

Véges automaták, reguláris nyelvek

Kiegészítő anyag az Algoritmusképzés tárgyhoz
(a Rónyai–Ivanyos–Szabó: Algoritmusok könyv mellé)

Friedl Katalin
BME SZIT
friedl@cs.bme.hu

2017. augusztus 3.

A véges automata egy igen egyszerű számítási modell. Bár vannak általánosabb formái is, mi itt csak olyan változatokkal foglalkozunk, amelyek egy eldöntési problémára (igen/nem kérdésre) segítenek válaszolni. A későbbiekben vizsgált automatatípusok (veremautomaták, Turing-gépek) esetében is többnyire erre szorítkozunk majd.

1. Alapfogalmak, jelölések

A véges, nem üres Σ halmaz jelöli az ábécét, elemeit általában *betűknek* vagy *karaktereknek* hívjuk.

A Σ elemeiből képzett véges hosszú sorozatok a *szavak*. Egy x szó hossza az, hogy hány betűből áll, a hossz jele $|x|$.

Az n hosszú szavak halmazát Σ^n jelöli. Így $\Sigma^1 = \Sigma$. A 0 hosszú szó (*üres szó*) is egy szó, jele: ε . Természetesen $|\varepsilon| = 0$ és $\Sigma^0 = \{\varepsilon\}$.

Az összes Σ feletti szó halmazának jele Σ^* . Felírhatjuk, hogy $\Sigma^* = \bigcup_{n \geq 0} \Sigma^n$.

Egy Σ feletti L nyelv szavak tetszőleges halmaza, $L \subseteq \Sigma^*$.

1. Példa. Legyen $\Sigma = \{0, 1\}$.

- A 0-val kezdődő 0/1 sorozatok egy nyelvet alkotnak.
- $L = \{00, 101\}$ egy nyelv, aminek 2 eleme van.

- Az $L = \emptyset$ üres nyelvben nincs egyetlen szó sem.
- Az $L = \{\varepsilon\}$ nyelvnek egyetlen eleme van, az üres szó.
- $L = \Sigma^4$ az összes olyan 0/1 sorozatból álló nyelv, melyek hossza 4.
- $L = \Sigma^*$ az összes véges hosszú 0/1 sorozatból álló nyelv.

A továbbiakban a nyelvek felismerésére szolgáló véges automatáknak három különböző változatával ismerkedünk meg. Először az algoritmikus szempontból legkézenfekvőbb *determinisztikus*, majd ennek bizonyos értelemben egyszerűsített változatával (*hiányos*). Végül a sokszor tömörebb és világosabb *nemdeterminisztikus* véges automatával. Látni fogjuk, hogy ezek mind ekvivalensek egymással.

2. Determinisztikus véges automata

1. Definíció. A determinisztikus véges automata *vagy röviden DVA egy olyan* $M = (Q, \Sigma, \delta, q_0, F)$ ahol:

- Q egy véges, nem üres halmaz. Ez az automata állapotainak halmaza.
- Σ egy véges, nem üres halmaz. Ez az automata ábécéje.
- $\delta : Q \times \Sigma \rightarrow Q$, az automata állapotátmeneti függvénye.
- $q_0 \in Q$ a kezdő állapot.
- $F \subseteq Q$ az elfogadó állapotok halmaza.

A determinisztikus véges automata *működése* egy adott $w \in \Sigma^*$ szón a következőképpen írható le. Az automatát a q_0 állapotból indítjuk. Ha w az n hosszú $a_1 a_2 \cdots a_n$ karaktersorozat ($a_i \in \Sigma$), akkor először az $r_1 = \delta(q_0, a_1)$ állapotba lép. A második lépésben az $r_2 = \delta(r_1, a_2)$ állapotba kerül, és így tovább, amíg az a_n karaktert is feldolgozza. Ekkor megáll, a számítás véget ér. Legyen $r_n \in Q$ a végső állapot.

2. Definíció. Azt mondjuk, hogy az M DVA elfogadja az n hosszú $w \in \Sigma^*$ szót, ha a w -hez tartozó számítás végén az utolsó, r_n állapotra teljesül, hogy $r_n \in F$. Egyébként M nem fogadja el *vagy* elutasítja a $w \in \Sigma^*$ szót.

3. Definíció. Az M DVA által elfogadott nyelv azoknak a szavaknak a halmaza, amelyeket M elfogad. Jele: $L(M)$.

A definícióból következik, hogy $L(M) \subseteq \Sigma^*$.

1. Feladat. Legyen $\Sigma = \{0, 1\}$. Adjunk meg egy olyan determinisztikus véges automatát, amely a nullára végződő szavakból álló nyelvet fogadja el!

Megoldás: Legyen $Q = \{A, B\}$, a kezdő állapot A , az elfogadó állapotok halmaza egy elemű, $F = \{B\}$, az automata $M = (Q, \Sigma, \delta, A, F)$, ahol az állapotátmeneti függvény a következő:

$$\delta(A, 0) = B$$

$$\delta(A, 1) = A$$

$$\delta(B, 0) = B$$

$$\delta(B, 1) = A$$

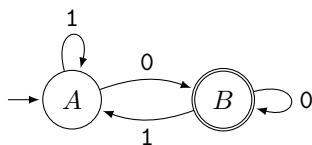
Vegyük észre, hogy ez a δ függvény olyan, hogy mindegy, hogy az automata éppen melyik állapotban van: ha 0 karakter következik, akkor az automata a B állapotba, míg ha 1 karakter jön, akkor az A állapotba kerül. Mivel B az egyetlen elfogadó állapot, ez az automata valóban pontosan azokat a szavakat fogadja el, amelyek a 0 karakterre végződnek. \square

Sokszor áttekinthetőbb, ha az állapotátmeneti függvényt táblázattal adjuk meg. A fenti esetben ez így néz ki:

	0	1
A	B	A
B	B	A

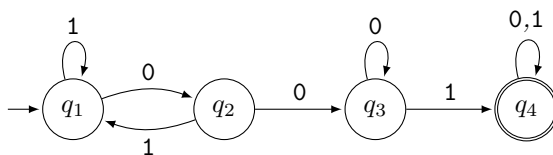
A DVA-t irányított gráffal is ábrázolhatjuk. A gráf csúcsai az állapotoknak felelnek meg, az állapotátmeneti függvényt az élek segítségével adjuk meg. A $\delta(q, a) = q'$ átmenetnek egy, a q állapotnak megfelelő csúcsból a q' állapotnak megfelelő csúcsba mutató irányított él felel meg, amihez az a címke tartozik. A kezdő állapotot egy bemenő nyíl jelöli, az elfogadó állapotokat dupla kör jelzi.

Az előző automata ebben a formában:



Általában feltesszük, hogy az így kapott gráfban a kezdőállapotból minden csúcs elérhető irányított úton (azaz minden állapot elérhető valamilyen bemeneti szóval). Adott esetben ezt a gráfon végzett bejárással könnyen ellenőrizhetjük, és ha vannak elérhetetlen állapotok, azokat, a hozzájuk tartozó átmenetekkel együtt elhagyjuk az automatából.

2. Feladat. Határozzuk meg az alábbi determinisztikus véges automata által elfogadott nyelvet!



Megoldás: Nem nehéz megsejteni, hogy a nyelv az összes olyan szóból áll, amelyik tartalmazza a 001 részszt. Ennek bizonyítására egy máskor is sokszor alkalmazható módszer a következő: ahelyett, hogy csak az elfogadó állapotot néznénk, határozzuk meg minden egyes állapotra, mely szavakra ér ott véget a számítás!

Sejtés:

q_4 : L_4 , ami a 001-et tartalmazó szavakból áll;

q_3 : L_3 , ami az olyan szavakból áll, melyek 00-ra végződnek és nincs bennük 001;

q_2 : L_2 , ami az olyan szavakból áll, melyek 0-ra végződnek, de nincs bennük 00;

q_1 : L_1 , ami az üres szóból és az olyan 1-re végződő szavakból áll, melyekben nincs 00.

Vegyük észre, hogy ezek a nyelvek diszjunktak és $L_1 \cup L_2 \cup L_3 \cup L_4 = \{0, 1\}^*$. Ha ez nem teljesülne, a sejtésünk biztosan nem lehetne jó, hiszen minden szónak benne kell lennie a halmazok közül pontosan egyben.

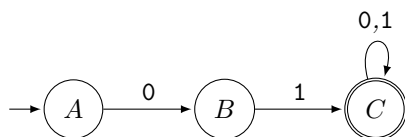
Azt, hogy a sejtés igaz, a szavak hossza szerinti teljes indukcióval lehet belátni. Rövid, pl. 0, 1, 2 hosszú szavakra könnyű ellenőrizni, hogy valóban a megfelelő L_i -be tartoznak. Tegyük fel, hogy a legfeljebb k hosszú szavakra a sejtésünk igaz ($k \geq 2$). Legyen $w = a_1 a_2 \cdots a_k a_{k+1}$. Ha az utolsó előtti karakter (a_k) feldolgozása után a q_1 állapotban vagyunk, akkor az indukciós feltevés miatt $a_1 a_2 \cdots a_k \in L_1$, azaz nincs benne 00 és $a_k = 1$. Így biztos, hogy w -ben nincs 00. Amennyiben az utolsó karakter $a_{k+1} = 1$, akkor w 1-re végződik, azaz $w \in L_1$, ami jó, hiszen az automata marad a q_1 állapotban. Ha viszont az utolsó karakter $a_{k+1} = 0$, akkor w 0-ra végződik (és továbbra sincs benne 00), ami miatt $w \in L_2$ teljesül. Ez megfelel annak, hogy ilyenkor az automata átkerül a q_2 állapotba. Hasonlóan kell ellenőrizni a többi esetet is, hogy ha az első k karaktert feldolgozva a q_i állapotban voltunk, akkor az a_{k+1} hatására a sejtésnek megfelelő állapotba kerülünk. \square

3. Hiányos véges automata

Ha a δ átmeneti függvény nincs mindenhol definiálva, akkor *hiányos* véges automatáról beszélünk.

Ha egy ilyen automata a számítása során egy q állapotban olyan a karaktert olvas, amire a (q, a) helyen δ nincs definiálva, akkor a számítás itt megáll, *elakad*. Ilyenkor a szót *nem fogadja el*, függetlenül attól, hogy az elakadáskor elfogadó állapotban van vagy nem.

2. Példa. Az alábbi hiányos véges automata azokat a szavakat fogadja el, amelyek a 01 sorozattal kezdődnek ($\Sigma = \{0, 1\}$).



Például a 001 szónál ez az automata az első 0 hatására átlép a B állapotba, de ott a második 0 feldolgozásakor elakad, hiszen nincs megfelelő átmenet definiálva. Így természetesen nem is fogadja el ezt a szót.

A következő tétel mutatja, hogy minden hiányos véges automata teljessé tehető.

1. Tétel. Minden M hiányos véges automata kiegészíthető, egy M' determinisztikus véges automatává úgy, hogy M és M' ugyanazt a nyelvet fogadja el.

Bizonyítás: Egészítsük ki M -et egy új t állapottal, ami nem lesz elfogadó állapot, és minden hiányzó átmenetet irányítsunk ide. Ezzel a korábbi hiányokat megszüntettük, de újabbak keletkeztek. Ezért minden $a \in \Sigma$ esetén legyen $\delta'(t, a) = t$. Könnyen látható, hogy ez az M' már nem hiányos és $L(M) = L(M')$. \square

4. Nemdeterminisztikus véges automata

Nyelvek leírását sokszor megkönnyíti, ha nem csak elhagyhatunk állapotokat és átmeneteket, hanem az átmeneteknek nem is kell egyértelműeknek lenni, ugyanahhoz az állapot és karakter párhoz több következő állapot is lehetséges. Ez egy, a bonyolultságelméletben szokásos, nemdeterminisztikus számítási modellt eredményez. Bár ez az automatatípus nem ad közvetlenül algoritmust a nyelv szavainak felismerésére, de időnként sokkal kisebb automatát, tömörebb, áttekinthetőbb leírást biztosít.

4. Definíció. A nemdeterminisztikus véges automata vagy röviden NVA egy olyan $M = (Q, \Sigma, \delta, q_0, F)$, amelyben Q , Σ , q_0 és F jelentése ugyanaz, mint az 1. definícióban, azonban a δ átmeneti függvény értéke az állapotok egy halmaza lehet, minden $q \in Q$ és $a \in \Sigma$ esetén $\delta(q, a) \subseteq Q$.

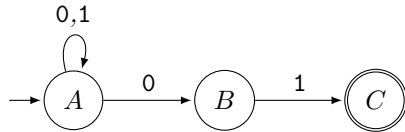
A nemdeterminisztikus véges automata működése egy adott $w = a_1 a_2 \dots a_n \in \Sigma^*$ szón a következőképpen írható le: Az automatát a q_0 állapotból indítjuk, ahonnan először valamelyik $r_1 \in \delta(q_0, a_1)$ állapotba lép, a második lépésben valamelyik $r_2 \in \delta(r_1, a_2)$ állapotba kerül, és így tovább, amíg az a_n karaktert is feldolgozza. Legyen r_n az az állapot, ahova utoljára lépett. A számítás *elfogadó*, ha $r_n \in F$. Ha egy adott helyzetben nem tud lépni, mert $\delta(q_i, a_{i+1}) = \emptyset$, akkor a számítás elakad, és így nem elfogadó.

Úgy lehet elképzelni, hogy ugyanazon a bemeneten többféle lefutás (*számítási út*) is van – ezért hívják nemdeterminisztikusnak.

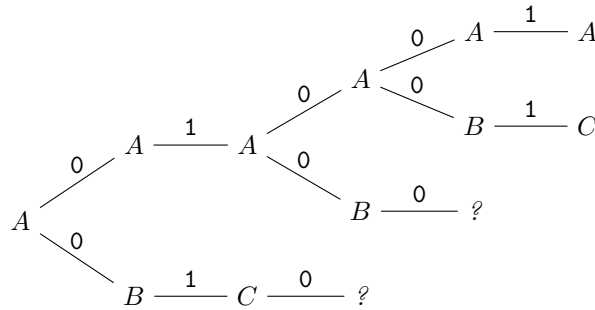
5. Definíció. Az automata akkor fogadja el a szót, ha van olyan számítási út (lefutás), ami elfogadó állapottal ér véget.

Egy adott bemeneten a számítások leírhatók egy *számítási fával*, amiben az elágazások a lehetséges következő állapotoknak felelnek meg. Az NVA akkor fogad el, ha van olyan ág, ami mentén a számítás nem akad el és a levélhez elfogadó állapot tartozik.

3. Példa. A következő nemdeterminisztikus véges automata a 01-re végződő szavakat fogadja el:



Számítási fa a 01001 szóhoz:



Látszik, hogy ebben az esetben csak kétféleképpen nem akad el a számítás és ezekből az A állapottal végződő nem elfogadó, a C-re végződő viszont elfogadó. Mivel van elfogadással végződő számítási út (azaz tudunk „jó utat” választani), az automata elfogadja a szót

A fenti példában a nemdeterminizmust arra használjuk, hogy nem kell tudnunk, mikor jön az utolsó két karakter. Ha a számítás során korábban lépünk a B állapotba, akkor vagy a B vagy a C állapotban a számítás elakad (a szót nem tudjuk végigolvasni), azaz nem kapunk elfogadó számítást. Azonban, ha jókor hagyjuk el az A állapotot, akkor a 01-re végződő szavak esetében elfogadó számítást kapunk.

Persze nem nehéz egy determinisztikus automatát felrajzolni a megadott nyelvre.

Vegyük észre, hogy a determinisztikus és a hiányos véges automata is speciális esete a nemdeterminisztikusnak. Ha minden $q \in Q$ és $a \in \Sigma$ párra $\delta(q, a)$ egy 1 elemű halmaz, akkor az eredeti DVA definíciójához jutunk. A hiányos automatát akkor kapjuk, ha minden $\delta(q, a)$ egy legfeljebb 1 elemű halmaz (tehát előfordulhat $\delta(q, a) = \emptyset$ is valamely q állapotra és a karakterre).

A következő tételben megmutatjuk, hogy minden nemdeterminisztikus véges automatához van vele ekvivalens determinisztikus véges automata. A bizonyítás egyben egy eljárást is megad egy véges automata determinizálására.

2. Tétel. Minden M nemdeterminisztikus véges automatához van olyan M' determinisztikus véges automata, amire $L(M) = L(M')$.

Bizonyítás: Legyen $M = (Q, \Sigma, \delta, q_0, F)$ egy tetszőleges nemdeterminisztikus véges automata. Ebből készítünk egy $M' = (Q', \Sigma, \delta', q'_0, F')$ DVA-t. A konstrukció alap gondolata, hogy az új automata párhuzamosan követi M összes lehetséges számítását, és akkor fogad el, ha M -nek legalább az egyik számítása elfogadó.

Legyen $Q' = \{R : R \subseteq Q\}$, azaz M' állapotai az M állapotainak részhalmazai. Az M' kezdő állapota az M kezdő állapotából álló egy elemű halmaz, $q'_0 = \{q_0\}$. Az elfogadó állapotok pedig azok a részhalmazok, melyekben van M -beli elfogadó állapot, azaz $F' = \{R : R \cap F \neq \emptyset\}$.

Az állapotátmenetet az M' minden $R \subseteq Q$ állapotára és minden $a \in \Sigma$ karakterre definiáljuk a következőképpen:

$$\delta'(R, a) = \bigcup_{r \in R} \delta(r, a)$$

Végül minden, a $\{q_0\}$ kezdőállapotból nem elérhető állapotot, a hozzá tartozó átmenetekkel együtt hagyjunk el M' -ből.

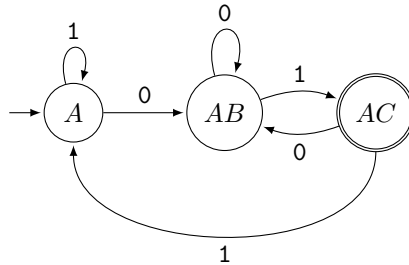
Az így kapott M' egy DVA, hiszen minden (R, a) állapot és karakter párhoz a δ' átmeneti függvény az eredeti M automata állapotainak egy halmazát, azaz M' egyetlen állapotát rendeli hozzá.

Most megmutatjuk, hogy az így leírt M' megfelel a feltételnek. Legyen $w = a_1 a_2 \cdots a_n \in \Sigma^*$ egy tetszőleges szó, és R_0, R_1, \dots, R_n az az állapotsorozat, amit ezen a szón az M' bejár. Ekkor minden $1 \leq i \leq n$ indexre és $r \in R_i$ elemre igaz, hogy a nemdeterminisztikus M automatának van olyan számítása a w szón, melynél az a_i karakter után M az r állapotba kerül. Ez azért igaz, mert egyrészt $R_0 = \{q_0\}$, és mivel $R_1 = \delta(q_0, a_1)$, ezért $i = 1$ -re igaz az állítás. Másrészt, ha már az i -nél kisebb indexekre tudjuk, akkor $R_i = \delta'(R_{i-1}, a_i)$ miatt minden $r \in R_i$ egy M -beli átmenettel megkapható egy $q \in R_{i-1}$ állapotból, ami mutatja, hogy az állítás teljesül.

A definíció szerint $w \in L(M')$ pontosan akkor teljesül, ha $R_n \cap F \neq \emptyset$, azaz ha van $r \in R_n \cap F$. Ami az előzőek szerint akkor és csak akkor lehetséges, ha a nemdeterminisztikus M -nek van elfogadó számítása, vagyis $w \in L(M)$. \square

1. Megjegyzés. A determinizált automata állapotainak száma a konstrukció alapján legfeljebb $2^{|Q|}$ lehet (ha minden állapota megmarad). Az esetek nagy részében azonban ezeknek jelentős részét elhagyjuk, mert nem elérhetőek. Egyszerűbb, és kevesebb munkával jár, ha ezeket a „felesleges állapotokat” létre sem hozzuk. Ahelyett, hogy mind a lehetséges $2^{|Q|}$ állapotot létrehoznánk a megfelelő átmenetekkel együtt, célszerű az új, $\{q_0\}$ kezdőállapotból indulva mindig csak az átmeneti függvény aktuális értékéhez szükséges állapotot felvenni. Így csak az elérhető állapotokat hozzuk létre.

4. Példa. A tétel bizonyítása szerint az előző példára adódó DVA ábrája (az állapotokban a halmazjelet elhagytuk)



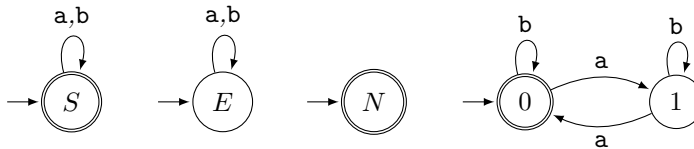
Vegyük észre, hogy itt az elméletileg lehetséges $2^3 = 8$ állapot helyett csak 3 állapota van a kapott DVA-nak.

5. Reguláris nyelvek

6. Definíció. Egy $L \subseteq \Sigma^*$ nyelvet regulárisnak hívunk, ha van olyan M véges automata, amire $L(M) = L$.

Itt, és általában is, *véges automata* alatt az eddig felsorolt változatok (és még néhány további lehetőség) bármelyikét értik, hiszen ezek ekvivalensek abban az értelemben, hogy átalakíthatók egymásba.

5. Példa. Legyen $\Sigma = \{a, b\}$. Reguláris nyelv például a Σ^* , az üres nyelv, $L = \{\varepsilon\}$, az $L = \{w \mid w\text{-ben páros sok } a \text{ betű van}\}$, amit például a következő automaták igazolnak:



Azonban sok olyan nyelv is van, amihez nincs véges automata. Itt mutatunk egy ilyen, a gyakorlaton lesz még több is Bizonyítáunk annak a formális megfogalmazásán alapul, hogy egy véges automatának a szó hosszától független, korlátos méretű memóriája van (az állapotok), ezért nem tud emlékezni a már feldolgozott karaktersorozat minden részletére.

Ha $a \in \Sigma$, és $n \geq 0$ egész szám, akkor jelölje a^n a csupa a betűből álló n hosszú sorozatot, $a^n b^n$ az n darab a betűből és utána n darab b betűből álló szót.

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

Bizonyítás: Indirekt bizonyítunk. Tegyük fel, hogy az L nyelv reguláris, azaz van olyan M determinisztikus véges automata, ami L -et fogadja el. Jelölje t az M állapotainak számát. Tekintsük a $0, 1, 2, \dots, t$ hosszú, csupa a betűből álló szavakat. Mivel csak t állapot van, ez a $t + 1$ szó között van legalább kettő, amelyek ugyanabban az állapotban érnek véget. Ha a k és az ℓ hosszú a -sorozat

is a q állapotban ér véget ($k \neq \ell$), akkor innen folytatva k darab a és ℓ darab b betűvel is elfogadó állapotba kell érjünk. Viszont ekkor az $a^k b^\ell$ szóval is elfogadó állapotba jutunk, pedig ez nincs is a megadott nyelvben. \square

6. Reguláris kifejezések

A reguláris nyelveknek egy sok helyen előforduló megadási módja a reguláris kifejezés. Ezeket, mint mindjárt látható, rekurzív definícióval írhatjuk le. A reguláris kifejezésekre többféle jelölési rendszer van, itt ezekből egyet fogunk használni. Hasonlóval találkozhatunk szövegszerkesztőkben, programozási nyelvekben és általában keresési kifejezések megadásánál is.

7. Definíció. A Σ ábécé feletti reguláris kifejezések a következők:

- \emptyset
- ε
- a minden $a \in \Sigma$ karakterre.

Továbbá, ha r_1 és r_2 reguláris kifejezés, akkor

- $r_1 + r_2$ összeg
- $r_1 r_2$ összefűzés
- r_1^* csillag

is reguláris kifejezés.

Egy összetettebb reguláris kifejezés megadásánál zárójeleket használunk annak feltüntetésére, hogyan keletkezett a kifejezés, pl. $(0+1)^*(00)$. Általános szabály, hogy ha nincs zárójel, akkor a $*$ műveletet kell először elvégezni, utána az összefűzést és végül a $+$ műveletet. Ha azonos műveletek vannak egymás után, akkor balról jobbra hajtjuk ezeket végre. Pl. az $a + bc^*(a + c) + b$ kifejezésnek egy, a definíció szerinti előállítás $r_1 = a, r_2 = b, r_3 = c, r_4 = r_3^*, r_5 = r_2 r_4, r_6 = r_1 + r_3, r_7 = r_5 r_6, r_8 = r_1 + r_7, r_9 = r_8 + r_2$.

A reguláris kifejezésben szereplő jelöléseket, műveleteket természetes módon meg lehet feleltetni nyelveknek, és ezeken végzett műveleteknek. A kapcsolódó műveletek

unió $L_1 \cup L_2 = \{x : x \in L_1 \text{ vagy } x \in L_2\}$

összefűzés $L_1 L_2 = \{x : x = y_1 y_2, \text{ ahol } y_1 \in L_1 \text{ és } y_2 \in L_2\}$

tranzitív lezárt $L^* = \{x : x = y_1 y_2 \cdots y_k, \text{ ahol } k \geq 0 \text{ és } y_i \in L\} = \{\varepsilon\} \cup L \cup LL \cup LLL \cup \cdots = \bigcup_{n \geq 0} L^n$

8. Definíció. A Σ ábécé feletti r reguláris kifejezés által leírt $L(r)$ nyelv legyen

- $L(\emptyset) = \emptyset$,
- $L(\varepsilon) = \{\varepsilon\}$.
- $L(\mathbf{a}) = \{\mathbf{a}\}$, ha $\mathbf{a} \in \Sigma$.

Továbbá, ha r_1 és r_2 reguláris kifejezés, akkor

- $L(r_1 + r_2) = L(r_1) \cup L(r_2)$,
- $L(r_1 r_2) = L(r_1)L(r_2)$,
- $L(r_1^*) = L(r_1)^*$

6. Példa. Reguláris kifejezések és az általuk leírt nyelvek:

- Ha $r = (a + b)^*$, akkor $L(r) = \{\mathbf{a}, \mathbf{b}\}^*$.
- Ha $r = 0^*10^*$, akkor $L(r) \subset \{0, 1\}^*$ a pontosan egy darab 1 karaktert tartalmazó szavakból álló nyelv.
- Ha $r = (0 + 1)^*1(0 + 1)^*$, akkor $L(r)$ a legalább egy darab 1-t tartalmazó szavakból álló nyelv.
- Ha $r = 0(0 + 1)^*0 + 1(0 + 1)^*1 + 0 + 1$, akkor $L(r) \subset \{0, 1\}^*$ azokból a legalább egy hosszú szavakból áll, amelyekben az első karakter azonos az utolsóval.

A reguláris kifejezéseknél használhatjuk az alábbi műveleti tulajdonságokat (melyeket a nyelvek megfelelő műveletei indokolnak):

- $r_1 + r_2 = r_2 + r_1$
- $r + \emptyset = \emptyset + r = r$
- $r\emptyset = \emptyset r = \emptyset$
- $r\varepsilon = \varepsilon r = r$
- $r + \varepsilon = r$ akkor és csak akkor ha $\varepsilon \in L(r)$
- általában $r_1 r_2 \neq r_2 r_1$

A reguláris kifejezés elnevezést indokolja, hogy nem nehéz egy reguláris kifejezés által leírt nyelvhez egy véges automatát megadni, azaz a nyelv reguláris nyelv. A másik irányban, adott véges automatából reguláris kifejezést felírni már lényegesen bonyolultabb, de ez is lehetséges:

4. Tétel. Az $L \subseteq \Sigma^*$ nyelv akkor és csak akkor reguláris, ha van olyan r reguláris kifejezés, amire $L = L(r)$.

Ezt a tételt itt nem bizonyítjuk.

.....