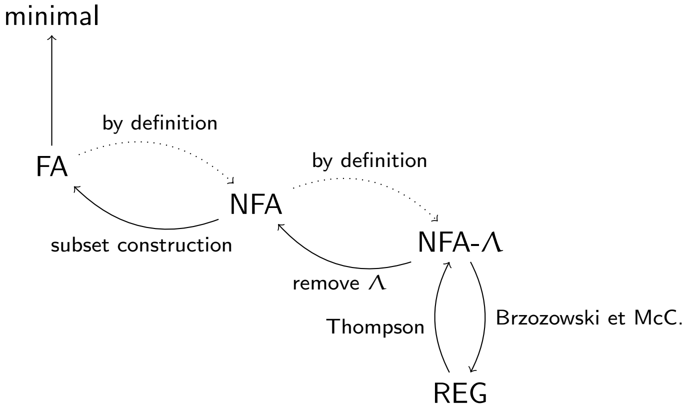

minimal
by definition
FA
by definition
NFA
subset construction
NFA-Λ
remove Λ
Thompson
Brzozowski et McC.
REG

Definition (REG)

– $\varnothing$ is in REG.

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

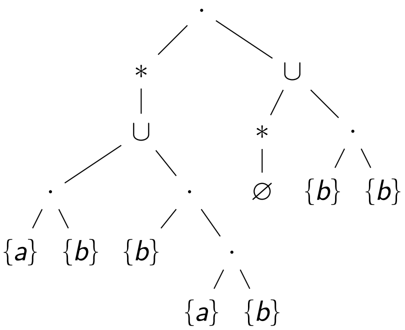
– if L_1 and L_2 in REG,
then so are $L_1 \cup L_2$, $L_1 \cdot L_2$, and L_1^*.

[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

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

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



– $\varnothing$, 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

$bba_0 ba_1 bbba_2 bba_1 a_2 bb$

[M] E 3.2

– Odd number of *a*

$bba_0 ba_1 bbba_2 bba_1 a_2 bb$

$b^* ab^* (ab^* a)^* b^*$ 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

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

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

[M] cf. E 3.3, see ↪E. 2.3

– No aa may also end in a

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

– Even number of both a and b

two letters together

aa and bb keep both numbers even [odd]

ab and ba switch between even and odd, for both numbers

– Even number of both *a* and *b*

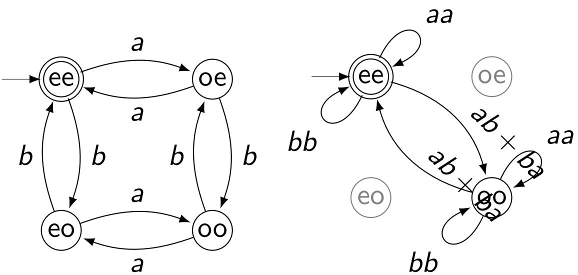
two letters together

aa and *bb* keep both numbers even [odd]

ab and *ba* switch between even and odd, for both numbers

$$(\ aa + bb + (ab + ba)(aa + bb)^*(ab + ba)\ )^*$$

[M] E 3.4, see ↪Brzozowski *et* McCluskey



– Numeric constants in programming language

14, +1, -12, 14.3, -.99, 16., 3E14, -1.00E2, 4.1E-1, .3E+2

[M] E 3.5

– Numeric constants in programming language

14, +1, -12, 14.3, -.99, 16., 3E14, -1.00E2, 4.1E-1, .3E+2

Use d for $(0+1+2+3+4+5+6+7+8+9)$

Use s for $(\Lambda + \text{'+'} + \text{'-'})$

Use p for '.'

$$(sdd^*(\Lambda + pd^*) + pdd^*)(\Lambda + Esdd^*)$$

[M] E 3.5

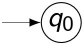
Theorem (Kleene)

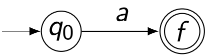
Finite automata and regular expressions specify the same familiy of languages.

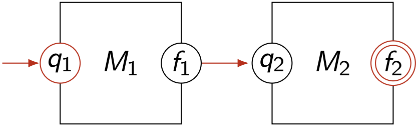
- from RegEx to FA
  $\hookrightarrow$Thompson's construction
- from FA to RegEx
  $\hookrightarrow$McNaughton and Yamada
  State elimination $\hookrightarrow$Brzozowski et McCluskey

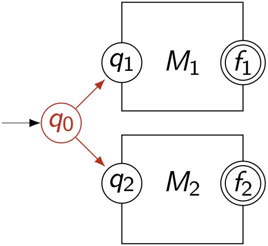
Theorem

If L is a regular language, then there exists an NFA that accepts L.







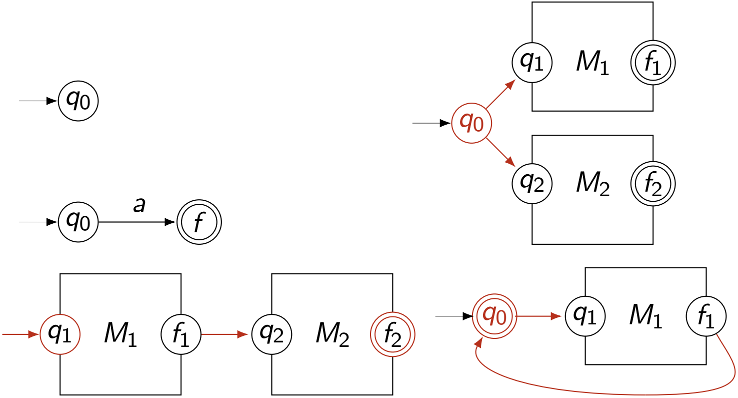


NFA for L^*...

[M] Th 3.25 [L] Th 3.1

Theorem

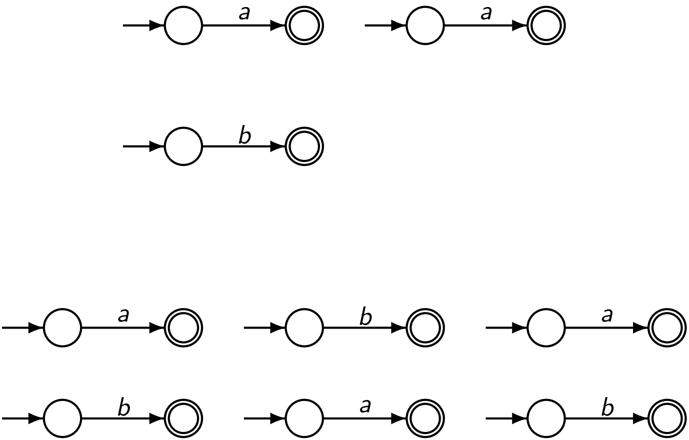
If L is a regular language, then there exists an NFA that accepts L.



[M] Th 3.25 [L] Th 3.1

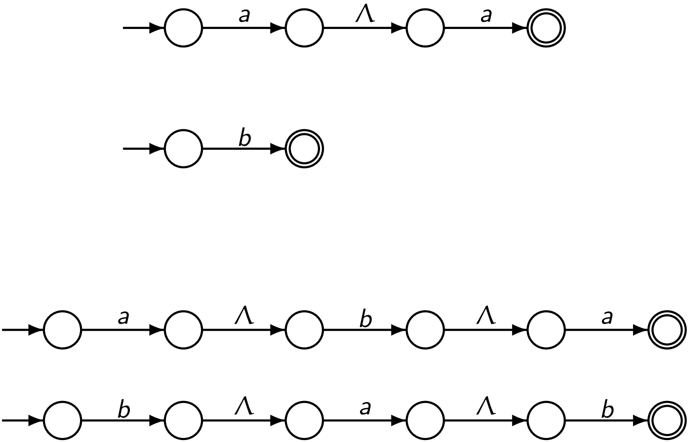
Example 3.28. An NFA Corresponding to $((aa+b)^*(aba)^*bab)^*$

Step 1



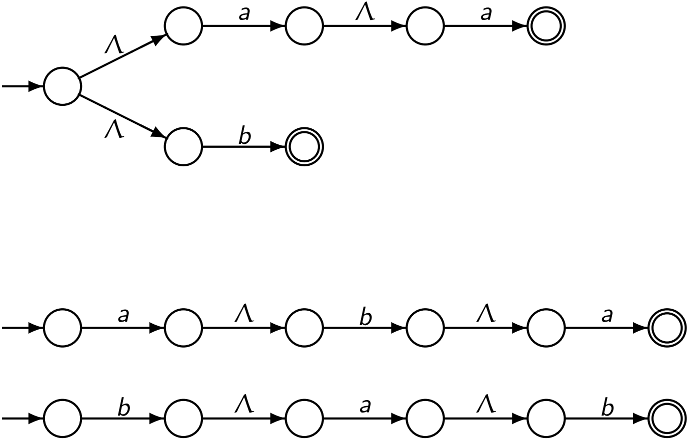
Example 3.28. An NFA Corresponding to $((aa + b)^*(aba)^*bab)^*$

Step 2



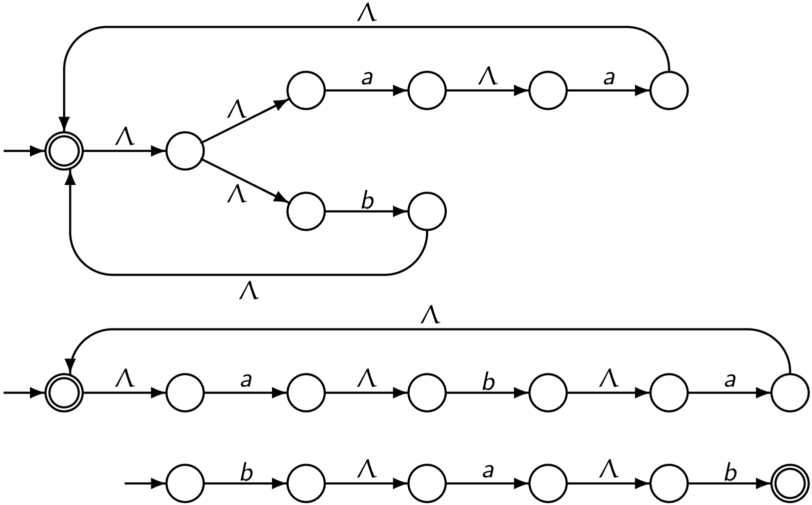
Example 3.28. An NFA Corresponding to $((aa+b)^*(aba)^*bab)^*$

Step 3



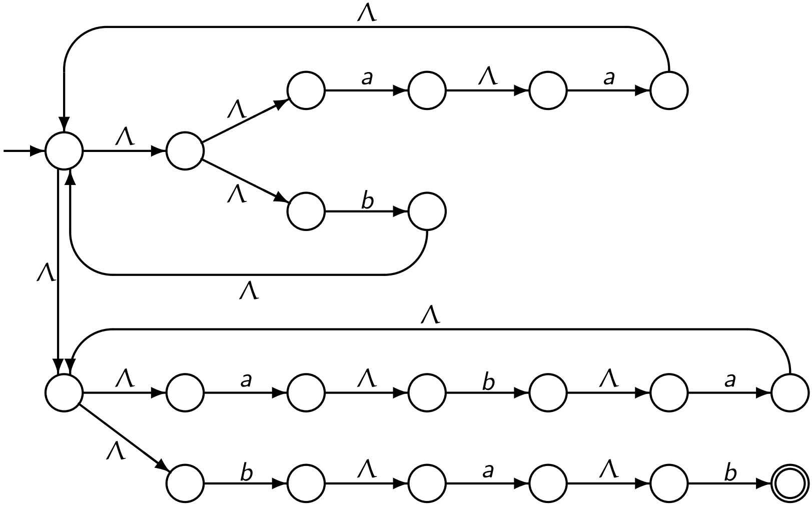
Example 3.28. An NFA Corresponding to $((aa + b)^*(aba)^*bab)^*$

Step 4



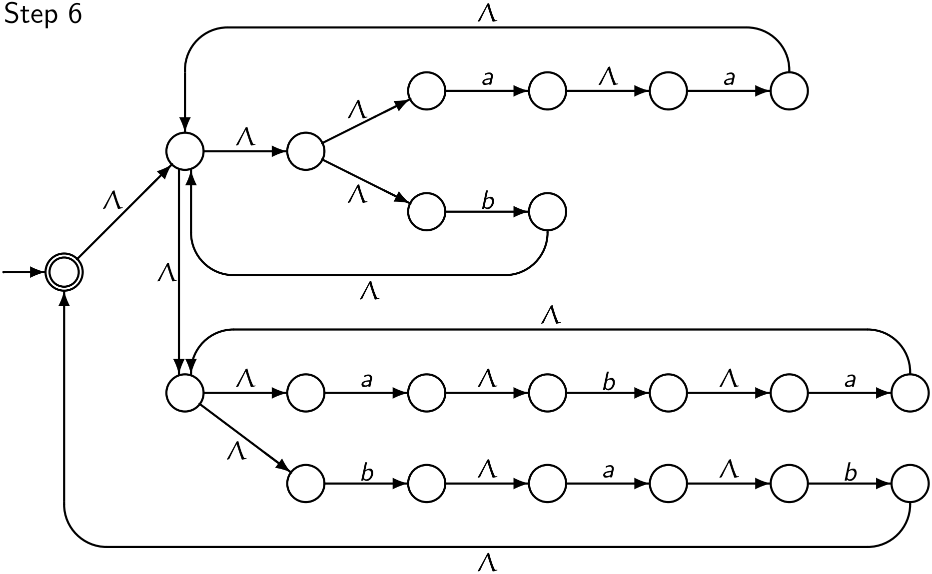
Example 3.28. An NFA Corresponding to $((aa + b)^*(aba)^*bab)^*$

Step 5

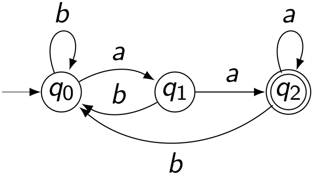


Example 3.28. An NFA Corresponding to $((aa+b)^*(aba)^*bab)^*$

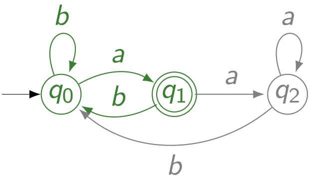
Step 6

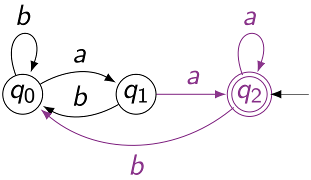


Intro: finding a regular expression



Intro: finding a regular expression





$\underbrace{(b + ab)^*}_{\text{loop on } q_0} a$

$\underbrace{b\,[(b + ab)^*a]\,a + a}_{\text{single loop on } q_2}$

$\underbrace{[(b + ab)^*aa]}_{\text{from } q_0 \text{ to } q_2}\underbrace{[b(b + ab)^*aa + a]^*}_{\text{loop on } q_2}$

short answer $(a + b)^*aa$ see ↪FA example

ABOVE

It is possible to construct an expression for a small automaton "by hand" by starting with a restricted version of the automaton, and slowly adding nodes and edges.

BELOW

Next a formal proof how this can be done generally, referred to as the McNaughtonYamada algorithm.

The expression is built iteratively. First we consider only paths in the automaton that can not pass any node: we only consider single edges. Then we add the nodes one by one. Regular expression $r^k(i,j)$ includes all strings from paths from i to j that only pass by nodes from 1 to k. (We always may exit or enter any other node, but only as first or last node of the path.)

LATER

The method of Brzozowski and McCluskey below "implements" this proof, using a generalized automaton. It features graphs with edges that carry regular expressions.

Theorem

If M is an FA, then $L(M)$ is regular.

PROOF

$M = (Q, \Sigma, q_0, A, \delta)$ assume $Q = \{1, 2, \ldots, n\}$ $q_0 = 1$

$L^k(i, j)$ only paths $i, p_1, \ldots, p_\ell, j$ with $1 \leqslant p_\ell \leqslant k$

[M] Th 3.30

cf. Floyd's algorithm for all-pairs shortest path problem

Theorem

If M is an FA, then $L(M)$ is regular.

PROOF

$M = (Q, \Sigma, q_0, A, \delta)$ assume $Q = \{1, 2, \ldots, n\}$ $q_0 = 1$

$L^k(i, j)$ only paths $i, p_1, \ldots, p_\ell, j$ with $1 \leqslant p_\ell \leqslant k$

$L^0(i, j) = \{a \mid \delta(i, a) = j\}$ $i \neq j$ basis

$L^0(i, j) = \{a \mid \delta(i, a) = j\} \cup \{\Lambda\}$ $i = j$

[M] Th 3.30

Theorem

If M is an FA, then $L(M)$ is regular.

PROOF

$M = (Q, \Sigma, q_0, A, \delta)$ assume $Q = \{1, 2, \ldots, n\}$ $q_0 = 1$

$L^k(i,j)$ only paths $i, p_1, \ldots, p_\ell, j$ with $1 \leqslant p_\ell \leqslant k$

$L^0(i,j) = \{a \mid \delta(i,a) = j\}$ $i \neq j$ basis

$L^0(i,j) = \{a \mid \delta(i,a) = j\} \cup \{\Lambda\}$ $i = j$

one by one add nodes, k from 1 to n:

$L^k(i,j) = \ldots$

[M] Th 3.30

Theorem

If M is an FA, then L(M) is regular.

PROOF

$M = (Q, \Sigma, q_0, A, \delta)$ assume $Q = \{1, 2, \ldots, n\}$ $q_0 = 1$

$L^k(i,j)$ only paths $i, p_1, \ldots, p_\ell, j$ with $1 \leqslant p_\ell \leqslant k$

$L^0(i,j) = \{a \mid \delta(i,a) = j\}$ $i \neq j$ basis

$L^0(i,j) = \{a \mid \delta(i,a) = j\} \cup \{\Lambda\}$ $i = j$

one by one add nodes, k from 1 to n:

$$L^k(i,j) = L^{k-1}(i,j) \;\cup\; \underbrace{L^{k-1}(i,k)}_{\text{from } i \text{ to } k} \cdot \underbrace{\left(L^{k-1}(k,k) \right)^*}_{\text{loop from } k \text{ to } k} \cdot \underbrace{L^{k-1}(k,j)}_{\text{from } k \text{ to } j}$$

[M] Th 3.30

Theorem

If M is an FA, then $L(M)$ is regular.

PROOF

$M = (Q, \Sigma, q_0, A, \delta)$ assume $Q = \{1, 2, \ldots, n\}$ $q_0 = 1$

$L^k(i,j)$ only paths $i, p_1, \ldots, p_\ell, j$ with $1 \leqslant p_\ell \leqslant k$

$L^0(i,j) = \{a \mid \delta(i,a) = j\}$ $i \neq j$ basis

$L^0(i,j) = \{a \mid \delta(i,a) = j\} \cup \{\Lambda\}$ $i = j$

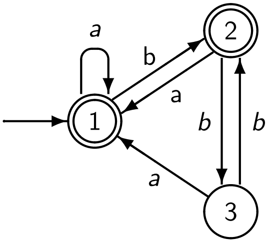
one by one add nodes, k from 1 to n:

$$L^k(i,j) = L^{k-1}(i,j) \cup \underbrace{L^{k-1}(i,k)}_{\text{from } i \text{ to } k} \cdot \underbrace{\left(L^{k-1}(k,k) \right)^*}_{\text{loop from } k \text{ to } k} \cdot \underbrace{L^{k-1}(k,j)}_{\text{from } k \text{ to } j}$$

$$L(M) = \bigcup_{j \in A} L^n(1,j)$$ full language, all nodes

[M] Th 3.30

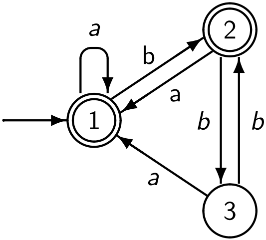
$r^k(i,j)$ expression for $L^k(i,j)$



$r^0(i,j)$	$j=1$	2	3
$i=1$	$a+\Lambda$	b	$\varnothing$
2	a	Λ	b
3	a	b	Λ

[M] E 3.32

$r^k(i,j)$ expression for $L^k(i,j)$



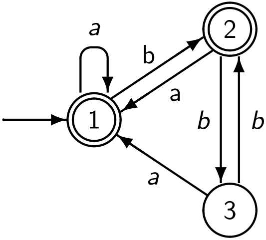
$r^0(i,j)$	$j=1$	2	3
$i=1$	$a+\Lambda$	b	$\varnothing$
2	a	Λ	b
3	a	b	Λ

$r^1(i,j)$	$j=1$	2	3
$i=1$	a^*	a^*b	$\varnothing$
2	aa^*	$\Lambda+aa^*b$	b
3	aa^*	a^*b	Λ

Simplified

[M] E 3.32

$r^k(i,j)$ expression for $L^k(i,j)$

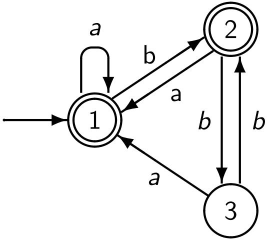


$r^1(i,j)$	$j=1$	2	3
$i=1$	a^*	a^*b	$\varnothing$
2	aa^*	$\Lambda + aa^*b$	b
3	aa^*	a^*b	Λ

$r^2(i,j)$	$j=1$	2	3
$i=1$	$a^*(baa^*)^*$	$a^*(baa^*)^*b$	$a^*(baa^*)^*bb$
2	$aa^*(baa^*)^*$	$(aa^*b)^*$	$(aa^*b)^*b$
3	$aa^* + a^*baa^*(baa^*)^*$	$a^*b(aa^*b)^*$	$\Lambda + a^*b(aa^*b)^*b$

Simplified / rewritten

$r^k(i,j)$ expression for $L^k(i,j)$



$r^2(i,j)$	$j=1$	2	3
$i=1$	$a^*(baa^*)^*$	$a^*(baa^*)^*b$	$a^*(baa^*)^*bb$
2	$aa^*(baa^*)^*$	$(aa^*b)^*$	$(aa^*b)^*b$
3	$aa^* + a^*baa^*(baa^*)^*$	$a^*b(aa^*b)^*$	$\Lambda + a^*b(aa^*b)^*b$

$$r^3(1,1) = r^2(1,1) + r^2(1,3)r^2(3,3)^*r^2(3,1)$$
$$r^3(1,2) = r^2(1,2) + r^2(1,3)r^2(3,3)^*r^2(3,2)$$

[M] E 3.32