Închidere la substituții:

Def: Σ_1, Σ_2 alfabete, o substituție (vezi pag 33 din PDF) este o funcție $\sigma : \Sigma_1^* \to 2^{\Sigma_2^*}$ cu două proprietăți:

1. $\sigma(\lambda) = \{\lambda\}$
2. $\sigma(xy) = \sigma(x)\sigma(y), \ \forall x, y \in \Sigma_1^*$.

Evident definirea funcției σ pe literele din Σ_1 definește complet σ.
Se poate extinde la limbaje: $\sigma(L) = \bigcup_{x \in L} \sigma(x)$ pentru $L \subseteq \Sigma_1^*$.

Exemplu: $\Sigma_1 = \{a, b\}$, $\Sigma_2 = \{a, b, c\}$ și o substituție $\sigma : \Sigma_1^* \to 2^{\Sigma_2^*}$
$\sigma(a) = \{ab, ac, b\}$ $\sigma(b) = \{b, ba\}$
$\sigma(ba) = \{b, ba\} \cdot \{ab, ac, b\} = \{bab, bac, bb, baab, baac\}$

Def: O substituție $h : \Sigma_1^* \to 2^{\Sigma_2^*}$ se numește morfism dacă $|h(a)| = 1$, $\forall a \in \Sigma_1$ (adică fiecare literă are asociat un limbaj de un cuvânt. Morfismele se definesc și ca:

$h : \Sigma_1^* \to \Sigma_2^*$ cu proprietățile:

1. $h(\lambda) = \lambda$
2. $h(xy) = h(x) \cdot h(y)$, $\forall x, y \in \Sigma_1^*$

Morfisme inverse:

Fie $h : \Sigma_1^* \to \Sigma_2^*$ un morfism. Pentru $w \in \Sigma_2^*$, $h^{-1}(w) = \{x \mid x \in \Sigma_1^*, h(x) = w\}$.
extindem la limbaje: $h^{-1}(L) = \{x \mid h(x) \in L, x \in \Sigma_1^*\}$, $L \subseteq \Sigma_2^*$.

Teoremă: Limbajele regulate sunt închise la
$\begin{cases} 1. \text{ substituții regulate} \\ 2. \text{ morfisme} \\ 3. \text{ morfisme inverse.} \end{cases}$

Dem: 1. REG închisă la substituții regulate:
Fie $\sigma : \Sigma_1^* \to 2^{\Sigma_2^*}$ substituție cu proprietatea că $\sigma(a)$ este regulat $\forall a \in \Sigma_1$.
Fie $L_1 \subseteq \Sigma_1^*$ un limbaj regulat.
Trebuie să demonstrăm că $\sigma(L_1)$ este regulat.
Deoarece L_1 este regulat, există o expresie regulată r_1 care descrie L_1.
Deci $L(r_1) = L_1$. Deoarece fiecare $\sigma(a)$ este regulat există expresiile r_a expresii regulate care descriu $\sigma(a)$, $\forall a \in \Sigma_1$.
r_1 este ER peste Σ_1, r_a este ER peste Σ_2.
Construim expresia r_2 din r_1 înlocuind fiecare simbol a din r_1 cu expresia r_a. Pentru că r_1 este ER și toate r_a-urile sunt expresii regulate rezultă că și r_2 este expresie regulată (formată din $\cup, \cdot, *$ de ER) peste Σ_2.
Trebuie să demonstrăm că $L(r_2) = \sigma(L_1) \Leftrightarrow L(r_2) = \sigma(L(r_1))$.
Dem. prin inducție după nr. de operatori din r_1:
Baza: r_1 are 0 operatori $\Rightarrow r_1 \in \{\emptyset, \lambda\} \cup \Sigma_1$. ~~Dacă $r_1 = \emptyset \Rightarrow r_2 = \emptyset \Rightarrow L(r_2) = \sigma(\emptyset) = \emptyset$~~

Dacă $r_1 = \emptyset \Rightarrow r_2 = \emptyset \Rightarrow L(r_2) = L(\emptyset) = \emptyset = \sigma(\emptyset)$.
$r_1 = \lambda \Rightarrow r_2 = \lambda \Rightarrow L(r_2) = L(\lambda) = \{\lambda\} = \sigma(\{\lambda\})$.
$r_1 = a \in \Sigma_1 \Rightarrow r_2 = r_a \Rightarrow L(r_a) = \sigma(a)$ din definiția lui r_a.

ipoteza inductivă: Presupunem că $L(r_2) = \sigma(L_1)$ pentru expresia r_1 cu cel mult k operatori.
Saltul inductiv: Demonstrăm pentru $k+1$ operatori:
Caz 1. $r_1 = r_1' + r_1''$ (sau $r_1 = r_1' \cup r_1''$). Din construcția lui r_2 avem că
$r_2 = r_2' + r_2''$ (dacă înlocuim în r_1' și r_1'' fiecare $a \in \Sigma_1$ cu r_a).
Din ipoteza inductivă rezultă că $L(r_2') = \sigma(L(r_1'))$ și $L(r_2'') = \sigma(L(r_1''))$.
Așadar $L(r_2) \overset{\text{def R.E.}}{=} L(r_2') \cup L(r_2'') \overset{\text{i.i.}}{=} \sigma(L(r_1')) \cup \sigma(L(r_1'')) \overset{\text{def. subst.}}{=} \sigma(L(r_1') \cup L(r_1'')) =$
$\overset{\text{def R.E}}{=} \sigma(L(r_1' \cup r_1'')) \overset{\text{def } r_1}{=} \sigma(L(r_1))$.
Caz 2. $r_1 = r_1' \cdot r_1'' \Rightarrow r_2 = r_2' \cdot r_2''$ similar.
Caz 3. $r_1 = r_1'^* \Rightarrow r_2 = r_2'^*$
$L(r_2) \overset{\text{def R.E}}{=} (L(r_2'))^* \overset{\text{i.i.}}{=} (\sigma(L(r_1')))^* = \sigma((L(r_1'))^*) = \sigma(L(r_1'^*)) = \sigma(L(r_1))$

Dem 2: Reg este închisă la morfisme:

Fie $L \subseteq \Sigma_1^*$ limbaj regulat și $h: \Sigma_1^* \to \Sigma_2^*$ morfism. Să dem că $h(L) \in Reg$.
Este imediat din Dem 1 pentru că limbajele finite sunt regulate, deci caz particular pentru demonstrația anterioară.

Dem 3: Reg este închisă la morfisme inverse:

Fie $L \subseteq \Sigma_2^*$ limbaj regulat și $h: \Sigma_1^* \to \Sigma_2^*$ morfism. Să dem că $h^{-1}(L) \in Reg$.

Fie $A = (Q, \Sigma_2, \delta, q_0, F)$ un DFA cu $L(A) = L$. Construim automatul M cu $L(M) = h^{-1}(L)$.

$M = (Q, \Sigma_1, \delta', q_0, F)$ unde $\delta'(q, a) = \overset{\in Q}{\delta(q, h(a))}$. ← extinderea lui δ la cuv.
$\forall q \in Q, \forall a \in \Sigma_1$.

Dem că $\delta'(q, x) = \delta(q, h(x)), \forall x \in \Sigma_1^*$ ← extinderile lui δ' și δ la cuvinte.

Inducție după lungimea lui x:

$|x| = 0 \Rightarrow x = \lambda \Rightarrow \delta'(q, \lambda) = q = \delta(q, \lambda)$, h morfism $\Rightarrow h(\lambda) = \lambda$.

Pp. adev. pt n dem. pt $n+1$: fie $|x| = n+1$, $x = x'a$, $a \in \Sigma_1$.

$$\delta'(q, x) = \delta'(q, x'a) = \delta'(\delta'(q, x'), a) \overset{i.i.}{=} \delta'(\delta(q, h(x')), a) \overset{def\ \delta'}{=} \delta(\delta(q, h(x')), h(a)) =$$
$$\overset{def.ext.\delta}{=} \delta(q, h(x') \cdot h(a)) \overset{h\ morfism}{=} \delta(q, h(x'a)) = \delta(q, h(x)).$$

Deci $\delta'(q, x) = \delta(q, h(x))$ pentru orice $q \in Q$ și $x \in \Sigma_1^*$.

Avem că $\delta'(q_0, w) \in F \Leftrightarrow \delta(q_0, h(w)) \in F$ deci ~~$L(M) =$~~
$\forall w \in L(M) \Leftrightarrow h(w) \in L$.

$\Rightarrow L(M) = h^{-1}(L)$. q.e.d.

Cum se folosește:

Să se demonstreze că $L = \{a^n b a^n \mid n \geq 1\}$ nu e regulat.

Pp. că L este regulat $\Rightarrow$ pentru orice morfism h avem $h^{-1}(L)$ este regulat:

Fie $h_1: \{a,b,c\}^* \to \{a,b\}^*$ cu $h_1(a) = a$, $h_1(b) = ba$, $h_1(c) = a$ $\Big| \Rightarrow h_1^{-1}(L)$ este regulat.

$h_1^{-1}(L) = \{x^n b y^{n-1} \mid x, y \in \{a,c\}, n \geq 1\}$ este regulat. $h_1(abc) = abaa$

Fie încă un morfism $h_2: \{a,b,c\}^* \to \{0,1\}^*$, $h_2(a) = 0$, $h_2(b) = 1$, $h_2(c) = 1$.

Atunci $h_2(h_1^{-1}(L)) \cap 0^* 1^* = \{a^n b^n \mid n \geq 1\}$. Contradicție.

sau $h_1^{-1}(L) \cap a^* b c^* = \{a^n b c^{n-1} \mid n \geq 1\} = L'$. $h_2(L') = \{a^n b^n \mid n \geq 1\}$

Minimizarea DFA:

1. echivalența pe cuvinte: Pentru $L \subseteq \Sigma^*$ un limbaj definim $\equiv_L$ astfel:

$$x \equiv_L y \Leftrightarrow \forall z \in \Sigma^* \text{ avem } xz \in L \Leftrightarrow yz \in L.$$

$\equiv_L$ este relație de echivalență.

2. invarianța la dreapta la concatenare:
O relație s.n. invariantă la dreapta față de concatenare dacă

$$x R y \Rightarrow \forall z \in \Sigma^*,\ xz R yz.$$

3. echivalența dată de un automat:
Fie $M = (Q, \Sigma, \delta, q_0, F)$ un DFA. Definim $\equiv_M$:

$$x \equiv_M y \Leftrightarrow \delta(q_0, x) = \delta(q_0, y).$$

$\equiv_M$ este relație de echivalență și invariantă la dreapta.

4. indicele unei relații de echivalență:

$|\Sigma^*/R| =$ nr. de clase de echivalență ale relației.

$\equiv_M$ este de indice finit (numărul de stări din M care sunt accesibile) evident în clasa unei stări $q \in Q$ avem cuvintele $x \in \Sigma^*$ cu $\delta(q_0, x) = q$.

Teorema Myhill-Nerode:
Următoarele trei propoziții sunt echivalente:
1. $L \subseteq \Sigma^*$ este regulat
2. L este reuniunea unor clase de echivalență ale unei relații de echivalență invariantă la dreapta de indice finit.
3. Relația $\equiv_L$ definită pentru L este de indice finit.

Demonstrație:

$1 \Rightarrow 2$: L este regulat $\Rightarrow$ există un DFA $M = (Q, \Sigma, \delta, q_0, F)$ a.î. $L(M) = L$.
Construim M' din M eliminând stările inaccesibile și complet.
Relația $\equiv_{M'}$ este relație de echivalență invariantă la dreapta de indice finit.
Folosim $\equiv_{M'}$ pentru **2**: $L = \bigcup_{q \in F} [q'] = \{x \in \Sigma^* \mid \delta(q_0, x) = f \in F\}$ pentru că $L(M') = L$.
Deci L se poate scrie ca reuniune de clase de echivalență ale $\equiv_{M'}$.

$2 \Rightarrow 3$

Demonstrăm că orice relație R care satisface 2 este o rafinare a $\equiv_L$.
Adică $x R y \Rightarrow x \equiv_L y$, cu alte cuvinte: clasele de echivalență ale lui R sunt incluse în clasele lui $\equiv_L$. În acest caz $|\Sigma^*/R| \geq |\Sigma^*/\equiv_L|$, deci am avea că $\equiv_L$ este de indice finit.

Fie $x R y \overset{\text{invariantă}}{\Rightarrow}$ ~~$(\forall z \in \Sigma^* \; xz \in L \Leftrightarrow yz \in L)$~~ $\forall z \in \Sigma^* \; xz R yz$.
Pentru că L este reuniunea unor clase de echivalență ale lui R și pentru că $\forall z$, $xzRyz \Rightarrow xz$ și yz sunt în aceeași clasă de echivalență $\Rightarrow$
$\Rightarrow xz \in L \Leftrightarrow yz \in L \Rightarrow x \equiv_L y$.

$3 \Rightarrow 1$.

Dem că $\equiv_L$ este invariantă la dreapta: Fie $x \equiv_L y$ și fie $z \in \Sigma^*$.
$xz \overset{?}{\equiv}_L yz$

$\forall w \in \Sigma^* \; (xz)w \in L \Leftrightarrow (yz)w \in L$ pentru că $x(zw) \in L \Leftrightarrow y(zw) \in L$ (din $\equiv_L$).
Deci $xz \equiv_L yz \Rightarrow \equiv_L$ este invariantă la dreapta.
Fie $[x]$ clasa lui x : $[x] = \{w \mid w \equiv_L x\}$.
$\equiv_L$ are clasele : $[\lambda], [x_1], [x_2], \dots [x_n]$ (indice finit).
$\overline{Q} = \{[\lambda], [x_1], [x_2], \dots, [x_n]\}$.
Obs: dacă $x \in L \Rightarrow \forall y \in [x]$ avem $y \in L$ pentru că $y \equiv_L x$, $\lambda \in \Sigma^* \Rightarrow$
$y\lambda \in L \Leftrightarrow x\lambda \in L$. $\left.\begin{matrix} \\ x \in L \end{matrix}\right| \Rightarrow y \in L$.

Definim automatul : $A = (\overline{Q}, \overline{\Sigma}, \overline{\delta}, [\lambda], \overline{F})$ cu
$\overline{Q} = \{[\lambda], [x_1], [x_2], \dots [x_n]\}$ finită.
$\overline{F} = \{[x] \mid x \in L\}$
$\overline{\delta}([x], a) = [xa]$. este bine definită pt că $\equiv_L$ e invariantă la dreapta (adică pentru $x \equiv_L y$, $\overline{\delta}([x], a) = \overline{\delta}([y], a)$).

Din definiția lui $\overline{\delta}$ avem $\overline{\delta}([\lambda], x) = [x]$, deci $x \in L(A) \Leftrightarrow [x] \in \overline{F} \Leftrightarrow x \in L$.
Deci L este regulat q.e.d.

Teoremă!

Minimizarea DFA:

Automatul DFA cu număr minim de stări care acceptă L este unic abstracție de un izomorfism și este dat de automatul A de mai sus.

Dem:

Am văzut că pentru orice DFA $M=(Q,\Sigma,\delta,q_0,F)$ cu $L(M)=L$, automatul M definește $\equiv_M$ echivalență invariantă la dreapta de indice finit ($1 \Rightarrow 2$).

Din $2 \Rightarrow 3$ am văzut că $\equiv_M$ rafinează $\equiv_L$.

nr de stări din $M \geq |\Sigma^*/\equiv_M|$ (egalitate dacă M nu are stări inaccesibile)

și $|\Sigma^*/\equiv_M| \geq |\Sigma^*/\equiv_L|$ $\Rightarrow$ orice automat M cu $L(M)=L$ are cel puțin atâtea stări ca automatul A din $3 \Rightarrow 1$.

Dacă nr. de stări din M = nr de stări din $A \Rightarrow |\Sigma^*/\equiv_M| = |\Sigma^*/\equiv_L|$ și $\equiv_M$ era o rafinare a $\equiv_L$

$\Rightarrow x \equiv_M y \Rightarrow x \equiv_L y \Rightarrow \equiv_M = \equiv_L$.

Definim izomorfismul dintre M și A: $f: Q \to \overline{Q}$ și $f(q)=[x] \Leftrightarrow \delta(q_0,x)=q$

f bine definită, izomorfism.

□.

Teorema ne dă existența și unicitatea automatului minim, dar nu și cum să îl găsim.

Dăm un algoritm de complexitate $O(|\Sigma|\cdot|Q|^2)$ → Σ alfabet, Q stările

cel mai bun algoritm cunoscut: Algoritmul lui Hopcroft $O(|\Sigma|\cdot|Q|\log|Q|)$.

Pentru limbaje finite: Krivoi, Revuz $O(|\Sigma|\cdot|Q|)$.

Echivalența pe stări: pentru $M=(Q,\Sigma,\delta,q_0,F)$ un DFA fără stări inaccesibile

$$p \equiv q \Leftrightarrow (\forall w \in \Sigma^* \ \delta(p,w) \in F \Leftrightarrow \delta(q,w) \in F).$$

$\equiv$ este relație de echivalență și avem o bijecție φ de la clasele lui $\equiv$ la $\overline{Q}$:

$$\varphi(\hat{q}) = [w] \Leftrightarrow \delta(q_0,w) \in \hat{q}.$$

Deci putem construi $A=(\overline{Q},\overline{\Sigma},\overline{\delta},[\lambda],\overline{F})$ dacă calculăm clasele lui $\equiv$.

Căutăm stările neechivalente (în felul ăsta găsim echivalențele de stări

$p \not\equiv q \iff \exists x \in \Sigma^*$ cu $\delta(p,x) \in F$ și $\delta(q,x) \notin F$ sau invers.

Algoritm:

1. pentru $p \in F$ și $q \in Q - F$ pun 1 în matricea $A[p,q]$, 0 altfel.
2. pentru $p,q \in Q$ construiesc o listă goală
3. pentru orice pereche (p,q) nemarcată în A ($A[p,q]==0$).
4. Dacă $\exists a \in \Sigma$ a. î. $(\overbrace{\delta(p,a)}^{p'}, \overbrace{\delta(q,a)}^{q'})$ e marcată în A ($A[\delta...\delta]==1$)
5. marcăm (p,q) ($A[p,q]=1$).
6. marcăm toate perechile de stări din listele (p,q) și din listele perechilor marcate în acest pas. $(p'',q'')-(p''',q''')$
7. else
8. pentru toate $a \in \Sigma$
9. punem (p,q) în lista lui $(\delta(p,a), \delta(q,a))$.

structuri folosite: matrice $|Q| \times |Q|$ în care marcăm cu 1 stările neechivalente pentru fiecare pereche (p,q) o listă L_{pq} de perechi de stări: perechile neechivalente DACĂ aflăm că p și q sunt neechivalente.

Lema: pentru un DFA $A = (Q, \Sigma, \delta, q_0, F)$. $p \not\equiv q \iff$ în matricea calculată de alg. la poziția (p,q) avem 1.

Dem. inducție după lungimea șirului cel mai scurt care face diferența

Lema: complexitatea algoritmului este $O(|\Sigma| \cdot |Q|^2)$:

Dem: Liniile 1, 2: $O(|Q|^2)$.

Liniile 3-9 executate de $O(|Q|^2)$.

Linia 6: în timp proporțional cu lungimile tuturor listelor.

Fiecare pereche (p_1,p_2) apare cel mult în $O(|\Sigma|)$ liste $\Rightarrow$ în total linia 6 se execută în $O(|\Sigma| \cdot |Q|^2)$.

Am găsit stările echivalente, cum minimizăm?

$p \equiv q \Rightarrow$ putem elimina q : $\forall r \in Q$ cu $\delta(r,a) = q$ definim $\delta'(r,a) = p$.

$\hat{M} = (Q/\equiv, \Sigma, \hat{\delta}, \hat{q}_0, F/\equiv)$ cu $Q/\equiv = \{\hat{q} \mid q \in Q\}$, $\hat{q} = \{p \mid p \equiv q, p \in Q\}$

$F/\equiv = \{\hat{f} \mid f \in F\}$

$\hat{\delta}(\hat{q}, a) = \widehat{\delta(q,a)}$.

acest automat este bine definit și izomorf cu automatul minimal pt L.

Dem. $q \equiv p \Rightarrow \delta(q,a) = \delta(p,a)$ $\forall q,p \in Q$, $\forall a \in \Sigma$

$\hat{\delta}(\hat{q}_0, w) = \widehat{\delta(q_0, w)} \Rightarrow L(M) = L(\hat{M})$.

Pentru minimalitatea lui $\hat{M}$:

presupunem că $\hat{M}$ are mai multe stări decât aut. minimal $\Rightarrow$

$\Rightarrow \exists\, p,q$ din M cu $\hat{p} \neq \hat{q}$ și $\exists\, x,y \in \Sigma^*$ cu $x \equiv_L y$ și $\delta(q_0, x) = p$

$\delta(q_0, y) = q$.

Din $\hat{p} \neq \hat{q} \Rightarrow \exists w \in \Sigma^*$ cu $\delta(q,w) \in F$ și $\delta(p,w) \notin F$ sau invers $\Rightarrow$

$\Rightarrow \delta(q_0, xw) \in F$ și $\delta(q_0, yw) \notin F$ sau invers $\Rightarrow (xw, yw) \notin \equiv_{L(M)}$

contradicție pt. că $\equiv_{L(M)}$ e invariantă la dreapta. q.e.d.