

Formális nyelvek I/1.

Véges automaták és reguláris nyelvek

Fülöp Zoltán

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

- 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 *környezetfüggetlen 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

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

szabályok alkalmazásával felépíthető jelsorozatok (szavakat) nevezzük. A | jel választási lehetőséget jelent, olvassuk "vagy"-nak. Ez egy *környezetfüggetlen nyelv*.

A formális nyelvek egy alkalmazása

Levezetés (vagy deriváció):

$$\begin{aligned} & \langle \textit{kif} \rangle \\ \Rightarrow & \langle \textit{tag} \rangle \\ \Rightarrow & \langle \textit{tag} \rangle * \langle \textit{fakt} \rangle \\ \Rightarrow & \langle \textit{tag} \rangle * (\langle \textit{kif} \rangle) \\ \Rightarrow & \langle \textit{tag} \rangle * (\langle \textit{kif} \rangle + \langle \textit{tag} \rangle) \\ \Rightarrow & \langle \textit{tag} \rangle * (\langle \textit{tag} \rangle + \langle \textit{tag} \rangle) \\ \Rightarrow & \langle \textit{fakt} \rangle * (\langle \textit{tag} \rangle + \langle \textit{tag} \rangle) \\ \Rightarrow & \langle \textit{fakt} \rangle * (\langle \textit{fakt} \rangle + \langle \textit{tag} \rangle) \\ \Rightarrow & \langle \textit{fakt} \rangle * (\langle \textit{fakt} \rangle + \langle \textit{fakt} \rangle) \end{aligned}$$

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

A formális nyelvek egy alkalmazása

Levezetés (vagy deriváció):

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

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

A formális nyelvek egy alkalmazása

Egy másik példa: a FONYA programozási nyelv szintaxisa:

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

A formális nyelvek egy alkalmazása

Egy másik példa: a FONYA programozási nyelv szintaxisa:

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

A formális nyelvek egy alkalmazása

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

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

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

```
A := 0;  
while A + C do  
    begin A := A + 1  
        B := B * C  
    end;  
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 $\langle \textit{program} \rangle \Rightarrow^* w$?

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

A formális nyelvek egy alkalmazása

Megjegyzés:

A környezetfüggetlen nyelvtanok csak a programozási nyelvek szintaktikai szabályainak a megadására alkalmasak. Nem alkalmasak olyan szemantikai feltételek kezelésére, mint például: "a programban előforduló minden változót előzetesen deklarálni kell". Ez utóbbihoz további eszközök szükségesek (pl attribútum nyelvtan).

Általános fogalmak, jelölések

Használni fogjuk:

- ▶ görög betűk: $\alpha, \beta, \gamma, \delta, \dots, \varepsilon, \dots, \rho, \sigma$, stb
- ▶ Γ, Δ, Σ (nagy gamma, delta és szigma)
- ▶ rövidítések: $\implies$ (logikai implikáció, "ha ... akkor"),
 $\iff$ (ekvivalencia, "akkor és csak akkor")
- ▶ $A \iff B$ azt jelenti, hogy $A \implies B$ és $B \implies A$, ahol A és B állítások
- ▶ elemi halmazelméleti fogalmak: $A \subseteq B$ (A részhalmaza B -nek), $A \subset B$ (A valódi részhalmaza B -nek)
- ▶ halmazelméleti műveletek: $A \cup B$ (egyesítés vagy unió), $A \cap B$ (metszet), $A - B$ (kivonás), $\overline{A}$ (komplementer)
- ▶ a fentiekre egy példa: $A \subset B \iff A \subseteq B$ és $B - A \neq \emptyset$

Általános fogalmak, jelölések

Használni fogjuk:

- ▶ Tétel (= egy állítás) és Bizonyítás (= az állítás igazolása).
- ▶ Lemma (= egy egyszerűbb állítás), általában egy tétel előkészítése.
- ▶ A bizonyítást nem azért végezzük el, mert nem hisszük el, hogy az állítás igaz, hanem azért, mert a bizonyításból ismerjük meg az állítás valódi tartalmát, értékét.
- ▶ A bizonyítások általában konstruktívak: meg is konstruáljuk az állításban szereplő objektumokat (pl nyelvtant vagy automatát).

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:

$$\Sigma^* = \{a_1 \dots a_k \mid k \geq 0, a_1, \dots, a_k \in \Sigma\}$$

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

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

$\Sigma^+ = \{a, b, aa, ab, ba, bb, aaa, \dots\}$

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

Általános fogalmak, jelölések

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

Példa: abb pefixe 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

Egy w szó megfordításán (vagy tükörképén) az alábbi módon definiált w^R szót értjük:

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

Példa: $a^R = a$, $ab^R = ba$, $aabbb^R = bbbaa$, stb.

Észrevétel: $(xy)^R = y^R x^R$ minden $x, y \in \Sigma^*$ -ra.

Általános fogalmak, jelölések

Nyelv: Σ^* tetszőleges L részhalmazát Σ feletti nyelvnek nevezzük (L mint language). Ha L véges számú szóból áll, akkor véges nyelvnek nevezzük.

Példa $\Sigma = \{a, b\}$ feletti nyelvekre:

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

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

Általános fogalmak, jelölések

Az összes Σ feletti nyelvek halmaza a Σ^* összes részhalmazainak a halmaza, vagyis:

$$\mathcal{P}(\Sigma^*) = \{L \mid L \subseteq \Sigma^*\}.$$

Egy megjegyzés: Minden Σ -ra a Σ^* megszámlálhatóan végtelen halmaz (mert elemei a hosszuk szerint, azonos hosszúságon belül pedig abc szerint sorba rendezhetők). Mivel a megszámlálhatóan végtelen halmaznak kontinuum számosságú részhalmaza van, ezért kontinuum számosságú Σ feletti nyelv van.

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

- ▶ $\bar{L} = \Sigma^* - L$ az L komplementere,
- ▶ $L^R = \{w^R \mid w \in L\}$ az L megfordítása (vagy tükörképe).
Például: $\{\varepsilon, ab, abba, babb\}^R = \{\varepsilon, ba, babba, bbab\}$ és
 $\{a^n b^n \mid n \geq 0\}^R = \{b^n a^n \mid n \geq 0\}$

Műveletek nyelvekkel

- $L_1 L_2 = \{uv \mid u \in L_1, v \in L_2\}$ az L_1 és L_2 *konkatenációja*

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

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

$$\{b, ba\}\{\varepsilon, ab, bab\} = \{b, bab, bbab, ba, baab, babab\}$$

Ebből az is látszik, hogy a konkatenáció nem kommutatív.

Még egyszerűbb ellenpélda: legyen $L_1 = \{a\}$ és $L_2 = \{b\}$.
Ekkor

$$L_1 L_2 = \{ab\} \text{ és } L_2 L_1 = \{ba\}.$$

Műveletek nyelvekkel

További példák konkatenációra:

$$\{a^n \mid n \geq 0\} \{b^n \mid n \geq 0\} = \{a^n b^m \mid n, m \geq 0\}$$

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

A nyelvek konkatenációja is asszociatív: $L_1(L_2L_3) = (L_1L_2)L_3$,
ezért a zárójelezés elhagyható.

► *Hatványozás*: $L^n = \overbrace{LL \dots L}^n$, $L^0 = \{\varepsilon\}$

például: $bbb, bbab, babab, bababa \in \{b, ba\}^3$

Műveletek nyelvekkel

- *Iteráció:* $L^* = \{\varepsilon\} \cup L \cup LL \cup LLL \cup \dots$ az L nyelv iteráltja.

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

Ekvivalens definíció:

$$L^* = \{w_1 \dots w_n \mid n \geq 0, w_1, \dots, w_n \in L\}$$

Példa:

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

- $L^+ = L \cup LL \cup LLL \cup \dots$ (a 0 eset kizárva) vagy
 $L^+ = \{w_1 \dots w_n \mid n \geq 1, w_1, \dots, w_n \in L\}$

Műveletek nyelvekkel

Megjegyzés:

- ▶ $L^* = \{\varepsilon\} \cup L^+$.
- ▶ Minden L nyelvre $\varepsilon \in L^*$.
- ▶ $\emptyset^* = \{\varepsilon\}$ és $\emptyset^+ = \emptyset$
- ▶ Ha $\varepsilon \in L$, akkor $\varepsilon \in L^+$, tehát $L^+ = L^*$.
- ▶ Ha $\varepsilon \notin L$, akkor $\varepsilon \notin L^+$, tehát $L^+ \neq L^*$.

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:

$$L_1 \cup (L_2 \cup L_3) = (L_1 \cup L_2) \cup L_3 \quad L_1(L_2L_3) = (L_1L_2)L_3$$

$$L_1 \cup L_2 = L_2 \cup L_1 \quad L\{\varepsilon\} = \{\varepsilon\}L = L$$

$$L \cup L = L$$

$$L \cup \emptyset = \emptyset \cup L = L \quad L\emptyset = \emptyset L = \emptyset$$

Továbbá:

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

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

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

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

$$(L_1L_2)^R = L_2^RL_1^R$$

Nyelvtanok

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

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

Környezetfüggetlen nyelvtanok

Környezetfüggetlen 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 $A \rightarrow \alpha$ alakú ún. *átírási szabályok* véges halmaza, ahol $A \in N$ és $\alpha \in (N \cup \Sigma)^*$.

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

Környezetfüggetlen nyelvtanok

Példa:

$$G_1 = (\{S\}, \{a, b\}, \{S \rightarrow aSb, S \rightarrow \varepsilon\}, S)$$

egy nyelvtan, ahol

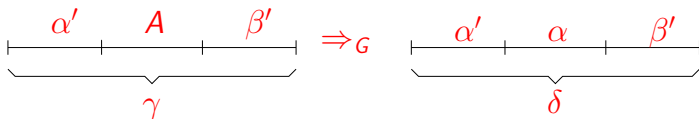
- ▶ $\{S\}$ a nemterminális ábécé (ez esetben csak egy nemterminális van, általában nem),
- ▶ $\{a, b\}$ a terminális ábécé,
- ▶ S a kezdő (start) szimbólum,
- ▶ $\{S \rightarrow aSb, S \rightarrow \varepsilon\}$ a szabályok halmaza.

$S \rightarrow aSb$ egy szabály, aminek a bal oldala S , a jobb oldala pedig aSb

Környezetfüggetlen nyelvtanok

Közvetlen levezetés (deriváció):

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



$A \Rightarrow$ jelölés nem keverendő össze a $\implies$ (logikai implikáció) rövidítéssel és a szabályokban alkalmazott $\rightarrow$ jelöléssel.

Környezetfüggetlen nyelvtanok

Példa közvetlen levezetésre:

$$G_1 = (\{S\}, \{a, b\}, \{S \rightarrow aSb, S \rightarrow \varepsilon\}, S)$$

$bSa \Rightarrow_{G_1} baSba$ az $S \rightarrow aSb$ szabállyal

$baaSa \Rightarrow_{G_1} baaa$ az $S \rightarrow \varepsilon$ szabállyal

Környezetfüggetlen nyelvtanok

Levezetések:

$\gamma \Rightarrow_G \delta$: egy lépés (= a közvetlen levezetés)

$\gamma \Rightarrow_G^n \delta$: $n \geq 0$ 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 csak $\Rightarrow$ -t írunk.

Környezetfüggetlen nyelvtanok

Példa több lépéses levezetésekre:

$$G_1 = (\{S\}, \{a, b\}, \{S \rightarrow aSb, S \rightarrow \varepsilon\}, S)$$

$$\begin{aligned} S &\Rightarrow aSb && \Rightarrow aaSbb \\ &\Rightarrow aaaSbbb && \Rightarrow aaabbbb \end{aligned}$$

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

Továbbá: $aSb \Rightarrow aaSbb$, $aSb \Rightarrow^2 aaaSbbb$, tehát $aSb \Rightarrow^* aaSbb$, $aSb \Rightarrow^* aaaSbbb$ mindegyike teljesül.

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

Környezetfüggetlen nyelvtanok

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

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

Az $w \in \Sigma^*$ feltétel miatt w -ben nincsenek nemterminálisok, tehát nem lehet belőle "tovább lépni".

A $G_1 = (\{S\}, \{a, b\}, \{S \rightarrow aSb, S \rightarrow \varepsilon\}, S)$ példában

$$L(G_1) = \{a^n b^n \mid n \geq 0\}.$$

Környezetfüggetlen nyelvtanok

Fontos megjegyzés:

A nyelv és a nyelvtan két különböző fogalom. Egy nyelv szavak egy (véges vagy végtelen) halmaza, míg egy nyelvtan egy olyan végesen specifikált (adott) eszköz, amellyel nyelvet lehet generálni.

Egy adott nyelvtan egy nyelvet generál, de egy adott nyelvhez több olyan nyelvtan is megadható, amely őt generálja.

A $G = (N, \Sigma, P, S)$ és $G' = (N', \Sigma', P', S')$ környezetfüggetlen nyelvtanok *ekvivalensek*, ha $L(G) = L(G')$.

Környezetfüggetlen nyelvtanok

Például, az előbbi $G_1 = (\{S\}, \{a, b\}, \{S \rightarrow aSb, S \rightarrow \varepsilon\}, S)$ nyelvtan ekvivalens a

$$G_2 = (\{A, S\}, \{a, b\}, \{S \rightarrow \varepsilon, S \rightarrow aAb, A \rightarrow aAb, A \rightarrow \varepsilon\}, S)$$

nyelvtannal, mert ugyancsak

$$L(G_2) = \{a^n b^n \mid n \geq 0\}.$$

Például:

$$\begin{array}{llll} S & \Rightarrow_{G_2} & aAb & \Rightarrow_{G_2} & aaAbb \\ & \Rightarrow_{G_2} & aaaAbbb & \Rightarrow_{G_2} & aaabbbb \end{array}$$

Környezetfüggetlen nyelvtanok

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

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

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

Környezetfüggetlen nyelvtanok

Az $A \rightarrow \alpha_1, \dots, A \rightarrow \alpha_n$ szabályok felírását a következőképpen rövidítjük: $A \rightarrow \alpha_1 \mid \dots \mid \alpha_n$.

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

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

Egy levezetés G_{ar} -ban:

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

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

Környezetfüggetlen nyelvtanok

Még egy példa: adjunk meg olyan cf nyelvtant, amelyik az

$$L = \{a^n b^n c^m \mid n, m \geq 0\}$$

nyelvet generálja.

Környezetfüggetlen nyelvtanok

Még egy példa: adjunk meg olyan cf nyelvtant, amelyik az

$$L = \{a^n b^n c^m \mid n, m \geq 0\}$$

nyelvet generálja. Megoldás a következő G nyelvtan:

- $S \rightarrow AC$,
- $A \rightarrow aAb \mid \varepsilon$,
- $C \rightarrow Cc \mid \varepsilon$.

Példa egy levezetésre:

$$\begin{aligned} S &\Rightarrow AC \Rightarrow aAbC \Rightarrow aaAbbC \Rightarrow aabbC \Rightarrow aabbCc \\ &\quad aabbCcc \Rightarrow aabbCccc \Rightarrow aabbcccc \end{aligned}$$

Jelöléssel: $L(G) = L$.

Könyezetfüggetlen nyelvek

Egy L nyelvet *könyezetfüggetlen nyelvnek* hívunk, ha van olyan G könyezetfüggetlen nyelvtan, melyre $L = L(G)$. Az összes könyezetfüggetlen nyelvek halmazát CF-fel jelöljük. Például az $\{a^n b^n \mid n \geq 0\}$ nyelv, és az aritmetikai kifejezésből álló nyelv könyezetfüggetlen.

Általános jelölések

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

fogják jelölni. ha ezt megjegyezzük, akkor a jelöléseink sokkal rövidebbek lesznek.

Például $A \in N$, $\alpha \in (N \cup \Sigma)^*$ és $x \in \Sigma^*$ helyett írhatunk A -t, α -t és x -et.

Reguláris nyelvtanok

Egy $G = (N, \Sigma, P, S)$ nyelvtan *reguláris* (vagy *jobblineáris*), ha P -ben minden szabály $A \rightarrow xB$ vagy $A \rightarrow x$ alakú.

A jobblineáris elnevezés abból ered, hogy minden szabály jobb oldalán legfeljebb egy nemterminális van és ha van, akkor az a szó jobb oldali végén helyezkedik el.

Reguláris nyelvtan esetén minden levezetés

$$A_1 \Rightarrow x_1 A_2 \Rightarrow x_1 x_2 A_3 \Rightarrow \dots \Rightarrow x_1 x_2 \dots x_n A_{n+1}$$

alakú, ahol az alkalmazott szabályok

$$A_1 \rightarrow x_1 A_2, A_2 \rightarrow x_2 A_3, \dots, A_n \rightarrow x_n A_{n+1}$$

Levezetést csak $A \rightarrow x$ alakú szabállyal fejezhetünk be.
Ugyanez érvényes minden olyan szóra, amelyet az S kezdőszimbólumból vezetünk le.

Reguláris nyelvtanok

Definíció szerint minden reguláris nyelvtan környezetfüggetlen, mert az $A \rightarrow xB$ szabályok esetén $xB \in (N \cup \Sigma)^*$ és az $A \rightarrow x$ szabályok esetén $x \in (N \cup \Sigma)^*$.

Más szóval, a reguláris nyelvtanok speciális környezetfüggetlen nyelvtanok.

Reguláris nyelvtanok

A továbbiakban példákat adunk reguláris nyelvtanokra.

Az első példa a G_3 nyelvtan, melynek szabályai:

$$\blacktriangleright S \rightarrow aS \mid bS \mid \varepsilon$$

Tehát a G_3 nyelvtan reguláris és $L(G_3) = \Sigma^*$, ahol $\Sigma = \{a, b\}$.

Például az abb szó levezetése:

$$S \Rightarrow aS \Rightarrow abS \Rightarrow abbS \Rightarrow abb$$

Derivációt csak az $S \rightarrow \varepsilon$ szabállyal tudunk befejezni!

Reguláris nyelvtanok

Legyen G_4 az a nyelvtan, melynek szabályai:

$$\blacktriangleright S \rightarrow aaS \mid abS \mid baS \mid bbS \mid \varepsilon$$

Akkor a G_4 nyelvtan reguláris és

$$L(G_4) = \{w \in \Sigma^* \mid |w| \text{ páros szám} \},$$

ahol $\Sigma = \{a, b\}$. Például az $abbbbaa$ szó levezetése:

$$S \Rightarrow abS \Rightarrow abbbS \Rightarrow abbbbaaS \Rightarrow abbbbaa$$

Ugyancsak megadható olyan reguláris nyelvtan, amelyik a

$$\{w \in \Sigma^* \mid |w| \text{ páratlan szám} \}$$

nyelvet generálja.

Reguláris nyelvtanok

Legyen G_5 az a nyelvtan, melynek szabályai:

$$\blacktriangleright S \rightarrow aA \mid B$$

$$\blacktriangleright A \rightarrow aA \mid B$$

$$\blacktriangleright B \rightarrow bB \mid \varepsilon$$

Akkor a G_5 nyelvtan reguláris. Derivációk:

$$S \Rightarrow aA \Rightarrow aaA \Rightarrow aaB \Rightarrow aabB \Rightarrow aab$$

$$S \Rightarrow aA \Rightarrow aaA \Rightarrow aaB \Rightarrow aa, S \Rightarrow B \Rightarrow bB \Rightarrow bbB \Rightarrow bb$$

$$L(G_5) = \{a^n b^m \mid n, m \geq 0\}$$

Derivációt csak a $B \rightarrow \varepsilon$ szabállyal tudunk befejezni!

Reguláris nyelvtanok

További példa a G_6 nyelvtan, melynek szabályai:

- ▶ $S \rightarrow aB,$
- ▶ $B \rightarrow bB \mid a$

A G_6 nyelvtan ugyancsak reguláris. Derivációk:

$S \Rightarrow aB \Rightarrow aa, S \Rightarrow aB \Rightarrow abB \Rightarrow aba$

$S \Rightarrow aB \Rightarrow abB \Rightarrow abbB \Rightarrow abba$

$$L(G_6) = \{ab^n a \mid n \geq 0\}$$

Derivációt csak a $B \rightarrow a$ szabállyal lehet befejezni!

Reguláris nyelvtanok

További példa a G_7 nyelvtan, melynek szabályai:

$$\blacktriangleright S \rightarrow aS \mid bS \mid A$$

$$\blacktriangleright A \rightarrow abaB$$

$$\blacktriangleright B \rightarrow aB \mid bB \mid \varepsilon$$

A G_7 nyelvtan ugyancsak reguláris. Egy deriváció:

$$S \Rightarrow bS \Rightarrow baS \Rightarrow baA \Rightarrow baabaB \Rightarrow \\ baababB \Rightarrow baababbB \Rightarrow baababbaB \Rightarrow baababba$$

Egy szó akkor és csak akkor deriválható S -ből, ha előfordul benne az aba rész-szó. Ezért $L(G_7) = \{uabav \mid u, v \in \Sigma^*\}$, ahol $\Sigma = \{a, b\}$.

Reguláris nyelvtanok

Végül két megjegyzés:

1) Minden olyan $G = (N, \Sigma, P, S)$ nyelvtan reguláris, melyben $P = \emptyset$. Nyilvánvaló, hogy minden ilyen G nyelvtan esetén $L(G) = \emptyset$.

2) Minden véges L nyelvhez van olyan reguláris nyelvtan, amely ezt a nyelvet generálja. Ugyanis, ha minden $w \in L$ szóra felvesszük az $S \rightarrow w$ szabályt, akkor az így kapott nyelvtan reguláris lesz.

Például, az $\{\varepsilon, ab, bab, abba\}$ véges nyelvet az

► $S \rightarrow \varepsilon \mid ab \mid bab \mid abba$

reguláris nyelvtan generálja.

Reguláris nyelvek

Egy L nyelvet *reguláris nyelvnek* hívunk, ha van olyan G reguláris nyelvtan, melyre $L = L(G)$. Az összes reguláris nyelvek halmazát **REG**-gel jelöljük.

Például az $\emptyset$, az $\{a^n b^m \mid n, m \geq 0\}$ nyelv, és minden véges nyelv reguláris.

Könyezetfüggetlen nyelvek és reguláris nyelvek

Mivel minden reguláris nyelvtan egyúttal könyezetfüggetlen is, ezért minden reguláris nyelv egyúttal könyezetfüggetlen is,

A most bevezetett jelölésekkel: $\text{REG} \subseteq \text{CF}$.

Később meg fogjuk mutatni, hogy

$$\text{REG} \subset \text{CF},$$

vagyis

$$\text{REG} \subseteq \text{CF} \text{ és } \text{CF} - \text{REG} \neq \emptyset.$$

Más szóval, vannak olyan könyezetfüggetlen nyelvek, amelyek nem regulárisak.

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:

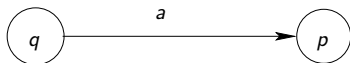
1. Q egy nem üres, véges halmaz, az *állapotok halmaza*,
2. Σ egy ábécé, az *input ábécé*,
3. $q_0 \in Q$ a *kezdő állapot*,
4. $F \subseteq Q$ a *végállapotok halmaza*,
5. $\delta : Q \times \Sigma \rightarrow Q$ egy leképezés, az *átmenetfüggvény*.

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.

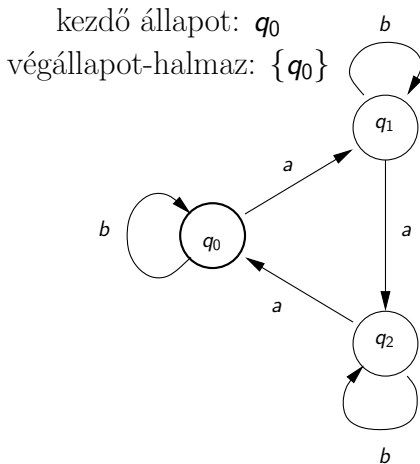
Véges automaták

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:



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

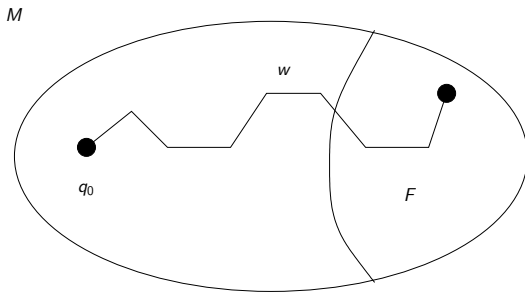
Véges automaták

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.

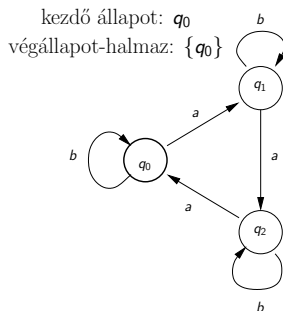
Szavakkal: q_0 -ból a w hatására valamelyik $q \in F$ végállapotba jutunk.



$$w \in L(M)$$

Véges automaták

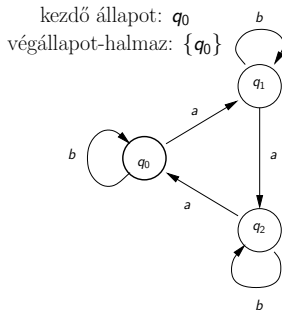
Átmenetek az M_3 automatában:



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

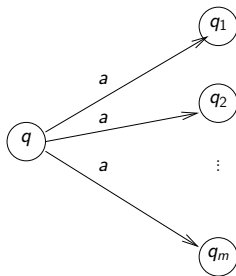
1. Q egy nem üres, véges halmaz, az *állapotok halmaza*,
2. Σ egy ábécé, az *input ábécé*,
3. $q_0 \in Q$ a *kezdő állapot*,
4. $F \subseteq Q$ a *végállapotok halmaza*,
5. $\delta : Q \times \Sigma \rightarrow \mathcal{P}(Q)$ egy leképezés, az *átmenetfüggvény*.

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:

$$\delta(q, a) = \{q_1, \dots, q_n\}$$

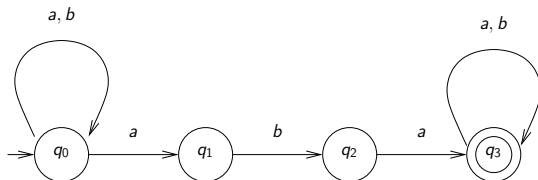


Az is megengedett, hogy $\delta(q, a) = \emptyset$.

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 $q' \in \delta(q, a)$.

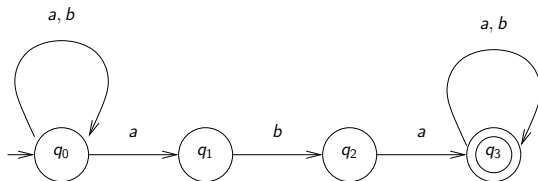
Az $M = (Q, \Sigma, \delta, q_0, F)$ automata által felismert nyelven az

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

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!

Vegyük észre, hogy $L(M_{aba})$ ugyanaz, mint a G_5 reguláris nyelvtan által generált nyelv.

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 $\delta(q, a)$ legalább 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 automatóhoz 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).

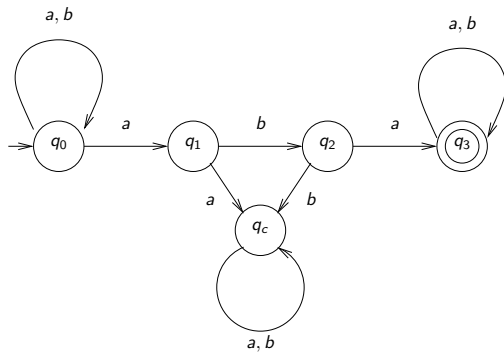
Továbbá, minden $q \in Q$ és $a \in \Sigma$ esetén, legyen

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

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.

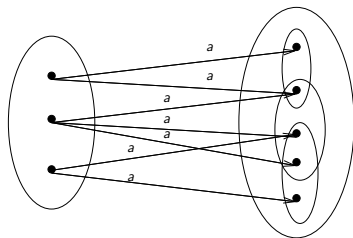
b) Fordítva: legyen $M = (Q, \Sigma, \delta, q_0, F)$ egy nemdeterminisztikus automata. Megadunk egy $M' = (Q', \Sigma, \delta', q'_0, F')$ determinisztikus automatát, amelyre $L(M') = L(M)$.

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,
- ▶ $q'_0 = \{q_0\}$,
- ▶ $F' = \{S \subseteq Q \mid S \cap F \neq \emptyset\}$,
- ▶ $\delta' : Q' \times \Sigma \rightarrow Q'$ az a leképezés amelyre tetszőleges $S \in Q'$ és $a \in \Sigma$ esetén $\delta'(S, a) = \bigcup_{q \in S} \delta(q, a)$.



S

$\delta'(S, a) = \bigcup_{q \in S} \delta(q, a)$

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

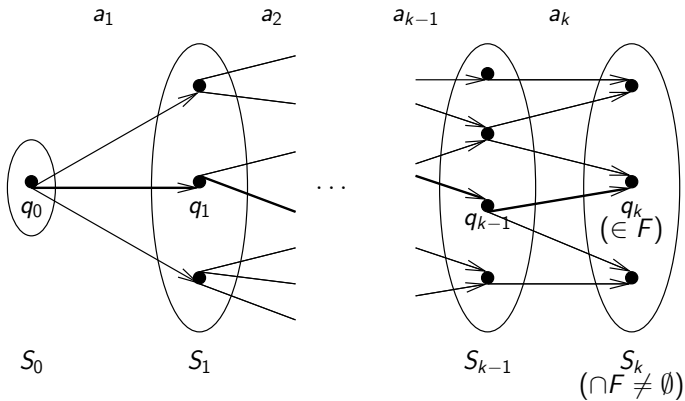
$$(\{q_0\}, w) \vdash_{M'}^* (S, \varepsilon) \text{ akkor és csak akkor, ha} \\ S = \{q \in Q \mid (q_0, w) \vdash_M^* (q, \varepsilon)\}.$$

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

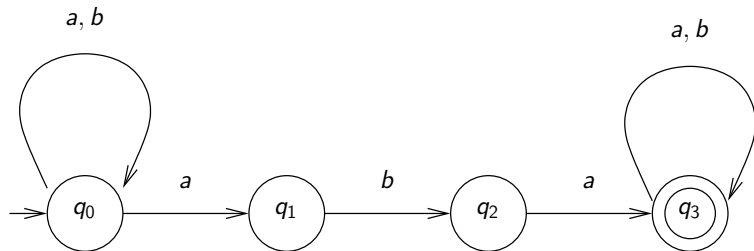
Egy fontos megjegyzés:

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

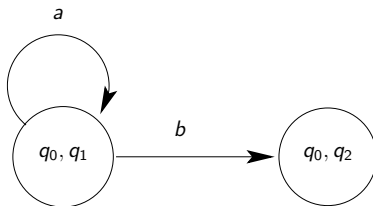
Példa. Determinizáljuk az M_{aba} nemdeterminisztikus automatát!



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

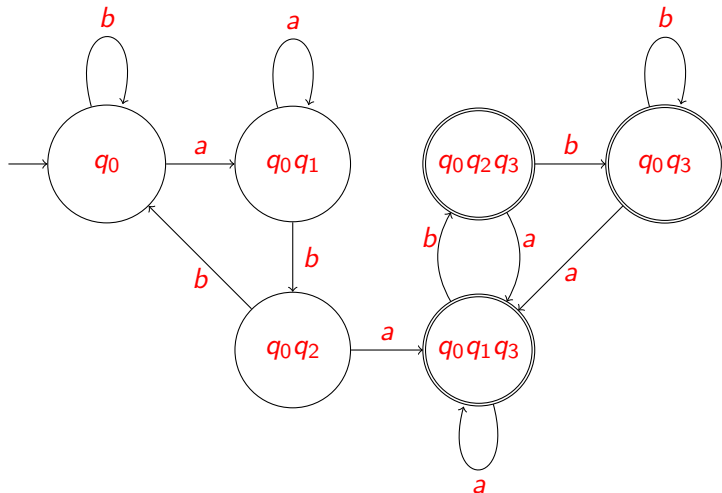
Véges automaták

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



Véges automaták

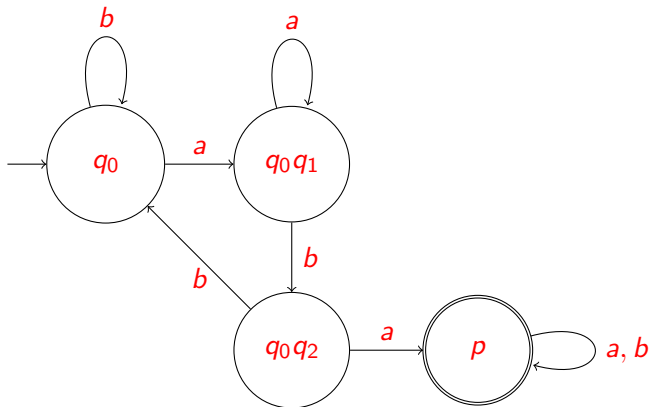
Példa. Az M_{aba} automata determinizálása:



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

Véges automaták

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



Az így kapott automatának már csak **4** állapota van.

Véges automaták

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

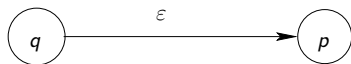
Az $M = (Q, \Sigma, \delta, q_0, F)$ rendszert *nemdeterminisztikus ε -automatának* nevezzük, ahol:

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. $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \rightarrow \mathcal{P}(Q)$ egy leképezés, az *átmenetfüggvény*.

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

Véges automaták

Egy ε -átmenet



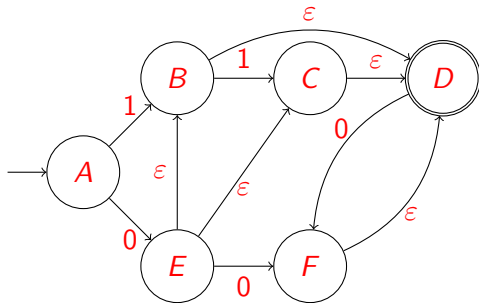
Átmeneti reláció:

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

A felismert nyelv definíciója ugyanaz, mint a nemdeterminisztikus
esetében: 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 egy ε -átmenettel elérhető állaptoakat, 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 $CI(q)$ lezárások ismeretében, legyen:

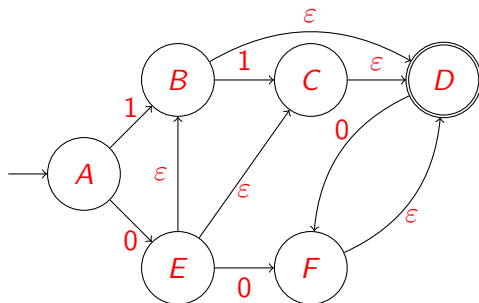
- ▶ $\delta'(q, a) = \bigcup_{p \in CI(q)} \delta(p, a)$ és
- ▶ $F' = \{q \in Q \mid CI(q) \cap F \neq \emptyset\}$.

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

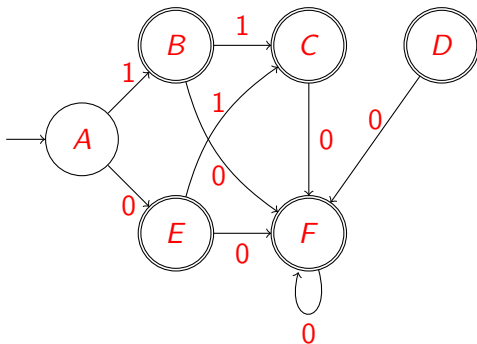
Egy példa:



- ▶ $CI(A) = \{A\}$, $CI(B) = \{B, D\}$, $CI(C) = \{C, D\}$,
- ▶ $CI(D) = \{D\}$, $CI(E) = \{E, B, C, D\}$, $CI(F) = \{F, D\}$
- ▶ végállapotok az ε -mentes automatában: $\{E, B, C, F, D\}$

Véges automaták

Egy példa:



Az ekvivalens nemdeterminisztikus automata.

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

Véges automaták minimalizálása

Külön fájlban...

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 determinisztikusát. Ez még inkább igaz a nemdeterminisztikus ϵ -automatákra. Például két automatóhoz nagyon könnyű megadni egy olyan nemdeterminisztikus ϵ -automatát, amely az eredeti automaták által felismert nyelvek egyesítését vagy konkatenációját ismeri fel.

Véges automaták

Néhány alkalmazás hardver és szoftver modellezésére:

- ▶ Digitális áramkörök tervezésére és viselkedésének vizsgálatára alkalmas szoftverek modellezése.
- ▶ Fordítóprogramok "lexikális elemzés" fázisa: az input szöveg (program) olvasása és logikai egységekre bontása (azonosító, kulcsszó, elválasztójel).
- ▶ Szoftverek nagyméretű szövegek (pl. weboldalak tömege) olvasására és azokban szavak, kifejezések és egyéb sablonok megtalálására.
- ▶ Véges állapotú rendszerek (pl. kommunikációs protokollok, biztonságos információ cserét biztosító protokollok) verifikálására alkalmas szoftverek modellezése.

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 legsűkebb 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 $|R| = |R_1|^*$.

Definíció: Egy $L \subseteq \Sigma^*$ nyelv reprezentálható reguláris kifejezéssel (röviden: reprezentálható), 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

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

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

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

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 kifejezésre

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

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 $L = \{x_1, \dots, x_n\}$, $n \geq 1$. Akkor

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

ahol

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

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 `+` jelet a betűk között: az $a_1 + \dots + a_n$ reguláris kifejezést $[a_1 \dots a_n]$ formában rövidíti. Például a $< + > + =$ helyett $[< > =]$ -t ír.
- Kihasználva, hogy az ASCII rendezett, halmazokat rövidítve definiál. Például $[0-9]$ jelenti a $0 + \dots + 9$ 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

A következőkben az alábbi három nyelvosztályt vizsgáljuk meg:

- (1) a reguláris nyelvek **REG** osztálya (azok a nyelvek, amelyek generálható reguláris nyelvtannal),
- (2) az automatával felismerhető nyelvek osztálya,
- (3) a reguláris kifejezésekkel reprezentálható nyelvek osztálya.

Ki fog derülni, hogy a három nyelvosztály megegyezik (vagyis valójában egyetlen nyelvosztály három különböző megadási módjáról beszélünk).

Az ekvivalencia tétel

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

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

Bizonyítás. Megmutatjuk, hogy

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

Akkor (1) $\iff$ (2) $\iff$ (3).



A reprezentálható nyelvek regulárisak

1. Lemma. $(3) \implies (1)$: Ha $L \subseteq \Sigma^*$ nyelv reprezentálható reguláris kifejezéssel, akkor generálható reguláris 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 $G = (\{S\}, \Sigma, \emptyset, S)$ reguláris nyelvtannal.

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

A reprezentálható nyelvek regulárisak

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 $G_i = (N_i, \Sigma, P_i, S_i)$, reguláris nyelvtannal, $i = 1, 2$. ($N_1 \cap N_2 = \emptyset$.)

A reprezentálható nyelvek regulárisak

Akkor L generálható a

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

reguláris nyelvtannal, ahol S egy új szimbólum.

$$S \Rightarrow_G^* w$$

akkor és csak akkor, ha

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

A reprezentálható nyelvek regulárisak

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 $G_i = (N_i, \Sigma, P_i, S_i)$, reguláris nyelvtannal, $i = 1, 2$. ($N_1 \cap N_2 = \emptyset$.)

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

A reprezentálható nyelvek regulárisak

- ▶ Ha $A \rightarrow xB \in P_1$, akkor $A \rightarrow xB \in P$,
- ▶ Ha $A \rightarrow x \in P_1$, akkor $A \rightarrow xS_2 \in P$,
(G_1 "befejező" szabályainak kezelése)
- ▶ P_2 minden eleme P -nek is eleme.

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

akkor és csak akkor, ha

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

A reprezentálható nyelvek regulárisak

Indukciós lépés.

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

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

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

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

A reprezentálható nyelvek regulárisak

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

$$S \Rightarrow_G \varepsilon$$

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

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

$$w_1 w_2 S \Rightarrow_G w_1 w_2 S_1 \dots$$

Az ekvivalencia tétel

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

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

Bizonyítás.

- 1. Lemma: (3) $\implies$ (1) $\checkmark$
- 2. Lemma: (1) $\implies$ (2)
- 3. Lemma: (2) $\implies$ (3)

A reguláris nyelvek felismerhetők automatával

2. Lemma. $(1) \implies (2)$: 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 reguláris nyelvtan.

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

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

A reguláris nyelvek felismerhetők automatával

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

Bizonyítás. Konstruáljuk meg P' -t a következőképpen:

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

A reguláris nyelvek felismerhetők automatával

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

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

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

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

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

szabályokat, ahol $A_1, \dots, A_n$ új nemterminálisok.

Legyen $N' = N \cup \{ \text{új nemterminálisok} \}$.

A reguláris nyelvek felismerhetők automatával

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

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

Ugyanis

$A \rightarrow a_1 \dots a_n B \in P$ 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

$A \rightarrow a_1 \dots a_n \in P$ 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 reguláris nyelvek felismerhetők automatával

Példa a szabályok szétdarabolására

$$\begin{array}{lcl} G : & S & \rightarrow abA \mid bB \\ & A & \rightarrow bbB \mid \varepsilon \\ & B & \rightarrow ab \mid bA \end{array}$$

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

A reguláris nyelvek felismerhetők automatával

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

Bizonyítás. Konstruáljuk meg M -et a következőképpen:

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

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

A reguláris nyelvek felismerhetők automatával

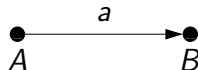
G -ben:

$$A \rightarrow aB \in P$$

$\Leftrightarrow$

$$B \rightarrow \varepsilon \in P$$

M -ben:



$$B \in F$$

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

A reguláris nyelvek felismerhetők automatával

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

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

Részletesebben:

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

akkor és csak akkor, ha

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

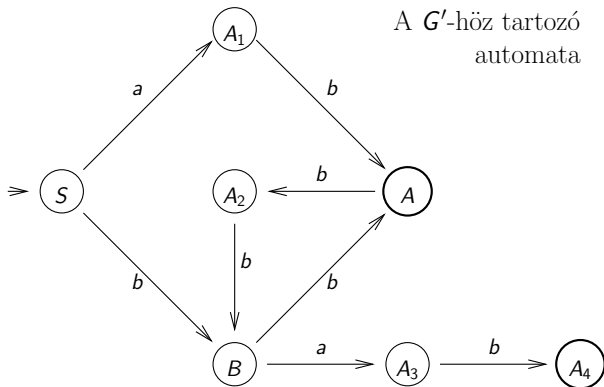
Az $A = S$, $B \in F$ választással adódik, hogy $L(M) = L(G)$.

A reguláris nyelvek felismerhetők automatával

Például, vegyük az előbbi G' nyelvtant:

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

A reguláris nyelvek felismerhetők automatával



Az automatával felismerhető nyelvek regulárisak

Egy megjegyzés a 2.2 Lemmához: a lemma bizonyításában alkalmazott konstrukció könnyen "megfordítható":

Lemma. Tetszőleges M nemdeterminisztikus ε -automatához megadható egy olyan G reguláris nyelvtant, amelyre $L(G) = L(M)$.

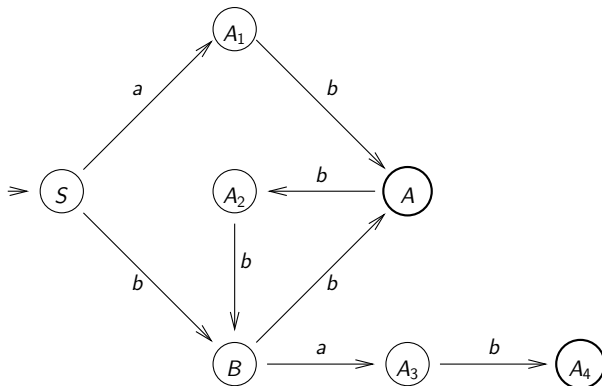
Bizonyítás. A nyelvtan nemterminálisai az automata állapotai lesznek. Továbbá, ha az automatában valamely q állapotból egy $a \in (\Sigma \cup \{\varepsilon\})$ szimbólum hatására egy p állapotba jutunk (azaz $p \in \delta(q, a)$), akkor a nyelvtanba felvesszük a $q \rightarrow ap$ szabályt. Végül minden q végállapot esetén a nyelvtanba felvesszük a $q \rightarrow \varepsilon$ szabályt.

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

Az automatával felismerhető nyelvek regulárisak

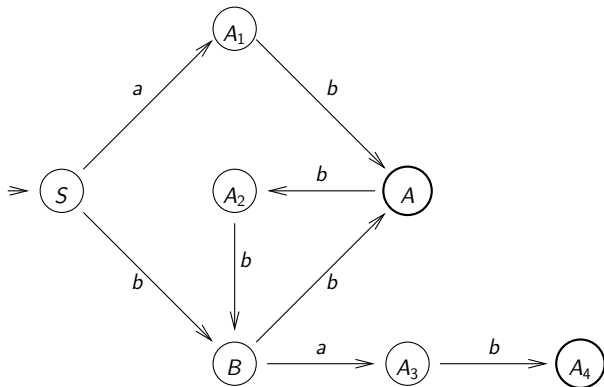
Példa 1 az előbbi megjegyzéshez (és lemmához):

Ha az előző példában kapott automatára alkalmazzuk a lemma bizonyításában szereplő konstrukciót, akkor visszkapjuk a G' reguláris nyelvtant.



Az automatával felismerhető nyelvek regulárisak

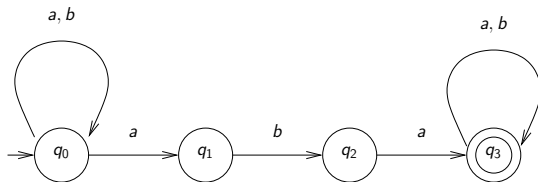
Az automata és a G' reguláris nyelvtant:


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

Az automatával felismerhető nyelvek regulárisak

Példa 2 az előbbi megjegyzéshez (és lemmához):

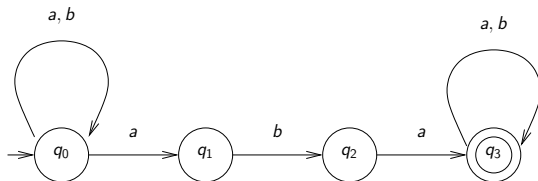
Vesszük a már ismert M_{aba} automatát



és megadjuk a vele ekvivalens reguláris nyelvtant.

Az automatával felismerhető nyelvek regulárisak

Az M_{aba} automata:



és a vele ekvivalens G reguláris nyelvtan:

$$q_0 \rightarrow aq_0 \mid bq_0 \mid aq_1$$

$$q_1 \rightarrow bq_2$$

$$q_2 \rightarrow aq_3$$

$$q_3 \rightarrow aq_3 \mid bq_3 \mid \varepsilon$$

Fontos: a $q_3 \rightarrow \varepsilon$ szabály kell, mert az automatában q_3 végállapot!

Az ekvivalencia tétel

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

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

Bizonyítás.

- 1. Lemma: (3) $\implies$ (1) $\checkmark$
- 2. Lemma: (1) $\implies$ (2) $\checkmark$
- 3. Lemma: (2) $\implies$ (3)

Az automatával felismerhető nyelvek reprezentálhatók

Lemma 3. (2) $\implies$ (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 $Q = \{1, \dots, n\}$ 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:

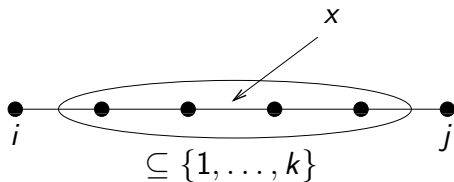
$$x \in L_{i,j}^{(k)} \iff (i, x) \vdash_M^* (j, \varepsilon) \text{ és}$$

minden $(i, x) \vdash_M^+ (i', x') \vdash_M^+ (j, \varepsilon)$ esetén

$$i' \in \{1, \dots, k\}.$$

Az automatával felismerhető nyelvek reprezentálhatók

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



Az automatával felismerhető nyelvek reprezentálhatók

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

$$L(M) = |R_{1,j_1}^{(n)} + \dots + R_{1,j_l}^{(n)}|,$$

ahol $F = \{j_1, \dots, j_l\}$.

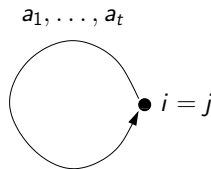
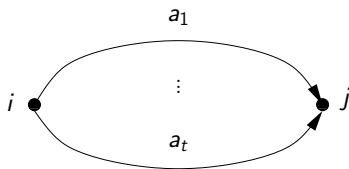
Többet bizonyítunk: minden $0 \leq k \leq n$ -ra és $1 \leq i, j \leq n$ -re megadunk egy $R_{i,j}^{(k)}$ reguláris kifejezést, melyre

$$|R_{i,j}^{(k)}| = L_{i,j}^{(k)}.$$

Az automatával felismerhető nyelvek reprezentálhatók

$R_{i,j}^{(k)}$ megadása k szerinti indukcióval. $k = 0$:

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



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

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

Az automatával felismerhető nyelvek reprezentálhatók

$k \Rightarrow k + 1$:

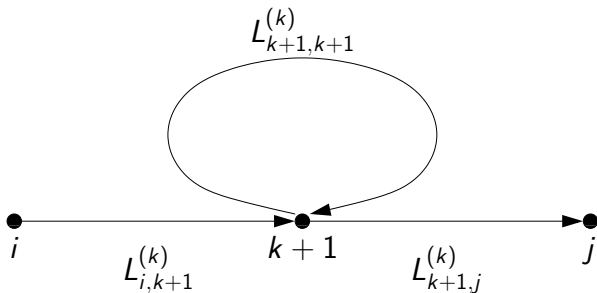
tfh minden i, j -re megadtuk $R_{i,j}^{(k)}$ -t, melyre $|R_{i,j}^{(k)}| = L_{i,j}^{(k)}$
 $\Rightarrow$ megadjuk minden i, j -re $R_{i,j}^{(k+1)}$ -t

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 reprezentálhatók

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



Az automatával felismerhető nyelvek reprezentálhatók

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

A reprezentálható nyelvek felismerhetők automatával

Egy megjegyzés a 3. Lemmához ($(2) \Rightarrow (3)$ bizonyítása): Tudjuk, hogy a lemma "megfordítható", mert $(3) \Rightarrow (1) \Rightarrow (2)$. De van közvetlen $(3) \Rightarrow (2)$ konstrukció is:

Lemma. Tetszőleges R reguláris kifejezéshez megadható egy olyan M (nemdeterminisztikus ε -automata), amelyre $|R| = L(M)$.

Bizonyítás. Minden R reguláris kifejezéshez megadunk egy olyan M automatát, melynek

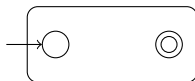
- ▶ pontosan egy végállapota van,
- ▶ a kezdőállapotába nincs bemenő él,
- ▶ az egyetlen végállapotából nincs kimenő él

és melyre $|R| = L(M)$.

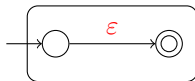
A bizonyítást R felépítése szerinti indukcióval végezzük.

A reprezentálható nyelvek felismerhetők automatával

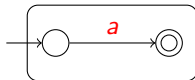
Az (i), (ii) és (iii) alapesetek:



(i) $R = \emptyset$



(ii) $R = \epsilon$



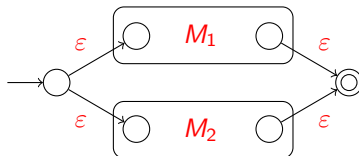
(iii) $R = a$

A reprezentálható nyelvek felismerhetők automatával

Indukciós lépés.

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

Indukciós feltevés: az R_i reguláris kifejezéshez megadtunk olyan M_i automatát mely rendelkezik a megadott tulajdonságokkal $i = 1, 2$.
Ekkor R -hez a következő automatát adjuk meg:



(iv) a)

A reprezentálható nyelvek felismerhetők automatával

Indukciós lépés.

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

Indukciós feltevés: ugyanaz, mint az előbb. Ekkor R -hez a következő automatát adjuk meg:



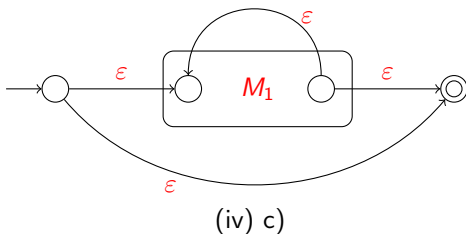
(iv) b)

A reprezentálható nyelvek felismerhetők automatával

Indukciós lépés.

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

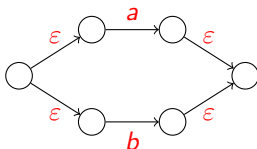
Indukciós feltevés: az R_1 reguláris kifejezéshez megadtunk olyan M_1 automatát mely rendelkezik a megadott tulajdonságokkal.
Ekkor R -hez a következő automatát adjuk meg:



A reprezentálható nyelvek felismerhetők automatával

Példa: adjunk meg a $(a + b)^*b(a + b)$ reguláris kifejezéssel ekvivalens automatát.

Megoldás: először megadjuk az $(a + b)$ -vel ekvivalens (1) automatát:

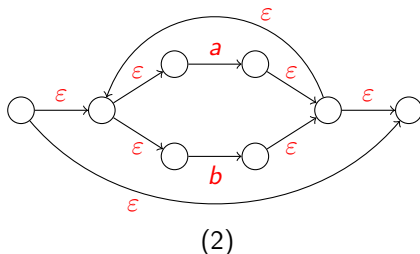


(1)

A reprezentálható nyelvek felismerhetők automatával

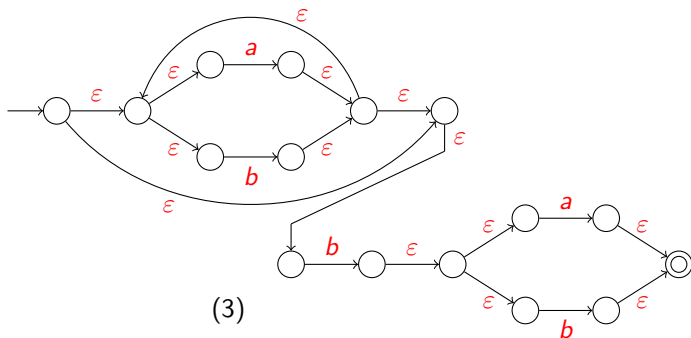
Példa: adjunk meg a $(a + b)^*b(a + b)$ reguláris kifejezéssel ekvivalens automatát.

Megoldás: (1)-ből megadjuk az $(a + b)^*$ -gal ekvivalens (2) automatát:



A reprezentálható nyelvek felismerhetők automatával

Megoldás: végül összerakjuk az $(a + b)^*b(a + b)$ -vel ekvivalens (3) automatát:



Az ekvivalencia tétel

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

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

Bizonyítás.

- 1. Lemma: (3) $\implies$ (1) $\checkmark$
- 2. Lemma: (1) $\implies$ (2) $\checkmark$
- 3. Lemma: (2) $\implies$ (3) $\checkmark$

Tehát (1) $\iff$ (2) $\iff$ (3).



Az ekvivalencia tétel

Összefoglalás:

A következő eszközök mindegyikével a reguláris nyelveket reprezentálhatjuk:

- ▶ reguláris nyelvtanok,
- ▶ reguláris nyelvtanok, amelyek minden szabálya $A \rightarrow B$, $A \rightarrow aB$ vagy $A \rightarrow \varepsilon$ 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 reguláris és a környezetfüggetlen nyelvek felismerő automatái

Nyelvosztályok és felismerő automatáik:

Nyelvosztály	Felismerő	Det. vs nemdet.
REG (reg. nyelvek)	véges automaták	det. = nemdet. = nemdet. ϵ átmenetekkel
CF (cf 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, $w_1 w_2^n w_3 \in L$.

Másik alak: Ha egy $L \subseteq \Sigma^*$ nyelv reguláris, akkor megadható

Tehát, ha egy $L \subseteq \Sigma^*$ nyelvhez nem adható meg olyan $k > 0$ egész szám, ami teljesíti a fenti feltételeket, akkor az $L \subseteq \Sigma^*$ nyelv nem reguláris.

Pumpáló lemma reguláris nyelvekre

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

Legyen $k = ||Q||$ (az állapotok száma). Megmutatjuk, hogy ez a k teljesíti a feltételeket.

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

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

továbbá $q_K \in F$.

Pumpáló lemma reguláris nyelvekre

Mivel $K \geq ||Q||$, a $q_0, q_1, \dots, q_K$ sorozaban 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 $w_1 = a_1 \dots a_i$, $w_2 = a_{i+1} \dots a_j$ és $w_3 = a_{j+1} \dots a_K$.
(Ha $i = 0$ akkor $w_1 = \varepsilon$, ha $j = K$ akkor $w_3 = \varepsilon$)

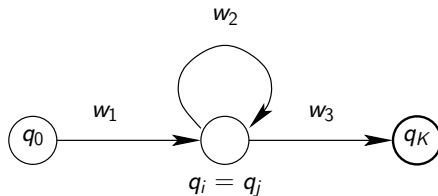
Nyilvánvaló, hogy $w = w_1 w_2 w_3$. Továbbá,

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 alkalmazása:

Lemma. Az $L = \{a^n b^n \mid n \geq 0\}$ nyelv nem reguláris.

Bizonyítás. Indirekt bizonyítás: tegyük fel, hogy L reguláris.

Akkor megadható olyan k szám, ami teljesíti a pumpáló lemmában szereplő feltételeket.

Vegyük az $a^k b^k \in L$ szót, melynek hossza $2k \geq k$.

A pumpáló lemmában szereplő feltételek szerint 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 nem létezik ilyen k szám. Akkor viszont a fenti L nyelv nem reguláris.

Pumpáló lemma reguláris nyelvekre

Intuíció:

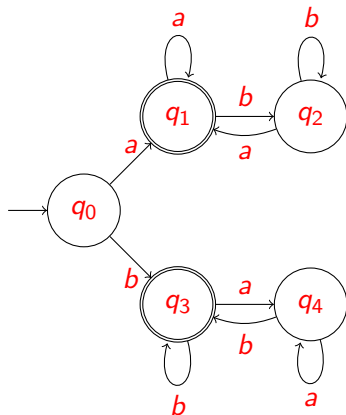
Egy automata nem képes számolni, hogy két betű ugyanannyiszor szerepel-e.

De kezeljük ezt óvatosan! Az alábbi nyelv reguláris:

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

Ld a következő ábrát.

Pumpáló lemma reguláris nyelvekre



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

Pumpáló lemma reguláris nyelvekre

A Pumpáló lemma egy következménye:

van olyan környezetfüggetlen nyelv, amelyik nem reguláris.

Tétel. $REG \subset CF$ (valódi része).

Bizonyítás.

a) $REG \subseteq CF$, mivel minden reguláris nyelvtan környezetfüggetlen.

b) $CF - REG \neq \emptyset$: az $\{a^n b^n \mid n \geq 0\}$ nyelv CF -beli, mivel generálható az

$$S \rightarrow aSb \mid \varepsilon$$

szabályokból álló környezetfüggetlen nyelvtannal. Ugyanakkor, mint láttuk $L \notin REG$.

Reguláris nyelvek zártsági tulajdonságai

Műveletekre való zártság általában:

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

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

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 egyváltozós műveletekre (pl. komplementer, iteráció, tükrözés) való zártság hasonlóan definiálható.

Reguláris nyelvek zártsági tulajdonságai

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

Tétel. REG zárt a reguláris műveletekre.

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

Legyen $L_1, L_2 \in \text{REG}$. 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

$L_1 \cup L_2 \in \text{REG}$.

A konkatenációra és a $*$ -ra való zártság hasonlóan bizonyítható.

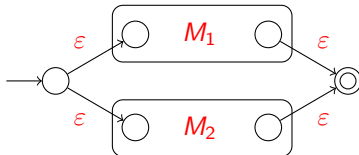
Reguláris nyelvek zártsági tulajdonságai

Tétel. REG zárt a reguláris műveletekre.

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

Legyen $L_1, L_2 \in REG$. Megadunk egy olyan M automatát, melyre $L(M) = L_1 \cup L_2$.

Mivel L_1, L_2 reguláris, vannak olyan M_1, M_2 automaták, melyekre $L_1 = L(M_1)$ és $L_2 = L(M_2)$. Ekkor a következő automata az $L_1 \cup L_2$ nyelvet ismeri fel:



A konkatenációra és a $*$ -ra való zártság hasonlóan bizonyítható.

Reguláris nyelvek zártsági tulajdonságai

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

Egy erősebb állítást bizonyítunk:

Tétel. REG zárt az $\cup$, $\cap$, és $-$ műveletekre.

Bizonyítás. A reguláris nyelveket most determinisztikus automatákkal reprezentáljuk.

A bizonyításban pedig 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

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

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.

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

Következmény: REG zárt a komplementer műveletre is.

Bizonyítás. Legyen $L \subseteq \Sigma^*$ egy reguláris nyelv. Mivel a Σ^* nyelv reguláris (nyilvánvaló) és $\bar{L} = \Sigma^* - L$, a kivonásra való zártság miatt $\bar{L}$ is reguláris.

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 $\bar{L}$ nyelvet ismeri fel, mert pontosan azok a szavak viszik végállapotba, amelyek M -et $Q - F$ -be, vagyis nem végállapotba viszik. Ezért az $\bar{L}$ nyelv is reguláris.

Reguláris nyelvek zártsági tulajdonságai

Összefoglalás:

Művelet	Zártság
$\cup$	igen
konkatenáció	igen
$*$	igen
$\cap$	igen
kivonás	igen
komplementer	igen
tükrözés	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, végeesség) értendő és úgy szól, hogy létezik-e olyan algoritmus, amelynek inputja a nyelvosztály tetszőleges eleme, outputja pedig "Igen" ha a nyelv rendelkezik az adott tulajdonsággal, 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 állapotot, amelybe az automata q_0 -ból a w hatására kerül. Ha $q \in F$, akkor a válasz "Igen", különben "Nem".

Eldöntési kérdések reguláris nyelvekre

Üres-e probléma

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

Tétel. Az Üres-e 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

Üres-e probléma

Algoritmus:

- (1) Számoljuk ki a q_0 -ból elérhető állapotok Q' halmazát a $Q_0, Q_1, \dots$ állapothalmaz-sorozat segítségével.
 - (i) Legyen $Q_0 = \{q_0\}$ és $i = 0$.
 - (ii) Legyen $Q_{i+1} = Q_i \cup \{\delta(q, a) \mid q \in Q_i \text{ és } a \in \Sigma\}$.
 - (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 $\exists x = x_1 x_2 x_3$ felbontás, melyre $0 < |x_2| \leq k$ és minden $n \geq 0$ -ra $x_1 x_2^n x_3 \in L(M)$.

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 $\exists x = x_1 x_2 x_3$ felbontás, melyre $0 < |x_2| \leq k$ és $x' = x_1 x_3 \in L(M)$. 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) $Q_i = \{\delta(q, a) \mid q \in Q_{i-1} \text{ és } a \in \Sigma\}$.

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 $L_1 - L_2 = \emptyset$.)



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

Megjegyzés: A minimalizálás részről tudjuk, hogy egy adott L nyelvet felismerő minimális automata (izomorfizmus erejéig) egyértelműen meghatározott.

Ebből következik, hogy az M_1 és M_2 automaták ekvivalenciáját a következőképpen is eldönthetjük: mindkettőt minimalizáljuk, majd ellenőrizzük, hogy a kapott M'_1 és M'_2 automaták "megegyeznek-e" (izomorfak-e).

A minimális automata egyértelműsége miatt ugyanis $L(M_1) = L(M_2)$ akkor és csak akkor, ha M'_1 "ugyanaz, mint" M'_2 .

Eldöntési kérdések reguláris nyelvekre

Összefoglalás

Kérdés	Eldönthető-e?
Eleme-e	igen
Üres-e	igen
Végtelen-e	igen
Tartalmazás	igen
Ekvivalencia	igen