

1 Definiții

Definiție:

Fie Σ_1, Σ_2 două alfabete. O substituție (vezi pagina 33 din PDF) este o funcție $\sigma : \Sigma_1^* \rightarrow 2^{\Sigma_2^*}$ cu două proprietăți:

1. $\sigma(\lambda) = \{\lambda\}$;
2. $\sigma(x \cdot y) = \sigma(x) \cdot \sigma(y)$.

Evident, definirea funcției σ pe literele din Σ_1 se definește complet pe σ . Se poate extinde la limbaje :

$$\sigma(L) = \bigcup_{x \in L} \sigma(x), \text{ pentru } L \subseteq \Sigma_1^*$$

Exemplu : $\Sigma_1 = \{a, b\}, \Sigma_2 = \{a, b, c\}$ și o substituție $\sigma : \Sigma_1^* \rightarrow 2^{\Sigma_2^*}$

$$\begin{aligned} \sigma(a) &= \{ab, ac, b\}, \sigma(b) = \{b, ba\} \\ \sigma(ba) &= \{b, ba\} \cdot \{ab, ac, b\} = \{bab, bac, bb, baab, baac\} \end{aligned}$$

O substituție $f : \Sigma_1^* \rightarrow 2^{\Sigma_2^*}$ se numește morfism dacă $|f(a)| = 1, \forall a \in \Sigma_1$ (adica fiecare litera are asociat un limbaj de un cuvânt)

Morfismele se definesc și ca: $k : \Sigma_1^* \rightarrow \Sigma_2^*$ cu proprietățile :

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

2 Închiderea la substituții și morfisme inverse

Fie $k : \Sigma_1^* \rightarrow \Sigma_2^*$ un morfism . Pentru $w \in \Sigma_2^*, k^{-1}(w) = \{x \mid x \in \Sigma_1^*, k(x) = w\}$

extindem la limbaje : $k^{-1}(L) = \{x \mid k(x) \in L, x \in \Sigma_1^*\}, L \subseteq \Sigma_2^*$

Teoremă : Limbajele regulate sunt închise la :

1. Substituții regulate
2. Morfisme
3. Morfisme inverse

Demonstrație:

1. REG închisă la substituții regulate :

Fie $\sigma : \Sigma_1^* \rightarrow 2^{\Sigma_2^*}$ substituție cu proprietatea că $\sigma(a)$ este regulat, $\forall a \in \Sigma_1$. Fie $L1 \subseteq \Sigma_1^*$ un limbaj regulat.

Trebuie să demonstrăm că $\sigma(L1)$ este regulat.

Deoarece $L1$ este regulat, există o expresie regulată r_1 care descrie $L1$.

Deci $L(r1) = L1$. Deoarece fiecare $\sigma(a)$ este regulat există expresiile r_a expresii regulate care descriu $\sigma(a)$, $\forall a \in \Sigma_1$.

$r1$ este ER peste Σ_1 , r 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 expresii regulate), peste Σ_2

Trebuie să demonstrăm că $L(r_2) = \sigma(L1) \iff L(r_2) = \sigma(L(r_1))$

Demonstrăm prin inducție după numărul de operatori din r_1 :

Baza: r_1 are 0 operatori $\implies r1 \in \{\emptyset, \lambda\} \cup \Sigma_1$

Daca: $r_1 = \emptyset \implies r2 = \emptyset \implies L(r_2) = L(\emptyset) = \emptyset = \sigma(\emptyset)$

$r_1 = \lambda \implies r2 = \lambda \implies L(r_2) = L(\lambda) = \{\lambda\} = \sigma(\{\lambda\})$

$r_1 = a \in \Sigma_1 \implies r2 = r_a \implies L(r_a) = \sigma(a)$ din definiția lui r_a

Ipoteza inductivă : Presupunem că $L(r_2) = \sigma(L1)$ pentru expresia r_1 cu cel mult k operatori

Saltul inductiv : Demonstrăm pentru $k+1$ operatori : $L(r_2)$

Caz 1 : $r_1 = r'_1 + r''_1$ (sau $r_1 = r'_1 \cup r''_1$). Din construcția lui r_2 avem că $r_2 = r'_2 + r''_2$ (daca 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 $L(r_2) \stackrel{def.R.E.}{=} L(r'_2) \cup L(r''_2) \stackrel{I.I.}{=} \sigma(L(r'_1)) \cup \sigma(L(r''_1)) \stackrel{def.subst.}{=} \sigma(L(r'_1) \cup L(r''_1)) \stackrel{def.R.E.}{=} \sigma(L(r'_1 \cup r''_1)) \stackrel{def.r1}{=} \sigma(L(r_1))$.

Caz 2: $r_1 = r'_1 \cdot r''_1 \implies r_2 = r'_2 \cdot r''_2$ similar

Caz 3: $r_1 = r'^*_1 \implies r_2 = r'^*_2$

$L(r_2) \stackrel{def.R.E.}{=} (L(r'_2))^* \stackrel{I.I.}{=} (\sigma(L(r'_1)))^* = \sigma((L(r'_1))^*) = \sigma(L(r'^*_1)) = \sigma(L(r_1))$

Demonstratie 2: Reg este închisă la morfisme:

Fie $L \subseteq \Sigma_1^*$ limbaj regulat si $h : \Sigma_1^* \rightarrow \Sigma_2^*$ morfism. Se demonstrează ca $h(L) \in Reg$. Este imediat din demonstrația 1 pentru ca limbajele finite sunt regulate, deci caz particular pentru demonstrația anterioară.

Demonstratie 3: Reg este închisă la morfisme inverse:

Fie $L \subseteq \Sigma_2^*$ limbaj regulat si $h : \Sigma_1^* \rightarrow \Sigma_2^*$ morfism. Se demonstrează ca $h^{-1}(L) \in Reg$.

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 $\delta'(q, a) = \delta(q, h(a)) \in Q$. \leftarrow extinderea lui δ la cuv $\forall q \in Q, \forall a \in \Sigma_1$.

Demonstrăm ca $\delta'(q, x) = \delta(q, h(x)), \forall x \in \Sigma_1^* \leftarrow$ extinderile lui δ' si δ la cuvinte.

Inducție după lungimea lui x :

$$|x| = 0 \implies x = \lambda \implies \delta'(q, \lambda) = q = \delta(q, \lambda), \text{ h morfism } \implies h(\lambda) = \lambda.$$

Presupunem adevărat pentru n . Demonstrăm pentru $n+1$:

Fie $|x| = n + 1, x = x'a, \forall a \in \Sigma_1$.

$$\begin{aligned} \delta'(q, x) &= \delta'(q, x'a) = \delta'(\delta'(q, x'), a) \stackrel{I.I.}{=} \delta'(\delta(q, h(x')), a) \stackrel{def.\delta'}{=} \delta(\delta(q, h(x')), h(a)) \stackrel{def.ext.\delta}{=} \\ &\delta(q, h(x') \cdot h(a)) \stackrel{h.morfism}{=} \delta(q, h(x'a)) = \delta(q, h(x)). \end{aligned}$$

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

Avem că $\delta'(q_0, w) \in F \iff \delta(q_0, h(w)) \in F$ deci $w \in L(M) \iff h(w) \in L$.

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

Cum se folosește:

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

Presupunem că L este regulat \implies pentru orice morfism h avem $h^{-1}(L)$ este regulat:

Fie $h_1 : \{a, b, c\}^* \rightarrow \{a, b, c\}^*$ cu $h_1(a) = a$,

$$h_1(b) = ba,$$

$$h_1(c) = a$$

$\implies h_1^{-1}(L)$ 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 $h_2 : \{a, b, c\}^* \rightarrow \{0, 1\}^*$, $h_2(a) = 0, h_2(b) = 1, h_2(c) = 1$.

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

sau $h_1^{-1}(L) \cap a^* b c^* = \{a^n b c^{n-1} \mid n \geq 1\} = L'$. $h_2(L') = \{0^n 1^n \mid n \geq 1\}$, deci contradicție.

3 Minimizarea DFA

1. Echivalență pe cuvinte:

Pentru $L \subseteq \Sigma^*$ un limbaj definim \equiv_L astfel:

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

\equiv_L este relație de echivalență.

2. Invarianta la dreapta la concatenare:

O relație se numește invariantă la dreapta față de concatenare dacă $xRy \implies \forall z \in \Sigma^*, xzRyz$.

3. Echivalența dată de un automat:

Fie $M = (Q, \Sigma, \delta, q_0, F)$ un DFA. Definim \equiv_M :

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

\equiv_M este relație de echivalență și invariantă la dreapta.

4. Indicele unei relații de echivalență:

$|\Sigma^*/R|$ = numărul 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

Demonstrația teoremei:

1 \implies 2 : L este regulat \implies există un DFA $M = (Q, \Sigma, \delta, q_0, F)$ astfel încât $L(M) = L$.

Construim M' din M eliminând stările inaccesibile și îl facem pe automat 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 ca $L(M') = L$.

Deci L se poate scrie ca reuniune de clase de echivalență ale relației $\equiv_{M'}$.

2 \implies 3 : Demonstrăm că orice relație R care satisface 2 este o rafinare a relației \equiv_L .

Adică $xRy \implies x \equiv_L y$, 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 $xRy \xrightarrow{\text{invarianta}} \forall z \in \Sigma^* xzRyz$.

Pentru că L este reuniunea unor clase de echivalență ale lui R și pentru că $\forall z, xzRyz \implies xz$ și yz sunt în aceeași clasă de echivalență $\implies xz \in L \iff yz \in L \implies x \equiv_L y$.

3 \implies 1 : Demonstrăm ca \equiv_L este invarianta la dreapta : Fie $x \equiv_L y$ și fie $z \in \Sigma^*$.
 $xz \equiv_L yz$?

$\forall w \in \Sigma^*, (xz)w \in L \iff (yz)w \in L$ pentru ca $x(zw) \in L \iff y(zw) \in L$ (din \equiv_L).

Deci $xz \equiv_L yz \implies \equiv_L$ 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).

$\overline{Q} = \{[\lambda], [x_1], [x_2], \dots, [x_n]\}$.

Observație : dacă $x \in L \implies \forall y \in [x]$ avem $y \in L$ pentru că $y \equiv_L x, \lambda \in \Sigma^* \implies y\lambda \in L \iff x\lambda \in L$,

$x \in L$

$\implies y \in L$

Definim automatul: $A = (\overline{Q}, \overline{\Sigma}, \overline{\delta}, [\lambda], \overline{F})$ cu

$\overline{Q} = \{[\lambda], [x_1], [x_2], \dots, [x_n]\}$ finita.

$\overline{F} = \{[x] \mid x \in L\}$

$\overline{\delta}([x], a) = [xa]$ este bine definită pentru ca \equiv_L este invarianta 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 $x \in L(A) \iff [x] \in \overline{F} \iff x \in L$.

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.

Demonstrație :

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

(1 \implies 2).

Din 2 \implies 3 am văzut că \equiv_M rafinează \equiv_L .

Numărul de stări din $M \geq |\Sigma^* / \equiv_M|$ (egalitate dacă M nu are stări inaccesibile) și $|\Sigma^* / \equiv_M| \geq |\Sigma^* / \equiv_L| \implies$ orice automat M cu $L(M) = L$ are cel puțin atâtea stări ca automatul A din 3 \implies 1.

Dacă numărul de stări din $M =$ numărul de stări din $A \implies |\Sigma^* / \equiv_M| = |\Sigma^* / \equiv_L|$ și \equiv_M era o rafinare a $\equiv_L \implies x \equiv_M y \implies x \equiv_L y \implies \equiv_M = \equiv_L$.

Definim izomorfismul dintre M și A : $f : Q \rightarrow \overline{Q}$ și $f(q) = [x] \iff \delta(q_0, x) = q$
 funcția f este bine definită, izomorfism.

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

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

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

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

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

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

\equiv este relație de echivalență și avem o bijecție φ de la clasele lui \equiv la $\overline{Q} : \varphi(\widehat{q}) = [w] \iff \delta(q_0, w) \in \widehat{q}$.

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

Căutăm stările neechivalente (în acest fel 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 $\exists a \in \Sigma$ astfel încât $(p', q') = (\delta(p, a), \delta(q, a))$ este marcată în A ($A[\delta... \delta] == 1$)
5. marcăm (p, q) ($A[p, q] = 1$). $(p'', q'') - (p''', q''')$
6. marcăm toate perechile de stări din listele (p, q) și din listele perechilor marcate în acest pas.
7. altfel, pentru toate $a \in \Sigma$ 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 algoritm la poziția (p, q) avem 1.

Demonstratie : inducție după lungimea șirului cel mai scurt care face diferența.

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

Demonstrație : 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 \implies în total linia 6 se execută în $O(|\Sigma| \cdot |Q|^2)$.

Am găsit stările echivalente, cum minimizăm?

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

$\widehat{M} = (Q / \equiv, \Sigma, \widehat{\delta}, \widehat{q}_0, F / \equiv)$ cu

$Q / \equiv = \{\widehat{q} \mid q \in Q\}, \widehat{q} = \{p \mid p \equiv q, p \in Q\}$

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

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

acest automat este bine definit si izomorf cu automatul minimal pentru L.

Demonstrație:

$$q \equiv p \implies \delta(q, a) = \delta(p, a) \quad \forall q, p \in Q \quad \forall a \in \Sigma$$

$$\widehat{\delta}(\widehat{q_0}, w) = \widehat{\delta(q_0, w)} \implies L(M) = L(\widehat{M}).$$

Pentru minimizarea lui \widehat{M} :

Presupunem că \widehat{M} are mai multe stări decat automatul minimal $\implies \exists p, q \in M$ cu $\widehat{p} \neq \widehat{q}$ și $\exists x, y \in \Sigma^*$ cu $x \equiv_L y$ și $\delta(q_0, x) = p, \delta(q_0, y) = q$.

Din $\widehat{p} \neq \widehat{q} \implies \exists w \in \Sigma^*$ cu $\delta(q, w) \in F$ si $\delta(p, w) \notin F$ sau invers $\implies \delta(q_0, xw) \in F$ si $\delta(q_0, yw) \notin F$ sau invers $\implies (xw, yw) \not\equiv_{L(M)}$

Constradicție pentru că $\equiv_{L(M)}$ este invariantă la dreapta. q.e.d.