Chapter 3

Finite Automata
and Regular Languages
finite automata and regular languages

3.0 Review
3.1 Moore and Mealy machines

3.2 Quotients
3.3 Morphisms and substitutions
3.4 Advanced closure properties of regular languages
3.5 Transducers

3.6 Two-way finite automata
3.7 The transformation automaton
3.8 Automata, graphs, and Boolean matrices

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
\[ A = (Q, \Sigma, \delta, q_0, F) \]

- \( Q \): states
- \( q_0 \in Q \): initial state
- \( F \subseteq Q \): final states
- \( \Sigma \): input alphabet \( a, b, w, x \)
- \( \delta : Q \times \Sigma \rightarrow Q \): transition function

\[ \delta^* : Q \times \Sigma^* \rightarrow Q \]

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

\[ L(A) = \{ x \in \Sigma^* \mid \delta(q_0, x) \in F \} \]
A nondeterministic finite automaton is defined as:

\[ A = (Q, \Sigma, \delta, q_0, F) \]

- **States** \( p, q \)
- **Initial state** \( q_0 \in Q \)
- **Final states** \( F \subseteq Q \)
- **Input alphabet** \( a, b, w, x \)

The transition function \( \delta : Q \times \Sigma \rightarrow 2^Q \) is given by:

\[
\delta(q, \epsilon) = \{q\}
\]

\[
\delta(q, xa) = \bigcup_{r \in \delta(q,x)} \delta(r, a)
\]

The language of the automaton is:

\[ L(A) = \{ x \in \Sigma^* \mid \delta(q_0, x) \cap F \neq \emptyset \} \]

A deterministic version of the transition function \( \delta' : 2^Q \times \Sigma \rightarrow 2^Q \) is defined as:

\[
\delta'(U, a) = \bigcup_{p \in U} \delta(p, a)
\]
subset construction

deterministic

\[ \delta' : 2^Q \times \Sigma \to 2^Q \]

\[ \delta'(U, a) = \bigcup_{p \in U} \delta(p, a) \]
nfa $\leadsto$ regular expression

$E_0 + E_1 \cdot E_2^* \cdot E_3$
long words can be pumped

∀ for every regular language $L$
∃ there exists a constant $n \geq 1$
such that
∀ for every $z \in L$
        with $|z| \geq n$
∃ there exists a decomposition $z = uvw$
        with $|uv| \leq n$, $|v| \geq 1$
such that
∀ for all $i \geq 0$, $uv^i w \in L$
- 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
<table>
<thead>
<tr>
<th></th>
<th>RLIN</th>
<th>REG</th>
<th>DPDA</th>
<th>PDAe</th>
<th>DLBA</th>
<th>MON</th>
<th>LBA</th>
<th>REC</th>
<th>TYPE0</th>
</tr>
</thead>
<tbody>
<tr>
<td>intersection</td>
<td>+</td>
<td>–</td>
<td>–</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>complement</td>
<td>+</td>
<td>+</td>
<td>–</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>–</td>
<td>-</td>
</tr>
<tr>
<td>union</td>
<td>+</td>
<td>–</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>concatenation</td>
<td>+</td>
<td>–</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>star, plus</td>
<td>+</td>
<td>–</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>ε-free morphism</td>
<td>+</td>
<td>–</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>morphism</td>
<td>+</td>
<td>–</td>
<td>+</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>+</td>
</tr>
<tr>
<td>inverse morphism</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>intersect reg lang</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>mirror</td>
<td>+</td>
<td>–</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
</tbody>
</table>

$\cap$, $\cup$, $\cdot$, $\epsilon$ boolean operations

$\cup$, $\cdot$, $\ast$ regular operations

$h$, $h^{-1}$, $\cap R$ (full) trio operations
3.1 Moore and Mealy machines
3.2 Quotients
Let $L_1, L_2 \subseteq \Sigma^*$.

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

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

$L_1 / L_2 = a^+$

$L_1 / L_3 = a^+bc^*$

**Ex.** $\text{Pref}(L) = L / \Sigma^*$
Example 3.2.2

$L_1, L_2 \subseteq \Sigma^*$

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

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

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

‘⊆’ $m^2 - n^2 = (m+n)(m-n)$

<table>
<thead>
<tr>
<th>m</th>
<th>n</th>
<th>m+n</th>
<th>m-n</th>
<th>$m^2 - n^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>e</td>
<td>e</td>
<td>e</td>
<td>e</td>
<td>mult four</td>
</tr>
<tr>
<td>e</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>odd</td>
</tr>
<tr>
<td>o</td>
<td>e</td>
<td>o</td>
<td>o</td>
<td>odd</td>
</tr>
<tr>
<td>o</td>
<td>o</td>
<td>e</td>
<td>e</td>
<td>mult four</td>
</tr>
</tbody>
</table>

‘⊇’ $(k+1)^2 - k^2 = 2k+1$ odd

$(k+2)^2 - k^2 = 4k+4$ multiple of four
$L_1, L_2 \subseteq \Sigma^*$

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

can 'hide' computations

**Ex.**

$L_1 = \{ a^{2n}c b a^n \mid n \geq 1 \} \{ b a^{2n} b a^n \mid n \geq 1 \}^* b a$

$L_2 = c \cdot \{ b a^n b a^n \mid n \geq 1 \}^*$

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

$a^{16} \ c \ b a^8 \ b a^8 \ b a^4 \ b a^4 \ b a^2 \ b a^2 \ b a \ b a$
**Thm.**  \( L, R \subseteq \Sigma^* \) If \( R \) regular, then \( R/L \) regular.

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

noncomputable ! (\( L \) arbitrary)

REG closed under quotient \( \text{REG / REG} = \text{REG} \) (see Ch.4)

CF not closed, even \( \text{CF / CF} = \text{RE} \) \( \text{CF / REG} = \text{CF} \)
3.3 Morphisms and substitutions
III 13

‘monoid’

\[ h : \Sigma \rightarrow \Delta^* \]
\[ h : \Sigma^* \rightarrow \Delta^* \quad h(xy) = h(x)h(y), \ h(\epsilon) = \epsilon \]
\[ h : 2^\Sigma^* \rightarrow 2^\Delta^* \quad h(L) = \bigcup_{x \in L} h(x) \]

0 ↦ ab, 1 ↦ ba, 2 ↦ \epsilon

00212 ↦ ababba

\[ \{0^n21^n \mid n \geq 0\} \mapsto \{(ab)^n(ba)^n \mid n \geq 0 \} \]

**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 ↦ b(aa)*b
1 ↦ a+b(aa)*ab

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

\[ s(K) = \{ x \in \{a, b\}^* \mid \#_a x, \#_b x \text{ are even} \} \]
Theorem 3.3.9

\[ h : \Sigma \to \Delta^*, \ K \subseteq \Delta^* \]

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

**Thm.** REG closed under inverse morphism

\[ \begin{array}{c}
  a \\
  \downarrow \\
  h(a) \\
\end{array} \]

\[ \delta'(p, a) = \delta(p, h(a)) \]

\[ h : 0 \mapsto ab, 1 \mapsto ba \]

\[ h^{-1}( \{bb, aba\}^*) = \{0011\}^* \]
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?
\[\text{abbba} \parallel \text{acac} \ni \text{abacbacba}\]

\[K \parallel L\] using morphisms, intersection

copies of alphabet

\[\Sigma, \Sigma_1 = \{ a_1 \mid a \in \Sigma \}, \Sigma_2 = \{ a_2 \mid a \in \Sigma \}\]

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

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

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

\[\begin{align*}
\text{abbba} & \xleftarrow{h_1} a_1b_1a_2c_2b_1a_2c_2b_1a_1 & \xrightarrow{h_2} \text{acac} \\
\end{align*}\]

\[\begin{align*}
\in K & \quad \downarrow g & \quad \in L \\
\text{abacbacba} & & \\
\end{align*}\]

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

\[ Q' = \{ q'_0 \} \cup Q \times Q \times Q \quad \text{middle, 1st, 2nd} \]

\[
\delta'(q'_0, \varepsilon) = \{ [q, q_0, q] \mid q \in Q \} \quad \varepsilon\text{-move}
\]

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

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

\[ \sqrt{L} = \{ x \in \Sigma^* \mid xx \in L \}. \]
$$\text{cut}_f L = \{ x \mid xy \in L \text{ for } y \text{ with } |y| = f(|x|) \}.$$ 

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

$$f(n) = 2^n \quad \log L \quad \text{p.76}$$

$$f(n) = n^2$$

which $f$?

see: transition matrix (Ch. 3.8)
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 \text{\( \epsilon \)-move}
\]

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

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

\[
F' = \{ [q, q, 1] \mid q \in Q \}
\]
Note that the construction introduces \( \epsilon \)-moves.

Is this a proof?

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

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

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

In informal notation,
\[
q_0 \xrightarrow{x} p \xrightarrow{y} q_f \text{ in } A, \text{ then } q'_0 \xrightarrow{\epsilon} [p, p, 0] \xrightarrow{y} [p, p_f, 0] \xrightarrow{\epsilon} [p, q_0, 1] \xrightarrow{x} [p, p, 1] \text{ in } A'.
\]

For the reverse implication we need that indeed all computations in \( A' \) are of this form.
3.5 Transducers
FST \sim \text{finite state automaton with output}

\[ A = (Q, \Sigma, \Delta, S, q_{in}, F) \]

\[ S \subseteq Q \times \Sigma^* \times \Delta^* \times Q \]

\[ \ldots (\Sigma \cup \{\epsilon\}) \times (\Delta \cup \{\epsilon\}) \ldots \]

\[ T(A) \subseteq \Sigma^* \times \Delta^* \quad \text{transduction (translation)} \]

\[ x \rightarrow_A y \quad \text{rational relation} \]

\[ K \subseteq \Sigma^* \]

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

erase every 2nd \(a\) (keeping words ending in \(b\))
finite state transductions

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

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

$$T(K) = \{ x \mid xy \in K \text{ and } \#_a y \text{ even} \}$$
\( K, L \subseteq \Sigma^* \)

\[ \Sigma' = \{ a' \mid a \in \Sigma \} \quad \Sigma \cap \Sigma' = \emptyset \]

\[ f : \Sigma \cup \Sigma' \to \Sigma \quad f(a) = f(a') = a \]

\( f^{-1} \) non-det colouring

\[ h : \Sigma \to \Sigma' \quad h(a) = a' \]

\[ g : \Sigma \cup \Sigma' \to \Sigma \quad g(a) = a, \ g(a') = \varepsilon \]

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

basic full trio operations:
- morphism
- inverse morphism
- intersection regular
$h : \{\begin{array}{ll}
  a & \rightarrow 100 \\
  b & \rightarrow 10 \\
  c & \rightarrow 010 \\
\end{array}\}$

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

⇒ sequence of full trio op's is FST
Theorem 3.5.6

FST $A_i = (Q, \Sigma_i, \Sigma_{i+1}, S_i, q_{io}, F_i)$

$T(A_1)T(A_2) \Rightarrow$ FST $A' = (Q', \Sigma_1, \Sigma_3, S', q'_0, F')$

formally – $Q' = Q_1 \times Q_2$
- $q'_0 = \langle q_{10}, q_{20} \rangle$
- $F' = F_1 \times F_2$, 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 $(p, \epsilon, \epsilon, p)$’
Nivat's theorem

every full trio operation is a fs transduction

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

\( R_M \) regular language over 'transitions'

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

\( h \) and \( g \) select input and output

\[
\begin{align*}
K & \ni b \ b \ a \ a \ b \ a \\
R_M & \ni b:01 \ b:01 \ a:\epsilon \ a:1 \ b:01 \ a:\epsilon \\
T_M(K) & \ni 01 \ 01 \ \epsilon \ 1 \ 01 \ \epsilon
\end{align*}
\]

\[
T_M(K) = g( h^{-1}(K) \cap R_M )
\]
closure properties

\textit{trio} ≡ faithful cone:
morphism, $\epsilon$-free morphism, intersection regular

\textit{full trio} ≡ cone:
\ldots, (arbitrary) morphism, \ldots

\textit{(full) semi-AFL} : (full) trio & union

\textit{(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

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

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

\( \delta(\cdot, \triangleright) = (\cdot, R), \quad \delta(\cdot, \triangleleft) = (\cdot, L) \)

configuration \( \triangleright \Sigma^* Q \Sigma^* \triangleleft \cup Q \triangleright \Sigma^* \triangleleft \)

\( wqax \vdash wapx \) when \( \delta(q, a) = (p, R) \) move

\( waqx \vdash wpax \) \( \delta(q, a) = (p, L) \)

infinite loops possible!

\( L(\mathcal{M}) = \{ w \in \Sigma^* \mid q_0 \triangleright w \triangleleft \vdash^* \triangleright wp \triangleleft, p \in F \} \)
Shefferdson [1959]  

**2DFA ⊆ DFA**  

- keep track of ‘excursions’ to the left  
- \( \tau : Q \cup \{ \bar{q} \} \to Q \cup \{ \ell \} \)  
- \( \bar{q} \) final, \( \ell \) for loop  

- updating \( \tau \) to \( \tau_{xb} \)  
- \( \delta(p, b) = (q, R) \) then \( \tau_{xb}(p) = q \)  
- \( \delta(p, b) = (p_1, L), \tau_x(p_1) = q_1, \)  
  \( \delta(q_1, b) = (p_2, L), \ldots, \tau_x(p_k) = q_k \)  

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 \( \tau_{xb}(p) = \ell \)  

- \( \tau_x(q) = \delta(q_0, x) \)
root($L$) = \{ $w \in \Sigma^* | w^n \in L$ 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

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

exists connection

\[
\begin{pmatrix}
1 & 1 & 0 \\
0 & 1 & 1 \\
1 & 0 & 0
\end{pmatrix}
\begin{pmatrix}
0 & 1 & 0 \\
1 & 0 & 0 \\
1 & 0 & 1
\end{pmatrix}
= \begin{pmatrix}
0 & 1 & 0 \\
1 & 0 & 1 \\
0 & 1 & 0
\end{pmatrix}
\]
Theorem 3.8.1

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

$M_a$ Boolean matrix

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

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

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

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

**Thm.** for $w = a_1a_2 \ldots a_t$, $a_i \in \Sigma$

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

**Cor.** $M_{xy} = M_xM_y$
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

\[ u_0 = [1, 0, 0, \ldots, 0] \quad \text{(row)} \]

\[(u_F)_i = 1 \text{ iff } q_i \in F \quad \text{(column)}\]

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

\textbf{Thm.} \ x \in L(A) \ \text{iff} \ u_0 \ M_x \ u_F = 1

\textit{matrix represents computation}
nondeterministic case

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

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

$$w = a_1 a_2 \ldots a_t$$

$$M_w = M_{a_1} M_{a_2} \cdots M_{a_t}$$

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

transformation automaton

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

$$q'_0 = I \quad \text{identity matrix}$$

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

no final states specified

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

i.e., \((M)_{ij} = 1 \text{ iff } \delta(q_i, w) \ni q_j\)
\[ \sqrt{L} = \{ x \in \Sigma^* \mid xx \in L \}. \]

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

\( M_x \) after reading \( x \)

final: \( (M_x)_{q_0p} = 1 \) for some \( p \in Q \) [unique]

and \( (M_x)_{pq} = 1 \) for some \( q \in F \)

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

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

\[ M^{k+1} = M^k M \]

Prop. \( \log L = \{ x \mid xy \in L \text{ for } y \text{ with } |y| = 2|x| \} \).

\[ M^{2^k} = (M^{2^{k-1}})^2 \]
product: $\text{aut} \times \text{transformation aut}$

$$(p, M^k) \xrightarrow{a} (\delta(p, a), M^{k+1})$$
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.** \( \text{REC} = \text{REG} \) (for strings)

monoid as automaton
\( A_M = (M, \Sigma, \delta, 1, S) \)
\( \delta(m, a) = m \circ h(a) \quad m \in M, \ a \in \Sigma \)

\( x \in L(A_M) \) 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 \( R \)

DFA \( M = (Q, \Sigma, \delta, q_0, F) \)
ending in the same state
\( xR_M y \) iff \( \delta(q_0, x) = \delta(q_0, y) \)

equivalence relation on \( \Sigma^* \)
- finite index \( |Q| \)
- right invariant \( xR_M y \) implies \( xzR_M yz \)
right congruence
- \( L(M) \) union of equivalence classes
\( R_M \) saturates \( L \)
Fundamental observation

\[ L \subseteq \Sigma^* \]

\[ xR_L y \] when, for all \( u \), \( (xu \in L \iff yu \in L) \)

- equivalence relation on \( \Sigma^* \)
  - index may be infinite
  - right invariant \( xR_L y \) implies \( xzR_L yz \)
  - \( L \) union of equivalence classes

\[ R_1, R_2 \] equivalence relations
\[ R_1 \text{ refinement of } R_2: R_1 \subseteq R_2 \]

**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 \( z \) \( \Rightarrow xR_L y \)
$L \subseteq \Sigma^*$

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

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

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

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

$x^{-1}L$ may contain

$\varepsilon$

$\{a, b\}^a$

even $b$'s ($\geq 2$)

odd $b$'s

\[
\begin{array}{c|cccc}
& \varepsilon & a & b & bb \\
\hline
\varepsilon & \checkmark & \checkmark & - & \checkmark \\
b & - & \checkmark & \checkmark & - \\
ba & \checkmark & \checkmark & \checkmark & - \\
\end{array}
\]

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

[\varepsilon] even number $b$'s

[a] = [\varepsilon], [b]

[b] odd $b$'s, ending in $b$

[ba], [bb] = [\varepsilon]

[ba] odd $b$'s, ending in $a$

[bba] = [ba], [bab] = [\varepsilon]
**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

\( a. \Rightarrow b. \) \( R_M \) for automaton \( M \)

\( b. \Rightarrow c. \) \( E \) is a refinement of \( R_L \). index \( R_L \leq \text{index } E \)

\( c. \Rightarrow a. \) use equivalence classes as states

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

*automaton is ‘inside’ the language*
number of states

- \( n \) state nfa
- \( 2^n \) state dfa
- all reachable
- all nonequivalent
3.10 Minimization of finite automata
Theorem 3.10.1

\[ L \subseteq \Sigma^* \]

\[ xR_L y \text{ when, for all } u, \ (xu \in L \text{ iff } yu \in L) \]

\[ xR_M y \text{ when } \delta(q_0, x) = \delta(q_0, y) \]

\[ xR_M y \then xR_L y \]

Myhill-Nerode: \( R_L \)-classes \( \sim \) 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 \ (\( R_M \) refines \( R_L \))

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

injective \ (surjective, same number states)

respects transitions \ (right invariant)
\( M = (Q, \Sigma, \delta, q_0, F) \) dfa for \( L \)

\( xR_L y \) when, for all \( u, (xu \in L \iff yu \in L) \)

\( xR_M y \) when \( \delta(q_0, x) = \delta(q_0, y) \)

\( xR_M y \) then \( xR_L y \)

\( p \equiv q \quad \text{indistinguishable} \)

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

\( \mu : Q \to \Sigma^*/R_L \)

\( q \mapsto [x], \text{ such that } \delta(q_0, x) = q \)

well-defined \( (R_M \text{ refines } R_L) \)

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

may not be injective

respects transitions \( (\text{right invariant}) \)

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

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

\( \triangleright \) 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. $T = U$
8. if $T\{\delta(p, a), \delta(q, a)\} = 1$ then $U\{p, q\} = 1$
   for all $a \in \Sigma$, all $p, q$ with $T\{p, q\} = 0$
. until no changes
9. return($U$)

$U\{p, q\} = 1$ iff $p \not\equiv q$
\[ \begin{array}{cccc}
1 & 2 & 3 & 4 \\
\epsilon & . & . & X & .
1 & . & X & .
2 & X & .
3 & X \\
\end{array} \]

\[ \delta(\epsilon, b) = 3, \quad \delta(4, b) = 2 \]

\[ \begin{array}{cccc}
1 & 2 & 3 & 4 \\
\epsilon & . & . & X & X \\
1 & . & X & X \\
2 & X & X \\
3 & X \\
\end{array} \]
<table>
<thead>
<tr>
<th>algorithm</th>
<th>worst-case</th>
<th>practice</th>
<th>implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAIVE</td>
<td>$O(n^3)$</td>
<td>reasonable</td>
<td>easy</td>
</tr>
<tr>
<td>MINIMIZE</td>
<td>$O(n^2)$</td>
<td>good</td>
<td>moderate</td>
</tr>
<tr>
<td>FAST</td>
<td>$O(n \log n)$</td>
<td>very good</td>
<td>difficult</td>
</tr>
<tr>
<td>BRZOZOWSKI</td>
<td>$O(n2^n)$</td>
<td>often good</td>
<td>easy</td>
</tr>
</tbody>
</table>
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:


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

\[ M = (Q, \Sigma, \delta, q_0, F) \]

\[ R(M) = (Q, \Sigma, \delta^R, F, q_0) \] reversing arrows

\[ q \in \delta(p, a) \text{ iff } p \in \delta^R(q, a) \]

multiple initial states

\[ S(M) \text{ subset, only } \textit{reachable} \text{ states} \]

\textbf{Thm.} \[ S(R(S(R(M)))) \] minimal DFA equivalent \( M \)
example Brzozowski

even number b’s or ending in a
\[ \mathcal{M} = (Q, \Sigma, \delta, q_0, F) \]
\[ R(\mathcal{M}) = (Q, \Sigma, \delta^R, F, q_0) \text{ reversing arrows} \]
\[ S(R(\mathcal{M})) = (Q'', \Sigma, \delta'', q_0'', F'') \]
\[ q \in \delta''(X, w^R) \iff \delta(q, w) \in X \]

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

\( A, B \in Q'': A \equiv B \text{ then } A = B \)

\( p \in A \text{ then } \delta(q_0, w) = p \text{ some } w \in \Sigma^* \)
so \( \delta''(A, w^R) \ni q_0 \iff \delta''(A, w^R) \in F'' \)

\( A \equiv B \text{ so } \delta''(B, w^R) \in F'' \iff \delta''(B, w^R) \ni q_0 \)
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
\[ \{ a^n b^{n^2} \mid n \geq 0 \} \mapsto a^* b^* \]
subsequence ordering

- reflexive $x \sqsubseteq x$ for all $x$
- antisymmetric $x \sqsubseteq y$ 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 $x|y$

$x = x_1x_2\ldots x_n$ and $y = y_1x_1y_2x_2\ldots ynx_ny_{n+1}$

$ab^n a$ 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 $\Sigma^*$

($\sim$ Higman’s Lemma)
subsequences
\[ \text{sub}(L) = \{ x \in \Sigma^* \mid x|y \text{ where } y \in L \} \]

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

\[ L = \{ a^n b^n \mid n \geq 1 \} \]
sub\( (L) = a^*b^* \)
sup\( (L) = \{a, b\}^*ab\{a, b\}^* \)

3.12.6 \( P_3 = \{ 2, 10, 12, 21, 102, 111, 122, 201, 212, 1002, \ldots \} \)
sub\( (P_3) = \{0, 1, 2\}^* \)
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 $\Sigma^*$

**Prf.** good sequences $(w_1, w_2, \ldots)$ st. $w_i \ntriangleright w_j$ ($i < j$)
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 $v_k | v_\ell$ 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
(⇒) contradiction
\[ \sqsubseteq \text{ partial order on } S \]
reflexive, antisymmetric, transitive

\[ x \text{ minimal: } \quad y \sqsubseteq x \text{ implies } y = x \]

**Lem.** minimal elements are incomparable

\[ \text{min}(L) \text{ minimal elements of } L \]

if no infinite descending chain \( x_1 \sqsubseteq x_2 \sqsubseteq x_3 \ldots \)
well-founded

then for \( y \in L \) some \( y' \in \text{min}(L) \) with \( y' \sqsubseteq y \)

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

**Lem.** \( \text{sup}(L) = \text{sup}(\text{min}(L)) \)

special case: \( \mid \text{ on } \Sigma^*, \leq \text{ on } \mathbb{N}^n \).
not \( \leq \text{ on } \mathbb{Z}^n \).

\[ S - \text{sub}(L) = \text{sup}(\text{min}(S - \text{sub}(L))) \]

because \( \text{sup}(S - \text{sub}(L)) = S - \text{sub}(L) \)
Theorem 3.12.5

- $\sup(L) = \sup(\min(L))$
- $\min(L)$ finite and incomparable

**Thm.** $\sup(L)$ regular (for arbitrary $L$)

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

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

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

finite union

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

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