Formális nyelvek I.
Véges automaták és reguláris nyelvek

Fülöp Zoltán

SZTE TTIK Informatikai Intézet
Számítástudomány Alapjai Tanszék
6720 Szeged, Árpád tér 2.

Irodalom

• Fülöp Zoltán, Formális nyelvek és szintaktikus elemzésük, Polygon, 2004.
• Ésik Zoltán, Gombás Éva és Iván Szabolcs: Automaták és formális nyelvek példatár, Typotex Kiadó, 2011.
• J. E. Hopcroft és R. Motwani, J. D. Ullman, Introduction to Automata Theory, Languages, and Computation, Addison Wesley, 2001 és Pearson Education Limited 2014. (A biblia.)
• Peter Linz, An Introduction to Formal Languages and Automata, Jones & Barlett Learning, 2012.

Ajánlott az előadások látogatása és a jegyzetelés! A vizsgán az előadáson elhangzottakat is tudni kell.

A formális nyelvek egy alkalmazása

Egy (programozási) nyelv szintaxisa: azon szabályok összessége, amelyek meghatározzák a nyelvet.

Hogyan, milyen módszerrel adható meg a programozási nyelvek szintaxisa?

A legelterjedtebb módszer a *generatív nyelvtannal* történő szintaxis megadás.

Adjuk meg az A, B és C változókból, a 0 és 1 konstansokból, a $+$ és a $*$ műveleti jelekből, valamint a (és) zárójelekből felépíthető aritmetikai kifejezések szintaxisát!

Ilyenek például az A, 1, $A+1$, $A+B$, $A*(B+1)$ aritmetikai kifejezések. Az összes ilyen kifejezés egy nyelvet alkot.

A formális nyelvek egy alkalmazása

Aritmetikai kifejezések

Aritmetikai kifejezéseknek az A, B és C változó jelekből, a 0 és 1 konstans jelekből, a $+$ és $*$ műveleti jelekből, valamint a (és) csoportosító jelekből, a

$$\begin{array}{lcl}
\langle kif\rangle & \rightarrow & \langle tag\rangle \mid \langle kif\rangle + \langle tag\rangle \\
\langle tag\rangle & \rightarrow & \langle fakt\rangle \mid \langle tag\rangle * \langle fakt\rangle \\
\langle fakt\rangle & \rightarrow & (\langle kif\rangle) \mid \langle valt\rangle \mid \langle konst\rangle \\
\langle valt\rangle & \rightarrow & A \mid B \mid C \\
\langle konst\rangle & \rightarrow & 0 \mid 1
\end{array}$$

szabályok alkalmazásával felépíthető jelsorozatokat (szavakat) nevezzük. A | jel választási lehetőséget jelent, olvassuk "vagy"-nak. Ez egy *generatív nyelvtan*.

A formális nyelvek egy alkalmazása

Levezetés:

$$
\begin{array}{ll}
 & \underline{\langle kif \rangle} \\
\Rightarrow & \underline{\langle tag \rangle} \\
\Rightarrow & \langle tag \rangle * \underline{\langle fakt \rangle} \\
\Rightarrow & \langle tag \rangle * (\underline{\langle kif \rangle}) \\
\Rightarrow & \langle tag \rangle * (\underline{\langle kif \rangle} + \langle tag \rangle) \\
\Rightarrow & \underline{\langle tag \rangle} * (\langle tag \rangle + \langle tag \rangle) \\
\Rightarrow & \langle fakt \rangle * (\underline{\langle tag \rangle} + \langle tag \rangle) \\
\Rightarrow & \langle fakt \rangle * (\langle fakt \rangle + \underline{\langle tag \rangle}) \\
\Rightarrow & \langle fakt \rangle * (\langle fakt \rangle + \langle fakt \rangle)
\end{array}
$$

Minden lépésben az aláhúzott szintaktikai egységet helyettesítjük a megfelelő szabály jobb oldalán álló valamelyik kifejezéssel.

A formális nyelvek egy alkalmazása

Levezetés:

$$
\begin{array}{ll}
\Rightarrow & \underline{\langle fakt \rangle} * (\langle fakt \rangle + \langle fakt \rangle) \\
\Rightarrow & \langle valt \rangle * (\underline{\langle fakt \rangle} + \langle fakt \rangle) \\
\Rightarrow & \langle valt \rangle * (\langle valt \rangle + \underline{\langle fakt \rangle}) \\
\Rightarrow & \underline{\langle valt \rangle} * (\langle valt \rangle + \langle konst \rangle) \\
\Rightarrow & A * (\underline{\langle valt \rangle} + \langle konst \rangle) \\
\Rightarrow & A * (B + \underline{\langle konst \rangle}) \\
\Rightarrow & A * (B + 1)
\end{array}
$$

Jelölés: $\langle kif \rangle \Rightarrow^* A * (B + 1)$

A formális nyelvek egy alkalmazása

Az A, B, C, 0, 1, $+$, $*$, $($ és $)$ jelekből álló aritmetikai kifejezés szintaxisa:

egy jelsorozat (vagy szó) akkor és csak akkor aritmetikai kifejezés, ha a $\langle kif \rangle$*-ből a fenti szintaktikai szabályok alkalmazásával történő levezetéssel megkapható.*

Röviden:

$$w \text{ szó aritmetikai kifejezés} \iff \langle kif \rangle \Rightarrow^* w.$$

A formális nyelvek egy alkalmazása

Egy másik példa: a FONYA programozási nyelv szintaxisa:

$$
\begin{array}{lll}
\langle program \rangle & \rightarrow & \langle ut.lista \rangle. \\
\langle ut.lista \rangle & \rightarrow & \langle ut \rangle \mid \langle ut \rangle ; \langle ut.lista \rangle \\
\langle ut \rangle & \rightarrow & \langle ert.ado \rangle \mid \langle ifut \rangle \mid \\
 & & \langle whileut \rangle \mid \langle blokk \rangle \\
\langle ert.ado \rangle & \rightarrow & \langle valt \rangle := \langle kif \rangle \\
\langle ifut \rangle & \rightarrow & \textbf{if } \langle relacio \rangle \textbf{ then } \langle ut \rangle \textbf{ else } \langle ut \rangle \\
\langle whileut \rangle & \rightarrow & \textbf{while } \langle relacio \rangle \textbf{ do } \langle ut \rangle \\
\langle blokk \rangle & \rightarrow & \textbf{begin } \langle ut.lista \rangle \textbf{ end} \\
\langle relacio \rangle & \rightarrow & \langle kif \rangle \langle relaciojel \rangle \langle kif \rangle \\
\langle relaciojel \rangle & \rightarrow & < \mid > \mid = \mid \neq
\end{array}
$$

A formális nyelvek egy alkalmazása

Egy másik példa: a FONYA programozási nyelv szintaxisa:

$$
\begin{array}{lcl}
\langle kif\rangle & \to & \langle tag\rangle \mid \langle kif\rangle + \langle tag\rangle \\
\langle tag\rangle & \to & \langle fakt\rangle \mid \langle tag\rangle * \langle fakt\rangle \\
\langle fakt\rangle & \to & (\langle kif\rangle) \mid \langle valt\rangle \mid \langle konst\rangle \\
\langle valt\rangle & \to & A \mid B \mid C \\
\langle konst\rangle & \to & 0 \mid 1
\end{array}
$$

A formális nyelvek egy alkalmazása

Egy w jelsorozat akkor és csakis akkor szintaktikusan helyes FONYA nyelvű program, ha $\langle program\rangle \Rightarrow^* w$.

Ilyen például a bal oldali jelsorozat és nem ilyen a jobb oldali:

$A := 0;$	$A := 0;$
while $A < C$ **do**	**while** $A + C$ **do**
begin $A := A + 1;$	**begin** $A := A + 1$
$B := B * C$	$B := B * C$
end;	**end**;
$C := C * B.$	$C := C * B.$

A jobb oldaliban két szintaktikus hiba van!

A formális nyelvek egy alkalmazása

Az elemzés alapkérdése:

Amennyiben adott egy programozási nyelv szintaxisa és adott egy – ezen a nyelven írott – program, akkor hogyan tudjuk eldönteni azt, hogy az adott program engedelmeskedik-e a szintaxisnak, vagyis szintaktikusan helyes-e?

Röviden: igaz-e, hogy $\langle program\rangle \Rightarrow^* w$?

Több algoritmus is létezik, lásd a "Szintaktikus elemzési módszerek" c. kurzus anyagát.

Általános fogalmak, jelölések

Ábécé: szimbólumoknak egy tetszőleges véges, nemüres halmaza. Általában Σ-val jelöljük.

Σ *ábécé feletti szó*: egy $a_1 \dots a_k$ alakú sorozat, ahol $k \geq 0$ és $a_1, \dots, a_k \in \Sigma$.

$k = 0$ eset: *üres szónak nevezzük*, jele ε.

Példa ábécére és szavakra: $\Sigma = \{a, b\}$, $\varepsilon, a, b, aba, aababba,$ stb.

Programozásban: ASCII, Unicode ábécék, BEGIN, END, IF, ALMA, K1, további kulcsszavak, azonosítók, 123, -412.2, K1+123, egyéb számok, kifejezések stb pedig szavak.

Általános fogalmak, jelölések

Összes szavak halmaza:

$$\Sigma^* = \{a_1 \ldots a_k \mid k \geq 0,\ a_1, \ldots, a_k \in \Sigma\}$$
$$\Sigma^+ = \{a_1 \ldots a_k \mid k \geq 1,\ a_1, \ldots, a_k \in \Sigma\} = \Sigma^* - \{\varepsilon\}$$

Példa: $\Sigma^* = \{\varepsilon, a, b, aa, ab, ba, bb, aaa, \ldots\}$,
$\Sigma^+ = \{a, b, aa, ab, ba, bb, aaa, \ldots\}$

Konkatenáció: az $u, v \in \Sigma^*$ szavak egymás után írásával kapott $uv \in \Sigma^*$ szó.
Példa: ha $u = abb$, $v = ba$, akkor $uv = abbba$, $u\varepsilon = \varepsilon u = u$.
A konkatenáció asszociatív: $u(vw) = (uv)w$ minden $u, v, w \in \Sigma^*$-ra

Hatványozás: $u^n = \overbrace{uu \ldots u}^{n}$, $u^0 = \varepsilon$, $u = abb$-re $u^3 = abbabbabb$.

Általános fogalmak, jelölések

Ha $w = xy$, akkor x a w *prefixe*, y a w *suffixe*.

Példa: abb pefixei ε, a, ab, abb, suffixei abb, bb, b, ε

Egy w szó *hosszán* a benne előforduló betűk multiplicitással vett számát értjük. A jele $|w|$, a pontos definíció a következő:

(i) ha $w = \varepsilon$, akkor $|w| = 0$,
(ii) ha $w = av$, valamely $a \in \Sigma$ és $v \in \Sigma^*$-ra, akkor $|w| = 1 + |v|$.

Példa: $|\varepsilon| = 0$, $|a| = 1$, $|ab| = 2$, $|abb| = 3$.

Általános fogalmak, jelölések

Nyelv: Σ^* tetszőleges részhalmazát Σ feletti nyelvnek nevezzük.

Példa $\{a, b\}$ feletti nyelvekre:

- $\{aa, bab, abba\}$ (véges nyelv),
- $\{w \in \Sigma^* \mid |w| \text{ páros }\}$,
- $\{\varepsilon, ab, aabb, aaabbb, \ldots\} = \{a^n b^n \mid n \geq 0\}$.

A már ismert aritmetikai kifejezések halmaza egy nyelv az $\{A, B, C, 0, 1, +, *, (,)\}$ ábécé felett.

Az összes Σ feletti nyelvek halmaza: $\mathcal{P}(\Sigma^*)$ (kontinuum számosságú).

Műveletek nyelvekkel

Legyenek $L, L_1, L_2 \subseteq \Sigma^*$ nyelvek. Az

$$L_1 \cup L_2, L_1 \cap L_2 \text{ és } L_1 - L_2$$

az ismert halmazelméleti műveletek. Továbbá

- $\overline{L} = \Sigma^* - L$ az L *komplementere*,
- $L_1 L_2 = \{uv \mid u \in L_1, v \in L_2\}$ az L_1 és L_2 *konkatenációja*,

  Minden lehetséges módon választunk L_1-ből és L_2-ből szavakat és "összeláncoljuk" (konkatenáljuk) őket:

  $\{\varepsilon, ab, bab\}\{b, ba\} = \{b, ba, abb, abba, babb, babba\}$

  $\Sigma^*\{aba\} = \{waba \mid w \in \Sigma^*\}$, az aba-ra végződő szavak halmaza.

Műveletek nyelvekkel

A nyelvek konkatenációja is asszociatív:

$$L_1(L_2L_3) = (L_1L_2)L_3,$$

ezért a zárójelezés elhagyható.

- *Hatványozás*: $L^n = \overbrace{LL\ldots L}^{n}$, $L^0 = \{\varepsilon\}$
- $L^* = \{\varepsilon\} \cup L \cup LL \cup LLL \cup \ldots$ az L *iteráltja*,

  Tetszőleges számú (0 is megengedett) L-beli szó konkatenációjaként megkapható szavak halmaza.

  $\{ab, ba\}^* = \{\varepsilon, ab, ba, abab, abba, baab, baba, ababab, \ldots\}$

- $L^+ = L \cup LL \cup LLL \cup \ldots$ (a 0 eset kizárva).

Műveletek nyelvekkel

Megjegyzés: $\emptyset^* = \{\varepsilon\}$ és minden L nyelvre $\varepsilon \in L^*$.

Továbbá: $L^* = \{\varepsilon\} \cup L^+$.

Az $\cup$, $\cap$ és komplementer műveleteket Boole műveleteknek, az $\cup$, konkatenáció és iteráció műveleteket pedig reguláris műveleteknek nevezzük.

Műveletek nyelvekkel

Néhány, nyelv műveletekre vonatkozó azonosság:

$L_1 \cup (L_2 \cup L_3) = (L_1 \cup L_2) \cup L_3$	$L_1(L_2L_3) = (L_1L_2)L_3$
$L_1 \cup L_2 = L_2 \cup L_1$	$L\{\varepsilon\} = \{\varepsilon\}L = L$
$L \cup L = L$	
$L \cup \emptyset = \emptyset \cup L = L$	$L\emptyset = \emptyset L = \emptyset$

Továbbá:

$$L_1(L_2 \cup L_3) = L_1L_2 \cup L_1L_3$$

$$(L_1 \cup L_2)L_3 = L_1L_3 \cup L_2L_3$$

$$(L_1 \cup L_2)^* = (L_1^*L_2)^*L_1^*$$

$$(L_1L_2)^* = \{\varepsilon\} \cup L_1(L_2L_1)^*L_2$$

Generatív nyelvtanok

Egy olyan, könnyen leírható eszközzel ismerkedünk meg, amely alkalmas (általában végtelen) nyelvek megadására. Az eszköz neve generatív nyelvtan (vagy generatív grammatika).

Más szóval: a generatív nyelvtanok olyan végesen specifikálható eszközök, melyekkel nyelveket tudunk reprezentálni.

Generatív nyelvtanok

Generatív nyelvtan: egy $G = (N, \Sigma, P, S)$ négyes, ahol:

- N egy ábécé, a *nemterminális ábécé*,
- Σ egy ábécé, a *terminális (befejező, végső)* ábécé, amire $N \cap \Sigma = \emptyset$,
- $S \in N$ a *kezdő szimbólum* (vagy *start szimbólum*),
- P pedig $\alpha \to \beta$ alakú ún. *átírási szabályok* véges halmaza, ahol $\alpha, \beta \in (N \cup \Sigma)^*$ és α-ban van legalább egy nemterminális betű.

(α a szabály bal oldala, β a jobb oldala.)

Generatív nyelvtanok

Példa:

$$G_1 = (\{S, A\}, \{a, b\}, \{S \to aAb, aA \to aaAb, A \to \varepsilon\}, S)$$

egy nyelvtan, ahol

- $\{S, A\}$ a nemterminális ábécé,
- $\{a, b\}$ a terminális ábécé,
- S a kezdő (start) szimbólum,
- $\{S \to aAb, aA \to aaAb, A \to \varepsilon\}$ a szabályok halmaza.

$aA \to aaAb$ egy szabály, aminek a bal oldala aA, a jobb oldala pedig $aaAb$

Generatív nyelvtanok

Közvetlen levezetés (*deriváció*):

tetszőleges $\gamma, \delta \in (N \cup \Sigma)^*$ esetén $\gamma \Rightarrow_G \delta$, ha van olyan $\alpha \to \beta \in P$ szabály és vannak olyan $\alpha', \beta' \in (N \cup \Sigma)^*$ szavak, amelyekre fennállnak, hogy $\gamma = \alpha' \alpha \beta'$, $\delta = \alpha' \beta \beta'$.

$$\underbrace{\alpha' \mid \alpha \mid \beta'}_{\gamma} \Rightarrow_G \underbrace{\alpha' \mid \beta \mid \beta'}_{\delta}$$

Generatív nyelvtanok

Példa:

$G_1 = (\{S, A\}, \{a, b\}, \{S \to aAb, aA \to aaAb, A \to \varepsilon\}, S)$

$b\underline{aA}Aa \Rightarrow b\underline{aaAb}Aa$ az $aA \to aaAb$ szabállyal

$baaAb\underline{A}a \Rightarrow baaAba$ az $A \to \varepsilon$ szabállyal

Generatív nyelvtanok

Levezetések:

$\gamma \Rightarrow_G \delta$: egy lépés

$\gamma \Rightarrow_G^n \delta$: $n \geq 0$ lépés ($\gamma \Rightarrow_G^0 \delta \iff \gamma = \delta$)

$\gamma \Rightarrow_G^+ \delta$: legalább egy lépés

$\gamma \Rightarrow_G^* \delta$: valamennyi (esetleg 0) lépés

Ha nem okoz félreértést, $\Rightarrow_G$ helyett $\Rightarrow$-t írunk.

Generatív nyelvtanok

Példa:

$G_1 = (\{S, A\}, \{a, b\}, \{S \to aAb, aA \to aaAb, A \to \varepsilon\}, S)$

$$\begin{array}{llllll} S & \Rightarrow & aAb & \Rightarrow & aaAbb \\ & \Rightarrow & aaaAbbb & \Rightarrow & aaabbb \end{array}$$

Tehát: $S \Rightarrow^* aaabbb$, $S \Rightarrow^+ aaabbb$, $S \Rightarrow^4 aaabbb$ mind teljesülnek.

Továbbá: $aAb \Rightarrow aaAbb$, $aAb \Rightarrow^2 aaaAbbb$, tehát $aAb \Rightarrow^* aaAbb$, $aAb \Rightarrow^* aaaAbbb$.

Leginkább azok a levezetések érdekelnek bennünket, amelyek a kezdő szimbólumból indulnak ki és terminális szóban végződnek.

Generatív nyelvtanok

$G = (N, \Sigma, P, S)$ *nyelvtan által generált nyelv*:

$$L(G) = \{u \in \Sigma^* \mid S \Rightarrow_G^* u\}.$$

A $G_1 = (\{S, A\}, \{a, b\}, \{S \to aAb, aA \to aaAb, A \to \varepsilon\}, S)$ példában

$$L(G_1) = \{a^n b^n \mid n \geq 1\}.$$

Egy nyelvet általában nem csak egy nyelvtannal lehet generálni: a $G = (N, \Sigma, P, S)$ és $G' = (N', \Sigma', P', S')$ nyelvtanok *ekvivalensek*, ha $L(G) = L(G')$.

Generatív nyelvtan

Példa:

Az előbbi G_1 nyelvtan ekvivalens a

$$G_2 = (\{S\}, \{a, b\}, \{S \to aSb, S \to ab\}, S)$$

nyelvtannal, mert ugyancsak

$$L(G_2) = \{a^n b^n \mid n \geq 1\}.$$

Generatív nyelvtanok

Példa: $G_{ar} = (N, \Sigma, P, S)$, ahol

- $N = \{K, T, F\}$,
- $\Sigma = \{+, *, (,), a\}$,
- $S = K$,
- $P = \{$
  - $K \to K + T$, $K \to T$,
  - $T \to T * F$, $T \to F$,
  - $F \to (K)$, $F \to a\}$.

Ekkor $L(G_{ar})$ az a-ból valamint a $(,), +$ és $*$ jelekből képezhető aritmetikai kifejezések halmaza.

Generatív nyelvtan

Az $A \to \alpha_1, \ldots, A \to \alpha_n$ szabályok felírását a következőképpen rövidítjük: $A \to \alpha_1 \mid \ldots \mid \alpha_n$.

Például, a G_{ar} nyelvtan szabályai megadhatók így is:

- $K \to K + T \mid T$,
- $T \to T * F \mid F$,
- $F \to (K) \mid a$.

Egy levezetés G_{ar}-ban:

$K \Rightarrow T \Rightarrow T * F \Rightarrow F * F \Rightarrow (K) * F \Rightarrow (K + T) * F \Rightarrow^*$

$(F + F) * F \Rightarrow^* (a + a) * a$

Általános jelölések

- Az $a, b, c, d, \ldots$ szimbólumok Σ elemeit,
- az $A, B, C, D, \ldots$, és S szimbólumok N elemeit,
- az $\ldots, U, V, W, X, Y, Z$ szimbólumok $N \cup \Sigma$ elemeit,
- az $\alpha, \beta, \gamma, \delta, \ldots$ szimbólumok $(N \cup \Sigma)^*$ elemeit,
- és az $\ldots, u, v, w, x, y, z$ szimbólumok Σ^* elemeit

fogják jelölni.

Chomsky nyelvosztályok

A $G = (N, \Sigma, P, S)$ nyelvtan

- 0 típusú (vagy *általános*), ha rá semmilyen korlátozás nincs.
- 1 típusú (vagy *környezetfüggő*), ha P-ben minden szabály $\alpha A \beta \to \alpha \delta \beta$ alakú, ahol $\delta \neq \varepsilon$. Kivétel, az $S \to \varepsilon$ szabály, ekkor azonban az S nem szerepelhet semelyik szabály jobb oldalán.
- 2 típusú (vagy *környezetfüggetlen*), ha P-ben minden szabály $A \to \alpha$ alakú.
- 3 típusú (vagy *reguláris*), ha P-ben minden szabály $A \to xB$ vagy $A \to x$ alakú.

Chomsky nyelvosztályok

Egy adott G nyelvtan esetén a legnagyobb olyan $i \in \{0, 1, 2, 3\}$ az érdekes, melyre a nyelvtan i típusú.

Pl. a G_1, G_2 és G_{ar} nyelvtanok rendre 0, 2 és 2 típusúak.

Egy adott $L \subseteq \Sigma^*$ nyelvet i típusúnak mondunk valamely $i \in \{0, 1, 2, 3\}$-re, ha van olyan i típusú G nyelvtan, amelyre $L = L(G)$. Itt is a legnagyobb olyan i az érdekes, melyre a nyelv i típusú.

Pl. az $L(G_1), L(G_2)$ és $L(G_{ar})$ nyelvek mindegyike 2 típusú és be lehet bizonyítani, hogy egyik sem 3 típusú.

Chomsky nyelvosztályok

Az $i \in \{0, 1, 2, 3\}$ típusú nyelvek osztályát $\mathcal{L}_i$-vel jelöljük.

Az nyilvánvaló, hogy $\mathcal{L}_3 \subseteq \mathcal{L}_2$ és $\mathcal{L}_1 \subseteq \mathcal{L}_0$, mert minden 3-típusú nyelvtan 2-típusú is és minden 1-típusú nyelvtan 0-típusú is.

Később látni fogjuk, hogy érvényes a *Chomsky nyelvhierarchia*:

$$\mathcal{L}_3 \subseteq \mathcal{L}_2 \subseteq \mathcal{L}_1 \subseteq \mathcal{L}_0,$$

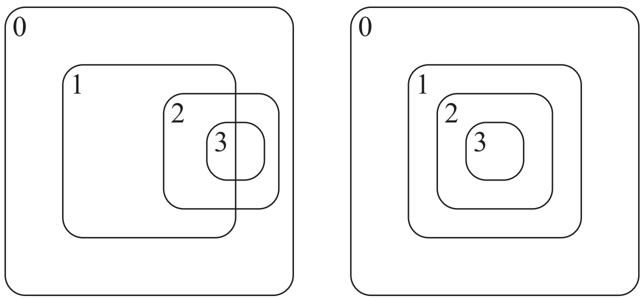
sőt az erősebb

$$\mathcal{L}_3 \subset \mathcal{L}_2 \subset \mathcal{L}_1 \subset \mathcal{L}_0$$

alakja is.

Chomsky nyelvosztályok

Összefoglalás



Nyelvtanok Nyelvek

A bal oldali az eddigiek alapján világos, a jobb oldalit fokozatosan, teljesen a félév végére látjuk be.

Véges automaták, reguláris kifejezések

További program:

Először a reguláris nyelveket (vagyis a reguláris nyelvtanokkal generálható nyelveket) vizsgáljuk.

Bevezetünk további két olyan eszközt, amelyekkel reguláris nyelveket lehet megadni (reprezentálni): a véges automatákat és a reguláris kifejezéseket.

Megmutatjuk, hogy mind a véges automatákkal felismerhető nyelvek, mind a reguláris kifejezésekkel reprezentálható nyelvek megegyeznek a reguláris nyelvekkel.

Véges automaták

Az $M = (Q, \Sigma, \delta, q_0, F)$ rendszert *determinisztikus automatának* nevezzük, ahol:

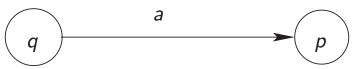
1. Q egy nem üres, véges halmaz, az *állapotok halmaza*,
2. Σ egy ábécé, az *input ábécé*,
3. $q_0 \in Q$ a *kezdő állapot*,
4. $F \subseteq Q$ a *végállapotok halmaza*,
5. $\delta : Q \times \Sigma \to Q$ egy leképezés, az *átmenetfüggvény*.

Véges automaták

Automata megadása irányított gráfként:

Az állapotok a gráf csúcsai,

ha $\delta(q, a) = p$, akkor a q csúcsból egy élet irányítunk a p csúcsba és az élet ellátjuk az a címkével,



és azt mondjuk, hogy az automata a q állapotból az a input szimbólum hatására átmegy a p állapotba.
A kezdő és a végállapotokat reprezentáló csúcsokat megjelöljük.

Véges automaták

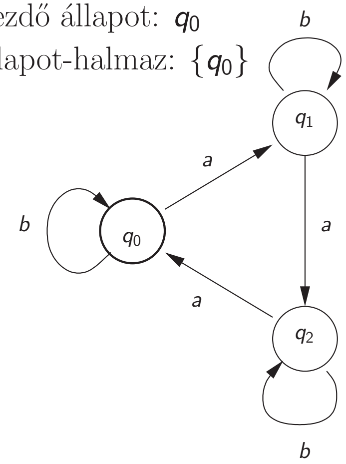
Példa: $M_3 = (Q, \Sigma, \delta, q_0, F)$ egy automata, ahol

- $Q = \{q_0, q_1, q_2\}$,
- $\Sigma = \{a, b\}$,
- $F = \{q_0\}$, továbbá
  - $\delta(q_0, a) = q_1$, $\delta(q_0, b) = q_0$,
  - $\delta(q_1, a) = q_2$, $\delta(q_1, b) = q_1$,
  - $\delta(q_2, a) = q_0$, $\delta(q_2, b) = q_2$.

Véges automaták

Az M_3 automata megadása irányított gráfként:

kezdő állapot: q_0
végállapot-halmaz: $\{q_0\}$



Véges automaták

Automata megadható táblázat formában is. A kezdő állapotot a táblázat első sorába írjuk, a végállapotokat pedig megjelöljük.

Az M_3 automata megadása táblázattal:

δ	a	b
$\star q_0$	q_1	q_0
q_1	q_2	q_1
q_2	q_0	q_2

Véges automaták

M konfigurációinak halmaza: $C = Q \times \Sigma^*$.

A $(q, a_1 \dots a_n)$ konfiguráció azt jelenti, hogy M a q állapotban van és az $a_1 \dots a_n$ szót kapja inputként.

Átmeneti reláció:

$(q, w), (q', w') \in C$ esetén $(q, w) \vdash_M (q', w')$ ha $w = aw'$, valamely $a \in \Sigma$-ra és $\delta(q, a) = q'$.

$(q, w) \vdash_M (q', w')$, egy lépés

$(q, w) \vdash_M^n (q', w')$, $n \geq 0$ lépés

$(q, w) \vdash_M^+ (q', w')$, legalább egy lépés

$(q, w) \vdash_M^* (q', w')$, valamennyi (esetleg 0) lépés

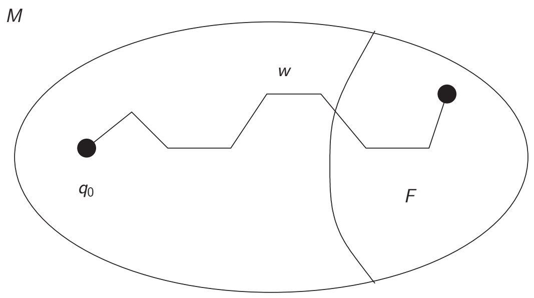
Véges automaták

Az $M = (Q, \Sigma, \delta, q_0, F)$ automata által felismert nyelven az

$$L(M) = \{w \in \Sigma^* \mid (q_0, w) \vdash_M^* (q, \varepsilon) \text{ és } q \in F\}$$

nyelvet értjük.

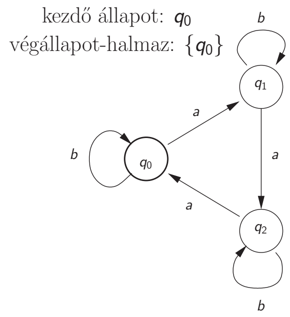
Szavakkal: q_0-ból a w hatására valamelyik $q \in F$ végállapotba jutunk.



$w \in L(M)$

Véges automaták

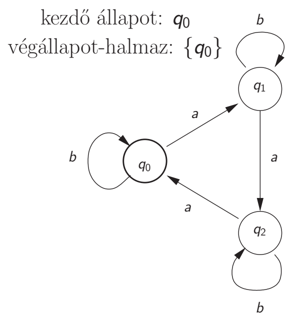
Átmenetek az M_3 automatában:



$(q_1, aabb) \vdash_{M_3} (q_2, abb) \vdash_{M_3} (q_0, bb) \vdash_{M_3}^2 (q_0, \varepsilon)$,
$(q_0, babaa) \vdash_{M_3} (q_0, abaa) \vdash_{M_3} (q_1, baa) \vdash_{M_3} (q_1, aa) \vdash_{M_3} (q_2, a) \vdash_{M_3} (q_0, \varepsilon)$ és
$(q_0, abb) \vdash_{M_3} (q_1, bb) \vdash_{M_3} (q_1, b) \vdash_{M_3} (q_1, \varepsilon)$

Véges automaták

Az M_3 automata által felismert $L(M_3)$ nyelv azon szavakból áll, amelyekben az a betűk száma osztható hárommal.



$babaa \in L(M_3)$, mert: $(q_0, babaa) \vdash_{M_3} (q_0, abaa) \vdash_{M_3}$
$(q_1, baa) \vdash_{M_3} (q_1, aa) \vdash_{M_3} (q_2, a) \vdash_{M_3} (q_0, \varepsilon)$

$abb \notin L(M_3)$, mert: $(q_0, abb) \vdash_{M_3} (q_1, bb) \vdash_{M_3} (q_1, b) \vdash_{M_3} (q_1, \varepsilon)$

Véges automaták

A determinisztikus automata általánosítása: nemdeterminisztikus automata.

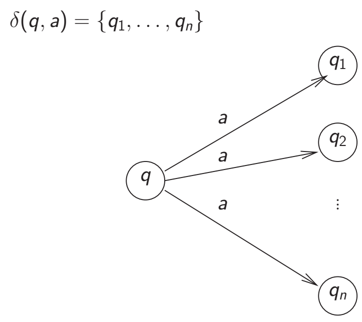
Az $M = (Q, \Sigma, \delta, q_0, F)$ rendszert *nemdeterminisztikus automatának* nevezzük, ahol:

1. Q egy nem üres, véges halmaz, az *állapotok halmaza*,
2. Σ egy ábécé, az *input ábécé*,
3. $q_0 \in Q$ a *kezdő állapot*,
4. $F \subseteq Q$ a *végállapotok halmaza*,
5. $\delta : Q \times \Sigma \to \mathcal{P}(Q)$ egy leképezés, az *átmenetfüggvény*.

Egy input szimbólum hatására egy állapotból több állapotba is átmehet. Az általánosítás valójában nem növeli meg a felismerő kapacitást.

Véges automaták

A nemdeterminisztikus automata egy input szimbólum hatására egy állapotból több állapotba is átmehet:

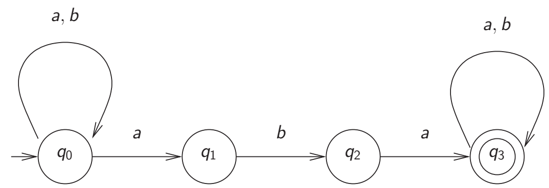


Az is megengedett, hogy $\delta(q, a) = \emptyset$.

Véges automaták

Példa: $M_{aba} = (Q, \Sigma, \delta, q_0, F)$ egy nemdeterminisztikus automata, ahol

- $Q = \{q_0, q_1, q_2, q_3\}$,
- $\Sigma = \{a, b\}$,
- $F = \{q_3\}$, továbbá
  - $\delta(q_0, a) = \{q_0, q_1\}$, $\delta(q_0, b) = \{q_0\}$,
  - $\delta(q_1, a) = \emptyset$, $\delta(q_1, b) = \{q_2\}$,
  - $\delta(q_2, a) = \{q_3\}$, $\delta(q_2, b) = \emptyset$,
  - $\delta(q_3, a) = \delta(q_3, b) = \{q_3\}$.



Véges automaták

Az M_{aba} automata megadása táblázattal:

δ	a	b
q_0	q_0, q_1	q_0
q_1		q_2
q_2	q_3	
$\star q_3$	q_3	q_3

Véges automaták

Az átmeneti reláció és a felismert nyelv nemdeterminisztikus automatákra:

Átmeneti reláció:

$(q, w), (q', w') \in C$ esetén $(q, w) \vdash_M (q', w')$ ha $w = aw'$, valamely $a \in \Sigma$-ra és $\underline{q' \in \delta(q, a)}$.

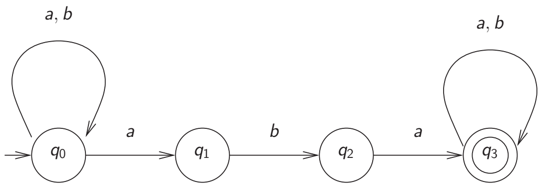
Az $M = (Q, \Sigma, \delta, q_0, F)$ automata által felismert nyelven az

$$L(M) = \{w \in \Sigma^* \mid (q_0, w) \vdash_M^* (q, \varepsilon) \text{ valamely } q \in F\text{-re}\}$$

nyelvet értjük.

Szavakkal: q_0-ból a w hatására elérhető valamely $q \in F$ végállapot (ugyanakkor esetleg nem végállapotok is elérhetők).

Véges automaták



$L(M_{aba}) = \{uabav \mid u, v \in \Sigma^*\}$, tehát M_{aba} pontosan azon Σ^*-beli szavakat ismeri fel amelyekben előfordul az aba rész-szó.

M_{aba} nemdeterminisztikus!

Véges automaták

Lehetnek olyan szavak, amelyeket egy nemdeterminisztikus automata nem tud végig olvasni, mert a $\delta(q, a) = \emptyset$ alakú átmenetek miatt "elakadhat". A teljesen definiált automaták viszont minden szót végig tudnak olvasni.

Az $M = (Q, \Sigma, \delta, q_0, F)$ nemdeterminisztikus automata *teljesen definiált (vagy: teljes)*, ha minden $q \in Q$ és $a \in \Sigma$ esetén $\delta(q, a)$ <u>legalább</u> egy elemű.

A determinisztikus automaták teljesen definiáltak. Továbbá, minden nemdeterminisztikus automata könnyen teljessé tehető egy ún. "csapda" állapot bevezetésével, anélkül, hogy a felismert nyelv megváltozna.

Véges automaták

Tétel. Tetszőleges $M = (Q, \Sigma, \delta, q_0, F)$ nemdeterminisztikus automatához megadható olyan $M' = (Q', \Sigma, \delta', q_0, F)$ teljesen definiált automata, melyre $L(M) = L(M')$.

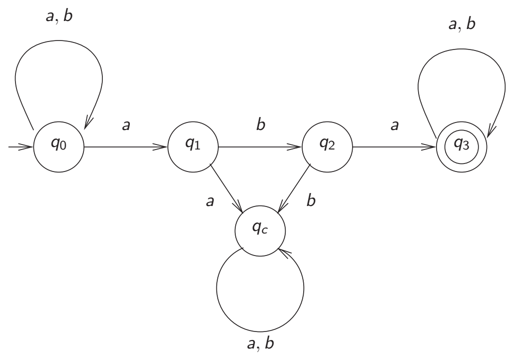
Bizonyítás. Ha M teljesen definiált, akkor legyen $M' = M$. Különben, legyen $Q' = Q \cup \{q_c\}$, ahol $q_c \notin Q$, vagyis egy új állapot (a "csapda" állapot).
Továbbá, minden $q \in Q$ és $a \in \Sigma$ esetén, legyen

$$\delta'(q,a) = \begin{cases} \delta(q,a) & \text{ha} \quad \delta(q,a) \neq \emptyset \\ \{q_c\} & \text{ha} \quad \delta(q,a) = \emptyset. \end{cases}$$

Végül, minden $a \in \Sigma$-ra, legyen $\delta'(q_c, a) = \{q_c\}$.

Véges automaták

Példa. Az M_{aba} automata teljessé tétele.



Véges automaták

Tétel. Egy nyelv akkor és csak akkor ismerhető fel nemdeterminisztikus automatával, ha felismerhető determinisztikus automatával.

Bizonyítás. a) Ha egy nyelv felismerhető determinisztikus automatával akkor felismerhető nemdeterminisztikus automatával is.

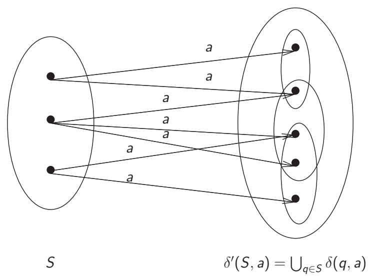
b) Fordítva: legyen $M = (Q, \Sigma, \delta, q_0, F)$ egy nemdeterminisztikus automata. Megadunk egy $M' = (Q', \Sigma, \delta', q_0', F')$ determinisztikus automatát, amelyre $L(M') = L(M)$.

A konstrukció neve: hatványhalmaz konstrukció.

Véges automaták

$M' = (Q', \Sigma, \delta', q_0', F')$, ahol

- $Q' = \mathcal{P}(Q)$ $(= \{S \mid S \subseteq Q\})$, a hatványhalmaz,
- $q_0' = \{q_0\}$,
- $F' = \{S \subseteq Q \mid S \cap F \neq \emptyset\}$,
- $\delta' : Q' \times \Sigma \to Q'$ az a leképezés amelyre tetszőleges $S \in Q'$ és $a \in \Sigma$ esetén $\delta'(S, a) = \bigcup_{q \in S} \delta(q, a)$.



Véges automaták

M' állapotai az M állapotaiból képzett halmazok. Nyilvánvaló, hogy M' determinisztikus. Az $L(M') = L(M)$ bizonyítása:

Állítás. Minden $w \in \Sigma^*$-ra és $S \subseteq Q$-ra

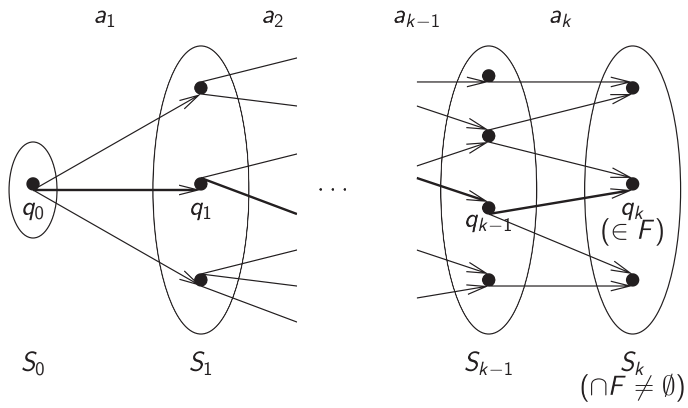
$(\{q_0\}, w) \vdash^*_{M'} (S, \varepsilon)$ akkor és csak akkor, ha
$$S = \{q \in Q \mid (q_0, w) \vdash^*_M (q, \varepsilon)\}.$$

Szavakkal: Az M-ben a w hatására elérhető állapotok halmaza megegyezik azzal az állapottal, melybe M' a w hatására jut. Az állítás könnyen igazolható $|w|$ szerinti indukcióval.

Az állításból azonnal következik, hogy $L(M') = L(M)$, mert mindkét automata akkor ismeri fel w-t, ha S-ben van legalább egy F-beli állapot.

Véges automaták

Az M és M' felismeri $w = a_1 \dots a_k$ szót.



Véges automaták

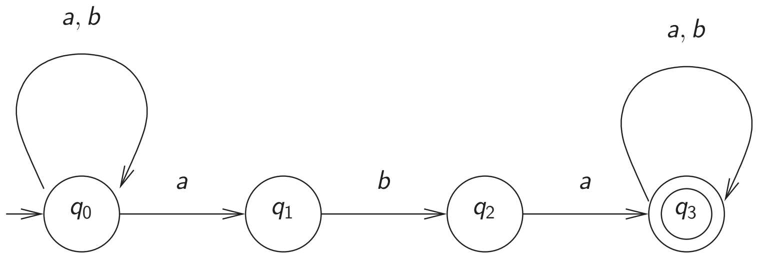
Egy fontos megjegyzés:

A hatványhalmaz konstrukcióval kapott determinisztikus automata állapotainak a száma exponenciálisan növekedhet az eredeti nemdeterminisztikus automata állapotaihoz képest (n-ről 2^n-re). Ez nagy állapotszámú rendszerek esetén állapottér robbanást eredményez.

Szerencsére a helyzet általában ennél sokkal jobb. A gyakorlatban úgy járunk el, hogy a determinisztikus automata $\{q_0\}$ kezdőállapotából kiindulva csak azokat az állapotokat konstruáljuk meg, amelyek ezen kezdőállapotból elérhetők. Ez a trükk általában a 2^n-nél jóval kevesebb állapotot eredményez. (Lásd a következő példát.)

Véges automaták

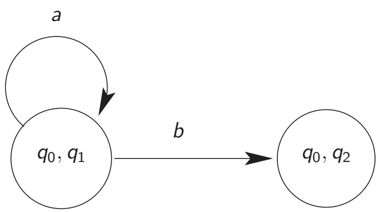
Példa. Determinizáljuk az M_{aba} nemdeterminisztikus automatát!



$L(M_{aba}) = \{uabav \mid u, v \in \Sigma^*\}$

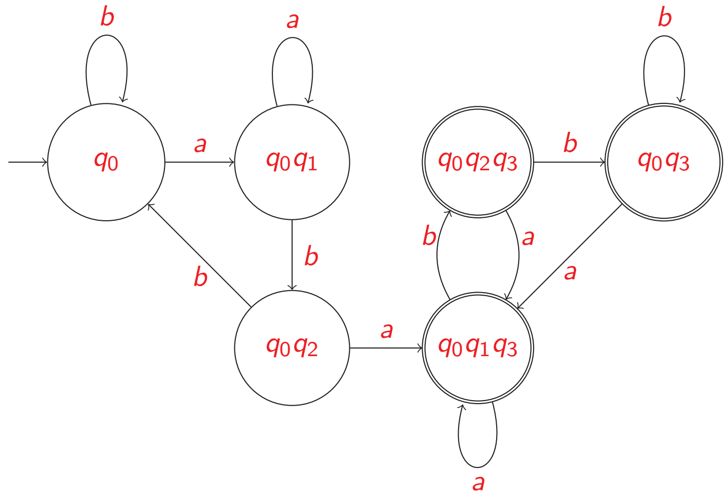
Véges automaták

Példa. Átmenetek a $\{q_0, q_1\}$ állapotból:



Véges automaták

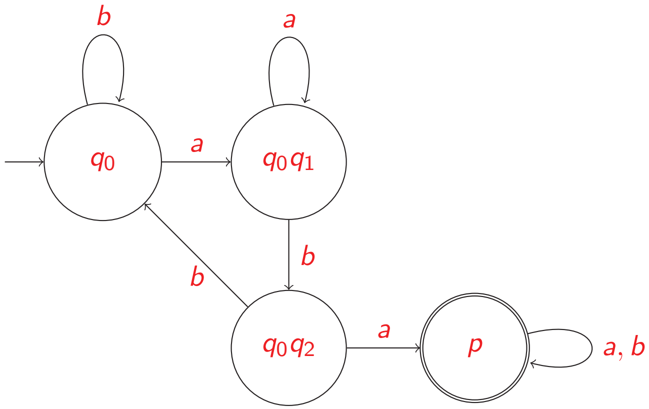
Példa. Az M_{aba} automata determinizálása:



A kapott automata determinisztikus és 6 állapota van (16 helyett).

Véges automaták

Példa. Észrevesszük, hogy a 3 végállapot összevonható egyetlen p végállapottá:



Az így kapott automatának már csak 4 állapota van.

Véges automaták

A nemdeterminisztkius automata általánosítása: nemdeterminisztikus automata ε-átmenettel, röviden nemdeterminisztikus ε-automata.

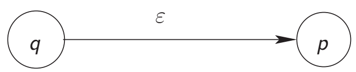
Az $M = (Q, \Sigma, \delta, q_0, F)$ rendszert *nemdeterminisztikus ε-automatának* nevezzük, ahol:

1. Q egy nem üres, véges halmaz, az *állapotok halmaza*,
2. Σ egy ábécé, az *input ábécé*,
3. $q_0 \in Q$ a *kezdő állapot*,
4. $F \subseteq Q$ a *végállapotok halmaza*,
5. $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \to \mathcal{P}(Q)$ egy leképezés, az *átmenetfüggvény*.

Olyan átmenet is lehetséges, amelyik "nem fogyasztja az inputot". Előnye, hogy néha kényelmes alkalmazni. Ugyanakkor, a nemdeterminizmushoz hasonlóan, nem növeli meg a felismerő kapacitást.

Véges automaták

Egy ε-átmenet



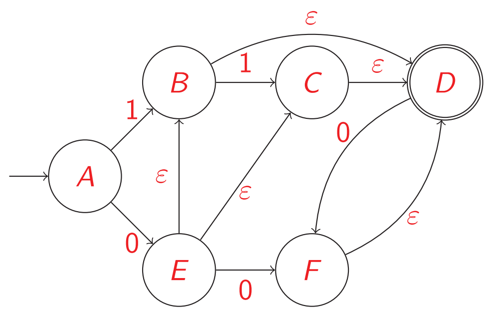
Átmeneti reláció:

$(q, w), (q', w') \in C$ esetén $(q, w) \vdash_M (q', w')$ ha $w = aw'$, valamely $\underline{a \in (\Sigma \cup \{\varepsilon\})}$-ra és $\underline{q' \in \delta(q, a)}$.

A felismert nyelv definíciója ugyanaz, mint a nemdeterminisztikus esetben: az M automata a w szót felismeri, ha q_0-ból a w hatására elérhető valamely $q \in F$ végállapot (esetleg a ε-átmenetek "segítségével").

Véges automaták

Egy példa:



Felismert nyelv: $\{1, 11, 0, 01, 00\}\{0\}^*$

δ	0	1	ε
$\rightarrow A$	E	B	
B		C	D
C			D
$\star D$	F		
E	F		B, C
F			D

Véges automaták

Tétel. Egy nyelv akkor és csak akkor ismerhető fel nemdeterminisztikus ε-automatával, ha felismerhető nemdeterminisztikus automatával.

Bizonyítás. a) Ha egy nyelv felismerhető nemdeterminisztikus automatával akkor felismerhető nemdeterminisztikus ε-automatával is.

b) Fordítva: legyen $M = (Q, \Sigma, \delta, q_0, F)$ egy nemdeterminisztikus ε-automata. Megadunk egy $M' = (Q, \Sigma, \delta', q_0, F')$ nemdeterminisztikus automatát, amelyre $L(M') = L(M)$.

Véges automaták

Az $M = (Q, \Sigma, \delta, q_0, F)$ nemdeterminisztikus ε-automatához megadunk egy $M' = (Q, \Sigma, \delta', q_0, F')$ nemdeterminisztikus automatát, amelyre $L(M') = L(M)$.

M' megadásához, ki kell számolni az állapotok ε-lezárását M-ben. Egy $q \in Q$ állapot ε-lezárása azon állapotokból áll, amelyek elérhetők q-ból ε-átmenetekkel:

$$Cl(q) = \{p \in Q \mid (q, \varepsilon) \vdash_M^* (p, \varepsilon)\}.$$

A $\{q\}$ halmazból kiindulva, hozzávesszük a q-ból <u>egy</u> ε-átmenettel elérhető állaptokat, és ezt az eljárást addig folytatjuk, amíg a halmaz bővíthető. Tehát $q \in Cl(q)$.

Véges automaták

Az $M = (Q, \Sigma, \delta, q_0, F)$ nemdeterminisztikus ε-automatához megadunk egy $M' = (Q, \Sigma, \delta', q_0, F')$ nemdeterminisztikus automatát, amelyre $L(M') = L(M)$.

A $Cl(q)$ lezárások ismeretében, legyen:

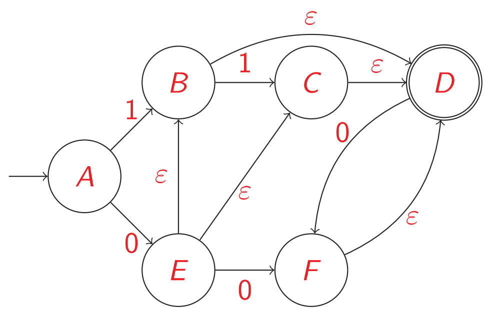
- $\delta'(q, a) = \bigcup_{p \in Cl(q)} \delta(p, a)$ és
- $F' = \{q \in Q \mid Cl(q) \cap F \neq \emptyset\}$.

Tehát M' a q állapotból az a hatására azon állapotokba megy át, amelyekbe M' valamennyi ε-átmenettel, majd egy a-átmenettel jut el. Továbbá M' végállapotai azon állapotok, amelyekből M valamennyi ε-átmenettel egy F-beli állapotba jut, vagyis felismer. (A "valamennyi" mindkét esetben lehet nulla is.)

Ezért $L(M') = L(M)$.

Véges automaták

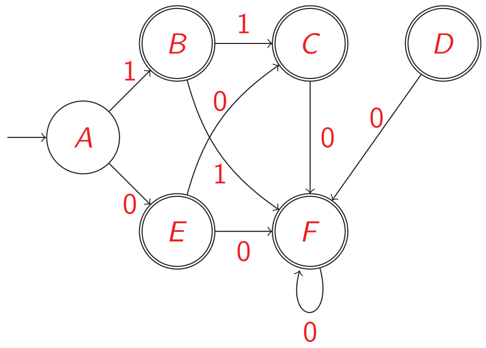
Egy példa:



- $Cl(A) = \{A\}$, $Cl(B) = \{B, D\}$, $Cl(C) = \{C, D\}$,
- $Cl(D) = \{D\}$, $Cl(E) = \{E, B, C, D\}$, $Cl(F) = \{F, D\}$
- $F' = \{E, B, C, F, D\}$

Véges automaták

Egy példa:



Az ekvivalens nemdeterminisztikus automata.
Felismert nyelv: $\{1, 11, 0, 01, 00\}\{0\}^*$.

Véges automaták

Összefoglalás:

Véges automatákkal nyelveket lehet definiálni, oly módon, hogy minden M automata felismer egy $L(M)$ nyelvet. A következő három fajta automatát ismertük meg:

- determinisztikus automata, nemdeterminisztikus automata, nemdeterminisztikus ε-automata.

A felismerő kapacitása mindhárom fajta automatának ugyanaz. Ugyanakkor, a nemdeterminisztikus automatákkal "könnyebb bánni", például egy adott nyelvhez általában könnyebb megadni az őt felismerő nemdeterminisztikus automatát, mint a determinisztikusat. Ez még inkább igaz a nemdeterminisztikus ε-automatákra. Például két atomatához nagyon könnyű megadni egy olyan nemdeterminisztikus ε-automatát, amely az eredeti automaták által felismert nyelvek egyesítését vagy konkatenációját ismeri fel.

Reguláris kifejezések

Program:

Nyelvek megadásának egy újabb formájával ismerkedünk meg.

Veszünk egy ábécét és hozzáveszünk néhány segédszimbólumot. Ezekből ún. reguláris kifejezéseket építünk fel bizonyos szabályok szerint. Minden reguláris kifejezés meghatároz (vagy: reprezentál) egy nyelvet. Az összes ilyen nyelvet vizsgáljuk.

Ki fog derülni, hogy a reguláris kifejezésekkel reprezentálható nyelvek nem mások, mint a reguláris nyelvek.

Reguláris kifejezések

Egy Σ ábécé feletti *reguláris kifejezések* halmaza a $(\Sigma \cup \{\emptyset, \varepsilon, (,), +, *\})^*$ halmaz legszűkebb olyan U részhalmaza, amelyre az alábbi feltételek teljesülnek:

(i) az $\emptyset$ (áthúzott nulla) szimbólum eleme U-nak;
(ii) a ε szimbólum eleme U-nak;
(iii) Minden $a \in \Sigma$-ra az a szimbólum eleme U-nak;
(iv) Ha $R_1, R_2 \in U$, akkor $(R_1) + (R_2)$, $(R_1)(R_2)$, és $(R_1)^*$ is elemei U-nak.

Példa: Legyen $\Sigma = \{a, b\}$. Akkor például a $(\emptyset)^*$, $((a) + (b))^*$ és $((a)(b))((a)^*)$, Σ feletti reguláris kifejezések.

Reguláris kifejezések

Az R reguláris kifejezés *által meghatározott* (*reprezentált*) $|R|$ *nyelvet* a következőképpen definiáljuk:

(i) Ha $R = \emptyset$ (áthúzott nulla), akkor $|R| = \emptyset$ (üres nyelv);
(ii) Ha $R = \varepsilon$ (mint szimbólum), akkor $|R| = \{\varepsilon\}$ (mint nyelv);
(iii) Ha $R = a$ (mint szimbólum), akkor $|R| = \{a\}$ (mint nyelv);
(iv) a) Ha $R = (R_1) + (R_2)$, akkor $|R| = |R_1| \cup |R_2|$;
(iv) b) Ha $R = (R_1)(R_2)$, akkor $|R| = |R_1||R_2|$;
(iv) c) Ha $R = (R_1)^*$, akkor $|R| = |R_1|^*$.

Egy $L \subseteq \Sigma^*$ nyelv reperzentálható reguláris kifejezéssel, ha van olyan Σ feletti R reguláris kifejezés, melyre $|R| = L$.

Reguláris kifejezések

A (gyakran zavaró) zárójelezés az egyértelmű kiolvashatóság miatt szükséges.

A zárójelek száma csökkenthető, ha megállapodunk abban, hogy a prioritási sorrend legyen *, konkatenáció, $+$. Továbbá, az $\cup$ és konkatenáció művelet asszociatív, azért a $+$ és az egymás után írás zárójelezése elhagyható. Végül (a) helyett a-t, $(\emptyset)$ helyett $\emptyset$-et írunk.

Így a reguláris kifejezések zárójelezése az alábbi módon egyszerűsödik:

$(\emptyset)^*$ helyett $\emptyset^*$
$((a) + (b))^*$ helyett $(a + b)^*$
$((a)(b))((a)^*)$ helyett aba^*
írhatunk $(a + b)^*aba(a + b)^*$-t, $a(a + b)^*b(a + b)^*a$-t, stb.

Reguláris kifejezések

Példák

- $|\emptyset^*| = |\emptyset|^* = \emptyset^* = \{\varepsilon\}$;
- $|(a+b)^*| = |a+b|^* = (|a| \cup |b|)^* = (\{a\} \cup \{b\})^* = \{a,b\}^*$;
- $|aba^*| = |ab||a^*| = |a||b||a^*| = |a||b||a|^* = \{a\}\{b\}\{a\}^* = \{ab\}\{\varepsilon, a, aa, \ldots\} = \{ab, aba, abaa, \ldots\}$.

Tehát a $\{\varepsilon\}$, $\{a,b\}^*$ és $\{ab, aba, abaa, \ldots\}$ nyelvek reprezentálhatók reguláris kifejezéssel.

Minden $\Sigma = \{a_1, \ldots, a_n\}$ ábécé esetén a Σ^* nyelv reprezentálható reguláris kifejezéssel, mert $\Sigma^* = |(a_1 + \ldots + a_n)^*|$.

Reguláris kifejezések

További példák: $\Sigma = \{a, b\}$

a) $L = \{uabav \mid u, v \in \Sigma^*\}$ ($w \in L \iff w$-ben előfordul az aba rész-szó) reprezentálható, mert

$$L = |(a+b)^* aba (a+b)^*|.$$

Gondoljunk az ugyanezen nyelvet felismerő determinisztikus automatára. Melyiket könnyebb megadni?

b) $L = \{aubva \mid u, v \in \Sigma^*\}$ ($w \in L \iff w$ a-val kezdődik és végződik és van benne legalább egy b) reprezentálható, mert

$$L = |a(a+b)^* b (a+b)^* a|.$$

Reguláris kifejezések

További példák: $\Sigma = \{a, b\}$

c) Az $(a+ba)^*(\varepsilon + b)$ reguláris kifejezére

$$|(a+ba)^*(\varepsilon+b)| =$$
$$\{w \in \{a,b\}^* \mid w\text{-ben nem fordul elő a } bb \text{ rész-szó}\}$$

Adjunk meg ehhez a nyelvhez egy determinisztikus automatát! Melyik az egyszerűbb?

Reguláris kifejezések

További példák reprezentálható nyelvekre:

d) Minden véges nyelv reprezentálható reguláris kifejezéssel. Valóban, legyen $L = \{x_1, \ldots, x_n\}$, $n \geq 1$. Akkor

$$L = |R_1 + \ldots + R_n|,$$

ahol

$$R_i = \begin{cases} a_{i_1} \ldots a_{i_{n_i}} & \text{ha } x_i = a_{i_1} \ldots a_{i_{n_i}} \\ \varepsilon & \text{ha } x_i = \varepsilon. \end{cases}$$

Például $|\varepsilon + a + abaa + abba| = \{\varepsilon, a, abaa, abba\}$. Adjunk meg ehhez a nyelvhez is egy determinisztikus automatát!

Reguláris kifejezések

e) Reguláris kifejezések a UNIX-ban: az ábécé az ASCII és különböző rövidítéseket enged meg.

Rövidítések karakter halmazokra:

• A . (pont) a "tetszőleges karakter" rövidítése.
• Elhagyja a + jeleket: az $a_1 + \ldots + a_n$ reguláris kifejezést $[a_1 \ldots a_n]$ formában rövidíti. Például a $< + > + =$ helyett [<>=]-t ír.
• Kihasználva, hogy az ASCII rendezett, halmazokat rövidítve definiál. Például [0-9] jelenti a $0 + \ldots + 9$ reguláris kifejezést. További példák: [A-Z] és [A-Za-z0-9].
• "Makrókat" használ. Például `[:digit:]` a [0-9] helyett és `[:alnum:]` a [A-Za-z0-9] helyett.

Reguláris kifejezések

e) Reguláris kifejezések a UNIX-ban: az ábécé az ASCII és különböző rövidítéseket és kiterjesztéseket enged meg.

Rövidítések műveletekre:

• A + helyett | jelet ír.
• A ? azt jelenti, hogy legfeljebb egy. Tehát az $R?$ UNIX kifejezés a $\varepsilon + R$ rövidítése.
• A + viszont azt jelenti, hogy legalább egy. Tehát az $R+$ UNIX kifejezés az RR^* rövidítése.
• Az $\{n\}$ rövidítés azt jelenti, hogy n példány. Tehát az $R\{5\}$ UNIX kifejezés az $RRRRR$ rövidítése.

Példák: .?, `[:digit:]+|[:alnum:]{3}`, stb.

Az ekvivalencia tétel

Tétel. Tetszőleges $L \subseteq \Sigma^*$ nyelv esetén a következő három állítás ekvivalens:

(1) L reguláris (generálható 3-típusú nyelvtannal).
(2) L felismerhető automatával.
(3) L reprezentálható reguláris kifejezéssel.

Bizonyítás. Megmutatjuk, hogy

1. Lemma: (3) $\Longrightarrow$ (1)
2. Lemma: (1) $\Longrightarrow$ (2)
3. Lemma: (2) $\Longrightarrow$ (3)

Akkor (1) $\Longleftrightarrow$ (2) $\Longleftrightarrow$ (3). $\diamond$

A reguláris nyelvek 3-típusúak

1. Lemma. (3) $\Longrightarrow$ (1): Ha $L \subseteq \Sigma^*$ nyelv reprezentálható reguláris kifejezéssel, akkor generálható 3-típusú nyelvtannal.
Bizonyítás. Az L-et reprezentáló R reguláris kifejezés struktúrája szerinti indukcióval.

Az indukció alapja.

(i) $R = \emptyset$ Ekkor $L = |R| = \emptyset$, mely generálható a $G = (\{S\}, \Sigma, \emptyset, S)$, 3-típusú nyelvtannal.

(ii) és (iii) $R = a$, ahol $a \in \Sigma$ vagy $a = \varepsilon$ Ekkor $L = |R| = \{a\}$, mely generálható a $G = (\{S\}, \Sigma, \{S \to a\}, S)$, 3-típusú nyelvtannal.

A reguláris nyelvek 3-típusúak

Indukciós lépés.

(iv) a) $R = (R_1) + (R_2)$

Ekkor $L = |R| = L_1 \cup L_2$, ahol $L_1 = |R_1|$ és $L_2 = |R_2|$.

Indukciós feltevés: L_i generálható a $G_i = (N_i, \Sigma, P_i, S_i)$, 3-típusú nyelvtannal, $i = 1, 2$. ($N_1 \cap N_2 = \emptyset$.)

A reguláris nyelvek 3-típusúak

Akkor L generálható a

$$G = (N_1 \cup N_2 \cup \{S\}, \Sigma, P_1 \cup P_2 \cup \{S \to S_1, S \to S_2\}, S),$$

3-típusú nyelvtannal, ahol S egy új szimbólum.

$$S \Rightarrow_G^* w$$

akkor és csak akkor, ha

$S_1 \Rightarrow_{G_1}^* w$ vagy $S_2 \Rightarrow_{G_2}^* w$.

A reguláris nyelvek 3-típusúak

Indukciós lépés.

(iv) b) $R = (R_1)(R_2)$

Ekkor $L = |R| = L_1 L_2$, ahol $L_1 = |R_1|$ és $L_2 = |R_2|$.

Indukciós feltevés: L_i generálható a $G_i = (N_i, \Sigma, P_i, S_i)$, 3-típusú nyelvtannal, $i = 1, 2$. ($N_1 \cap N_2 = \emptyset$.)

Akkor L generálható a $G = (N_1 \cup N_2, \Sigma, P, S_1)$, 3-típusú nyelvtannal, ahol P a legszűkebb olyan szabályhalmaz amire teljesülnek a következő feltételek:

A reguláris nyelvek 3-típusúak

- Ha $A \to xB \in P_1$, akkor $A \to xB \in P$,
- Ha $A \to x \in P_1$, akkor $A \to xS_2 \in P$,
- P_2 minden eleme P-nek is eleme.

$S_1 \Rightarrow_{G_1}^* w_1$ és $S_2 \Rightarrow_{G_2}^* w_2$

akkor és csak akkor, ha

$S_1 \Rightarrow_G^* w_1 S_2 \Rightarrow_G^* w_1 w_2$.

A reguláris nyelvek 3-típusúak

Indukciós lépés.

(iv) c) $R = (R_1)^*$

Ekkor $L = |R| = L_1^*$, ahol $L_1 = |R_1|$.

Indukciós feltevés: L_1 generálható a $G_1 = (N_1, \Sigma, P_1, S_1)$, 3-típusú nyelvtannal.

Akkor L generálható a $G = (N_1 \cup \{S\}, \Sigma, P, S)$, 3-típusú nyelvtannal, ahol S egy új szimbólum, P pedig a legszűkebb olyan szabályhalmaz amire teljesülnek a következő feltételek:

A reguláris nyelvek 3-típusúak

- $S \to S_1, S \to \varepsilon \in P$,
- Ha $A \to xB \in P_1$, akkor $A \to xB \in P$,
- Ha $A \to x \in P_1$, akkor $A \to xS \in P$.

$S \Rightarrow_G \varepsilon$

$S \Rightarrow_G S_1 \Rightarrow_G^* w_1 S \Rightarrow_G w_1 \quad (\in L_1)$

$w_1 S \Rightarrow_G w_1 S_1 \Rightarrow_G^* w_1 w_2 S \Rightarrow_G w_1 w_2 \quad (\in L_1 L_1)$

$w_1 w_2 S \Rightarrow_G w_1 w_2 S_1$ ...

Az ekvivalencia tétel

Tétel. Tetszőleges $L \subseteq \Sigma^*$ nyelv esetén a következő három állítás ekvivalens:

(1) L reguláris (generálható 3-típusú nyelvtannal).

(2) L felismerhető automatával.

(3) L reprezentálható reguláris kifejezéssel.

Bizonyítás.

1. Lemma: (3) $\Longrightarrow$ (1) $\surd$
2. Lemma: (1) $\Longrightarrow$ (2)
3. Lemma: (2) $\Longrightarrow$ (3)

A 3-típusú nyelvek felismerhetők automatával

2. Lemma. (1) $\Longrightarrow$ (2): Ha $L \subseteq \Sigma^*$ nyelv reguláris, akkor felismerhető automatával.
Bizonyítás. Legyen L egy reguláris nyelv és tegyük fel, hogy $L = L(G)$, ahol G egy 3-típusú nyelvtan.

2.1. Lemma. Minden $G = (N, \Sigma, P, S)$, 3-típusú nyelvtanhoz megadható vele ekvivalens $G' = (N', \Sigma, P', S)$, 3-típusú nyelvtan, úgy hogy P'-ben minden szabály $A \to B$, $A \to aB$ vagy $A \to \varepsilon$ alakú, ahol $A, B \in N$ és $a \in \Sigma$.

2.2. Lemma. Minden olyan $G = (N, \Sigma, P, S)$, 3-típusú nyelvtanhoz melynek csak $A \to B$, $A \to aB$ vagy $A \to \varepsilon$ alakú szabályai vannak megadható olyan $M = (Q, \Sigma, \delta, q_0, F)$ nemdeterminisztkius ε-automata, amelyre $L(M) = L(G)$.

A 3-típusú nyelvek felismerhetők automatával

2.1. Lemma. Minden $G = (N, \Sigma, P, S)$, 3-típusú nyelvtanhoz megadható vele ekvivalens $G' = (N', \Sigma, P', S)$, 3-típusú nyelvtan, úgy hogy P'-ben minden szabály $A \to B$, $A \to aB$ vagy $A \to \varepsilon$ alakú, ahol $A, B \in N$ és $a \in \Sigma$.

Bizonyítás. Konstruáljuk meg P'-t a következőképpen:

(i) Minden $A \to B$, $A \to aB$ és $A \to \varepsilon$ alakú P-beli szabályt vegyünk fel P'-be.

A 3-típusú nyelvek felismerhetők automatával

(ii) Minden $A \to a_1 \dots a_n B$, P-beli szabály esetén (ahol $n > 1$, $a_1, \dots, a_n \in \Sigma$) vegyük fel P'-be az

$$A \to a_1 A_1,\ A_1 \to a_2 A_2, \dots, A_{n-1} \to a_n B$$

szabályokat, ahol $A_1, \dots, A_{n-1}$ új nemterminális szimbólumok.

(iii) Minden $A \to a_1 \dots a_n$, P-beli szabály esetén (ahol $n \geq 1, a_1, \dots, a_n \in \Sigma$) vegyük fel P'-be az

$$A \to a_1 A_1,\ A_1 \to a_2 A_2, \dots, A_{n-1} \to a_n A_n, A_n \to \varepsilon$$

szabályokat, ahol $A_1, \dots, A_n$ új nemterminálisok.
Legyen $N' = N \cup \{$ új nemterminálisok $\}$.

A 3-típusú nyelvek felismerhetők automatával

Minden $A \in N$-re és $w \in \Sigma^*$-ra

$$A \Rightarrow_G^* w \text{ akkor és csak akkor, ha } A \Rightarrow_{G'}^* w.$$

Ugyanis

$A \to a_1 \dots a_n B \in P$ akkor és csak akkor, ha
$A \Rightarrow_{G'} a_1 A_1 \Rightarrow_{G'} \dots \Rightarrow_{G'} a_1 \dots a_{n-1} A_{n-1} \Rightarrow_{G'} a_1 \dots a_n B$

és

$A \to a_1 \dots a_n \in P$ akkor és csak akkor, ha
$A \Rightarrow_{G'} a_1 A_1 \Rightarrow_{G'} \dots \Rightarrow_{G'} a_1 \dots a_n A_n \Rightarrow_{G'} a_1 \dots a_n$.

Az $A = S$ választással kapjuk, hogy $L(G) = L(G')$.

A 3-típusú nyelvek felismerhetők automatával

Példa a szabályok szétdarabolására

$$\begin{array}{llllllllll}
G: & S & \to & abA & \mid & bB \\
 & A & \to & bbB & \mid & \varepsilon \\
 & B & \to & ab & \mid & bA \\
G': & S & \to & aA_1, & A_1 & \to & bA, & S & \to & bB \\
 & A & \to & bA_2, & A_2 & \to & bB, & A & \to & \varepsilon \\
 & B & \to & aA_3, & A_3 & \to & bA_4, & A_4 & \to & \varepsilon \\
 & B & \to & bA
\end{array}$$

A 3-típusú nyelvek felismerhetők automatával

2.2. Lemma. Minden olyan $G = (N, \Sigma, P, S)$, 3-típusú nyelvtanhoz melynek csak $A \to B$, $A \to aB$ vagy $A \to \varepsilon$ alakú szabályai vannak megadható olyan $M = (Q, \Sigma, \delta, q_0, F)$ nemdeterminisztkius ε-automata, amelyre $L(M) = L(G)$.

Bizonyítás. Konstruáljuk meg M-et a következőképpen:

- $Q = N$,
- $q_0 = S$,
- $F = \{B \in N \mid B \to \varepsilon \in P\}$,
- minden $A \in N$ és $a \in (\Sigma \cup \{\varepsilon\})$ esetén legyen

$$\delta(A, a) = \{B \in N \mid A \to aB \in P\}.$$

A 3-típusú nyelvek felismerhetők automatával

<u>G-ben</u>:		<u>M-ben</u>:
$A \to aB \in P$		$A \xrightarrow{a} B$
	$\Leftrightarrow$	
$B \to \lambda \in P$		$B \in F$

Az ábrán $a \in \Sigma$ vagy $a = \varepsilon$.

A 3-típusú nyelvek felismerhetők automatával

Ekkor minden $n \geq 1$, $A, B \in N$ és $w \in \Sigma^*$ esetén

$$A \Rightarrow_G^n wB \text{ akkor és csak akkor ha } (A, w) \vdash_M^n (B, \varepsilon).$$

Részletesebben:

$$A \Rightarrow_G a_1 A_1 \Rightarrow_G \ldots \Rightarrow_G a_1 \ldots a_{n-1} A_{n-1} \Rightarrow_G a_1 \ldots a_n B$$

akkor és csak akkor, ha

$$(A, a_1 \ldots a_n) \vdash_M (A_1, a_2 \ldots a_n) \vdash_M \ldots \vdash_M (B, \varepsilon).$$

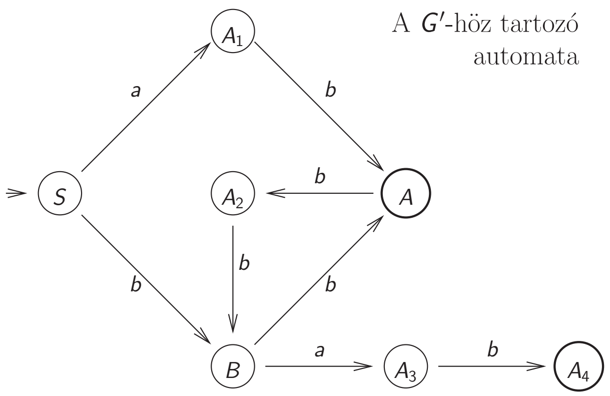
Az $A = S$, $B \in F$ választással adódik, hogy $L(M) = L(G)$.

A 3-típusú nyelvek felismerhetők automatával

Például, vegyük az előbbi G' nyelvtant:

$$\begin{array}{llllllllll}
G': & S & \to & aA_1, & A_1 & \to & bA, & S & \to & bB \\
 & A & \to & bA_2, & A_2 & \to & bB, & A & \to & \varepsilon \\
 & B & \to & aA_3, & A_3 & \to & bA_4, & A_4 & \to & \varepsilon \\
 & B & \to & bA & & & & & &
\end{array}$$

A 3-típusú nyelvek felismerhetők automatával



A 3-típusú nyelvek felismerhetők automatával

Egy megjegyzés a 2.2 Lemmához:

A lemma bizonyításában alkalmazott konstrukció könnyen "megfordítható", vagyis tetszőleges M nemdeterminisztikus ε-automatához meg tudunk adni egy olyan G reguláris nyelvtant, amelyre $L(G) = L(M)$.
Valóban, a nyelvtan nemterminálisai az automata állapotai lesznek.
Továbbá, ha az automatában valamely q állapotból egy $a \in (\Sigma \cup \{\varepsilon\})$ szimbólum hatására egy p állapotba jutunk (azaz $p \in \delta(q, a)$), akkor a nyelvtanba felvesszük a $q \to ap$ szabályt.
Végül minden q végállapot esetén a nyelvtanba felvesszük a $q \to \varepsilon$ szabályt.

Ezzel a bizonyítottuk, hogy (2) $\Longrightarrow$ (1). A tétel bizonyításához viszont még a (2) $\Longrightarrow$ (3) lépés hiányzik.

Az ekvivalencia tétel

Tétel. Tetszőleges $L \subseteq \Sigma^*$ nyelv esetén a következő három állítás ekvivalens:

(1) L reguláris (generálható 3-típusú nyelvtannal),
(2) L felismerhető automatával.
(3) L reprezentálható reguláris kifejezéssel.

Bizonyítás.

1. Lemma: (3) $\Longrightarrow$ (1) $\surd$
2. Lemma: (1) $\Longrightarrow$ (2) $\surd$
3. Lemma: (2) $\Longrightarrow$ (3)

Az automatával felismerhető nyelvek regulárisak

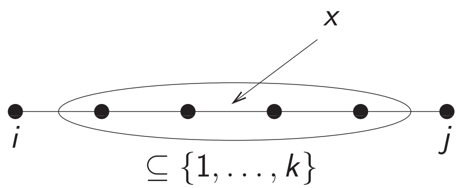
Lemma 3. (2) $\Longrightarrow$ (3): Minden, automatával felismerhető nyelv reprezentálható reguláris kifejezéssel. (S. C. Kleene tétele, 1956.)

Bizonyítás. Legyen $L = L(M)$, ahol $M = (Q, \Sigma, \delta, q_0, F)$ determinisztikus automata. Megadunk egy olyan reguláris kifejezést, amely L-et reprezentálja.
Tételezzük fel, hogy $Q = \{1, \ldots, n\}$ és $q_0 = 1$.
Minden $0 \leq k \leq n$ és $1 \leq i, j \leq n$ esetén definiáljuk a $L_{i,j}^{(k)} \subseteq \Sigma^*$ nyelvet a következőképpen:

$$x \in L_{i,j}^{(k)} \iff (i, x) \vdash_M^* (j, \varepsilon) \text{ és minden } (i,x) \vdash_M^+ (i', x') \vdash_M^+ (j, \varepsilon) \text{ esetén } i' \in \{1, \ldots, k\}.$$

Az automatával felismerhető nyelvek regulárisak

Az $L_{i,j}^{(k)}$ nyelvet alkotó x szavak:



Az automatával felismerhető nyelvek regulárisak

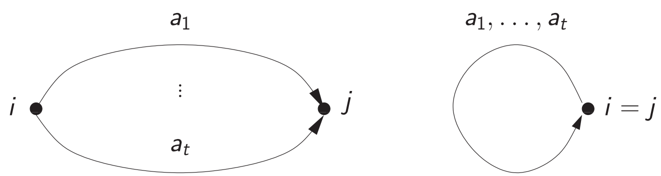
Mivel
$$L(M) = \bigcup_{j \in F} L_{1,j}^{(n)},$$
elegendő megadni minden $j \in F$-re egy $L_{1,j}^{(n)}$-t reprezentáló reguláris kifejezést.

Ha ugyanis $L_{1,j}^{(n)} = |R_{1,j}^{(n)}|$, akkor
$$L(M) = |R_{1,j_1} + \ldots + R_{1,j_l}|,$$
ahol $F = \{j_1, \ldots, j_l\}$.

Többet bizonyítunk: minden $0 \le k \le n$-ra és $1 \le i, j \le n$-re megadunk egy $R_{i,j}^{(k)}$ reguláris kifejezést, melyre
$$|R_{i,j}^{(k)}| = L_{i,j}^{(k)}.$$

Az automatával felismerhető nyelvek regulárisak

$R_{i,j}^{(k)}$ megadása k szerinti indukcióval. $k = 0$:
$$L_{i,j}^{(0)} = \begin{cases} \{a \in \Sigma \mid \delta(i,a) = j\}, & \text{ha } i \neq j \\ \{a \in \Sigma \mid \delta(i,a) = j\} \cup \{\varepsilon\}, & \text{ha } i = j. \end{cases}$$



$R_{i,j}^{(0)} = a_1 + \ldots + a_t$ vagy $R_{i,j}^{(0)} = a_1 + \ldots + a_t + \varepsilon$

Mindkét esetben $|R_{i,j}^{(0)}| = L_{i,j}^{(0)}$.

Az automatával felismerhető nyelvek regulárisak

$k \Rightarrow k+1$:

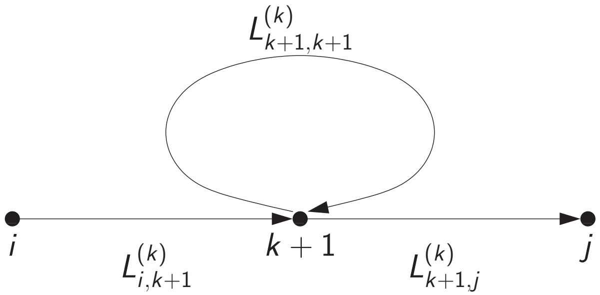
tfh minden i,j-re megadtuk $R_{i,j}^{(k)}$-t, melyre $|R_{i,j}^{(k)}| = L_{i,j}^{(k)}$

$\Longrightarrow$ megadjuk minden i,j-re $R_{i,j}^{(k+1)}$-t

Először is észrevesszük (!), hogy
$$L_{i,j}^{(k+1)} = L_{i,j}^{(k)} \cup L_{i,k+1}^{(k)} (L_{k+1,k+1}^{(k)})^* L_{k+1,j}^{(k)}.$$

Az automatával felismerhető nyelvek regulárisak

$L_{i,j}^{(k+1)} = L_{i,j}^{(k)} \cup L_{i,k+1}^{(k)}(L_{k+1,k+1}^{(k)})^* L_{k+1,j}^{(k)}$



Az automatával felismerhető nyelvek regulárisak

Legyen

$$R_{i,j}^{(k+1)} = R_{i,j}^{(k)} + R_{i,k+1}^{(k)}(R_{k+1,k+1}^{(k)})^* R_{k+1,j}^{(k)}.$$

Ekkor

$$\begin{aligned} |R_{i,j}^{(k+1)}| &= |R_{i,j}^{(k)}| \cup |R_{i,k+1}^{(k)}||R_{k+1,k+1}^{(k)}|^*|R_{k+1,j}^{(k)}| \\ &= L_{i,j}^{(k)} \cup L_{i,k+1}^{(k)}(L_{k+1,k+1}^{(k)})^* L_{k+1,j}^{(k)} = L_{i,j}^{(k+1)} \end{aligned}$$

Mint láttuk, $L(M) = |R_{1,j_1} + \ldots + R_{1,j_l}|$.

Az ekvivalencia tétel

Tétel. Tetszőleges $L \subseteq \Sigma^*$ nyelv esetén a következő három állítás ekvivalens:

(1) L reguláris (generálható 3-típusú nyelvtannal),

(2) L felismerhető automatával.

(3) L reprezentálható reguláris kifejezéssel.

Bizonyítás.

1. Lemma: (3) $\implies$ (1) $\surd$
2. Lemma: (1) $\implies$ (2) $\surd$
3. Lemma: (2) $\implies$ (3) $\surd$

Tehát (1) $\iff$ (2) $\iff$ (3). $\diamond$

Az ekvivalencia tétel

Összefoglalás:

A következő eszközök mindegyikével a reguláris nyelveket reprezentálhatjuk:

- reguláris (3-típusú) nyelvtanok,
- reguláris nyelvtanok, amelyek minden szabálya $A \to B$, $A \to aB$ vagy $A \to \varepsilon$ alakú,
- determinisztikus automaták,
- nemdeterminisztikus automaták,
- nemdeterminisztikus ε-automaták,
- reguláris kifejezések.

A továbbiakban, ha veszünk egy reguláris nyelvet valamilyen probléma megoldására, akkor mindig azt a reprezentációt választjuk, ami a legcélszerűbb az adott probléma szempontjából.

A Chomsky nyelvosztályok gépi reprezentációi

Gépi reprezentáció:

Nyelvosztály	Gépi repr.	Det. vs nemdet.
$\mathcal{L}_3$ (reguláris nyelvek)	véges automaták	det. = nemdet.
$\mathcal{L}_2$ (k. független nyelvek)	?	?
$\mathcal{L}_1$ (k. függő nyelvek)	?	?
$\mathcal{L}_0$ (általános nyelvek)	?	?

Pumpáló lemma reguláris nyelvekre

Lemma. (Pumpáló lemma reguláris nyelvekre.)
Minden $L \subseteq \Sigma^*$ reguláris nyelv esetén

- megadható olyan (L-től függő) $k > 0$ egész szám, hogy
- minden $w \in L$-re,
- ha $|w| \geq k$, akkor van olyan $w = w_1 w_2 w_3$ felbontás, melyre

1) $0 < |w_2|$ és $|w_1 w_2| \leq k$,
2) minden $n \geq 0$-ra, $w_1 w_2^n w_3 \in L$.

(Szükséges feltétele annak, hogy egy nyelv reguláris legyen. Ha egy nyelv nem teljesíti ezt a feltételt, akkor az nem reguláris.)

Pumpáló lemma reguláris nyelvekre

Bizonyítás. Mivel L reguláris, van olyan $M = (Q, \Sigma, \delta, q_0, F)$ determinisztikus automata, melyre $L = L(M)$.

Legyen $k = ||Q||$ (az állapotok száma). Ez a k jó lesz.

Vegyünk egy $w \in L$ szót úgy, hogy $|w| \geq k$. Akkor $w = a_1 a_2 \ldots a_K$, ahol $K \geq k$ és vannak olyan $q_1, q_2, \ldots, q_K \in Q$ állapotok, melyekre

$$\begin{array}{rcl} (q_0, a_1 \ldots a_K) & \vdash & (q_1, a_2 \ldots a_K) \\ & \vdash & (q_2, a_3 \ldots a_K) \\ & \ldots & \\ & \vdash & (q_{K-1}, a_K) \\ & \vdash & (q_K, \varepsilon), \end{array}$$

továbbá $q_K \in F$.

Pumpáló lemma reguláris nyelvekre

Mivel $K \geq ||Q||$, a $q_0, q_1, \ldots, q_K$ sorozatban legalább egy állapot kétszer is szerepel (skatulya elv = pigeon hole principle). Legyen q_j a legelső olyan állapot, amelyik a sorozatban már korábban is előfordul: van olyan $i < j$, hogy $q_i = q_j$.

Legyenek $w_1 = a_1 \ldots a_i$, $w_2 = a_{i+1} \ldots a_j$ és $w_3 = a_{j+1} \ldots a_K$.
(Ha $i = 0$ akkor $w_1 = \varepsilon$, ha $j = K$ akkor $w_3 = \varepsilon$)

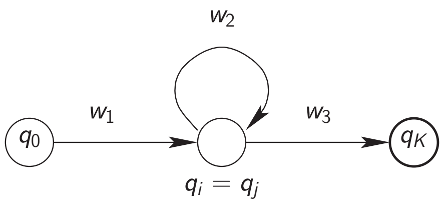
Nyilvánvaló, hogy $w = w_1 w_2 w_3$. Továbbá,

1) $0 < |w_2|$, mert $i < j$ és $|w_1 w_2| \leq k$, mert q_j a legelső olyan állapot, amelyik a sorozatban már korábban is előfordul.

és ...

Pumpáló lemma reguláris nyelvekre

2) Minden $n \geq 0$-ra $w_1 w_2^n w_3 \in L$:



Pumpáló lemma reguláris nyelvekre

A Pumpáló lemma egy következménye:

Lemma. Az $L = \{a^n b^n \mid n \geq 0\}$ nyelv nem reguláris.
Bizonyítás. Tegyük fel, hogy igen (indirekt bizonyítás).
Legyen k a pumpáló lemma szerint az L-hez tartozó szám, és vegyük az $a^k b^k \in L$ szót, melynek hossza $2k \geq k$.
A pumpáló lemma miatt létezik $a^k b^k = w_1 w_2 w_3$ felbontás, melyre $0 < |w_2|$, $|w_1 w_2| \leq k$ és minden $n \geq 0$-ra $w_1 w_2^n w_3 \in L$.

Mivel $|w_1 w_2| \leq k$, a középső $|w_2|$ szó csak a betűkből áll. Továbbá a $0 < |w_2|$ feltétel miatt a $w_1 w_2^2 w_3$, $w_1 w_2^3 w_3$, stb szavakban az a-k száma nagyobb mint a b-k száma, tehát ezen szavak egyike sincs L-ben. Ellentmondás, tehát L nem reguláris.

Pumpáló lemma reguláris nyelvekre

Intuíció:

Egy automata nem képes számolni, hogy két betű ugyanannyiszor szerepel-e.

De kezeljük ezt óvatosan! Az alábbi nyelv reguláris:

$\{w \in \{a, b\}^* \mid$
w-ben az ab és a ba alakú rész-szavak száma megegyezik$\}$.

Bizonyítás HF.

Pumpáló lemma reguláris nyelvekre

A Pumpáló lemma egy további következménye:
van olyan környezetfüggetlen nyelv, amelyik nem reguláris.

Tétel. $\mathcal{L}_3 \subset \mathcal{L}_2$.

Bizonyítás.

a) $\mathcal{L}_3 \subseteq \mathcal{L}_2$, mivel minden reguláris nyelvtan környezetfüggetlen.

b) Az $\{a^n b^n \mid n \geq 0\}$ nyelv $\mathcal{L}_2$-beli, mivel generálható az

$$S \to aSb \mid \varepsilon$$

szabályokból álló környezetfüggetlen nyelvtannal. Ugyanakkor, mint láttuk $L \notin \mathcal{L}_3$.

Reguláris nyelvek zártsági tulajdonságai

Műveletekre való zártság általában:

Legyen $\mathcal{C}$ nyelvek egy osztálya (pl. reguláris nyelvek, környezefüggetlen nyelvek) és

$$\bullet : \mathcal{P}(\Sigma^*) \times \mathcal{P}(\Sigma^*) \to \mathcal{P}(\Sigma^*), \qquad (L_1, L_2) \mapsto L_1 \bullet L_2$$

egy kétváltozós művelet nyelvekkel. (Például: egyesítés, konkatenáció, stb.)

Azt mondjuk, hogy $\mathcal{C}$ zárt a $\bullet$ műveletre, ha tetszőleges $L_1, L_2 \in \mathcal{C}$ esetén, $L_1 \bullet L_2 \in \mathcal{C}$.

Az egyáltozós műveletekre (pl. komplementer, iteráció) való zártság hasonlóan definiálható.

Reguláris nyelvek zártsági tulajdonságai

Reguláris műveletek: $\cup$, konkatenáció, *.

Tétel. A reguláris nyelvek osztálya zárt a reguláris műveletekre.

Bizonyítás. A reguláris nyelveket reguláris kifejezésekkel reprezentáljuk. Először az $\cup$-ra való zártságot bizonyítjuk.

Legyenek L_1, L_2 reguláris nyelvek. Megadunk egy olyan R reguláris kifejezést, melyre $|R| = L_1 \cup L_2$.

Mivel L_1, L_2 reguláris, vannak olyan R_1, R_2 reguláris kifejezések, melyekre $L_1 = |R_1|$ és $L_2 = |R_2|$. Legyen $R = (R_1) + (R_2)$.
Ekkor $|R| = |R_1 + R_2| = |R_1| \cup |R_2| = L_1 \cup L_2$, tehát $L_1 \cup L_2$ is reguláris.
A konkatenációra és a *-ra való zártság hasonlóan bizonyítható.

Reguláris nyelvek zártsági tulajdonságai

Boole műveletek: $\cup$, $\cap$, komplementer.

Tétel. A reguláris nyelvek osztálya zárt a Boole műveletekre.

Bizonyítás. A reguláris nyelveket most determinisztikus automatákkal reprezentáljuk.

A bizonyításban az *automaták direkt szorzata* konstrukciót alkalmazzuk.

Reguláris nyelvek zártsági tulajdonságai

Automaták direkt szorzata:

Az $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ és $M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ determinisztikus automaták egy direkt szorzata az $M = (Q_1 \times Q_2, \Sigma, \delta, [q_1, q_2], F)$ automata, ahol minden $p_1 \in Q_1$ és $p_2 \in Q_2$ állapot és $a \in \Sigma$ input szimbólum esetén

$$\delta([p_1, p_2], a) = [\delta_1(p_1, a), \delta_2(p_2, a)],$$

továbbá $F \subseteq Q_1 \times Q_2$ (később adjuk meg pontosan).

Az M állapotai tehát $[p_1, p_2]$ alakú párok. Az M_1 és M_2 automatákat párhuzamosan működtetjük: ha M_1 p_1-ből az a hatására r_1-be, M_2 p_2-ből az a hatására r_2-be megy át, akkor M $[p_1, p_2]$-ből az a hatására $[r_1, r_2]$-be megy át.

Reguláris nyelvek zártsági tulajdonságai

Ekkor minden $p_1, r_1 \in Q_1$, $p_2, r_2 \in Q_2$ és $x \in \Sigma^*$ esetében

$$([p_1, p_2], x) \vdash_M^* ([r_1, r_2], \varepsilon)$$
$$\Updownarrow$$
$$(p_1, x) \vdash_{M_1}^* (r_1, \varepsilon) \text{ és } (p_2, x) \vdash_{M_2}^* (r_2, \varepsilon).$$

Vagyis M a szavakon is M_1 és M_2 párhuzamos működését szimulálja.

Reguláris nyelvek zártsági tulajdonságai

A Boole műveletekre való zártság bizonyítása: legyenek $L_1 = L(M_1)$ és $L_2 = L(M_2)$ reguláris nyelvek, ahol M_1 és M_2 a fenti determinisztikus automaták. Konstruáljuk meg az M direkt szorzat automatát.

a) Ha $F = F_1 \times F_2$, akkor $L(M) = L(M_1) \cap L(M_2)$. Tehát $L_1 \cap L_2$ is reguláris.

b) Ha $F = F_1 \times (Q_2 - F_2)$, akkor $L(M) = L(M_1) - L(M_2)$. Tehát $L_1 - L_2$ is reguláris.
Következmény: Mivel a Σ^* nyelv reguláris (nyilvánvaló), az $\overline{L_1} = \Sigma^* - L_1$ is reguláris.

c) Ha $F = (F_1 \times Q_2) \cup (Q_1 \times F_2)$, akkor $L(M) = L(M_1) \cup L(M_2)$. Tehát $L_1 \cup L_2$ reguláris. (Újabb bizonyítás.)

Reguláris nyelvek zártsági tulajdonságai

Egy megjegyzés:

A komplementerre való zártság egyszerűbben is bizonyítható. Valóban, ha az L nyelv reguláris, akkor felismerhető egy $M = (Q, \Sigma, \delta, q_0, F)$ determinisztikus automatával. De akkor az $M' = (Q, \Sigma, \delta, q_0, Q - F)$ automata az $\overline{L}$ nyelvet ismeri fel, mert pontosan azok a szavak viszik végállapotba, amelyek M-et $Q - F$-be, vagyis nem végállapotba viszik. Ezért az $\overline{L}$ nyelv is reguláris.

Reguláris nyelvek zártsági tulajdonságai

Összefoglalás:

Művelet	**Zártság**
$\cup$	igen
konkatenáció	igen
$*$	igen
$\cap$	igen
komplementer	igen

Eldöntési kérdések reguláris nyelvekre

Az eldöntési kérdésekről általában.

Egy eldöntési kérdés mindig egy nyelvosztályra (pl reguláris nyelvek, környezetfüggetlen nyelvek) és egy nyelvekre vonatkozó tulajdonságra (pl. üresség, végesség) értendő és úgy szól, hogy létezik-e olyan algoritmus, amelynek inputja a nyelvosztály tetszőleges eleme, outputja pedig "Igen" ha a nyelv rendelkezik az adott tulandonsággal, különben pedig "Nem".

Például, létezik-e olyan algoritmus, amelynek inputja egy tetszőleges reguláris nyelv (vagyis e nyelvet reprezentáló determinisztikus automata), outputja pedig "Igen" ha a nyelv végtelen, különben pedig "Nem". Ugyanez kérdezhető a környezetfüggetlen nyelvekre is.

A kérdésnek csak akkor van értelme, ha az input nyelvet valamilyen eszközzel reprezenáljuk.

Eldöntési kérdések reguláris nyelvekre

Ebben a részben a reguláris nyelvek osztályára vonatkozó eldöntési kérdésekkel foglalkozunk.
A fejezet során a reguláris nyelveket determinisztikus automatákkal reprezentáljuk.

Tehát, ha azt mondjuk, hogy adott egy L reguláris nyelv, akkor ez alatt azt értjük, hogy adott egy determinisztikus M automata, amelyre $L = L(M)$.

Mint látni fogjuk, a reguláris nyelvek valamennyi, általunk vizsgált tulajdonsága eldönthető. Magasabb nyelvosztályok esetén pedig egyre több olyan tulajdonság lesz, ami nem eldönthető.

Eldöntési kérdések reguláris nyelvekre

Eleme-e probléma
Probléma: Eldönthető-e tetszőleges w szó és L reguláris nyelv esetén, hogy teljesül-e $w \in L$?

Tétel. Az eleme-e probléma reguláris nyelvekre eldönthető.

Input: Egy w szó és egy $M = (Q, \Sigma, \delta, q_0, F)$ automatával megadott L reguláris nyelv (tehát $L = L(M)$).
Output: "Igen" ha $w \in L$, különben "Nem".

Algoritmus: Határozzuk meg azt a q állpotot, amelybe az automata q_0-ból a w hatására kerül. Ha $q \in F$, akkor a válasz "Igen", különben "Nem".

Eldöntési kérdések reguláris nyelvekre

Ürességi probléma

Probléma: Eldönthető-e tetszőleges L reguláris nyelv esetén, hogy teljesül-e $L = \emptyset$?

Tétel. Az ürességi probléma reguláris nyelvekre eldönthető.

Input: Egy $M = (Q, \Sigma, \delta, q_0, F)$ automatával megadott L reguláris nyelv.

Output: "Igen", ha $L = \emptyset$, különben "Nem".

Eldöntési kérdések reguláris nyelvekre

Ürességi probléma

Algoritmus:

(1) Számoljuk ki a q_0-ból elérhető állapotok Q' halmazát a $Q_0, Q_1, \ldots$ állapothalmaz-sorozat segítségével.

(i) Legyen $Q_0 = \{q_0\}$ és $i = 0$.

(ii) Legyen $Q_{i+1} = Q_i \cup \{\delta(q,a) \mid q \in Q_i$ és $a \in \Sigma\}$.

(iii) Ha $Q_i = Q_{i+1}$, akkor $Q' = Q_i$ (és stop), különben $i = i+1$ és goto (ii).

(2) Ha $Q' \cap F = \emptyset$, akkor a válasz "Igen", különben "Nem".

Eldöntési kérdések reguláris nyelvekre

Végtelen-e probléma

Lemma. Legyen $M = (Q, \Sigma, \delta, q_0, F)$ egy automata és legyen $k = ||Q||$. Az $L(M)$ nyelv akkor és csak akkor végtelen, ha van olyan $x \in L(M)$, melyre $k \leq |x| < 2k$.

Bizonyítás. $\Leftarrow$-irány: Tfh $\exists x \in L(M)$, melyre $|x| \geq k$. Akkor a pumpáló lemma szerint $\exists\ x = x_1x_2x_3$ felbontás, melyre $0 < |x_2| \leq k$ és minden $n \geq 0$-ra $x_1x_2^nx_3 \in L(M)$. Következésképpen $L(M)$ végtelen.
$\Rightarrow$-irány: Tfh $L(M)$ végtelen. Akkor $\exists\ x \in L(M)$ melyre $|x| \geq 2k$. Mivel $|x| \geq k$, a pumpáló lemma miatt $\exists\ x = x_1x_2x_3$ felbontás, melyre $0 < |x_2| \leq k$ és $x' = x_1x_3 \in L(M)$. Ha ekkor $|x'| \geq 2k$, akkor ismételjük az eljárást x'-re addig, amíg $|x'| < 2k$ nem lesz.

Eldöntési kérdések reguláris nyelvekre

Végtelen-e probléma

Probléma: Eldönthető-e tetszőleges L reguláris nyelv esetén, hogy L végtelen-e?

Tétel. A végtelen-e probléma reguláris nyelvekre eldönthető.

Input: Egy $M = (Q, \Sigma, \delta, q_0, F)$ automatával megadott L reguláris nyelv.

Output: "Igen", ha L végtelen, különben "Nem".

Eldöntési kérdések reguláris nyelvekre

Végtelen-e probléma

Algoritmus: Legyen $k = ||Q||$. Számoljuk ki minden $0 \leq i < 2k$ esetén a q_0-ból az i hosszúságú szavakkal elérhető állapotok Q_i halmazát:

(i) Ha $i = 0$ akkor $Q_i = \{q_0\}$, különben

(ii) $Q_i = \{\delta(q,a) \mid q \in Q_{i-1}$ és $a \in \Sigma\}$.

Ezután minden i-re, melyre $k \leq i < 2k$, vizsgáljuk meg, hogy $Q_i \cap F \neq \emptyset$ teljesül-e. Ha valamely i-re teljesül, akkor $L(M)$ végtelen, tehát a válasz "Igen", különben a válasz "Nem".

Eldöntési kérdések reguláris nyelvekre

Tartalmazási probléma

Probléma: Eldönthető-e tetszőleges L_1, L_2 reguláris nyelvek esetén, hogy $L_1 \subseteq L_2$ teljesül-e?

Tétel. A tartalmazási probléma reguláris nyelvekre eldönthető.

Input: Az $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ és $M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ automatákkal adott $L_1 = L(M_1)$ és $L_2 = L(M_2)$ reguláris nyelvek.

Output: "Igen", ha $L_1 \subseteq L_2$, különben "Nem".

Eldöntési kérdések reguláris nyelvekre

Tartalmazási probléma

Algoritmus: Konstruáljuk meg az $M = (Q_1 \times Q_2, \Sigma, \delta, (q_1, q_2), F)$ automatát, ahol $F = F_1 \times (Q_2 - F_2)$. (Ismert, hogy ekkor $L(M) = L_1 - L_2$.) Döntsük el, hogy $L(M) = \emptyset$ teljesül-e (a reguláris nyelvek üressége eldönthető). Ha teljesül, akkor a válasz "Igen", különben a válasz "Nem".

(Az algoritmus azon a tényen alapul, hogy $L_1 \subseteq L_2$ akkor és csak akkor, ha $L_1 - L_2 = \emptyset$.) $\diamond$

Eldöntési kérdések reguláris nyelvekre

Ekvivalencia probléma

Probléma: Eldönthető-e tetszőleges L_1, L_2 reguláris nyelvek esetén, hogy $L_1 = L_2$ teljesül-e?

Tétel. Az ekvivalencia probléma reguláris nyelvekre eldönthető.

Input: Az $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ és $M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ automatákkal adott $L_1 = L(M_1)$ és $L_2 = L(M_2)$ reguláris nyelvek.

Output: "Igen", ha $L_1 = L_2$, különben "Nem".

Eldöntési kérdések reguláris nyelvekre

Ekvivalencia probléma

Algoritmus: Alkalmazzuk kétszer a tartalmazás eldöntési algoritmusát és döntsük el, hogy az $L_1 \subseteq L_2$ és az $L_2 \subseteq L_1$ tartalmazások teljesülnek-e. Ha mindkettő teljesül, akkor a válasz "Igen", különben a válasz "Nem".

(Az algoritmus azon a tényen alapul, hogy $L_1 = L_2$ akkor és csak akkor, ha $L_1 \subseteq L_2$ és $L_2 \subseteq L_1$.)

Eldöntési kérdések reguláris nyelvekre

Összefoglalás

Kérdés	**Eldönthető-e?**
Eleme-e	igen
Üresség	igen
Végtelen-e	igen
Tartalmazás	igen
Ekvivalencia	igen