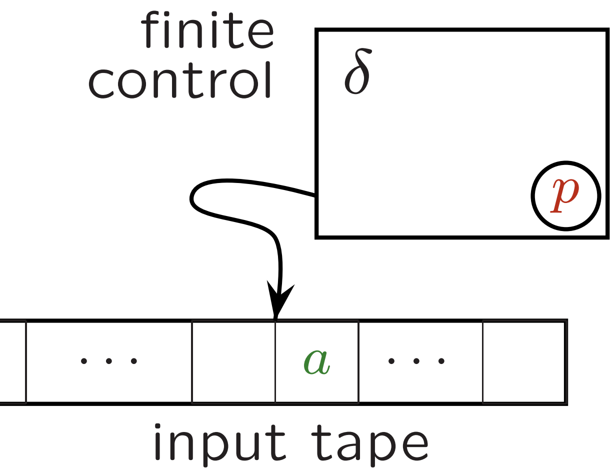
Chapter 3

Finite Automata and Regular Languages

3.0 Review
3.1 Moore and Mealy machines
closure 3.2 Quotients
3.3 Morphisms and substitutions
3.4 Advanced closure properties of regular languages
3.5 Transducers
machines 3.6 Two-way finite automata
3.7 The transformation automaton
3.8 Automata, graphs, and Boolean matrices
algebra 3.9 The Myhill-Nerode theorem
3.10 Minimization of finite automata
3.11 State complexity
3.12 Partial orders and regular languages

3.0 Review

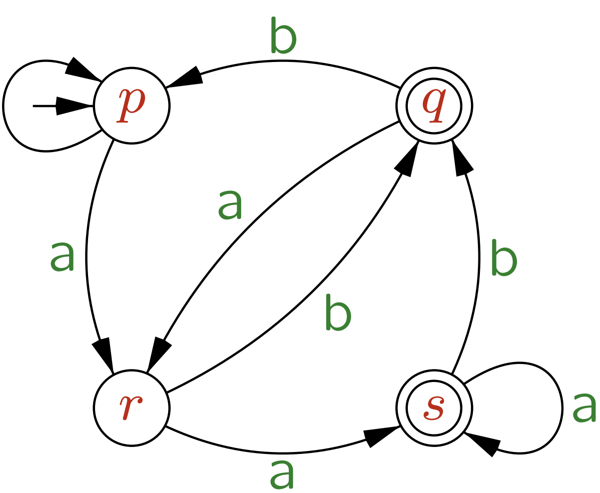


$$\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$$

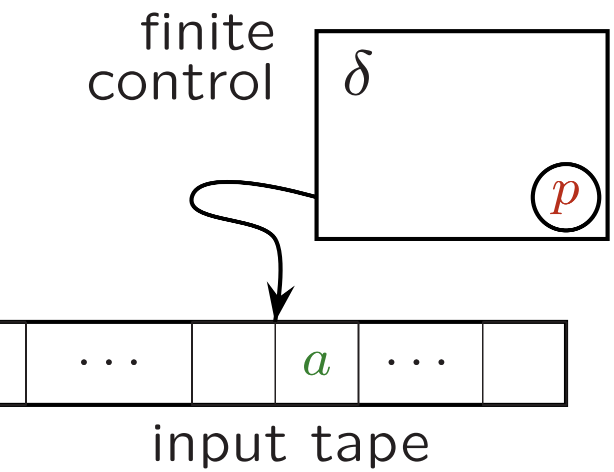
Q	*states*	p, q
$q_0 \in Q$	*initial state*	
$F \subseteq Q$	*final states*	
Σ	*input alphabet*	$a, b \quad w, x$
$\delta : Q \times \Sigma \to Q$	*transition function*	

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

$$\delta^*(q, \epsilon) = q$$
$$\delta^*(q, xa) = \delta(\ \delta^*(q, x),\ a)$$



$$L(\mathcal{A}) = \{\ x \in \Sigma^* \mid \delta(q_0, x) \in F\ \}$$



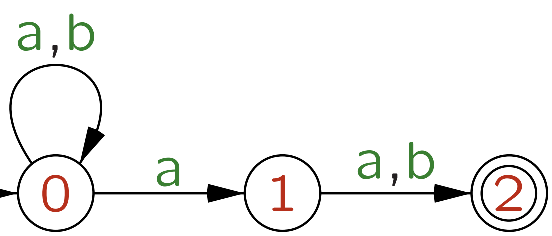
$$\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$$

Q	*states*	p, q
$q_0 \in Q$	*initial state*	
$F \subseteq Q$	*final states*	
Σ	*input alphabet*	$a, b \quad w, x$
$\delta : Q \times \Sigma \to 2^Q$	*transition function*	

$$\delta^* : Q \times \Sigma^* \to 2^Q$$

$$\delta^*(q, \epsilon) = \{q\}$$

$$\delta^*(q, xa) = \bigcup_{r \in \delta^*(q,x)} \delta(r, a)$$



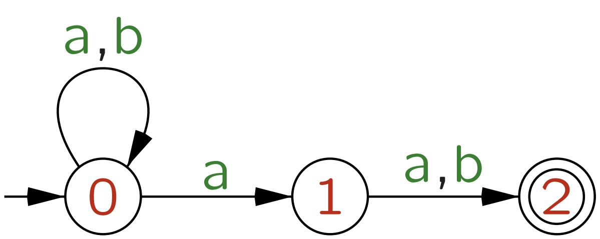
$$L(\mathcal{A}) = \{\ x \in \Sigma^* \mid \delta(q_0, x) \cap F \neq \varnothing\ \}$$

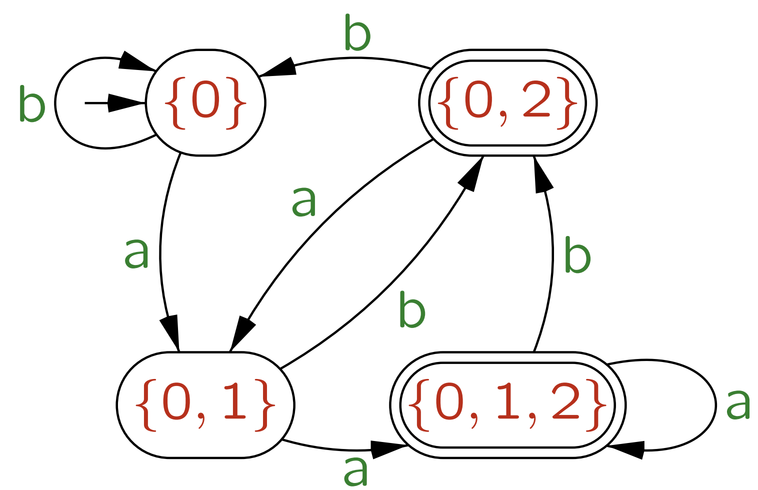
$$\delta(0, \mathrm{a}) = \{0, 1\}$$

$$\delta(2, \mathrm{a}) = \varnothing$$

$$\delta' : 2^Q \times \Sigma \to 2^Q \qquad \text{(deterministic)}$$

$$\delta'(U, a) = \bigcup_{p \in U} \delta(p, a)$$

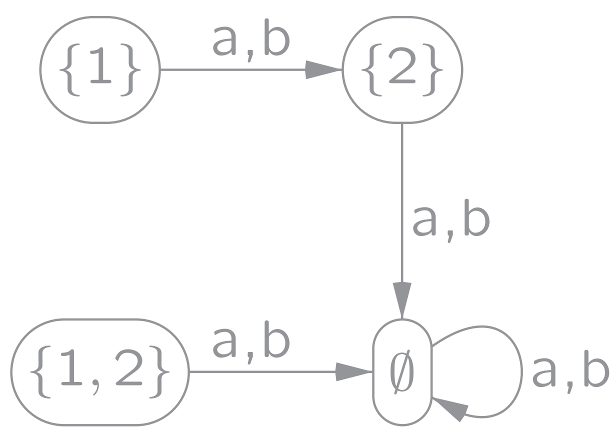


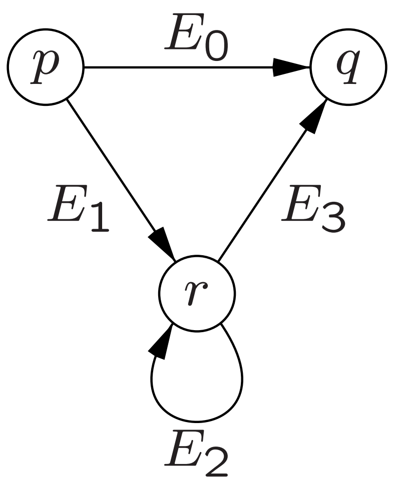


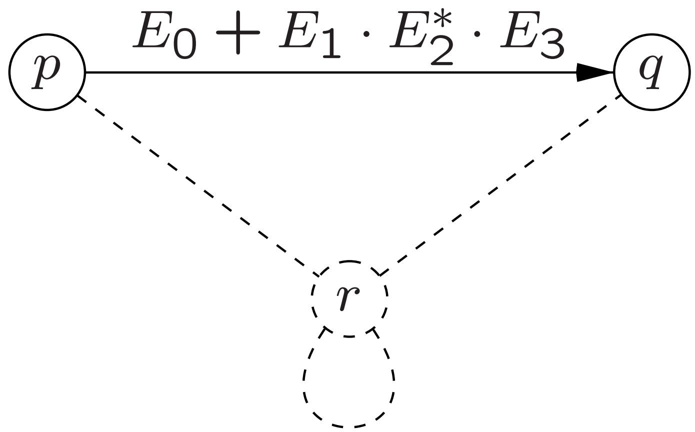
deterministic

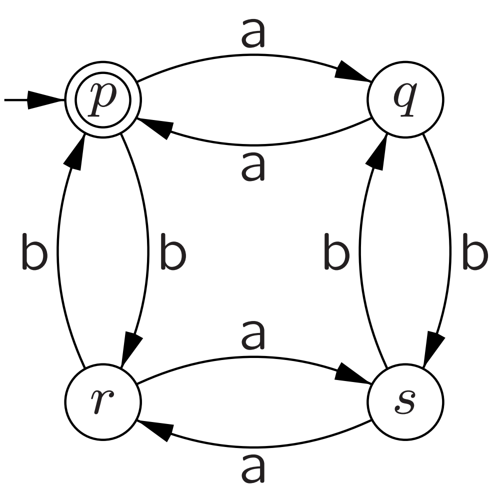
$\delta' : 2^Q \times \Sigma \ \to 2^Q$

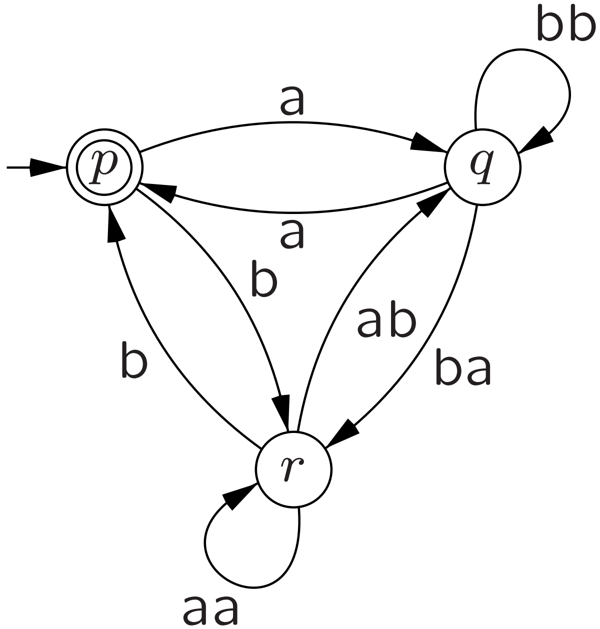
$\delta'(U, a) = \bigcup_{p \in U} \delta(p, a)$

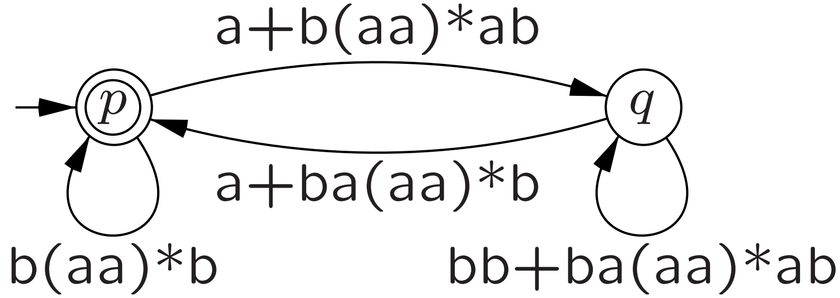



p
E_0
q
E_1
E_3
r
E_2


$E_0 + E_1 \cdot E_2^* \cdot E_3$
p
q
r


a
p
q
a
b
b
b
b
a
r
s
a


bb
a
p
q
a
b
ab
b
ba
r
aa


a+b(aa)*ab
p
q
a+ba(aa)*b
b(aa)*b
bb+ba(aa)*ab

long words can be pumped

$\forall$ for every regular language L
$\exists$ there exists a constant $n \geq 1$
such that
$\forall$ for every $z \in L$
with $|z| \geq n$
$\exists$ there exists a decomposition $z = uvw$
with $|uv| \leq n$, $|v| \geq 1$
such that
$\forall$ for all $i \geq 0$, $uv^iw \in L$

- clever idea, intuition
- formal construction, specification
- show it works, e.g., induction

once the idea is understood,
the other parts might be boring

but essential to test intuition

examples help to get the message

	RLIN REG	DPDA	CF PDAe	DLBA	MON LBA	REC	TYPE0 RE
intersection	+	−	−	+	+	+	+
complement	+	+	−	+	+	+	−
union	+	−	+	+	+	+	+
concatenation	+	−	+	+	+	+	+
star, plus	+	−	+	+	+	+	+
ϵ-free morphism	+	−	+	+	+	+	+
morphism	+	−	+	−	−	−	+
inverse morphism	+	+	+	+	+	+	+
intersect reg lang	+	+	+	+	+	+	+
mirror	+	−	+	+	+	+	+
	fAFL		fAFL	AFL	AFL	AFL	fAFL

$\cap$, ^{c}, $\cup$ boolean operations

$\cup$, $\cdot$, * regular operations

h, h^{-1}, $\cap R$ (full) trio operations

3.1 Moore and Mealy machines

3.2 Quotients

$$\overbrace{x \underbrace{\qquad\qquad\qquad}_{L_2}}^{L_1}$$

$L_1, L_2 \subseteq \Sigma^*$

$L_1/L_2 = \{\, x \in \Sigma^* \mid xy \in L_1 \text{ for some } y \in L_2 \,\}$

Ex. $L_1 = a^+bc^+$, $L_2 = bc^+$, $L_3 = c^+$

$L_1/L_2 = a^+$

$L1/L_3 = a^+bc^*$

Ex. $\text{Pref}(L) = L/\Sigma^*$

Example 3.2.1 and 3.2.5

$$\overbrace{x \quad \underbrace{\qquad\qquad}_{L_2}}^{L_1}$$

$L_1, L_2 \subseteq \Sigma^*$

$L_1/L_2 = \{\, x \in \Sigma^* \mid xy \in L_1$ for some $y \in L_2 \,\}$

Ex. $L = \{\, a^{n^2} \mid n \geq 0 \,\}$

$$L/L = \{\, a^{n^2-m^2} \mid n \geq m \geq 0 \,\} = a(aa)^* + (a^4)^*$$

'$\subseteq$' $m^2 - n^2 = (m+n)(m-n)$

m	n	$m+n$	$m-n$	m^2-n^2
e	e	e	e	mult four
e	o	o	o	odd
o	e	o	o	odd
o	o	e	e	mult four

'$\supseteq$' $(k+1)^2 - k^2 = 2k+1$ odd

$(k+2)^2 - k^2 = 4k+4$ multiple of four

Example 3.2.2

$$\overbrace{x \quad \underbrace{\qquad\qquad\qquad}_{L_2}}^{L_1}$$

$L_1, L_2 \subseteq \Sigma^*$

$L_1/L_2 = \{\, x \in \Sigma^* \mid xy \in L_1 \text{ for some } y \in L_2 \,\}$

can 'hide' computations

Ex. $L_1 = \{\, a^{2n}cba^n \mid n \geq 1 \,\}\{\, ba^{2n}ba^n \mid n \geq 1 \,\}^*ba$

$L_2 = c \cdot \{\, ba^nba^n \mid n \geq 1 \,\}^*$

$L_1/L_2 = \{\, a^{2^n} \mid n \geq 1 \,\}$

$a^{16}\, c\, ba^8\, ba^8\, ba^4\, ba^4\, ba^2\, ba^2\, ba\, ba$

cf. Thm 4.1.6

Thm. $L, R \subseteq \Sigma^*$ If R regular, then R/L regular.

$F' = \{\, q \in Q \mid \delta(q, y) \in F$ for some $y \in L \,\}$.

noncomputable ! (L arbitrary)

REG closed under quotient

REG / REG = REG

(see Ch.4) CF not closed, even

CF / CF = RE

CF / REG = CF

3.3 Morphisms and substitutions

'monoid'

$h : \Sigma \to \Delta^*$

$h : \Sigma^* \to \Delta^*$ $\quad h(xy) = h(x)h(y),\ h(\epsilon) = \epsilon$

$h : 2^{\Sigma^*} \to 2^{\Delta^*}$ $\quad h(L) = \bigcup_{x \in L} h(x)$

$0 \mapsto \mathsf{ab},\ 1 \mapsto \mathsf{ba},\ 2 \mapsto \epsilon$

$00212 \mapsto \mathsf{ababba}$

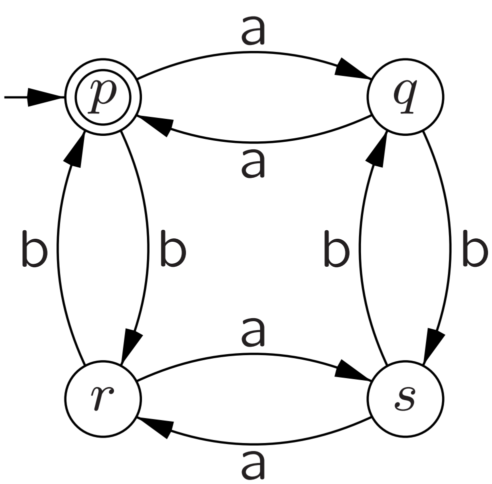
$\{\ 0^n 2 1^n \mid n \geq 0\} \mapsto \{(\mathsf{ab})^n(\mathsf{ba})^n \mid n \geq 0\ \}$

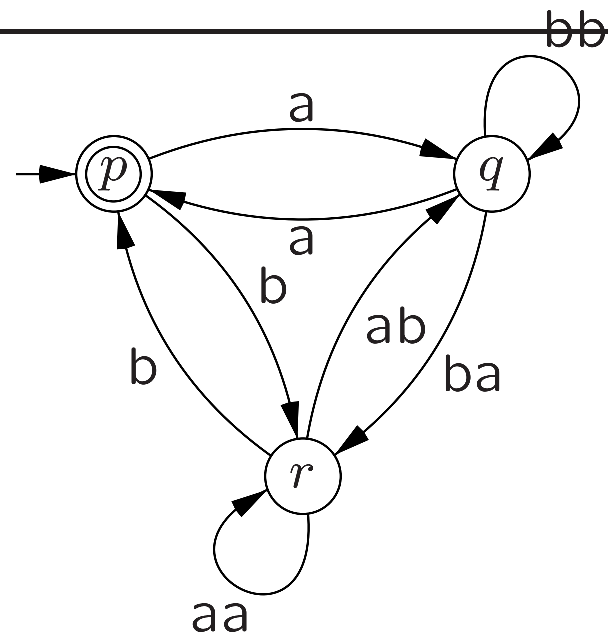
Thm. $h(K \cup L) = h(K) \cup h(L)$

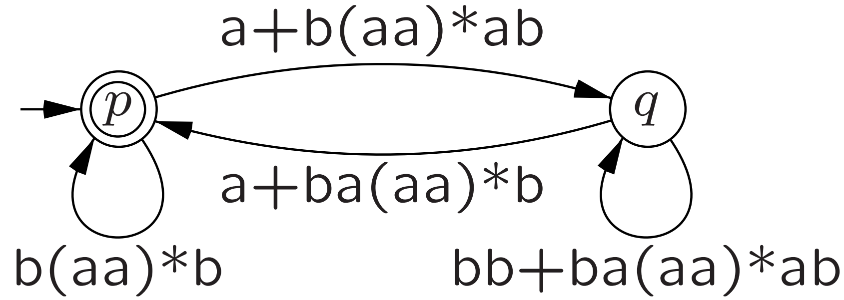
$h(K \cdot L) = h(K) \cdot h(L)$

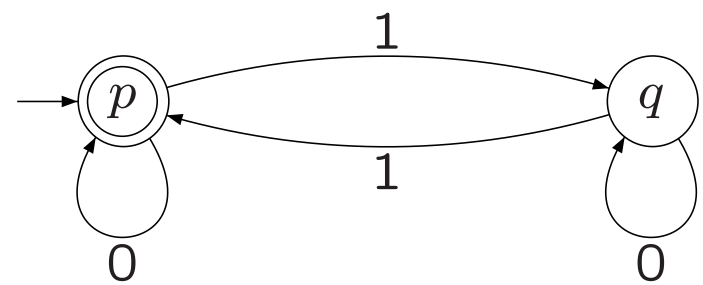
$h(K^*) = h(K)^*$

REG closed under morphisms









$0 \mapsto$ b(aa)*b

$1 \mapsto$ a+b(aa)*ab

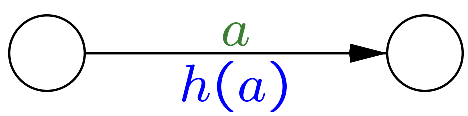
$K = \{x \in \{0,1\}^* \mid \#_1 x \text{ is even}\}$

$s(K) = \{x \in \{a,b\}^* \mid \#_a x, \#_b x \text{ are even}\}$

$$h : \Sigma \to \Delta^*,\ K \subseteq \Delta^*$$

$$h^{-1}(K) = \{\, x \in \Sigma^* \mid h(x) \in K \,\}$$

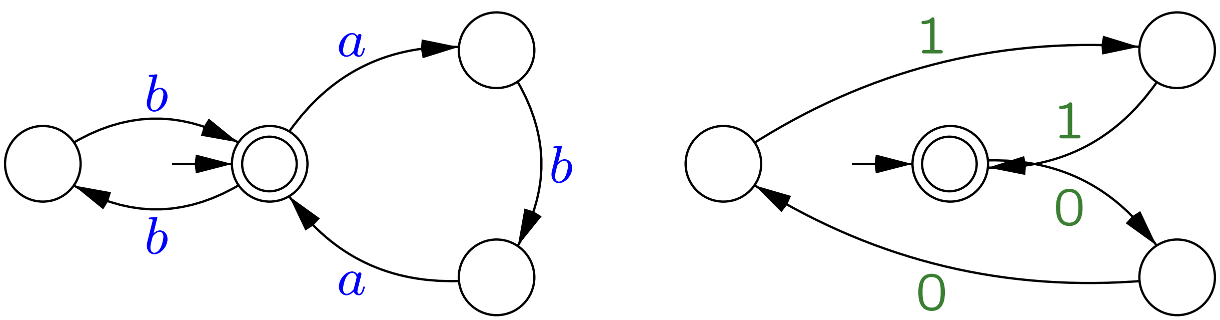
Thm. REG closed under inverse morphism



$$\delta'(p, a) = \delta(p, h(a))$$

$$h : 0 \mapsto ab,\ 1 \mapsto ba$$

$$h^{-1}(\ \{bb, aba\}^*\ ) = \{0011\}^*$$



Theorem 3.3.9

$\mathrm{shuff}(K, L) = K \parallel L$ shuffle

$abb \parallel aca =$

$\{aabbca, aabcba, aabcab, aacabb, aacbab, aacbba, abbaca,$
$ababca, abacba, abacab, acabba, acabab, acaabb\}$

$x \parallel \epsilon = \epsilon \parallel x = \{x\}$

$ax \parallel by = a(x \parallel by) \cup b(ax \parallel y)$

$K \parallel L = \bigcup_{x \in K, y \in L} x \parallel y$

Thm. K, L regular, then $K \parallel L$ regular.

but where is that stated?

Ex. 3.3.8

$abbba \parallel acac \ni abacbacba$

$K \parallel L$ using morphisms, intersection

copies of alphabet

Σ, $\Sigma_1 = \{ a_1 \mid a \in \Sigma \}$, $\Sigma_2 = \{ a_2 \mid a \in \Sigma \}$

$h_1 : \Sigma_1 \cup \Sigma_2 \to \Sigma^* \quad a_1 \mapsto a \quad a_2 \mapsto \epsilon$

$h_2 : \Sigma_1 \cup \Sigma_2 \to \Sigma^* \quad a_1 \mapsto \epsilon \quad a_2 \mapsto a$

$g : \Sigma_1 \cup \Sigma_2 \to \Sigma^* \quad a_1 \mapsto a \quad a_2 \mapsto a$

$$\begin{array}{ccccc} abbba & \xleftarrow{h_1} & a_1b_1a_2c_2b_1a_2c_2b_1a_1 & \xrightarrow{h_2} & acac \\ \in K & & \downarrow g & & \in L \\ & & abacbacba & & \end{array}$$

$$K \parallel L = g(\, h_1^{-1}(K) \cap h_2^{-1}(L)\,)$$

Ex. 3.3.8

3.4 Advanced closure properties of regular languages

$\frac{1}{2}L = \{\ x \in \Sigma^* \mid xy \in L \text{ for } y \text{ with } |y| = |x|\ \}.$

Thm. L regular, then $\frac{1}{2}L$ regular

guess middle state, simulate halves in parallel

$Q' = \{q_0'\} \cup Q \times Q \times Q$ middle, 1st, 2nd

$\delta'(q_0', \varepsilon) = \{[q, q_0, q] \mid q \in Q\}$ ε-move

$\delta'([q,p,r], a) = \{[q, \delta(p,a), \delta(r,b)] \mid b \in \Sigma\}$

$F' = \{[q,q,p] \mid q \in Q, p \in F\}$

$\sqrt{L} = \{\ x \in \Sigma^* \mid xx \in L\ \}.$

$\mathrm{cut}_f L = \{ \, x \mid xy \in L \text{ for } y \text{ with } |y| = f(|x|) \, \}.$

$f(n) = n \qquad \frac{1}{2}L$

$f(n) = 2^n \qquad \log L$ p.76

$f(n) = n^2$

which f ?

see: *transition matrix* (Ch. 3.8)

$\text{cyc}(L) = \{\, x_1x_2 \mid x_2x_1 \in L \,\}.$

Thm. If L is regular, then so is $\text{cyc}(L)$

guess middle,

simulate halves in opposite order

$Q' = \{q'_0\} \cup Q \times Q \times \{0, 1\}$

middle, state, phase

$\delta'(q'_0, \epsilon) = \{\, [q, q, 0] \mid q \in Q\}$ $\quad$ ϵ-move

$\delta'([q, p, i], a) = \{\, [q, \delta(p, a), i] \,\}$

$\delta'([q, q_f, 0], \epsilon) = \{\, [q, q_0, 1] \mid q_f \in Q \,\}$

$F' = \{\, [q, q, 1] \mid q \in Q \,\}$

$\triangleright$

Note that the construction introduces ϵ-moves.

Is this a proof?

The slide gives the intuition ('guess middle') and the formal construction ($\delta'([q, q_f, 0], \epsilon) = \{ [q, q_0, 1] \mid q_f \in Q \}$).

What is missing is the (formal) argument that the construction works, the correctness proof, i.e., that starting with automaton $\mathcal{A}$ for L the constructed automaton $\mathcal{A}'$ actually accepts $\text{cyc}(L)$.

Thus, if there is a computation for xy on $\mathcal{A}$, then there is a computation for yx on $\mathcal{A}'$ (and vice versa).

In informal notation,

$q_0 \overset{x}{\rightsquigarrow} p \overset{y}{\rightsquigarrow} q_f$ in $\mathcal{A}$, then $q_0' \overset{\epsilon}{\rightsquigarrow} [p, p, 0] \overset{y}{\rightsquigarrow} [p, p_f, 0] \overset{\epsilon}{\rightsquigarrow} [p, q_0, 1] \overset{x}{\rightsquigarrow} [p, p, 1]$ in $\mathcal{A}'$.

For the reverse implication we need that indeed all computations in $\mathcal{A}'$ are of this form. $\triangleleft$

3.5 Transducers

FST $\sim$ finite state automaton with output

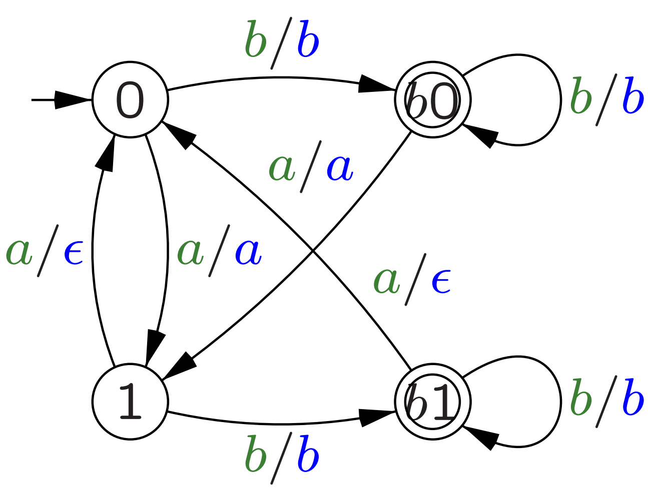
$\mathcal{A} = (Q, \Sigma, \Delta, S, q_{in}, F)$

$S \subseteq Q \times \Sigma^* \times \Delta^* \times Q$

$\ldots (\Sigma \cup \{\epsilon\}) \times (\Delta \cup \{\epsilon\}) \ldots$

$T(\mathcal{A}) \subseteq \Sigma^* \times \Delta^*$ transduction (translation)

$x \to_{\mathcal{A}} y$ rational relation

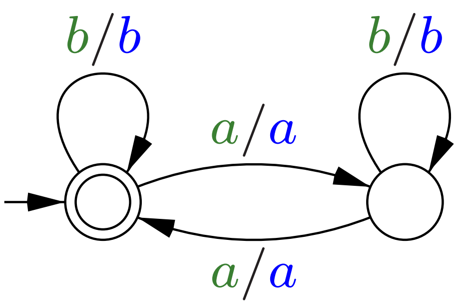


$K \subseteq \Sigma^*$

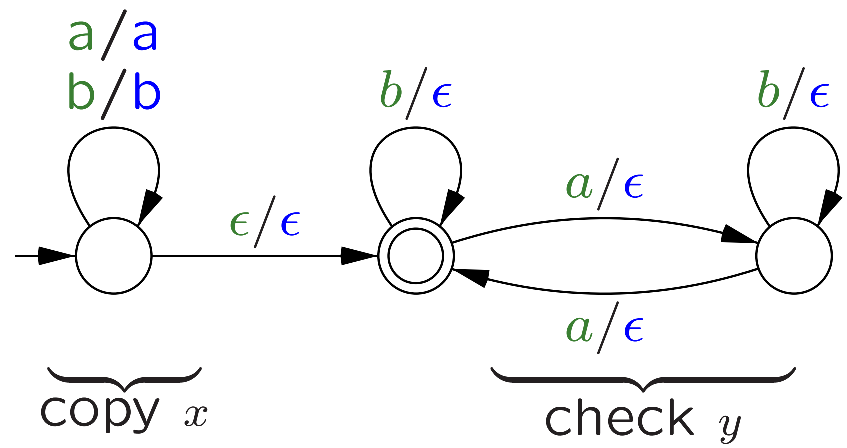
$T(K) = \{ y \in \Delta^* \mid (x, y) \in T(\mathcal{A}), x \in K \}$

erase every 2nd a (keeping words ending in b)

- intersection, quotient, concatenation with regular languages
- morphism, inverse morphism
- prefix, suffix
- . . . erasing every second a



$T(K) = K \cap \{ x \mid \#_a x \text{ even} \}$



$T(K) = \{ x \mid xy \in K \text{ and } \#_a y \text{ even} \}$

$K, L \subseteq \Sigma^*$

$\Sigma' = \{a' \mid a \in \Sigma\} \qquad \Sigma \cap \Sigma' = \varnothing$

$f : \Sigma \cup \Sigma' \to \Sigma \qquad f(a) = f(a') = a$

f^{-1} non-det colouring

$h : \Sigma \to \Sigma' \qquad h(a) = a'$

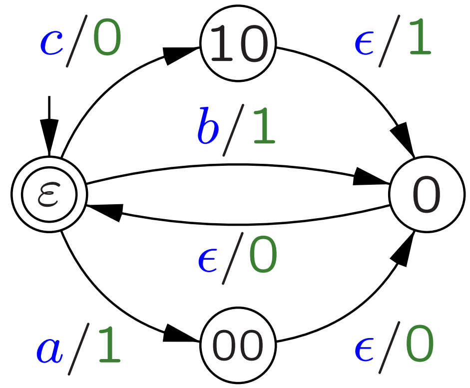
$g : \Sigma \cup \Sigma' \to \Sigma \qquad g(a) = a,\; g(a') = \varepsilon$

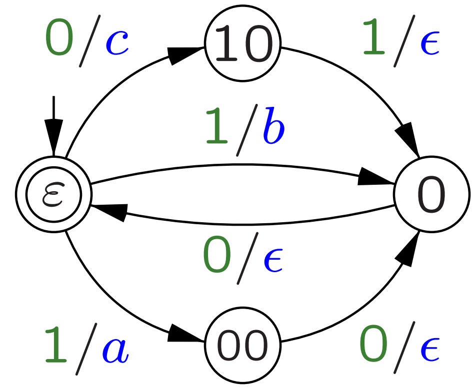
$K/L = g(\, f^{-1}(K) \cap \Sigma^* \cdot h(L)\,)$

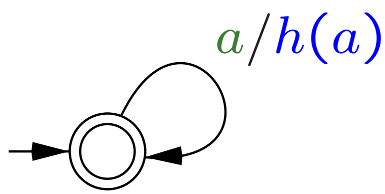
basic *full trio* operations:

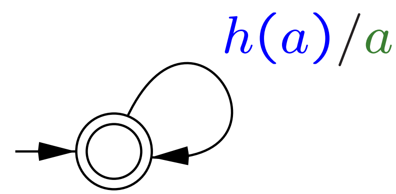
- morphism
- inverse morphism
- intersection regular

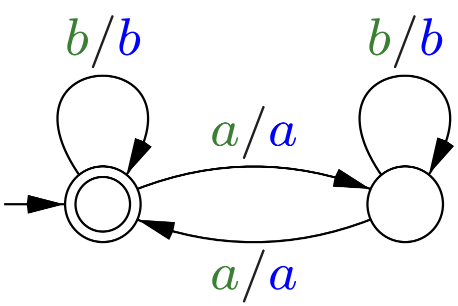
$$h : \begin{cases} a & \mapsto & 100 \\ b & \mapsto & 10 \\ c & \mapsto & 010 \end{cases}$$











- every 'basic' full trio operation is FST
- FST's are closed under composition

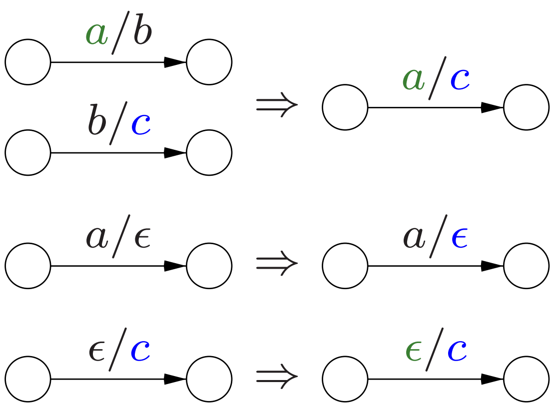
$\Rightarrow$ sequence of full trio op's is FST

$$\text{FST } \mathcal{A}_i = (Q, \Sigma_i, \Sigma_{i+1}, S_i, q_{io}, F_i)$$

$$T(\mathcal{A}_1)T(\mathcal{A}_2) \Rightarrow \text{FST } \mathcal{A}' = (Q', \Sigma_1, \Sigma_3, S', q_0', F')$$

formally – $Q' = Q_1 \times Q_2$
– $q_{in}' = \langle q_{10}, q_{20} \rangle$
– $F' = F_1 \times F_2$, and
– S' is defined by

if $(p_1, a, b, q_1) \in S_1$, and $(p_2, b, c, q_2) \in S_2$
(with $b \neq \epsilon$)
then $(\,\langle p_1, p_2\rangle, a, c, \langle q_1, q_2\rangle\,) \in S'$



if $(p_1, a, \epsilon, q_1) \in S_1$ and $p \in Q_2$,
then $(\,\langle p_1, p\rangle, a, \epsilon, \langle q_1, p\rangle,\,) \in S'$

if $p \in Q_1$ and $(p_2, \epsilon, c, q_2) \in S_2$,
then $(\,\langle p, p_2\rangle, \epsilon, c, \langle p, q_2\rangle) \in S'$

‘implicit $(p, \epsilon, \epsilon, p)$’

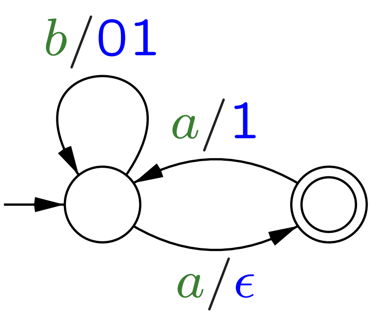
every full trio operation is a fs transduction

Thm. every FST is composition of full trio op's

$R_\mathcal{M}$ regular language over 'transitions'

$\{ \ a{:}\epsilon, \ a{:}1, \ b{:}01 \ \}$

h and g select input and output



$$\begin{array}{ccccccccc} K & \ni & b & b & a & a & b & a \\ & & & \uparrow h & & & & \\ R_\mathcal{M} & \ni & b{:}01 & b{:}01 & a{:}\epsilon & a{:}1 & b{:}01 & a{:}\epsilon \\ & & & \downarrow g & & & & \\ T_\mathcal{M}(K) & \ni & 01 & 01 & \epsilon & 1 & 01 & \epsilon \end{array}$$

$$\underset{\text{input}}{x} \xleftarrow{h} \underset{\text{computation}}{R} \xrightarrow{g} \underset{\text{output}}{y}$$

$$\boxed{T_\mathcal{M}(K) = g(\ h^{-1}(K) \ \cap \ R_\mathcal{M} \ )}$$

Theorem 3.5.3

closure properties

trio $\equiv$ faithful cone :

morphism, ϵ-free morphism, intersection regular

full trio $\equiv$ cone :

. . . , (arbitrary) morphism, . . .

(full) semi-AFL : (full) trio & union

(full) AFL : (full) semi-AFL & concatenation, Kleene plus

3.6 Two-way finite automata

like TM may move in both directions,
no writing, tape bounded

$\mathcal{M} = (Q, \Sigma, \delta, q_0, F)$

$\delta : Q \times (\Sigma \cup \{\triangleright, \triangleleft\}) \to Q \times \{L, R\}$

$\triangleright$tape markers$\triangleleft$ (**!**) $\quad \delta(\cdot, \triangleright) = (\cdot, R), \quad \delta(\cdot, \triangleleft) = (\cdot, L)$

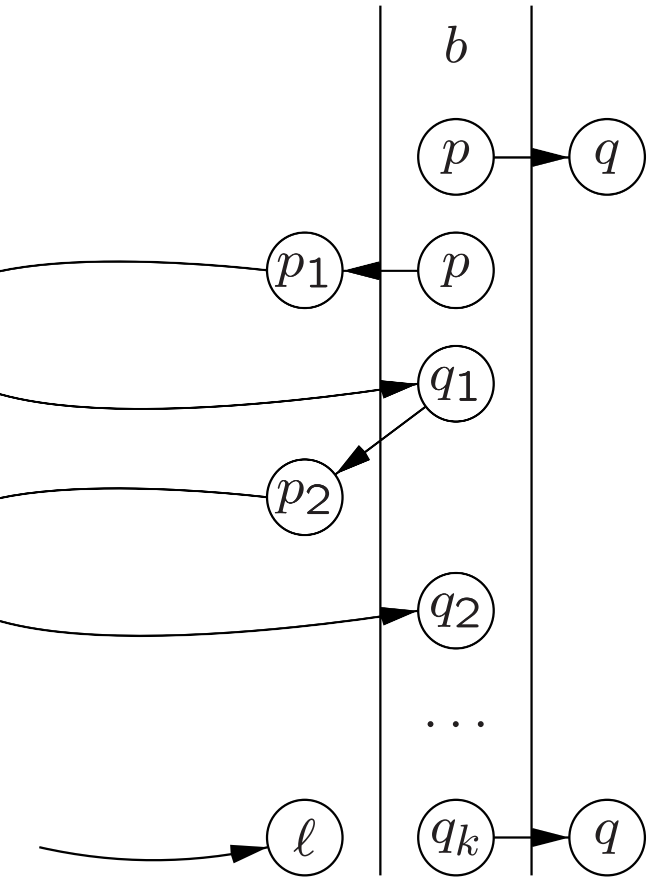
configuration $\triangleright \Sigma^* Q \Sigma^* \triangleleft \ \cup Q \triangleright \Sigma^* \triangleleft$

$wqax \vdash wapx$ when $\delta(q, a) = (p, R)$ move
$waqx \vdash wpax \qquad \delta(q, a) = (p, L)$

infinite loops possible !

$L(\mathcal{M}) = \{ \ w \in \Sigma^* \mid q_0 \triangleright w \triangleleft \vdash^* \triangleright wp \triangleleft, p \in F \ \}$

Shefferdson [1959] $\quad$ 2DFA $\subseteq$ DFA



keep track of 'excursions' to the left

$\tau : Q \cup \{\bar{q}\} \to Q \cup \{\ell\}$ $\quad$ $\bar{q}$ final, ℓ for loop

updating τ $\quad$ τ_x to τ_{xb}

$\delta(p, b) = (q, R)$ then $\tau_{xb}(p) = q$

$\delta(p, b) = (p_1, L)$, $\tau_x(p_1) = q_1$,

$\qquad \delta(q_1, b) = (p_2, L), \ldots, \tau_x(p_k) = q_k$

until one of the following occurs

if $q_k = \ell$ then $\tau_{xb}(p) = \ell$

if $\delta(q_k, b) = (q, R)$ then $\tau_{xb}(p) = q$

if $q_k = q_i$ or $q_k = p$ then $\tau_{xb}(p) = \ell$

$\tau_x(\bar{q}) = \delta(q_0, x)$

$\text{root}(L) = \{ w \in \Sigma^* \mid w^n \in L \text{ for some } n \geq 1 \}$

Thm. root(L) is regular (for regular L)

simulate $\mathcal{M}$ for L on $\triangleright w \triangleleft$
accept right when $\mathcal{M}$ accepts
otherwise continue left at state reached

also $\frac{1}{2}(L)$ can be solved this way

3.7 The transformation automaton
3.8 Automata, graphs, and Boolean matrices

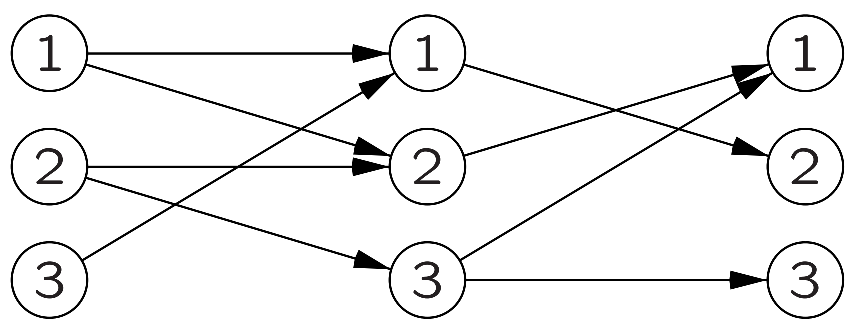
$C = AB$

$C_{ij} = \sum_{i=1}^{n} A_{ik} B_{kj}$

number of connections

Boolean

$C_{ij} = \bigvee_{k=1}^{n} A_{ik} \wedge B_{kj}$ exists connection

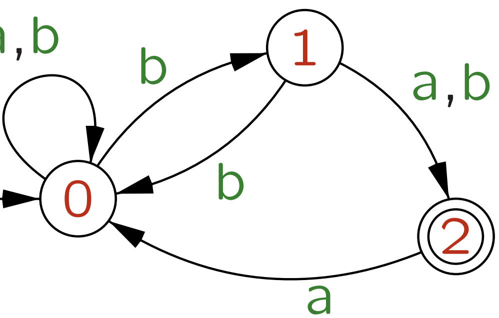


$$\begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

$Q = \{q_0, q_1, \ldots, q_{n-1}\}$ (ordered)

M_a Boolean matrix

$(M_a)_{ij} = 1$ iff $\delta(q_i, a) \ni q_j$



$$M_{\mathsf{a}} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix} \quad M_{\mathsf{b}} = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}$$

$(M_w)_{ij} = 1$ iff $\delta(q_i, w) \ni q_j$

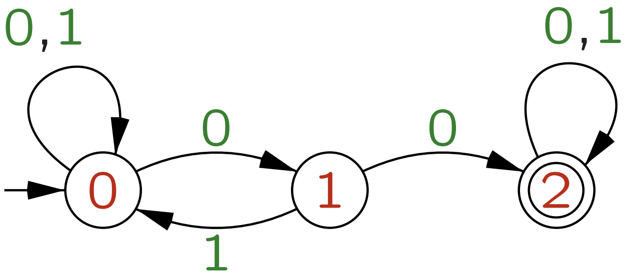
$$M_{\mathsf{abb}} = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}$$

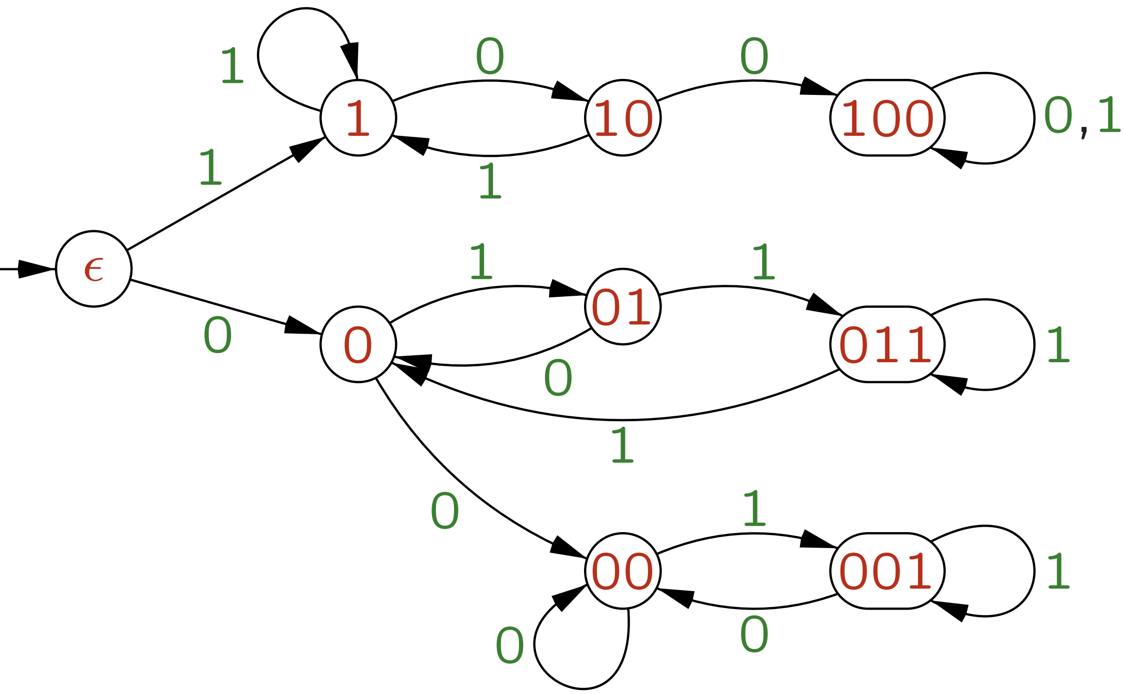
Thm. for $w = a_1 a_2 \ldots a_t$, $a_i \in \Sigma$

$$M_w = M_{a_1} M_{a_2} \cdot \ldots \cdot M_{a_t}$$

Cor. $M_{xy} = M_x M_y$

Theorem 3.8.1





From diagram $M_{000} = M_{00}$, etc.

transformation automaton

$$M_\epsilon = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

$$M_0 = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix},$$

$$M_1 = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

$$M_{00} = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix},$$

$$M_{01} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix},$$

$$M_{10} = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

$$M_{001} = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix},$$

$$M_{011} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix},$$

$$M_{100} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}$$

characteristic vectors

$u_0 = [1, 0, 0, \ldots, 0]$ (row)

$(u_F)_i = 1$ iff $q_i \in F$ (column)

$(M_w)_{ij} = 1$ iff $\delta(q_i, w) \ni q_j$

Thm. $x \in L(\mathcal{A})$ iff $u_0\, M_x\, u_F = 1$

matrix represents computation

Theorem 3.8.3

nondeterministic case

$\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$

$(M_a)_{ij} = 1$ iff $\delta(q_i, a) \ni q_j$

$w = a_1 a_2 \dots a_t$

$M_w = M_{a_1} M_{a_2} \cdot \ldots \cdot M_{a_t}$

$\mathcal{A} = (Q', \Sigma, \delta', q'_0, -)$

transformation automaton

$Q' = \{0, 1\}^{Q \times Q}$ $0, 1$-matrices

$q'_0 = I$ identity matrix

$\delta(M, a) = M \cdot M_a$

no final states specified

Thm. $\delta'(I, w) = M$, then $M = M_w$

i.e., $(M)_{ij} = 1$ iff $\delta(q_i, w) \ni q_j$

$$\sqrt{L} = \{\, x \in \Sigma^* \mid xx \in L \,\}.$$

states $Q' = \{0,1\}^{Q \times Q}$ (the M_x's)

M_x after reading x

final: $(M_x)_{q_0 p} = 1$ for some $p \in Q$ [unique]
and $(M_x)_{pq} = 1$ for some $q \in F$

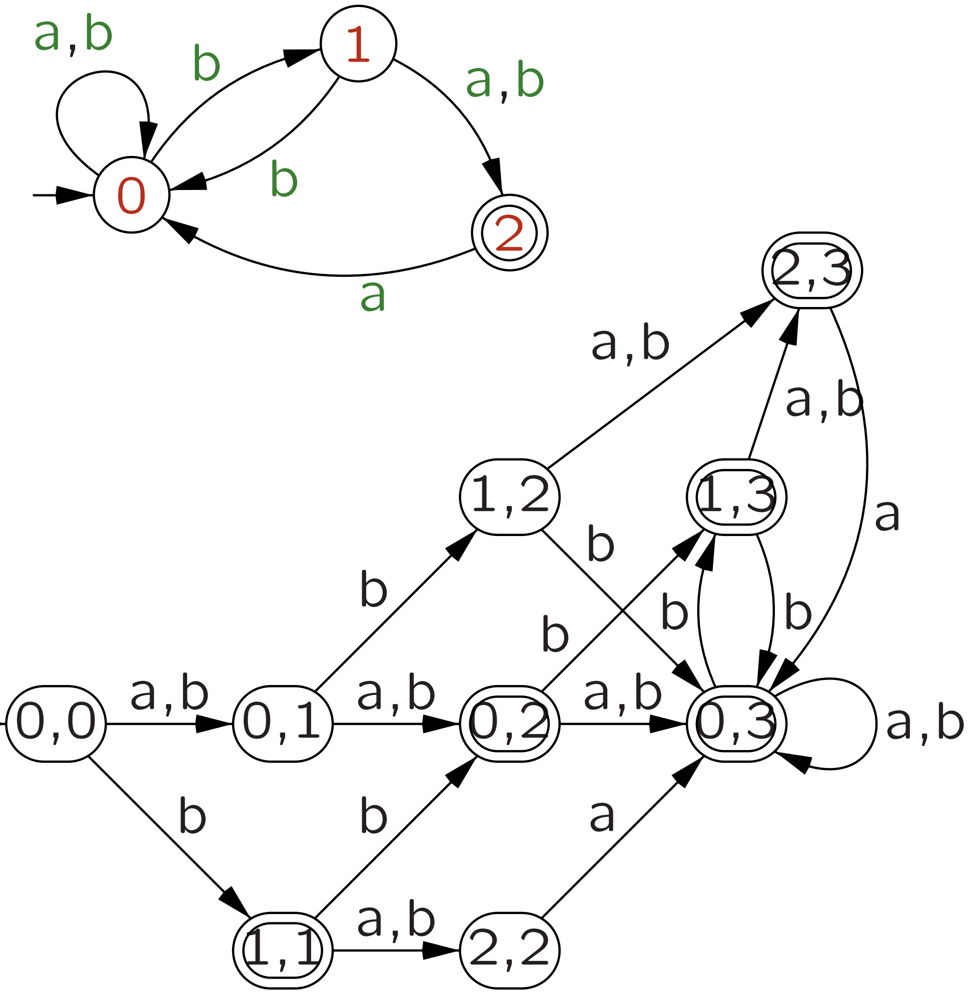
$$\tfrac{1}{2}L = \{\, x \in \Sigma^* \mid xy \in L \text{ for } y \text{ with } |y| = |x| \,\}.$$

$(p,q) \in M^k$ iff $\delta(p,u) = q$ for some u, $|u| = k$.

$$M^{k+1} = M^k M$$

Prop. $\log L = \{\, x \mid xy \in L \text{ for } y \text{ with } |y| = 2^{|x|} \,\}.$

$$M^{2^k} = (M^{2^{k-1}})^2$$



$$M = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}$$

$$M^2 = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \end{pmatrix}$$

$$M^k = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \quad (k \geq 3)$$

$M^k[i,j] = 1$ *iff*
length k path from i to j

product: aut $\times$
transformation aut
$(p, M^k) \xrightarrow{a} (\delta(p,a), M^{k+1})$

monoid $(M, \circ, 1)$
- closed $a \circ b \in M$
- associative $(a \circ b) \circ c = a \circ (b \circ c)$
- identity $a \circ 1 = 1 \circ a = a$

$(\Sigma^*, \cdot, \epsilon)$ strings

$(\mathbb{Z}^{n \times n}, \circ, I)$ $n \times n$-matrices
$(\{0, 1\}^{n \times n}, \circ, I)$ Boolean matrices:
finite monoid

monoid morphism $h : (M, \circ, 1) \to (M', \circ', 1')$
$h : M \to M'$
- $h(a) \circ' h(b) = h(a \circ b)$
- $h(1) = 1'$

Def. $L \subseteq \Sigma^*$ recognizable iff
finite monoid $(M, \circ, 1)$,
monoid morphism $h : \Sigma^* \to M$
$S \subseteq M$ such that $L = h^{-1}(S)$

Cor. $M_{xy} = M_x M_y$

automaton as monoid

$\mu : \Sigma^* \to \{0, 1\}^{Q \times Q}$
$x \mapsto M_x$ is a monoid morphism

Thm. REC = REG (for strings)

monoid as automaton

$\mathcal{A}_M = (M, \Sigma, \delta, 1, S)$
$\delta(m, a) = m \circ h(a) \qquad m \in M,\ a \in \Sigma$

$x \in L(\mathcal{A}_M)$ iff $\delta(1, x) \in S$ iff $h(x) = 1 \circ h(x) \in S$
iff $x \in h^{-1}(S)$

*

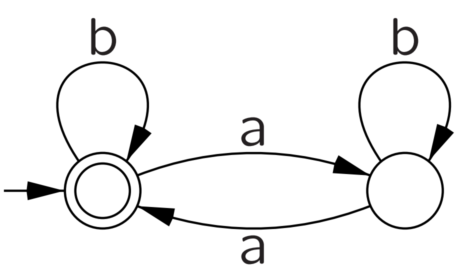
3.9 The Myhill-Nerode theorem

equivalence relation

- reflexive xRx for all x
- symmetric xRy implies yRx
- transitive xRy and yRz imply xRz

equivalence class $E = \{y \in S \mid xRy\}$

index of R



baabb $R_{\mathcal{M}}$ ϵ

baabba $R_{\mathcal{M}}$ a

DFA $\mathcal{M} = (Q, \Sigma, \delta, q_0, F)$

ending in the same state

$xR_{\mathcal{M}}y$ iff $\delta(q_0, x) = \delta(q_0, y)$

equivalence relation on Σ^*

- finite index $|Q|$
- right invariant $xR_{\mathcal{M}}y$ implies $xzR_{\mathcal{M}}yz$

right congruence

- $L(M)$ union of equivalence classes

R_M *saturates* L

$L \subseteq \Sigma^*$

xR_Ly when, for all u, ($xu \in L$ iff $yu \in L$)

equivalence relation on Σ^*

- index may be infinite
- right invariant xR_Ly implies xzR_Lyz
- L union of equivalence classes

R_1, R_2 equivalence relations

R_1 refinement of R_2: $R_1 \subseteq R_2$

Lem. L union of some classes of right-invariant equivalence relation E.
Then E refinement of R_L

Prf. xEy (right-invariant) $\Rightarrow xzEyz$ for all z (union of classes) $\Rightarrow xz \in L$ iff $yz \in L$ for all $z \Rightarrow xR_Ly$

$L \subseteq \Sigma^*$

xR_Ly when, for all u, ($xu \in L$ iff $yu \in L$)

xR_Ly iff $x^{-1}L = y^{-1}L$

$x^{-1}L = \{ u \mid xu \in L \}$

$x^{-1}L$ may contain

ϵ

$\{a,b\}^*a$

even b's (≥ 2)

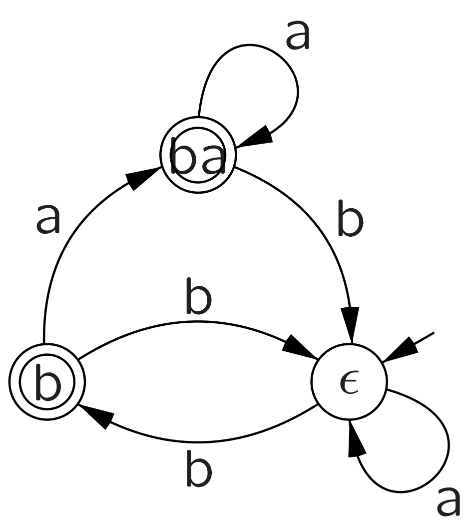
odd b's

	ϵ	a	b	bb
ϵ	$\surd$	$\surd$	–	$\surd$
b	–	$\surd$	$\surd$	–
ba	$\surd$	$\surd$	$\surd$	–

(x,y): $xy \in L$

$L = \{ x \in \{a,b\}^* \mid x$ ends in a or even b's $\}$

[ϵ]	even number b's	[a]=[ϵ], [b]
[b]	odd b's, ending in b	[ba], [bb]=[ϵ]
[ba]	odd b's, ending in a	[baa]=[ba], [bab]=[ϵ]



Theorem 3.9.5

Thm. $L \subseteq \Sigma^*$. equivalent:

a. L regular

b. L is union of equivalence classes of right-invariant equivalence relation E of finite index

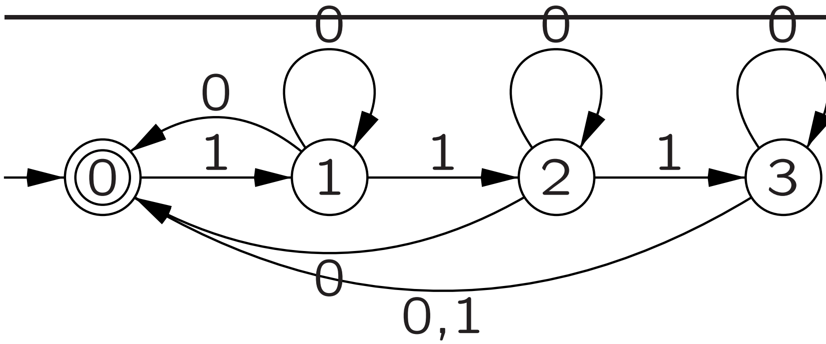
c. R_L has finite index

a.⇒b. $R_{\mathcal{M}}$ for automaton $\mathcal{M}$

b.⇒c. E is a refinement of R_L. index $R_L \leq$ index E

c.⇒a. use equivalence classes as states
$\delta([x], a) = [xa]$

automaton is 'inside' the language

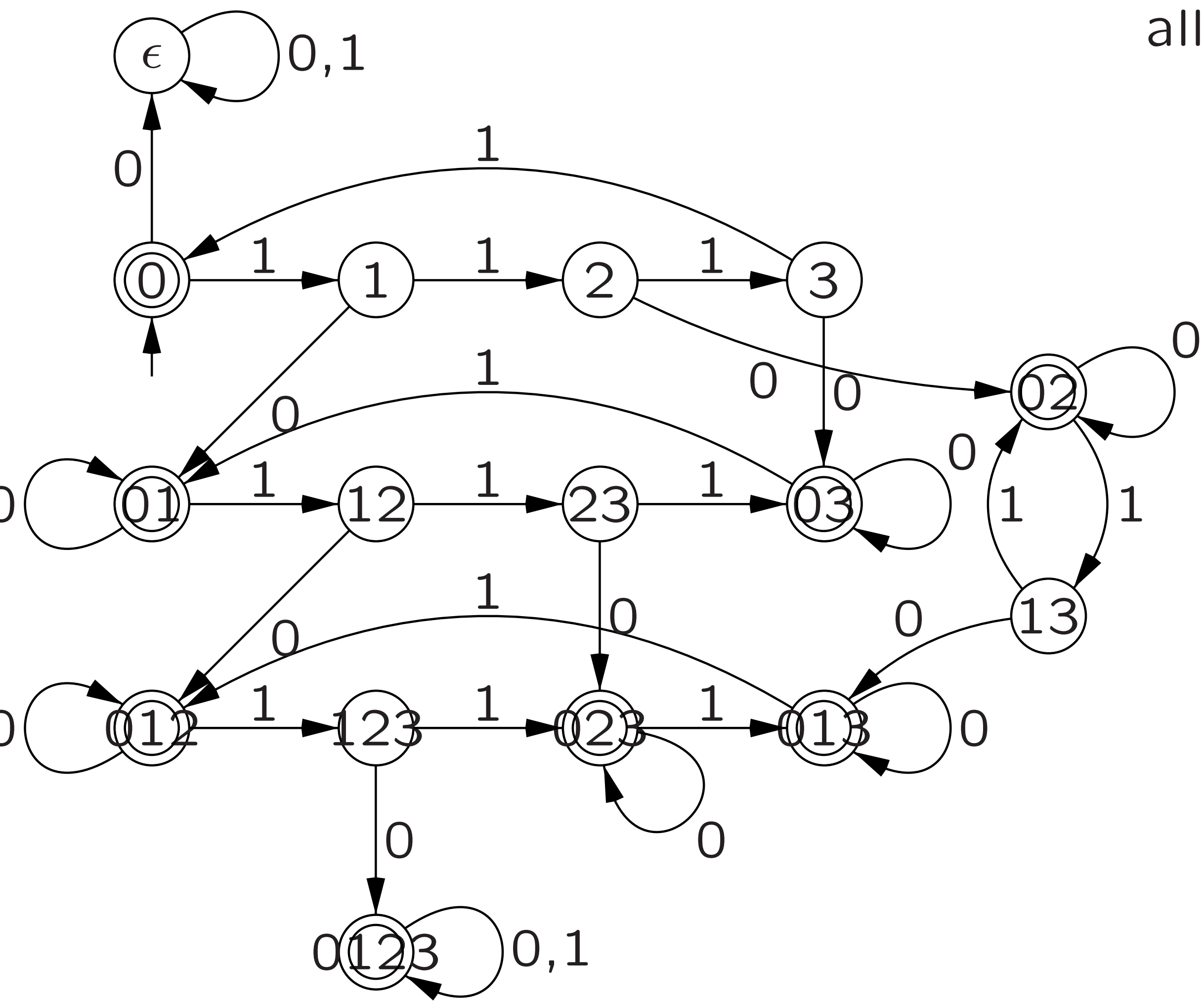


n state nfa

2^n state dfa

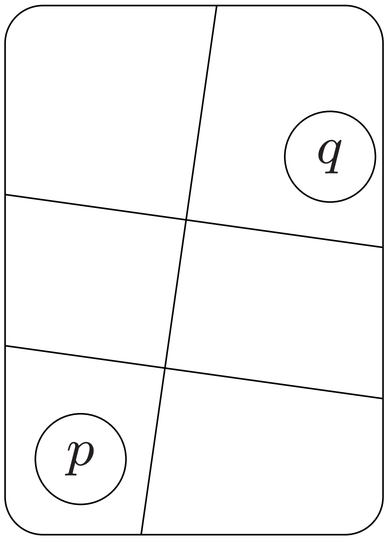
all reachable

all nonequivalent



Theorem 3.9.6

3.10 Minimization of finite automata



Σ^*/R_L

$L \subseteq \Sigma^*$

xR_Ly when, for all u, ($xu \in L$ iff $yu \in L$)

$xR_{\mathcal{M}}y$ when $\delta(q_0, x) = \delta(q_0, y)$

$xR_{\mathcal{M}}y$ then xR_Ly

Myhill-Nerode: R_L-classes $\rightsquigarrow$ automaton

$\delta([x], a) = [xa]$

Thm. unique minimal (det) automaton for L

$\mathcal{M} = (Q, \Sigma, \delta, q_0, F)$

$\mu : Q \to \Sigma^*/R_L$

$q \mapsto [x]$, such that $\delta(q_0, x) = q$

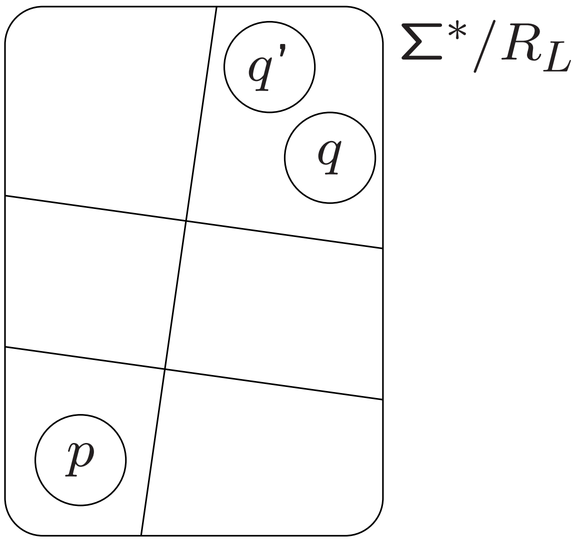
well-defined ($R_{\mathcal{M}}$ refines R_L)

surjective ($q = \delta(q_0, x) \mapsto [x]$)

injective (surjective, same number states)

respects transitions (right invariant)

Theorem 3.10.1



$\mathcal{M} = (Q, \Sigma, \delta, q_0, F)$ dfa for L

xR_Ly when, for all u, ($xu \in L$ iff $yu \in L$)

$xR_{\mathcal{M}}y$ when $\delta(q_0, x) = \delta(q_0, y)$

$xR_{\mathcal{M}}y$ then xR_Ly

$p \equiv q$ indistinguishable

$\delta(p, z) \in F$ iff $\delta(q, z) \in F$

$\mu : Q \to \Sigma^*/R_L$

$q \mapsto [x]$, such that $\delta(q_0, x) = q$

well-defined ($R_{\mathcal{M}}$ refines R_L)

surjective ($p = \delta(q_0, x) \mapsto [x]$)

may not be injective

respects transitions (right invariant)

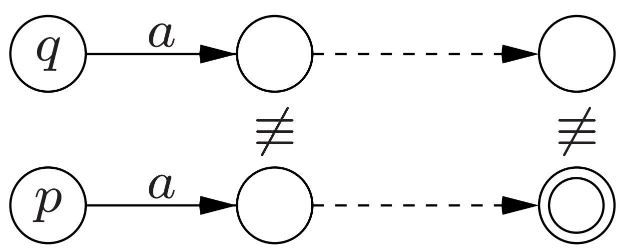
$p = \delta(q_0, x)$, $q = \delta(q_0, y)$

xR_Ly (or $[x] = [y]$) iff $p \equiv q$

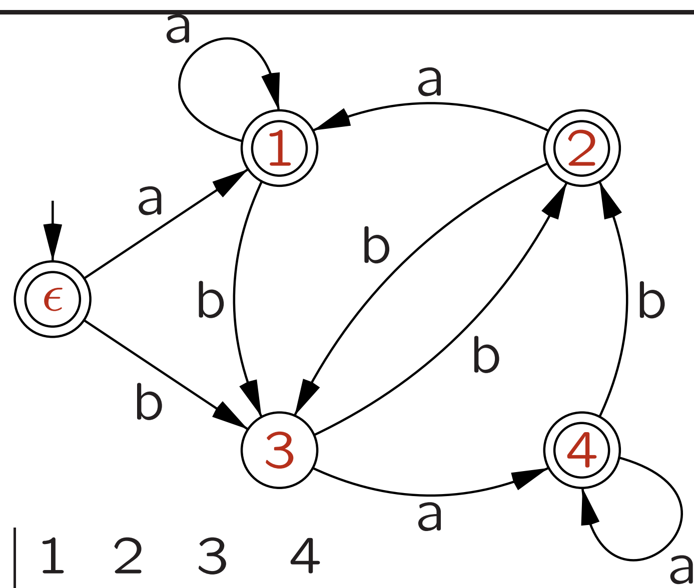
▷ find indistinguishable states $\equiv$

0. $U\{p,q\} = 0$ for all $p, q \in Q$
1. $U\{p,q\} = 1$ for all $p \in F$, $q \in Q - F$
3. repeat
5. $\quad T = U$
8. $\quad$ if $T\{\delta(p,a), \delta(q,a)\} = 1$ then $U\{p,q\} = 1$
$\quad\quad$ for all $a \in \Sigma$, all p, q with $T\{p,q\} = 0$
. until no changes
9. return(U)

$U\{p,q\} = 1$ iff $p \not\equiv q$



Theorem 3.10.2



	1	2	3	4
ϵ	.	.	X	.
1		.	X	.
2			X	.
3				X

$\delta(\epsilon, b) = 3,\ \delta(4, b) = 2$

	1	2	3	4
ϵ	.	.	X	X
1		.	X	X
2			X	X
3				X

algorithm	worst-case	practice	implementation
NAIVE	$\mathcal{O}(n^3)$	reasonable	easy
MINIMIZE	$\mathcal{O}(n^2)$	good	moderate
FAST	$\mathcal{O}(n \log n)$	very good	difficult
BRZOZOWSKI	$\mathcal{O}(n2^{2n})$	often good	easy

Figure 3.18

Bulletin of the EATCS 37, 1989, p.130

MINIMIZATION BY REVERSAL IS NOT NEW

J.A. Brzozowski
Department of Computer Science
University of Waterloo
Waterloo, Ontario, Canada

I read with interest W. Brauwer's note (Bulletin of EATCS, No. 35, June 1988, pp 113-116), about an algorithm, attributed to van de Snepscheut, for minimizing finite automata. I wholeheartedly agree with Dr. Brauer that the algorithm is simple and elegant; in fact, I considered it to be "rather surprising" when I discovered it in 1962. The key result is Theorem 13 in:

J.A. Brzozowski, "Canonical Regular Expressions and Minimal State Graphs for Definite Events", pp. 529-561 in *Mathematical Theory of Automata*, Vol. 12, of the MRI Symposia Series, Polytechnic Press of the Polytechnic Institute of Brooklyn; 1963.

The algorithm is also published in my Ph.D. Thesis:

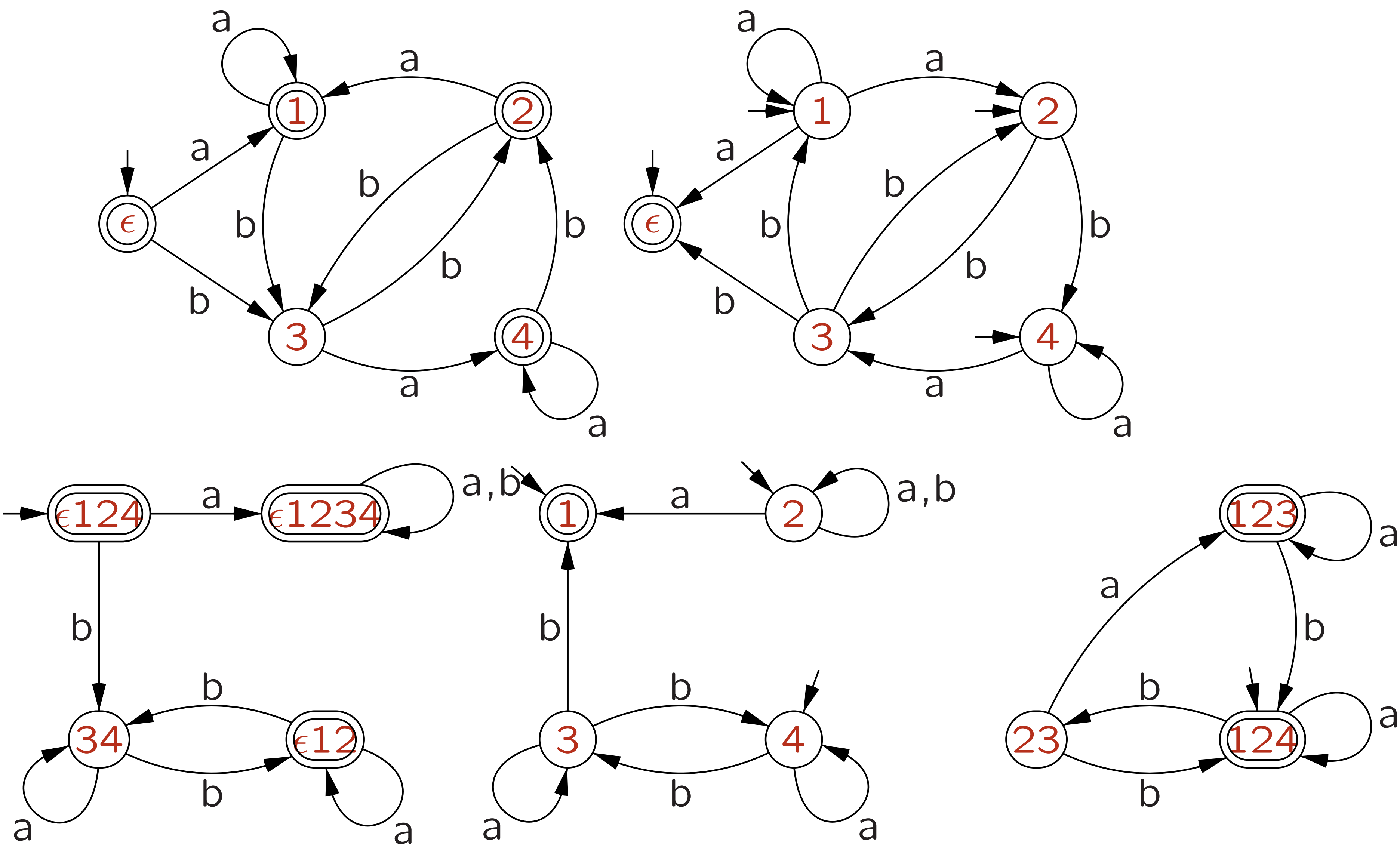
Regular Expression Techniques for Sequential Circuits, Ph.D. Dissertation, department of Electrical Engineering, Princeton University, Princeton, New Jersey; June 1962.

$\mathcal{M} = (Q, \Sigma, \delta, q_0, F)$

$R(\mathcal{M}) = (Q, \Sigma, \delta^R, F, q_0)$ reversing arrows
$q \in \delta(p, a)$ iff $p \in \delta^R(q, a)$
multiple initial states

$S(\mathcal{M})$ subset, only *reachable* states

Thm. $S(R(S(R(\mathcal{M}))))$ minimal DFA equivalent $\mathcal{M}$



even number b's or ending in a

$\mathcal{M} = (Q, \Sigma, \delta, q_0, F)$
$R(\mathcal{M}) = (Q, \Sigma, \delta^R, F, q_0)$ reversing arrows
$S(R(\mathcal{M})) = (Q'', \Sigma, \delta'', q_0'', F'')$

$q \in \delta''(X, w^R)$ iff $\delta(q, w) \in X$

Lem. $\mathcal{M}$ DFA, only reachable states.
$S(R(\mathcal{M}))$ minimal DFA for L^R

$A, B \in Q''$: $A \equiv B$ then $A = B$

$p \in A$ then $\delta(q_0, w) = p$ some $w \in \Sigma^*$
so $\delta''(A, w^R) \ni q_0 \Leftrightarrow \delta''(A, w^R) \in F''$
$A \equiv B$ so $\delta''(B, w^R) \in F'' \Leftrightarrow \delta''(B, w^R) \ni q_0$
so $p = \delta(q_0, w) \in B$

hence $A \subseteq B$ (for all p)

hence $A = B$ (symmetric)

3.11 State complexity

3.12 Partial orders and regular languages

motivation:

for (*any*) language, consider the language of all its *subsequences*

surprise:

it will be regular

$$\{ a^n b^{n^2} \mid n \geq 0 \} \mapsto a^* b^*$$

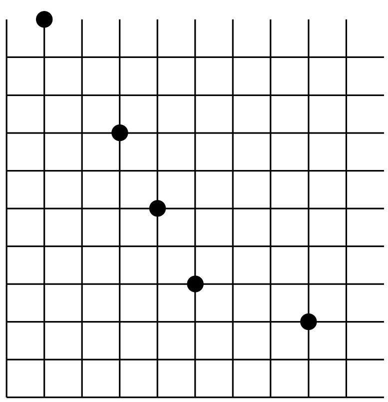
partial order

- reflexive $x \sqsubseteq x$ for all x
- *anti*symmetric $x \sqsubseteq y$ and $y \sqsubseteq x$ implies $x = y$
- transitive $x \sqsubseteq y$ and $y \sqsubseteq z$ imply $x \sqsubseteq z$

$\leq$ on $\mathbb{R}$, $\subseteq$ on $\mathcal{P}(V) = 2^V$, $\leq$ on $\mathbb{Z}^n$

incomparable neither $x \sqsubseteq y$ nor $y \sqsubseteq x$

subword ordering $x \leq y$ iff $y = uxv$



subsequence ordering $x|y$

$x = x_1 x_2 \dots x_n$ and $y = y_1 x_1 y_2 x_2 \dots y_n x_n y_{n+1}$

$ab^n a$ all comparable for $|$ (*chain*)

but all incomparable for $\leq$ (*antichain*)

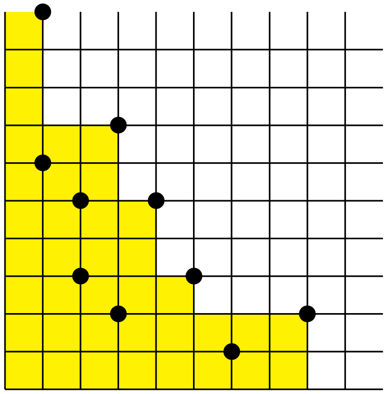
no infinite antichain for $\leq$ on $\mathbb{N}^n$

(Dickson's Lemma)

Thm. no infinite antichain for $|$ on Σ^*

($\sim$ Higman's Lemma)

Theorem 3.12.1

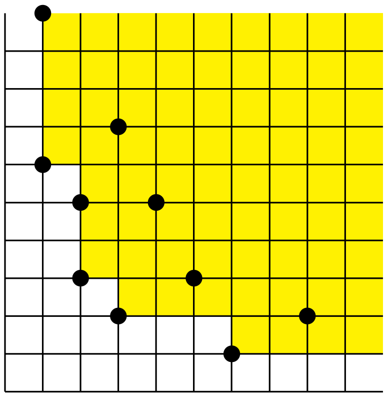


*sub*sequences

$$\text{sub}(L) = \{x \in \Sigma^* \mid x|y \text{ where } y \in L\}$$

*super*sequences

$$\text{sup}(L) = \{x \in \Sigma^* \mid y|x \text{ where } y \in L\}$$



$L = \{ a^n b^n \mid n \geq 1 \}$

$\text{sub}(L) = a^* b^*$

$\text{sup}(L) = \{a, b\}^* ab \{a, b\}^*$

3.12.6 $P_3 = \{ 2, 10, 12, 21, 102, 111, 122, 201, 212, 1002, \ldots \}$

$\text{sub}(P_3) = \{0, 1, 2\}^*$

$\text{sup}(P_3) =$

$$\Sigma^* 2 \Sigma^* \cup \Sigma^* 1 \Sigma^* 0 \Sigma^* \cup \Sigma^* 1 \Sigma^* 1 \Sigma^* 1 \Sigma^*$$

Thm. no infinite antichain for $|$ on Σ^*

Prf. *good* sequences $(w_1, w_2, \ldots)$ st. $w_i \not| w_j$ $(i < j)$

order good sequences

$(w_1, w_2, w_3, \ldots) < (v_1, v_2, v_3, \ldots)$ iff

$|w_1| = |v_1|, \ldots |w_k| = |v_k|$ but $|w_{k+1}| < |v_{k+1}|$

(1) every good sequence has a smaller one

$(w_1, w_2, w_3, \ldots)$

has infinite subsequence starting with same a

$w_{i_1} = av_1,\ w_{i_2} = av_2,\ \ldots$

$(w_1, \ldots, w_{i_1-1}, v_1, v_2, \ldots) < (w_1, w_2, w_3, \ldots)$

it is good $\quad v_k | v_\ell$ then $av_k = w_{i_k} | w_{i_\ell} = av_\ell$

it is smaller

(2) there is a minimal good sequence

w_1 shortest word with good continuation

w_1, w_2 shortest word with good continuation

etcetera

($\Rightarrow$) contradiction

$\sqsubseteq$ partial order on S

reflexive, antisymmetric, transitive

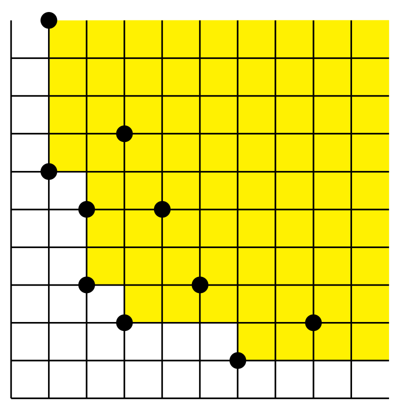
x minimal: $y \sqsubseteq x$ implies $y = x$

Lem. minimal elements are incomparable

min(L) minimal elements of L

if no infinite descending chain $x_1 \sqsupset x_2 \sqsupset x_3 \ldots$

well-founded



then for $y \in L$ some $y' \in \min(L)$ with $y' \sqsubseteq y$

$\sup(L) = \{x \in S \mid y \sqsubseteq x \text{ where } y \in L\,\}$

Lem. $\sup(L) = \sup(\min(L))$

special case: $\mid$ on Σ^*, $\leq$ on $\mathbb{N}^n$.
not $\leq$ on $\mathbb{Z}^n$.

$S - \mathrm{sub}(L) = \sup(\min(S - \mathrm{sub}(L)))$

because $\sup(S - \mathrm{sub}(L)) = S - \mathrm{sub}(L)$

Lemma 3.12.4

- $\text{sup}(L) = \text{sup}(\text{min}(L))$
- $\text{min}(L)$ finite   incomparable

Thm. $\text{sup}(L)$ regular (for arbitrary L)

$w = a_1 a_2 \ldots a_k \qquad (a_i \in \Sigma)$

$\text{sup}(\{w\}) = \Sigma^* a_1 \Sigma^* a_2 \Sigma^* \ldots \Sigma^* a_k \Sigma^*$

$\text{sup}(L) = \text{sup}(\text{min}(L)) = \bigcup_{w \in \text{min}(L)} \text{sup}(\{w\})$

finite union

Thm. $\text{sub}(L)$ regular (for arbitrary L)

$S - \text{sub}(L) = \text{sup}(\text{min}(S - \text{sub}(L)))$ regular transparencies made for

Second Course in Formal Languages and Automata Theory

based on the book by Jeffrey Shallit
of the same title

Hendrik Jan Hoogeboom, Leiden
http://www.liacs.nl/~hoogeboo/second/