

Automaták és Formális nyelvek

(2004)

by Data parancsnok
Based on (not so much auditted) lectures of
Dr. Radelecki Sándor

Determinisztikus véges automata (DFA – Deterministic Final Automata)

Elmélet:

A DFA véges számú állapotból és egy tranzíciós diagramból áll, melynek szakaszait ún. input szimbólumok jellemzik, amelyek mind ugyanabból a véges Σ ábécéből valóak. Minden egyes input szimbólum pontosan egy választ vált ki minden egyes állapotból, azaz egy állapothoz és egy inputszimbólumhoz pontosan egy nyíl tartozik. (Ezért nevezik determinisztikusnak.)

$A \ Q \times \Sigma = \{(q, a) \mid q \in Q, a \in \Sigma\}$ automata $\delta : Q \times \Sigma \rightarrow Q$ konfigurációinak halmaza.

A véges determinisztikus automata matematikai modellje:

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

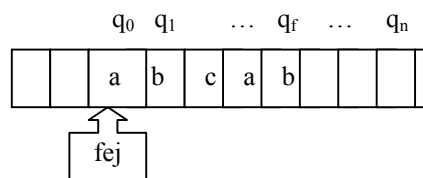
Q : az állapotok (pl. q_1, q_2) véges halmaza

Σ : véges input ábécé (pl. 0, 1)

δ : tranzíciós függvény, ami $\delta : Q \times \Sigma \rightarrow Q$ alakú

q_0 : kezdőállapot ($q_0 \in Q$)

F : a végállapotok nem üres, véges halmaza ($F \subseteq Q$)



A fej csak q_n -ig mehet, de maga az input szalag végtelen az egyik irányban. Az éppen beolvasott jel az $a \in \Sigma$. A jel beolvasása után a fej a q állapot függvényében jobbra mozdul a szalagon.

Ha a w szó beolvasása után a fej a $q_f \in F$ végállapotba jut, akkor az automata elfogadja a w szót.

Pl. $w = abcab$.

A DFA által elfogadott szavak:

Minden olyan $w = x_1x_2 \dots x_n$ szó, ahol $x_i \in \Sigma$, ha a tranzíciós diagramjában az x_i jeleknek megfelelő tranzíciós útvonal q_0 -ból egy végállapotba vezet.

Az automata által elfogadott szavak összessége az **automata nyelve**. Jelölése: $L(A)$

A_1 és A_2 **ekvivalens automaták**, ha $L(A_1) = L(A_2)$, azaz mindketten ugyanazokat a szavakat fogadják el.

Egy n állapotú DFA által elfogadott nyelv **nem üres**, ha van olyan $|w| < n$ hosszúságú szó, amit elfogad.

Egy n állapotú DFA által elfogadott nyelv **végtelen**, ha van olyan $n \leq |w| \leq 2n$ hosszúságú szó, amit elfogad.

Betű: Egy Σ véges ábécé elemei. Pl. Ha $\Sigma = \{a, b, c, \dots, z\}$, akkor $a, b, c, \dots, z$ betűk.

Szó: Betűk egymás után írásával (konkatenációjával) szavakat képezhetünk. Pl. $w = abcaabbca$

Üres szó: Olyan szó, melyben a betűk száma 0. Jelölése: ε

Szavak halmazai:

- az összes, nem üres szó halmaza. Jelölése: Σ^+
- az összes szó halmaza, beleértve az üres szavakat is (ezen halmazon értelmezzük, mint lineáris műveletet, a szavak összekapcsolását is; pl.: $wvxyw$, ahol $w \in \Sigma^*$). Jelölése: Σ^*
- $\Sigma^* = \Sigma^+ \cup \{\varepsilon\}$

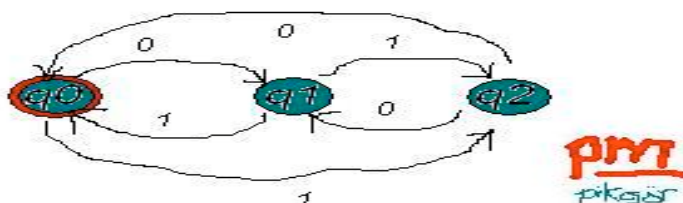
Szómonoid: A $(\Sigma^*, \cdot)$ nem kommutatív monoid, melynek semleges eleme az ε , a $\cdot$ művelet pedig a konkatenáció. Bevezetjük még az alábbi jelöléseket:

- $a^+ = \{a, a^2, a^3, \dots, a^n, \dots\} = \{a, aa, aaa, aaaa, \dots\}$
- $a^* = \{\varepsilon, a, a^2, a^3, \dots, a^n, \dots\} = \{\varepsilon, a, aa, aaa, aaaa, \dots\}$

Bármely $L \subseteq \Sigma^*$ részhalmazt **Σ feletti nyelvnek** nevezünk $\Rightarrow L(A) \subseteq \Sigma^*$

Működését **tranzíciós fával** szemléltetjük.

Példa ($L = (10+01)^*$ nyelvet előállító DFA):



Nem determinisztikus véges automata (NFA – Nondeterministic Finite Automata)

Elmélet:

Az NFA véges számú állapotból és egy tranzíciós diagramból áll, melynek szakaszait ún. input szimbólumok jellemzik, amelyek mind ugyanabból a véges Σ ábécéből valók. Bármely input szimbólum több választ is kiválthat bármely állapotból, de akár egyet sem, azaz egy állapothoz és egy inputszimbólumhoz 0 vagy több nyíl tartozhat. (Ezért nevezik nem determinisztikusnak.)

Jellemezhető az $N = (Q, \Sigma, \delta, q_0, F)$ modellel is.

Ahol minden jel ugyanazt jelenti, mint a DFA-nál, kivéve δ -t.

$\delta(q_i, a) \rightarrow \{q_{j1}, q_{j2}, \dots, q_{jn}\}$, tehát $\delta : Q \times \Sigma \rightarrow P(Q)$ alakú, ahol $P(Q)$ Q hatványhalmaza (ami 2^n számosságú). Magyarul a (q_i, a) párnak állapotok halmazát felelteti meg, ami lehet üres is.

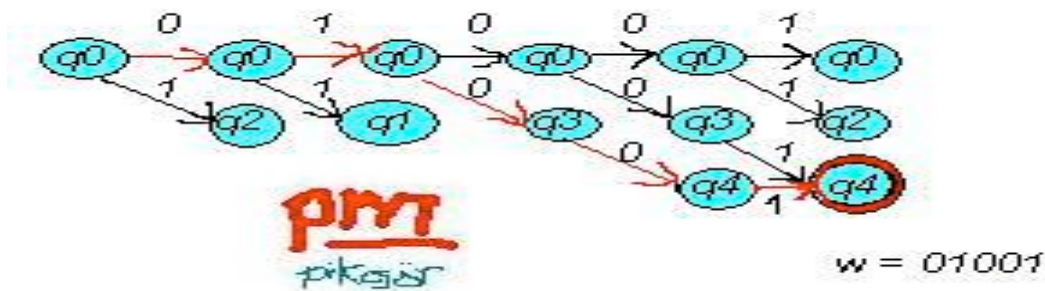
Az NFA által elfogadott szavak:

Minden olyan $w = x_1x_2\dots x_n$ szó, ahol $x_i \in \Sigma$, ha a tranzíciós diagrammában az x_i jeleknek megfelelő tranzíciós útvonal q_0 -ból egy olyan halmazba vezet, ami legalább egy F -beli elemet tartalmaz (elemei közül legalább egy végállapot).

Az NFA által elfogadott szavak összessége az **automata nyelve**. Jelölése: $L(N)$.

N_1 és N_2 **ekvivalens automaták**, ha $L(N_1) = L(N_2)$, azaz mindketten ugyanazokat a szavakat fogadják el.

Példa (a $w = 10110$ szót elfogadó NFA):



ϵ -os NFA: olyan NFA, melynek ábécéje $\Sigma \cup \{\epsilon\}$.

Az NFA és a DFA ekvivalenciája

Elmélet: Az A determinisztikus és az N nondeterminisztikus automata által elfogadott nyelv egy és ugyanaz, vagyis $L(A) = L(N)$.

Az NFA tranzíciós sorozata alapján a q_0 kezdőállapotból a szó hatására egy olyan halmazba jut, melyből egy biztos hogy végállapota a DFA-nak is, hiszen másként nem fogadná el a szót.

Ha DFA végállapota az NFA végállapot-halmazának egyetlen végállapota, akkor ez a DFA egyben NFA is.

Minden egyes NFA nyelvét tehát egy DFA is előállítja. Mindkét automata-típus ugyanazt a nyelvcsaládot állítja elő.

Minden ϵ -os NFA-hoz szerkeszthető egy vele ekvivalens NFA.

Gyakorlat:

1. Szerkesztünk az adott nyelvhez egy NFA-t.
2. Az azonos útvonalakat összevonjuk olyképpen, hogy a célállapotokból egy állapotot készítünk.
4. Ez már DFA is.

Reguláris kifejezések és nyelvek

(Regular expressions and languages)

Elmélet:

Reguláris műveletek:

- **egyesítés (unió):** $L_1 \cup L_2 = \{w \in \Sigma^* \mid w \in L_1 \text{ vagy } w \in L_2\}$
- **szorzat:** $L_1 \cdot L_2 = \{w \in \Sigma^* \mid w \in L_1 \text{ és } w \in L_2\}$
- **hatványozás:** $L^n = L \cdot L \cdot \dots \cdot L$ (vagyis n-szer L); $L^1 = L$; $L^0 = \{\varepsilon\}$
- **Kleen-féle pozitív lezárás:** $L^+ = L \cup L^2 \cup L^3 \cup \dots \cup L^n \cup \dots$
- **Kleen-féle lezárás:** $L^* = \{\varepsilon\} \cup L^+ = \{\varepsilon\} \cup L \cup L^2 \cup L^3 \cup \dots \cup L^n \cup \dots$

A Σ abc feletti reguláris kifejezések:

- $\emptyset$ (az üres nyelvet jelöli)
- ε (az $\{\varepsilon\}$ nyelvet jelöli)
- $\forall a \in \Sigma$ (az $\{a\}$ nyelvet jelöli)
- $r, s \in \Sigma$ (az R ill. S nyelvet jelölik)
- $r+s$ (az $R \cup S$ nyelvet jelöli)
- r^* (az R^* nyelvet jelöli)
- bármely olyan kifejezés, ami a fentiek véges számú ismétlésével megkapható

Példák:

- 01^*+0 nyelve az $L = \{0\} \cdot \{1\}^* \cup \{0\}$
- $(0+1)^*$ nyelve az összes bináris szót elfogadó nyelv
- $(0+1)^*00(0+1)^*$ nyelve az összes olyan bináris szó, mely legalább két egymást követő 0-t tartalmaz.

Reguláris nyelvek: minden olyan nyelv, ami reguláris kifejezésekkel megadható, vagyis az $\{a\}$, az $\{\varepsilon\}$ az üres nyelv és azok a nyelvek, amik ezekből az elemi nyelvekből véges számú egyesítés (unió) vagy szorzás vagy Kleen-féle lezárások műveletével megkaphatóak.

Tulajdonságok:

- minden DFA által elfogadott nyelv reguláris
- egy nyelv csak akkor reguláris, ha van olyan DFA, ami őt elfogadja
- minden reguláris nyelvhez szerkeszthető egy ε -os NFA, ami őt elfogadja

A reguláris nyelvek zártak

- az unió ($L_1 \cup L_2$)
- a szorzás ($L_1 \cdot L_2$), a hatványozás (L^n)
- a Kleen-féle lezárások (L^+ és L^*)
- halmazkülönbség ($L_1 \setminus L_2$)
- komplementerképzés ($\Sigma^* \setminus L$)
- metszet ($L_1 \cap L_2$)
- szimmetrikus differencia (diszjunkt unió) ($L_1 \Delta L_2$)
- homorfizmus ($h(L)$)
- behelyettesítés ($s(L)$)
- inverz homomorfizmus ($h^{-1}(L)$)

műveletére, azaz a zárójelben látható nyelvek mind regulárisak, ha L ill. L_1 és L_2 regulárisak.

A **behelyettesítés** olyan leképezés, melynek függvénye $s : \Sigma_1^* \rightarrow \Sigma_2^*$. Ennek hatására Σ_1 feletti szavakat reguláris kifejezésekre cseréli, úgy hogy $s(\varepsilon_1) = s(\varepsilon_2)$ illetve $s(xy) = s(x)s(y)$, ahol $x, y \in \Sigma_1^*$. A behelyettesítést a második összefüggés alapján elég csak a Σ_1 betűire megadni.

A $h : \Sigma_1^* \rightarrow \Sigma_2^*$ **homomorfizmusok** olyan behelyettesítésnek tekinthetők, ahol a reguláris kifejezések csak szavak lehetnek, azaz betűket szavakkal helyettesítünk. Ennélfogva tehát egy reguláris nyelv homomorf képe is reguláris.

Ha $h(L)$ homomorf képe L-nek, akkor $h^{-1}(L)$ az L nyelv **inverz homomorf képe**. $h^{-1}(L)$ lehet üres is.

Pumpáló lemma (Pumping Lemma)

Elmélet:

Minden reguláris nyelvhez létezik egy olyan n_L természetes szám, hogy ha $w \in L$ és $|w| \geq n_L$ akkor w felbontható egy $w = uvz$ szorzatra, amelyben $1 \leq |uv| \leq n_L$ és $w^{(i)} = uv^iz \in L$ teljesül ($\forall i \geq 0$ egész számra).

A lemma alapján megállapíthatjuk egy nyelvről, hogy az nem reguláris.

Példa:

$L = \{0^n 1^n \mid n \geq 1\}$ nyelv reguláris-e?

- 1.) Tfh. L reguláris.
- 2.) A pumpáló lemma alapján létezik hozzá egy $n_L \in \mathbb{N}$ küszöbszám, hogy $w \in L$ szó esetében ($|w| \geq n_L$) felírható a $w = 0^{n_L} 1^{n_L}$ alakban.
- 3.) Ekkor $|w| = 2n_L \geq n_L$, tehát a lemma alapján felírható $w = uvz$ alakban, ahol $1 \leq |w| \leq n_L$ és $w^{(i)} = uv^iz \in L$ ($\forall i \geq 0$ egész számra)
- 4.) Ha z -ben csak 1-es lenne, akkor minden 0-nak u -ban kell lennie és v -ben legalább egy darab egyesnek lennie kell. Ebből adódik, hogy $u=0$, és $w^{(0)} = uz$ esetén n darab 0 és legalább eggyel kevesebb 1-es van, ami szerint a $w^{(0)} \notin L$, hiszen az 1-esek és a 0-k számának is azonosan n -nek kell lennie.
- 5.) Ha v -ben csak 0 lenne, akkor minden 1-esnek z -ben kellene lennie és u -ban legalább egy 0-nak lennie kellene. A 4.) alapján igazolható, hogy ez nem lehetséges, tehát ellentmondásra jutunk.
- 6.) A konklúzió az, hogy **a fenti L nyelv nem reguláris.**

Automaták minimalizálása

Elmélet:

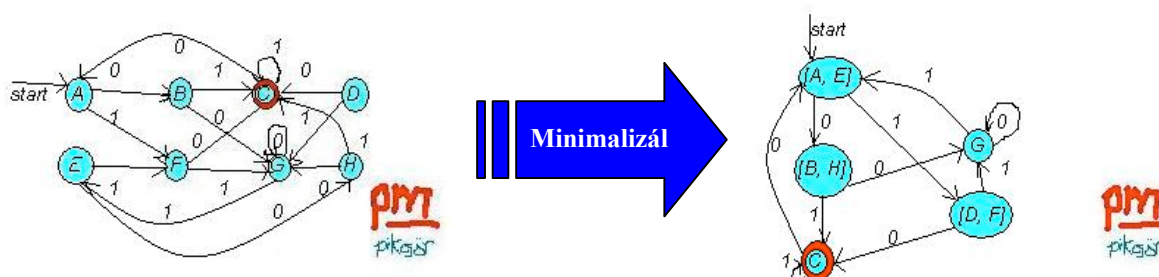
A p és q **ekvivalens állapotok**, ha tranzíciós diagrammján van két olyan egyforma út (szó), amelyen két különböző csúcsból (állapotból) ugyanabba a csúcsba (végállapotba) jutnánk.

Jelölés: $p \equiv q$

Két állapotot **megkülönböztethetőnek** nevezünk, ha nem ekvivalensek.

Egy automata **minimális**, ha nincs benne ekvivalens állapot.

Gyakorlat (példán keresztül):



Felrajzolunk egy táblázatot, s jelöljük a minimális ($\equiv$), illetve a megkülönböztethető ($\times$) állapotokat:

A								
B	$\times$							
C	$\times$	$\times$						
D	$\times$	$\times$	$\times$					
E	$\equiv$	$\times$	$\times$	$\times$				
F	$\times$	$\times$	$\times$	$\equiv$	$\times$			
G	$\times$	$\times$	$\times$	$\times$	$\times$	$\times$		
H	$\times$	$\equiv$	$\times$	$\times$	$\times$	$\times$	$\times$	
	A	B	C	D	E	F	G	H

Ekvivalensek tehát: $A \equiv E$; $B \equiv H$; $D \equiv F$. Ezeket az állapotokat összevonjuk, és így már minimális az automata.

Környezetfüggetlen nyelvtanok és nyelvek

Elmélet: Egy olyan $G = (V, T, P, S)$ négyes **környezetfüggetlen nyelvtannak** számít, amelyben V a változókat, T a terminálokat, S a kezdő (gyökér) szimbólumot és P a produkciós szabályokat jelöli. Angolul *Context Free Grammar (CFG)* elnevezésűek.

Sajátos jelölést használunk az ABC különböző betűire:

- $A, B, C, \dots, S$ változókat jelölnek
- $a, b, c, \dots, t$ terminálokat jelölnek
- u, v, w, x, y, z terminálokból kapott szvakt jelölnek, vagyis T elemeit
- $\alpha, \beta, \gamma, \dots, \omega$ mondatokat jelölnek, azaz olyan jelsorozatok, amelyek mind terminálokat, mind változókat is tartalmazhatnak, vagyis $(V \cup T)^*$
- X, Y, Z ismeretlenek jelölésére tartjuk őket fenn (lehetnek terminálok vagy változók)

A tömörítés érdekében, ha egy változóból valamilyen produkciós szabály alapján több mondatot is képezhetünk, akkor ezeket a változatokat $|$ (vagy) jellel elválasztva soroljuk fel a nyíl után.

Példa:

$V = \{A, \bar{A}, M, F\}$; $T = \{\text{fiú, kicsi, alszik}\}$

Itt S a mondat, A az alany, $\bar{A}$ az állítmány, M melléknév és F a főnév szimbolizálására használt változó.

$$P: \begin{cases} S \rightarrow A\bar{A} \\ A \rightarrow MF \\ F \rightarrow \text{fiú} \\ M \rightarrow \text{kicsi} \\ \bar{A} \rightarrow \text{alszik} \end{cases}$$

Ennek alapján a mondatunk így szól: „Kicsi fiú alszik.”

Egy G nyelvtanhoz tartozó **nyelven** a nyelvtan szabályai szerint előállított szavak összességét értjük.

Jelölésük: $L(G)$

G_1 és G_2 nyelvtan **ekvivalens**, ha $L(G_1) = L(G_2)$.

Környezetfüggetlen nyelvnek (*CFL – Context Free Language*) nevezünk minden környezetfüggetlen nyelvtannal előállítható nyelvet.

Minden reguláris nyelv környezetfüggetlen nyelv.

Bar-Hile lemma: Minden L környezetfüggetlen nyelvhez létezik egy olyan $n \in \mathbb{N}$ szám, hogy ha a $z \in \Sigma^*$ szó az L nyelvhez tartozik és $|z| \geq n$, akkor z felírható $z = uvwxy$ alakban az alábbi feltételekkel:

- $|vx| \geq 1$
- $|vwx| \leq n$
- $\forall i \geq 0$ -ra $uv^iwx^i y \in L$

R : reguláris nyelv

L, L_1, L_2 : környezetfüggetlen nyelvek

A CFG nyelvek zártak:

- unió: $L_1 \cup L_2$
- szorzás (konkatenáció): $L_1 \cdot L_2$
- hatványozás: L^n
- homomorfizmus: $h(L)$
- behelyettesítés: $L(G')$
- a reguláris nyelvekkel alkotott metszetre: $L \cap R$

A CFG nyelvek nem zártak:

- metszetképzés: $L_1 \cap L_2$
- komplementerképzés
- halmazkivonásra: $L_1 \setminus L_2$

Példák (L egy CFG nyelv):

Homomorfizmus:

$L = \{0^n 1^n \mid n \geq 1\}$; $w = 0^n 1^n$

$h(0) = a$

$h(1) = bb = b^2$

$h(w) = h(0)^n h(1)^n = a^n b^{2n}$

$h(L) = \{a^n b^{2n} \mid n \geq 1\}$ - ez is CFG nyelv

Behelyettesítés:

$L = \{a^i b^i \mid i \geq 0\}$

$L_a = \{0^n 1^n \mid n \geq 2\}$

$L_b = \{wwR \mid w \in (0+2)^*\}$

$S \rightarrow aSb \mid \varepsilon$

$S_a \rightarrow 0S_a 1 \mid 01$

$S_b \rightarrow 0S_b 0 \mid 2S_b 2 \mid \varepsilon$

$S \rightarrow S_a S_b \mid \varepsilon$

$\left. \begin{array}{l} L(G) \\ L(G') \end{array} \right\} L(G')$

Környezetfüggetlen nyelvtanok egyszerűsítése

Elmélet: Egy $G = \{V, T, P, S\}$ környezetfüggetlen nyelvtanban X szimbólum **nem felesleges szimbólum**, ha vagy egy előállított szóban szereplő terminál vagy részt vesz egy $S \Rightarrow^* \alpha X \beta \Rightarrow^* w$ ($w \in T^*$) derivációs sorozatban. Ellenkező esetben X **felesleges szimbólum**.

Minden nem üres környezetfüggetlen nyelv egy olyan környezetfüggetlen nyelvtannal származtatható, ami nem tartalmaz felesleges szimbólumokat.

- 1.) Először csak azokat a szimbólumokat tartjuk meg, amelyekre létezik egy olyan $w \in T^*$ és egy olyan deriváció, hogy $A \Rightarrow^* w$. (Ezen belül csak azokat a terminálokat, amelyek részt vesznek valamely előállított szóban.) A többi szimbólumot eltávolítjuk mindazokkal a produkciós szabályokkal együtt, amelyekben „meddő” szimbólumok valamelyike szerepel, s egy újabb $G' = (V', T', P', S')$ nyelvtant kapunk. Ha $S \notin V'$, akkor S -t is bele vesszük V' -be.
- 2.) V' -ből megtartjuk azokat az A változókat, amelyekre létezik valamilyen $S \Rightarrow^* \alpha A \beta$ deriváció, a többi szimbólumot eltávolítjuk a hozzájuk tartozó produkciós szabályokkal együtt. Eredmény: egy új környezetfüggetlen $G'' = (V'', T'', P'', S)$ nyelv.

Mivel $L(G) = L(G')$ és $L(G') = L(G'') \Rightarrow L(G) = L(G'')$.

Az 1. lépés azt jelenti, hogy

- ha $A \rightarrow w \in T^*$ produkciós szabály P -ben, vagy
- ha $A \rightarrow x_1 x_2 x_3 \dots x_n$ egy olyan produkciós szabály P -ből, hogy minden x_i ($i=1, \dots, n$) terminál vagy V' -beli,

akkor A -nak V' -be kell kerülnie.

A 2. lépés jelentése az, hogy ha valamelyik A változó nem érhető el az S gyökérből, akkor azokat ki kell venni.

Egy nem üres ($\epsilon \notin L$) környezetfüggetlen nyelvhez ($L = L(G)$) létezik egy L -t előállító $G^* = (V^*, T, P^*, S)$ nyelvtan, ami sem felesleges szimbólumokat, sem $A \rightarrow \epsilon$, sem $A \rightarrow B$ alakú produkciós szabályokat nem tartalmaz.

Gyakorlat:

- haszontalan nagybetűk azok, amik egyik nagybetűből sem elérhetőek;
- haszontalan kisbetűk azok, amelyek egyik nagybetűből sem elérhetőek;
- haszontalan komponensek, amik ezek bármilyen kombinációját tartalmazzák ill haszontalan nagybetűből érhetőek el.

Példa:

$S \rightarrow aAS \mid BC \mid a$
 $A \rightarrow SbA \mid aD \mid ba$
 $B \rightarrow b$
 $C \rightarrow a \mid b$
 $E \rightarrow bD \mid a$

Az egyszerűsített nyelvtan:

$S \rightarrow aAS \mid BC \mid a$
 $A \rightarrow SbA \mid ba$
 $B \rightarrow b$
 $C \rightarrow a \mid b$

Környezetérzékeny (környezetfüggő) nyelvtan (CSG - Context Sensitive Grammar): olyan nyelvtan, ami nem környezetfüggetlen. Voltaképpen az $\alpha A \beta \rightarrow \alpha \gamma \beta$ produkciós szabályú nyelvtan (ahol $A \in V$; $\alpha, \beta, \gamma \in V \cup T$), amilyen a nem determinisztikus Turing gép is, például.

Normálformák

Chomsky-féle normálforma

Elmélet: Minden környezetfüggetlen, ε -t nem tartalmazó L nyelv előállítható egy olyan G nyelvtannal, ami csak $A \rightarrow BC$ és $A \rightarrow a$ alakú produkciós szabályokat tartalmaz, ahol $A, B, C \in V$, $a \in T$.

Gyakorlat: Úgy kell átalakítani a nyelvtant, hogy az csak olyan szimbólumkombinációkat tartalmazzon, ahol

- vagy 2db nagybetű áll
- vagy 1db kisbetű áll
- Ha az új szimbólumból két azonos nagybetűt produkálunk, akkor az új szimbólum alsó indexébe írjuk be ezt a nagybetűt (pl. $D_A \rightarrow AA$)!
- Ha az új szimbólumból egy kisbetűt produkálunk, akkor az alsó indexbe ez a kisbetű kerül (pl. $C_a \rightarrow a$).

Példa:

$S \rightarrow AAS \mid aB \mid bAS$

$A \rightarrow ab$

$B \rightarrow b$

Ezt átalakítjuk Chomsky-féle normálformába:

$S \rightarrow D_AS \mid C_aB \mid XS$

$A \rightarrow C_aB$

$B \rightarrow b$

$C_a \rightarrow a$

$D_A \rightarrow AA$

$X \rightarrow BA$

Greibach-féle normálforma

Elmélet: Minden környezetfüggetlen, ε -t nem tartalmazó L nyelv előállítható egy olyan G nyelvtannal, ami csak $A \rightarrow a\alpha$ alakú produkciós szabályokat tartalmaz, ahol $A \in V$, $a \in T$, $\alpha \in (V \cup T)^*$.

Gyakorlat: Úgy kell átalakítani a nyelvtant, hogy az mindig kisbetűvel kezdődjön.

Példa:

$S \rightarrow ABS \mid BSA$

$A \rightarrow ASb \mid a$

$B \rightarrow b$

Ezt átalakítjuk Greibach-féle normálformába:

$S \rightarrow aBS \mid bSA$

$A \rightarrow aSb \mid a$

$B \rightarrow b$

Tükörszavak (polindrómák)

Elmélet: Olysn szavak, amiknek betűit fordított sorrendben leírva megkapjuk az eredeti szót (pl. kék, sas, abba, görög, korarasok)

Bináris polindrómák:

- 0, 1, ϵ
- ha w polindróma, akkor $0w0$ és $1w1$ is polindróma
- semmi egyéb nem polindróma
- az összes polindróma tehát felírható az $S \rightarrow 1S1 \mid 0S0 \mid 1 \mid 0 \mid \epsilon$ környezetfüggetlen nyelvtannal.

A w tükörképét általában w^R -rel jelöljük, így $w = w^R$.

Példák:

1.) $L = \{wcw^R \mid w \in (0+1)^*\}$ nyelvet írjuk fel CFG nyelvtan segítségével:
 $w = w^R$

$S \rightarrow 0S0 \mid 1S1 \mid c$

2.) $L = \{ww^R \mid w \in (0+1)^*\}$ nyelvet írjuk fel CFG nyelvtan segítségével:
 $w = w^R$

$S \rightarrow 00S00 \mid 11S11 \mid 01S10 \mid 10S01 \mid \epsilon$ (páros hosszúságú tükörszavakat elfogadó nyelv)

Derivációs fa

Egy $w \in T^*$ szó előállítását **derivációs fa** segítségével szemléltetünk. A fa szintjei egy-egy átalakítási lépésnek felelnek meg. Az átalakítás történhet balról-jobbra, jobbról-balra vagy e kettő kombinációja alapján; a fába a csúcsokat ennek megfelelő sorrend szerint írjuk ki.

Pl.: Állítsuk elő a $w = ababbaa$ szót az alábbi nyelvtan segítségével:

$S \rightarrow aAS \mid BC \mid a$

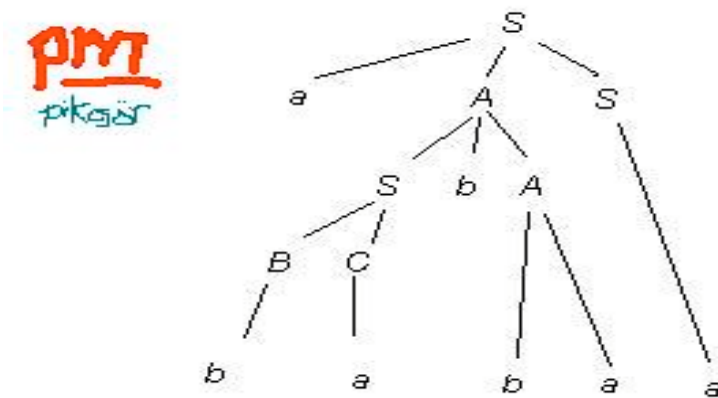
$A \rightarrow SbA \mid ba$

$B \rightarrow b$

$C \rightarrow a \mid b$

$S \rightarrow aAS \rightarrow aSbAS \rightarrow aBCbAS \rightarrow ababbaa$

A derivációs fa (balról-jobbra rendezve):



Az átalakítás végén, az ágak végpontjait összekötve, megkapjuk a szót. Ez a **fa alja**.

Veremautomaták

Elmélet:

A veremautomata matematikai modellje:

$$V = (Q, \Sigma, \Gamma, \delta, q_0, z_0, F) \quad \text{ahol}$$

Q : az állapotok (pl.: q_1, q_2) véges halmaza

Σ : inputábécé (pl.: $0, 1, \varepsilon$)

Γ : veremábécé (pl.: R, B)

δ : tranzíciós függvény; $\delta : (Q \times \Sigma \cup \{\varepsilon\} \times \Gamma) \rightarrow P(Q \times \Gamma^*)$

q_0 : kezdőállapot ($q_0 \in Q$)

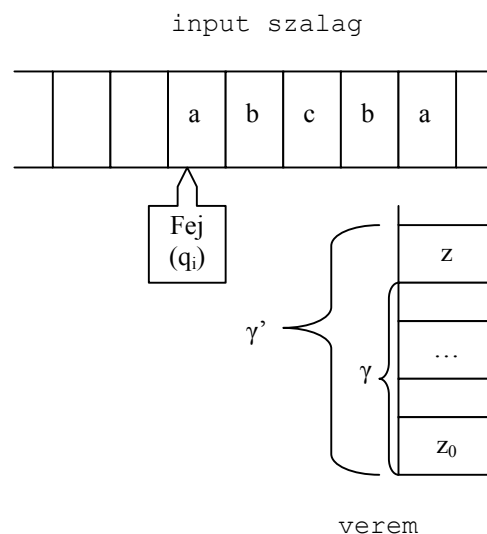
z_0 : startszimbólum, a verem aljának szimbóluma ($z_0 \in \Gamma$)

F : a végállapotok véges (vagy üres!) halmaza ($F \subseteq Q$); pl. $q_v \in F$

$w = bcab$

$w' = aw = abcab$

$\beta = z_1, z_2, \dots, z_k$



Működési leírás:

Adott egy inputszalag a Σ ábécé betűivel, amin a mozgó fej véges számú állapotba juthat el a tranzíciós függvény utasításai által.

A fej mozgását a $\delta(q_i, a, z_i)$ hármas határozza meg.

A tranzíciós lépés általában a $(q_i, a, z_i) \rightarrow (q_j, \gamma_k)$, ahol $\gamma_k \in \Gamma^*$.

A válasz után a fej jobbra lép.

Egy input hatására egyszerre több állapotba is eljuthat.

Hallgatólagosan úgy tekintjük, mintha ε is az inputábécé eleme lenne ($\varepsilon \in \Sigma$), azaz üres lépés is engedélyezett.

A verem csúcsának törlését technikailag úgy valósítjuk meg, hogy a verem csúcsát ε -nal helyettesítjük.

A teljes inputot és a teljes veremállapotot feltüntetjük:

$$(q_i, aw, z\gamma) \rightarrow (q_j, w, \beta\gamma) \rightarrow \dots \rightarrow (q_v, w^*, w\gamma)$$

A veremautomata által elfogadott nyelv az $L(V)$

- 1.) ha egy $w \in \Sigma^*$ szó beolvasása után valamilyen $q_v \in F$ végállapotba jutunk, azaz létezik olyan δ tranzíciós függvény, amelyre $\delta : (q_0, w, z_0) \xrightarrow{*} (q_v, \varepsilon, w)$

VAGY

- 2.) ha a w szó hatására a verem (és az inputszalag is) kiürül, azaz a $w \in \Sigma^*$ szót egy üres verem elfogadja, ha létezik hozzá egy δ tranzíciós függvény, amire $\delta : (q_0, w, z_0) \xrightarrow{*} (q_m, \varepsilon, \varepsilon)$.

A veremautomata akceptanciatétele® :

(i) Ha L -t elfogadja az M veremautomata a végállapotai révén, akkor létezik egy olyan M' veremautomata, amelyre L az M' üres verem által elfogadott nyelv.

(ii) Ha L -t egy M veremautomata üres verem elfogadja, akkor létezik egy olyan M^* veremautomata, mely végállapotai révén elfogadja L -t.

M veremautomata determinisztikus

- ha $\delta(q_0, w, z_0)$ legfeljebb egy elemű halmazt eredményez
- ha valamilyen (q, z) pár esetén a $\delta(q, \varepsilon, z)$ halmaz nemüres, akkor a ε -ra $\delta(q, a, z)$ üres.
- Pl. zsetonautomata, programozási nyelveket előállító automaták

Nem minden nondeterminisztikus veremautomata ekvivalens egy determinisztikus veremautomatával, így a determinisztikus veremautomaták által elfogadott nyelvek osztálya szűkebb, mint a CFL nyelvek osztálya.

Minden véges automata szimulálható egy determinisztikus veremautomatával.

Gyakorlat (+ PÉLDA):

Tranzíciós fán keresztül szemléltetjük, hogy egy veremautomata elfogad-e egy szót.

A veremautomata matematikai modelljéből megtudhatjuk, mikre kell figyelniünk. A δ az az egyik állapotból a másikba juttató függvény. Az R meg a B veremszimbólumok, amikre megy valahonnan valahova. Olyan mint az 1 meg a 0 csak a veremhez tartozik. A vizsgálati lépéseket egytől-egyig beírjuk a fába.

Pl.:

Adott az $M = (\{q_1, q_2\}, \{0, 1\}, \{B, R\}, \delta, q_1, R, 0)$ veremautomata, ahol

$$\delta(q_1, 0, R) = \delta(q_1, 1, R) = \{(q_1, BR)\}$$

$$\delta(q_1, 0, B) = \delta(q_1, 1, B) = \{(q_1, BB), (q_2, \varepsilon)\}$$

$$\delta(q_2, 0, B) = \delta(q_2, 1, B) = \{(q_2, \varepsilon)\}$$

$$\delta(q_1, \varepsilon, R) = \delta(q_2, \varepsilon, R) = \{(q_2, \varepsilon)\}$$

Elfogadja-e a $w = 001100$ szót?

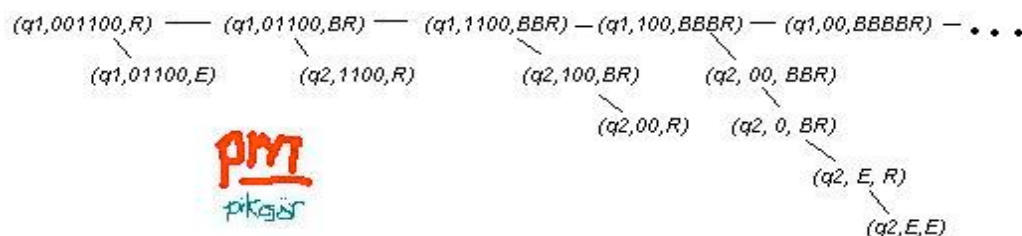
Az utasítások közül az, ami nincs $\{\}$ zárójelek közt, az kiinduló állapot, ami meg abban van, az célállapot.

Akkor fogadja el a w szót, ha olyan végállapotba jutunk, hogy $(q_m, \varepsilon, \varepsilon)$. Ha nem tudunk ilyen állapotba jutni, akkor nem fogadja el.

- 1.) Kezdetkor felírjuk a szót: 001100
 - 2.) Megnézzük, mi az első karaktere: 0
 - 3.) Megnézzük, mi a kiindulási állapot: $M = (\{q_1, q_2\}, \{0, 1\}, \{B, R\}, \delta, q_1, R, 0)$, tehát q_1 .
 - 4.) Megnézzük, mi a kiindulási szimbólum: $M = (\{q_1, q_2\}, \{0, 1\}, \{B, R\}, \delta, q_1, R, 0)$, tehát R.
 - 5.) A leírásba betesszük akkor az első elemet: $(q_1, 001100, R)$
 - 6.) Akkor keresünk a delták közt egy olyat, ami q_1 -ből megy R és 0 hatására valahova. Találunk ilyen: $\{(q_1, BR)\}$ -be megy.
 - 7.) Húzzunk egy nyilacska a $(q_1, 001100, R)$ -ből és megcélizzuk vele a $(q_1, 01100, BR)$ -t. Erre változott a kiindulási állapot $(q_1 \Rightarrow q_1, R \Rightarrow BR)$, a szó első karakterét vizsgáltuk, töröljük a vizsgálati listáról).
 - 8.) Most vizsgáljuk ε -re is. Keresünk olyan deltát, ami q_1 -ből ε és R hatására megy valahova. Megy: $(q_2, 01100, \varepsilon)$ -be. Ehhez kiinduló állapot már nincs, így ez az ág itt végetért.
 - 9.) Megyünk tovább a másik ágon. Vesszük az aktuális állapotot (q_1), a vizsgált szó első karakterét (0) és az első szimbólumot (B). Megnézzük, van-e ilyen delta. Van, $\{(q_1, BB), (q_2, \varepsilon)\}$ -be megy, tehát két nyilacska is kell húzni.
- ...
- Vizsgálni kell minden lépésnél az ε -os ágot is!
 - Miután vizsgáltad, lehúzd az első karaktert és csak a maradékot írod bele a célállapothoz tartozó részbe
 - A szimbólumnál is az első karaktert változtatod arra, amire a célállapotban jutott.
 - Ha elért a $(q_m, \varepsilon, \varepsilon)$ részhez, akkor nem kell több vizsgálat, elfogadja a szót.
 - ha ε -t helyettesítesz be valamelyik szimbólum helyére, akkor az azt jelenti, hogy törlöd azt a szimbólumot (üres karakterrel helyettesítet)

A tranzíciós fa (E az ε -nek felel meg):

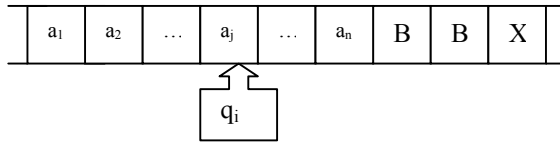
$w = 001100$



Turing gép (TM - Turing Machine)

Alan Turing (1912-1954), aki mellesleg megteremtette a virtuális valóságot, akinek nevével együtt emlegetik az Enigmát, összehozott egy jó kis gépet is. Természetesen az ő nevét örökölte, ez volt a Turing gép.

A Turing gép szintén egy végtelenített szalagon képzelhető el, csakúgy, mint a veremautomata.



A szalagon kezdetben a véges Σ ábécé a_i ($i = 1, \dots, n$) jelei vannak. $\Sigma \subseteq \Gamma$

A fej itt már nem csak olvas, hanem az aktuális cella jelét felülírja egy Γ -beli jellel (pl. B, X).

Ezután a fej elmozdul balra (L - left) vagy jobbra (R - right) vagy semerre (B - blank).

A fej a többi automatához hasonlóan csak véges számú különböző lépést tehet meg.

Matematikai modell:

$T = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ ahol

Q : az állapotok véges halmaza

Σ : az inputábécé véges halmaza

Γ : a Turing gép szimbólumai; $\Sigma \subseteq \Gamma$

δ : tranzíciós függvény, nem mindenütt értelmezhető; $\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$

q_0 : kezdőállapot; $q_0 \in Q$

B : blank szimbólum; $B \in \Gamma$

F : a végállapotok véges halmaza; $F \subseteq Q$

Ha egy nem akceptált szót olvasunk be a Turing géppel, akkor soha nem fog leállni.

A Turing gép által elfogadott nyelveket **rekurzívan felsorolható nyelveknek** nevezzük, mert $w \in L$ szót nem lehet egy véges algoritmussal eldönteni, hogy hozzátartozik-e a nyelvhez vagy sem. Ezeknél a szavaknál nem fog leállni soha.

Rekurzív halmazok: olyan halmazok, amelyeknél egy elem hovatartozása algoritmikusan (azaz véges lépésben) eldönthető.

Rekurzív nyelvek: olyan TM nyelvek, amelyeknél a $w \in L$ hozzátartozás algoritmikusan eldönthető. Ezek olyan TM-ek ill. olyan TM-ekkel ekvivalens automaták, amelyekre teljesül, hogy minden $\forall w \in \Sigma^*$ szó beolvasása után megállnak.

Rekurzív függvény: egy olyan $f : N^n \rightarrow N^R$ függvény, amelynek hatására a rekurzív TM nyelvhez tartozó szavakból üzenet lesz (pl. kódból dekódolt üzenet vagy fordítva).

Ha a TM nyelve csak rekurzívan felsorolt, akkor csak *parciálisan rekurzív függvényről* van szó. Ez esetben beszélhetünk **kiszámítható függvényről**, ha az $i_1, i_2, \dots, i_n$ behelyettesítési értékeken f értéke véges lépésben kiszámítható.

A kiszámítható függvények osztálya ekvivalens a parciálisan rekurzív egész függvények osztályával, azaz minden kiszámítható parciális egész függvény származtatható egy TM-pel.

Ha a TM nyelve rekurzív, akkor az általa származtatott kiszámítható függvény mindenütt értelmezett (nem parciális).

Chomsky hierarchia:

- **0-típusú nyelvek:** *rekurzívan felsorolható nyelvek*; minden olyan nyelvet előállítja, amit a Turing gép elfogad, azaz a nyelv szavaira megállhat.
- **1-típusú nyelvek:** *környezetérzékeny nyelvek*; a kötött szalagú nondeterminisztikus Turing automata által elfogadott nyelvek; produkciós szabályaira az $\alpha A \beta \rightarrow \alpha \gamma \beta$ jellemző, ahol α, β, γ mondatok (α, β lehet üres is, de γ nem), A pedig változó. Előfordulhat olyan produkciós szabály is, hogy $S \rightarrow \varepsilon$, de ekkor S nem szerepelhet semelyik másik szabály jobb oldalán.
- **2-típusú nyelvek:** *környezetfüggetlen nyelvek*; a produkciós szabály $A \rightarrow \gamma$, ahol A változó, γ mondat; nondeterminisztikus veremautomata által elfogadott nyelvek ezek; a programozási nyelvek elméleti alapját képezik ezek a nyelvek.
- **3-típusú nyelvek:** *reguláris nyelvek*; produkciós szabályukra az $A \rightarrow a$ jellemző, ahol A egy változó, a pedig egy terminál. Lehet $S \rightarrow \varepsilon$ is, de ekkor S nem szerepelhet semelyik másik szabály jobb oldalán. Végtes automaták által elfogadott nyelvek ezek; programozási nyelvek keresési mintáihoz használják.

Gyakorlat:

$Q = \{ q_0, q_1, q_2, q_3, \mathbf{q_4} \}$

$\Sigma = \{0, 1\}$

$\Gamma = \{0, 1, X, Y, B\}$

$F = \{ \mathbf{q_4} \}$

$W = 0011$

δ	0	1	X	Y	B
q_0	(q_1, X, R)	-	-	(q_3, Y, R)	-
q_1	$(q_1, 0, R)$	(q_2, Y, L)	-	(q_1, Y, R)	-
q_2	$(q_2, 0, L)$	-	(q_0, X, R)	(q_2, Y, L)	-
q_3	-	-	-	(q_4, Y, R)	(q_4, B, R)
$\mathbf{q_4}$	-	-	-	-	(q_4, B, B)

El kell jutni a szót beolvassva a (q_4, B, B) végállapotba (innen már nem megy sehova). Ha el tudunk jutni, akkor leáll a gép és ekkor elfogadja a szót.

A folyamatos leírás a következőképpen néz ki:

- 1.) felírjuk a szót és az első karakter elé beírjuk a aktuális állapotot (kezdetben ugye a q_0).
- 2.) megnézzük merre mozdul a fej (L – balra, R – jobbra, B – semerre) (kezdetben általában R)
- 3.) elmozdítjuk a fejet arra amerre kell és beírjuk a korábbi helyére a Γ szimbólumát
- 4.) ha a fej végigment a szó valamennyi karakterén és nem megy ismét a szóra rá, akkor elfogadta a szót
- 5.) a szó határain akkor lép ki, ha az első karakteren állva L mozgást vagy az utolsó karakteren állva R mozgást végez. Ekkor a további L ill R mozgások értelmetlenek, tehát nem kell őket figyelembe venni.
- 6.) A fej új állapotát mindig talpas nyíllal tüntetjük fel a korábbi után

Nézzük a levezetést:

$q_0 0011 \rightarrow q_1 011 \rightarrow X q_1 11 \rightarrow X 0 q_2 1 \rightarrow X q_2 Y 1 \rightarrow q_2 0 Y 1 \rightarrow 0 q_0 Y 1 \rightarrow 0 X q_3 1 \rightarrow 0 X Y q_3 \rightarrow 0 X Y Y q_4$

Túljutott a szón, és nem megy tovább a szóra ismét, tehát elfogadja.

További információk és érdekességek az angol matematikusról / filozófusról / fizikusról / biológusról / programozóról / hackerről / látnokról, valamint fenomenális gépéről: <http://www.turing.org.uk/turing>