Véges automaták, reguláris nyelvek

Kiegészítő anyag az Algoritmuselmélet tárgyhoz V.

(a Rónyai–Ivanyos–Szabó: Algoritmusok könyv mellé)

Friedl Katalin
BME SZIT
friedl@cs.bme.hu

2017. február 15.

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 *betűk*nek vagy *karakterek*nek 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* $\Sigma = \{\mathtt{0}, \mathtt{1}\}$.

- *A* 0*-val kezdődő 0/1 sorozatok egy nyelvet alkotnak.*
- $L = \{\mathtt{00}, \mathtt{101}\}$ *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.*

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.
- $\delta : Q \times \Sigma \to Q$, *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 $w = a_1a_2 \cdots a_n \in \Sigma^*$ szón a következőképpen írható le: Az automatát a q_0 állapotból indítjuk, ahonnan az először a $\delta(q_0, a_1)$ állapotba lép. Ha ez az állapot az r_1, akkor a második lépésben a $\delta(r_1, a_2)$ állapotba kerül, és így tovább, amíg az a_n karaktert is feldolgozza. Legyen r_n az az állapot, ahova az a_n hatására lép.

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* $\Sigma = \{\mathtt{0}, \mathtt{1}\}$. *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:

$$\delta(A, \mathtt{0}) = B$$
$$\delta(A, \mathtt{1}) = A$$
$$\delta(B, \mathtt{0}) = B$$
$$\delta(B, \mathtt{1}) = A$$

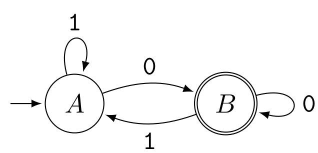
Vegyük észre, hogy ez a δ függvény olyan, hogy mindegy, hogy az automata éppen melyik állapotban van: ha `0` karakter következik, akkor az automata a B állapotba, míg ha `1` karakter jön, akkor az A állapotba kerül. Mivel B az egyetlen elfogadó állapot, ez az automata valóban pontosan azokat a szavakat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

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

	`0`	`1`
A	B	A
B	B	A

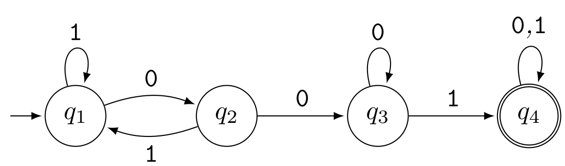
A DVA-t irányított gráffal is ábrázolhatjuk. A gráf csúcsai az állapotoknak felelnek meg, az állapotátmeneti függvényt az élek segítségével adjuk meg. A $\delta(q, \mathtt{a}) = q'$ átmenetnek egy, a q állapotnak megfelelő csúcsból a q' állapotnak megfelelő csúcsba menő irányított él felel meg, amihez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

Az előző automata ebben a formában:



1. Megjegyzés. *Általában feltehető, 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 elhagyhatjuk 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` részszót. 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 $L_1 \cup L_2 \cup L_3 \cup L_4 = \{\texttt{0}, \texttt{1}\}^*$ (ha ez nem teljesülne, a sejtésünk biztosan nem lehetne jó, minden szónak benne kell lennie a halmazok közül pontosan egyben).

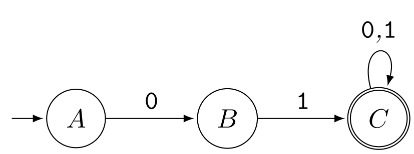
Azt, hogy a sejtés igaz, a szavak hossza szerinti teljes indukcióval lehet belátni. Rövid, pl. 0, 1, 2 hosszú szavakra könnyű látni, hogy valóban a megfelelő L_i-be tartoznak. Tegyük fel, hogy a legfeljebb k hosszú szavakra a sejtésünk igaz ($k \geq 2$). Legyen $w = a_1 a_2 \cdots a_k a_{k+1}$. Ha az utolsó előtti karakter 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 $a_k = \texttt{1}$. Így biztos, hogy w-ben sincs `00`. Amennyiben az utolsó karakter $a_{k+1} = \texttt{1}$, 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 $a_{k+1} = \texttt{0}$, 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. □

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` betűt olvas, amire a $(q, \texttt{a})$ helyen δ nincs definiálva, akkor *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* ($\Sigma = \{\texttt{0}, \texttt{1}\}$)*.*



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

Bizonyítás: Az ötlet, hogy kiegészítjük M-et egy új t állapottal és minden hiányzó átmenetet ide irányítunk. Továbbá minden $a \in \Sigma$ esetén legyen $\delta'(t, a) = t$. Természetesen t nem elfogadó állapot. Könnyen látható, hogy $L(M) = L(M')$. $\square$

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 $\delta(q, a) \subseteq Q$.*

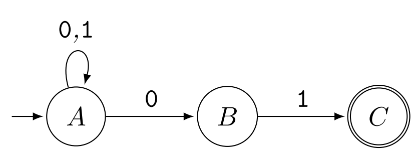
A nemdeterminisztikus véges automata *működése* egy adott $w = a_1 a_2 \dots a_n \in \Sigma^*$ szón a következőképpen írható le: Az automatát a q_0 állapotból indítjuk, ahonnan először valamelyik $r_1 \in \delta(q_0, a_1)$ állapotba lép, a második lépésben valamelyik $r_2 \in \delta(r_1, a_2)$ állapotba 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 kell elképzelni, hogy ugyanazon a bemeneten többféle lefutás is van (ezért hívják nemdeterminisztikusnak).

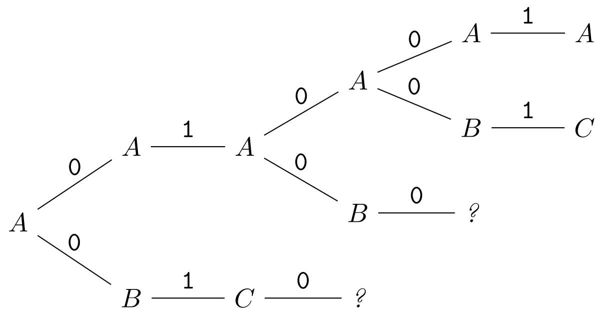
5. Definíció. *Az automata akkor* fogadja el *a szót, ha van olyan számítási út (lefutás), ami elfogadó állapottal ér véget.*

Egy adott bemeneten a számítások leírhatók egy *számítási fá*val, amiben az elágazások a lehetséges következő állapotoknak felelnek meg. Az NVA akkor fogad el, ha van olyan ág, ami mentén a számítás nem akadt 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. Persze nem nehéz egy determinisztikus automatát felrajzolni a megadott nyelvre.

Vegyük észre, hogy ez a definíció a hiányos automatát is magában foglalja, hiszen az állapotátmeneti függvény értékére megengedett az üres halmaz is, azaz lehet $\delta(q, \mathtt{a}) = \emptyset$ valamely q állapotra és $\mathtt{a}$ karakterre.

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

Bizonyítás: Legyen $M = (Q, \Sigma, \delta, q_0, F)$ a nemdeterminisztikus véges automata. Ebből készítünk egy $M' = (Q', \Sigma, \delta', q_0', F')$ DVA-t. A konstrukció alapgondolata, 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 állapotátmenet az M' minden $R \subseteq Q$ állapotára és $a \in \Sigma$ karakterre definiáljuk a következőképpen:

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

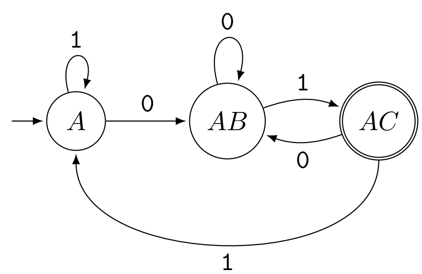
Ez az M' egy DVA, hiszen minden (R, a) állapot és karakter párhoz a δ' átmeneti függvény M' egy állapotát rendeli hozzá. Most megmutatjuk, hogy az így leírt M' megfelel a feltételnek. Legyen $w = a_1 a_2 \cdots a_n \in \Sigma^*$ egy tetszőleges szó, és $R_0, R_1, \ldots, R_n$ az az állapotsorozat, amit 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 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 $w \in L(M)$. $\square$

2. Megjegyzés. *A determinizált automata állapotainak száma a konstrukció alapján exponenciális, $2^{|Q|}$, de sokszor ezeknek az állapotoknak csak kis része érhető el a kezdőállapotból. Amikor a determinizálást végrehajtjuk, akkor elég csak ezekkel foglalkozni. Ahelyett, hogy a végén töröljük az el nem érhető állapotokat, célszerű az új kezdőállapotból indulva mindig csak az átmeneti függvény aktuális értékéhez szükséges állapotokat 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)*



5. Reguláris nyelvek

6. Definíció. *Egy $L \subseteq \Sigma^*$ nyelvet* reguláris*nak 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 $\Sigma = \{\mathtt{a}, \mathtt{b}\}$. Reguláris nyelv például a Σ^*, az üres nyelv, $L = \{\varepsilon\}$, az $L = \{w \mid w$-ben páros sok $\mathtt{a}$ betű van $\}$.*

Ha $\mathtt{a} \in \Sigma$, és $n \geq 0$ egész szám, akkor jelölje $\mathtt{a}^n$ a csupa $\mathtt{a}$ betűből álló n hosszú sorozatot.

3. Tétel. *Az $L = \{\mathtt{a}^n\mathtt{b}^n \mid n \geq 0\}$ 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 $0, 1, 2, \ldots, t$ hosszú, csupa $\mathtt{a}$ betűből álló szavakat. Mivel csak t állapot van, a $t + 1$ szó között van legalább kettő, amelyek ugyanabban az állapotban érnek véget. Ha a k és az ℓ hosszú $\mathtt{a}$-sorozat is a q állapotban ér véget ($k \neq \ell$), akkor innen folytatva k darab és ℓ darab $\mathtt{b}$ betűvel is elfogadó állapotba kell érjünk. Viszont ekkor az $\mathtt{a}^k\mathtt{b}^\ell$ szóval is elfogadó állapotba jutunk, pedig ez nincs is a megadott nyelvben. □

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. *Egy* Σ *ábécé feletti* reguláris kifejezés*ek a következők:*

- $\emptyset$
- ε
- $\mathtt{a}$ *minden* $\mathtt{a} \in \Sigma$ *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. $(\mathtt{0}+\mathtt{1})^*(\mathtt{00})$. 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 őket végre. Pl. az $\mathtt{a}+\mathtt{bc}^*\mathtt{a}+\mathtt{b}$ kifejezés zárójelekkel ellátva így néz ki: $(\mathtt{a} + ((\mathtt{b}((\mathtt{c})^*))\mathtt{a})) + \mathtt{b}$, azaz a definíció szerinti előállítása $r_1 = \mathtt{c}$, $r_2 = r_1^*$, $r_3 = \mathtt{b}r_2$, $r_4 = r_3\mathtt{a}$, $r_5 = \mathtt{a} + r_4$, $r_6 = r_5 + \mathtt{b}$.

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

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

összefűzés $L_1L_2 = \{x : x = y_1y_2, \text{ ahol } y_1 \in L_1 \text{ és } y_2 \in L_2\}$

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

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

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

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

A műveletek néhány fontos tulajdonsága (amiket 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 $r_1 r_2 \neq r_2 r_1$

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.