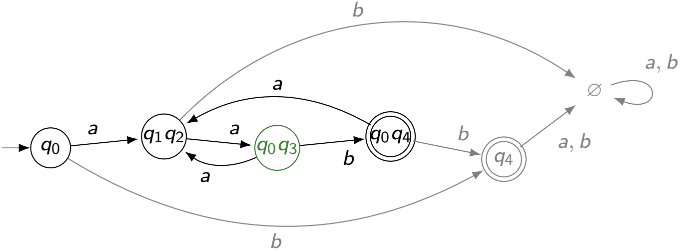
The precise inductive proof of this result does not have to be known for the exam. However, the construction in the next slides has to be known.

Folding the computation tree

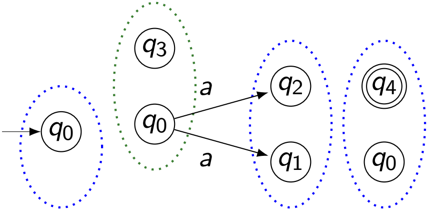


$\{q_0\} \quad \{q_1, q_2\}\,\{q_0, q_3\}\,\{q_1, q_2\}\,\{q_0, q_3\}\,\{q_0, q_4\}\,\{q_1, q_2\}\,\{q_0, q_3\}\,\{q_0, q_4\}$





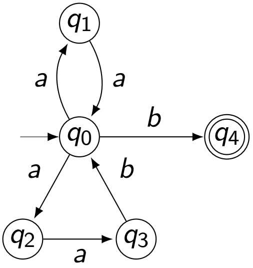
[M] E 3.6 and E 3.21



Subset construction

NFA $M = (Q, \Sigma, q_0, A, \delta)$ without Λ-transitions
construct FA $M_1 = (Q_1, \Sigma, \delta_1, q_1, A_1)$
– $Q_1 = 2^Q$, i.e., sets of states of M
– $q_1 = \{q_0\}$
– $A_1 = \{\ q \in Q_1 \mid q \cap A \neq \varnothing\ \}$
– $\delta_1(q, \sigma) = \bigcup_{p \in q} \delta(p, \sigma) \qquad \forall q \in Q_1, \forall \sigma \in \Sigma$

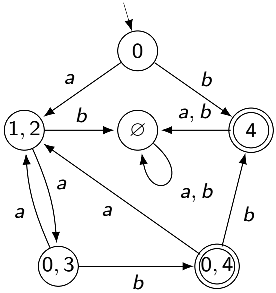
[M] Th 3.18



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, yet not always

[M] E 3.21. also ↪E 3.6

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.

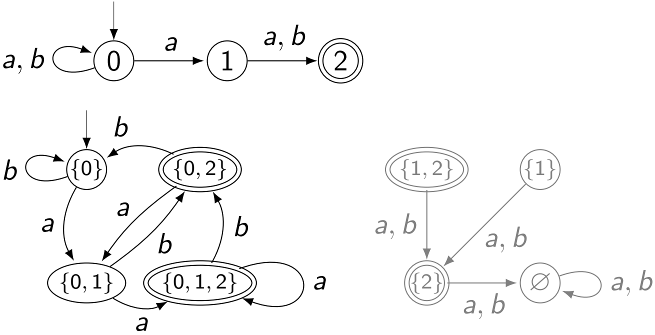
REFERENCE

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

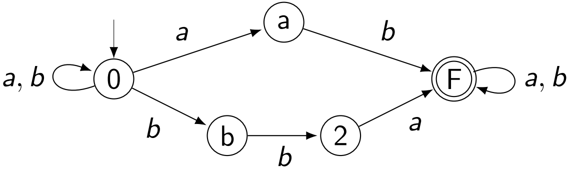
BELOW

Unreachable states can be omitted.

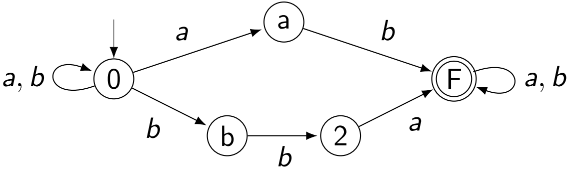
Example: *a* second symbol from end

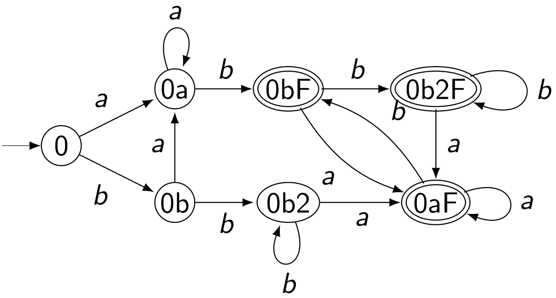


also ↪3rd from the end



[M] language from ↪E 2.18

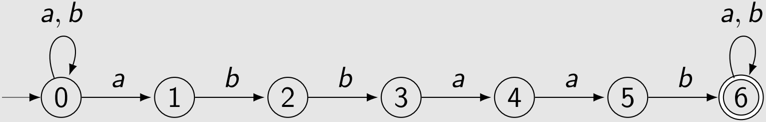




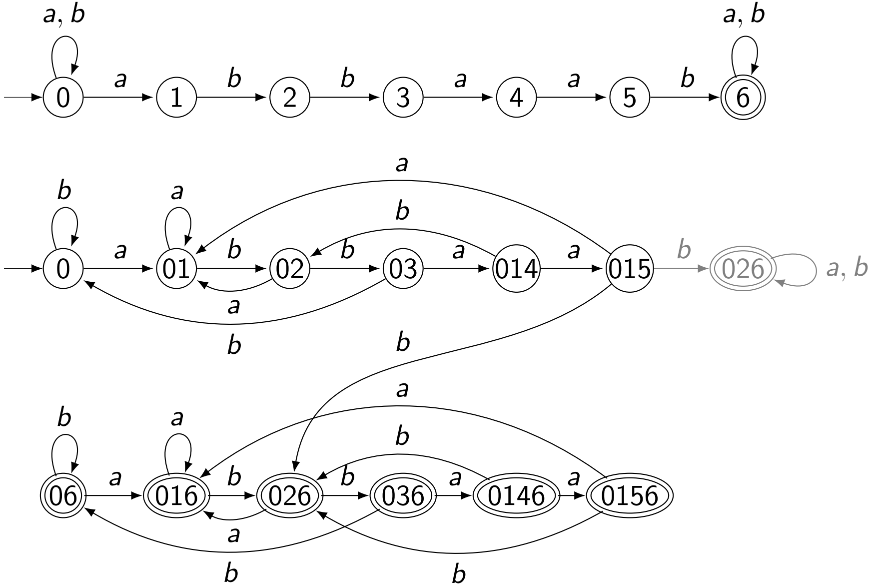
[M] language from ↪E 2.18

Example

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



[M] $\hookrightarrow$E. 2.5 (deterministic)

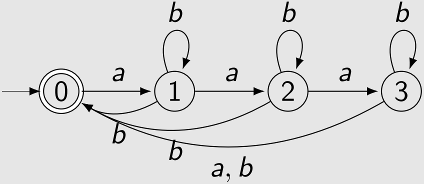

a, b
a, b
0
a
1
b
2
b
3
a
4
a
5
b
6
a
b
b
a
0
a
01
b
02
b
03
a
014
a
015
b
026
a, b
a
b
b
a
b
b
06
a
016
a
b
026
b
036
a
0146
a
0156
a
b
b

ABOVE

Illustration.

The subset construction 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 ($n = 4$)



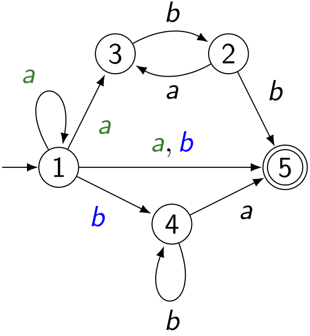
All 2^n subsets are reachable, nonequivalent, states.

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.

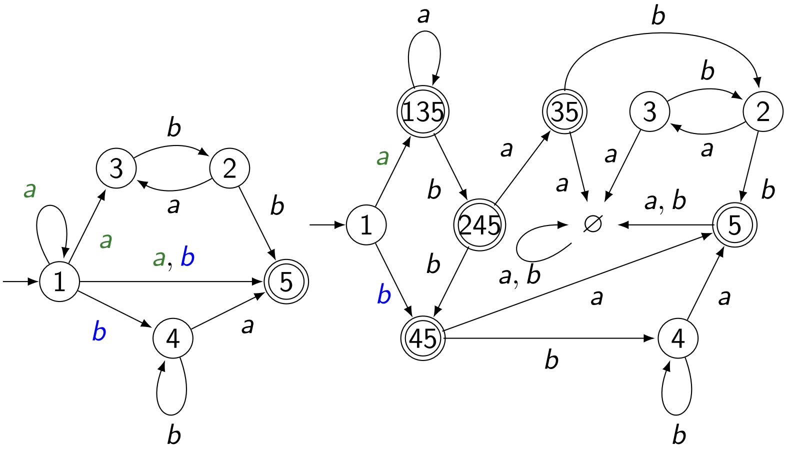
$\{a\}^*[\ \{ab\}^*\{b\} \cup \{b\}^*\{a\}\ ]$



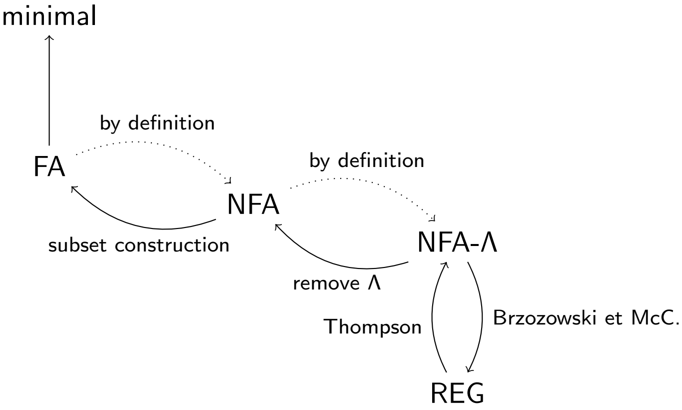
q	$\delta(q, a)$	$\delta(q, b)$
1	1, 3, 5	4, 5
2	3	5
3	—	2
4	5	4
5	—	—

[M] E 3.23 ctd.

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



[M] E 3.23 ctd.


minimal
by definition
FA
by definition
NFA
subset construction
NFA-Λ
remove Λ
Thompson
Brzozowski et McC.
REG

Definition (REG)

– $\varnothing$ is in REG.

– $\{a\}$ in REG, for every $a \in \Sigma$

– if L_1 and L_2 in REG,
then so are $L_1 \cup L_2$, $L_1 \cdot L_2$, and L_1^*.

[M] D. 3.1 $\mathcal{R}$

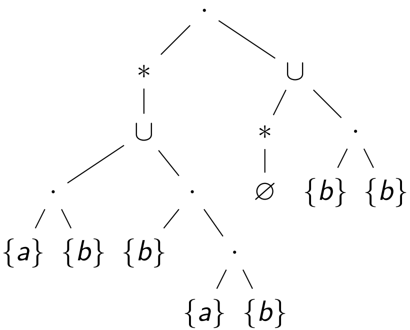
Smallest set[family] of languages that
– contains $\varnothing$ and $\{a\}$ for $a \in \Sigma$, and — basis
– is *closed under* union, concatenation and star. — induction

[M] cf. E 1.20

$\{ab, bab\}^*\{\Lambda, bb\}$

$\{ab, bab\}^*\{\Lambda, bb\}$

$(\ (\{a\} \cdot \{b\}) \cup (\{b\} \cdot \{a\} \cdot \{b\})\ )^* \cdot (\ \varnothing^* \cup (\{b\} \cdot \{b\})\ )$



– $\varnothing$, Λ, and a are RegEx (for all $a \in \Sigma$)
– if E_1 and E_2 are RegEx, then so are E_1^*, $(E_1 + E_2)$, and $(E_1 E_2)$

expression [syntax] vs its language [semantics]

E string	$L(E)$ language
$\varnothing$	$\varnothing$
Λ	$\{\Lambda\}$
a	$\{a\}$
$(E_1 + E_2)$	$L(E_1) \cup L(E_2)$
$(E_1 E_2)$	$L(E_1) \cdot L(E_2)$
E_1^*	$L(E_1)^*$

we say

$E_1 = E_2$ iff $L(E_1) = L(E_2)$

– Odd number of a

$bba_0ba_1bbba_2bba_1a_2bb$

[M] E 3.2

– Odd number of *a*

$bba_0ba_1bbba_2bba_1a_2bb$

$b^*ab^*(ab^*a)^*b^*$	not correct
$b^*ab^*(ab^*ab^*)^*$	$b^*ab^*(ab^*ab^*)(ab^*ab^*)$
$b^*a(b^*ab^*ab^*)^*$	not correct
$b^*a(b^*ab^*a)^*b^*$	$b^*a(b^*ab^*a)(b^*ab^*a)b^*$
$b^*a(b+ab^*a)^*$	$b^*ab^*(ab^*a)b^*(ab^*a)b^*$

[M] E 3.2

– Ending with b, no aa

$bb(ab)bbb(ab)(ab)b$

– Ending with b, no aa

$bb(ab)bbb(ab)(ab)b$

$(b + ab)^*(b + ab)$ at least once

[M] cf. E 3.3, see $\hookrightarrow$E. 2.3

– No aa may also end in a

$(b + ab)^*(\Lambda + a)$