Formális nyelvek és Automaták

*(Elmélet és Gyakorlat)*

**Oktatók:** Dr. Jenei Sándor

**Terem**: C/V.

**Óra**:2022.09.06.

[**Automaták és Formális nyelvek**](#_heading=h.468gdzog11e1) **3**

[Ábécék, szavak, formális nyelvek, valamint a velük való műveletek](#_heading=h.oug1zwbcpguz) 5

[Halmaz jelölések](#_heading=h.xo5kaba44pms) 7

[Véges, determinisztikus automaták](#_heading=h.ha952raydpt5) 9

[Fogalom](#_heading=h.mv09f5ulxabe) 9

[Egy automata](#_heading=h.l6djzi20zbel) 9

[Mikor egyenlő kettő automata?](#_heading=h.2rsxafxjnh5j) 9

[**Vizsgatételek**](#_heading=h.ck1xk2bn1dqt) **11**

[Első tétel](#_heading=h.714qm7th50mu) 11

[Leírás](#_heading=h.ab3n8xcpjs4u) 11

[Tétel](#_heading=h.bxyyjcbwbwwh) 11

[Második tétel](#_heading=h.xoif7wn1s93q) 15

[Leírás](#_heading=h.klamghij1cdp) 15

[Tétel](#_heading=h.cs6qy79zjuv5) 15

[Harmadik tétel](#_heading=h.bpjpyvrrbtbh) 18

[Leírás:](#_heading=h.vj9575qqoea1) 18

[Tétel](#_heading=h.tix2c3rndx89) 18

[Negyedik tétel](#_heading=h.lad17pedcxis) 23

[Leírás](#_heading=h.3op2o59z8km2) 23

[Tétel](#_heading=h.4134pwvmckwj) 23

[Ötödik tétel](#_heading=h.s2p8ls5pk5ny) 25

[Leírás:](#_heading=h.hkp5b0kb7uwv) 25

[Tétel](#_heading=h.isdjf3v7le07) 25

[Hatodik tétel](#_heading=h.zhsk4n2u31r3) 27

[Leírás:](#_heading=h.2p0eb07a1i1c) 27

[Tétel](#_heading=h.imlethefhw89) 27

# 

# Automaták és Formális nyelvek

Automaták elméletét az információs gépek, absztrakt matematikai elméletét értjük. Az információs gép egy olyan berendezés, mely az őket körülvevő környezetből kapnak bemeneti információt, és a kimenő adatként pedig új információt szolgáltat a rendszernek.

A tárgyban vizsgáljuk az automaták és formális nyelvek kapcsolatát vizsgáljuk.

A formális nyelvek az információ feldolgozásban azért is fontosak, mivel - ha egyes adathalmazok úgy tekintünk, mint véges sok jelből alkotott szavakra - feldolgozásuk során alkalmazhatjuk a formális nyelvek kezelésére kidolgozott automatikus módszereket. Éppen ezért a formális nyelvek fogalmát úgy célszerű bevezetni, hogy mindjárt meg is adjuk a rájuk alkalmazandó módszereket.

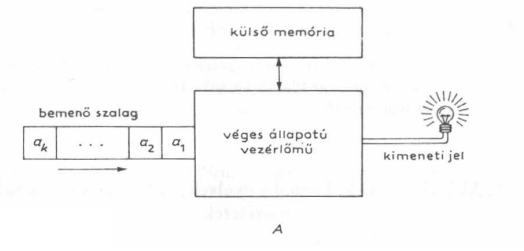
Autómaták alatt - a fogalom legtágabb értelmében - olyan absztrakt gépeket értünk, amelyek bemeneti adatok alapján, új információt szolgáltatnak kimenetként a környezetüknek, s mindezt úgy valósítják meg, hogy összesen csak véges sok különböző belső állapotba kerülhetnek, azaz véges a “memóriájuk”. Így a formális nyelvektől azt kívánjuk, hogy a vizsgált nyelvet véges eszközökkel írják le, függetlenül, hogy véges-e vagy végtelen. A továbbiakban csak a véges leírású formális nyelveket fogjuk vizsgálni.

A formális nyelvek véges leírása céljából az automaták három alapvető típusával foglalkozunk:

1. A formális nyelvek felismerőivel
2. A formális nyelvek átalakítóval
3. A formális nyelvek generátorával

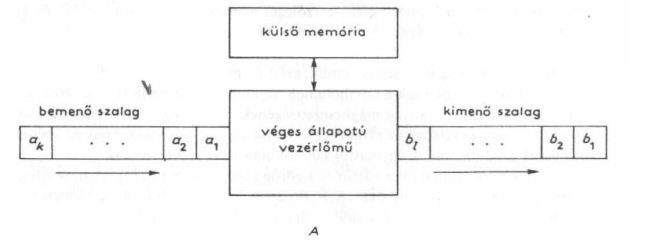
**A formális nyelvek felismerői (automata, amely felismeri a formális nyelveket):**

* Olyan A automatákat nevezünk, amelyek ellenőrzik, hogy a bemenetként megadott szó (azaz egy jelsorozat), pl. a1, a2 .. an , az adott nyelvhez tartozik-e vagy sem.
  + Megfelelés esetében az automata véges sok lépés után megáll, és valahogyan közli azt velünk.
  + Ellenkező esetben közli velünk az eredményt, véges sok lépés és felismerés után, vagy vég nélkül dolgozik.
* A bemenő adatokat egy adott véges halmazból, ún. bemenő ábécéből alkotott véges jelsorozatok (szavak), kimenő adata pedig egy bináris szimbólum (pl. igen–nem, 0–1, “világít” –”nem világít” stb.)
* Bekapcsoláskor a felismerő automata egy meghatározott kezdő állapotban van, illetve működése determinisztikus, azaz ha ugyanazt a szót adjuk be bemenetként különböző pillanatokban, a gép ugyanúgy fog válaszolni.



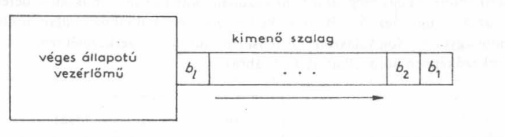
**Formális nyelvek átalakítói (Automata, amely átalakítja a formális nyelveket.)**

* Egy A automatát átalakítónak nevezünk, ha egy adott bemeneti ábécéből felépített tetszőleges a1a2 … an jelsorozatot átvált egy másik jelsorozattá, amelynek elemeit az ún. kimeneti ábécéből , {b1, b2 …, bl}-ből veszi.
  + Az átalakító tehát egy adott bemeneti nyelv szavait alakítja át a kimeneti ábécé szavaira, méghozzá determinisztikus módon. Az új szavak egyúttal egy új kimeneti nyelvet is meghatároznak.



**Formális Nyelvek Generátora:**

* Egy olyan A autómatát nevezünk, ami bekapcsoláskor, az adott kimeneti ábc szavaiból egy új szót hoz létre. Inter Determinisztikus, tehát különböző pillanatokban, különböző szavak keletkeznek benne.
  + A kimenet, amit egy automata a működése során alkot, új nyelvet alkotnak.



## Ábécék, szavak, formális nyelvek, valamint a velük való műveletek

* Legyen megadva valamilyen jeleknek egy véges halmaza, és tekintsük azokat a véges sorozatokat, amelyeket az adott jelekből valamilyen szabály alapján alkottunk meg. Ezeknek a sorozatoknak a halmazát formális nyelveknek, magukat a sorozatokat pedig szavaknak és mondatoknak nevezünk.
* **DEF**: Szimbólumok (jelek) tetszőleges nemüres véges halmazát ábécének, az ábécét alkotó jeleket az ábécé betűinek nevezzünk.
* **DEF**: A jel fogalmát alapfogalomnak tekintjük, ezért formálisan nem is definiáljuk.
  + A szimbólumok lehetnek írott jelek, egy anyag mágnesezettségének a foka, egy jelzés vagy állapot megléte, ill. hiánya stb.
  + Az ábécék jelölésére általában a latin és a görög ábécé nagybetűit fogjuk használni, s esetleg indexszel is elláthatjuk őket: *K, V, Wn, W’*
  + Azt, hogy egy jel eleme egy adott ábécének, a szokásos módon az E szimbólummal jelezzük: a ∈ *W*, c*k* ∈ *V’*
  + Az ábécékkel mint véges halmazokkal különféle halmazműveletek végezhetők. Így beszélhetünk pl. ábécék egyesítéséről vagy különbségéről stb.
* **DEF**: Legyen adva egy tetszőleges ábécé. Ekkor az ábécé betűiből alkotott véges sorozatokat szavaknak, az adott ábécéből alkotott szavaknak nevezzük.
  + Amikor ilyen betű sorozatokat írunk le, akkor a sorozat tagjait általában fogjuk vesszővel elválasztani egymástól, hanem – a szokott sorrendben – szorosan egymás után írjuk őket. Így pl. az a1, a2 …, an sorozat jelölése a1 a2 … an, az 1, 1, 1, 0, 0 sorozatá pedig 11100.
  + A “szó” és a “sorozat” kifejezéseket a továbbiakban ugyanabban az értelemben fogjuk használni, közülük is általában az elsőt.
  + A szavakat a görög kisbetűkkel jelöljük: α, β, γ, δ … ; néha esetleg latin kis betűket is használunk, elsősorban az ábécé végéről: u, v, w, x, y …
  + Az α szót alkotó sorozat elemeinek a számát a szó hosszának nevezzük, amelynek jelölése d(α).
* **DEF**: Az egyetlen betűt sem tartalmazó sorozatot üres szónak nevezzük, és ε-nal jelöljük. Természetesen, értelemszerűen d(ε) = 0.
  + Ha adva van egy *V ábécé, akkor a belőle alkotott szavakat, valamint azok hosszát rekurzívan is definiálhatjuk.*
    - *1.* ε-t a *V ábécéből alkotott szónak tekintjük, amelynek a hossza 0;*
    - 2. ha α egy *V-ből alkotott szó, amelynek hossza n, továbbá a egy V-beli betű, akkor* αa is *V*-ből alkotott szó lesz, amelynek hossza *n + 1*;
    - 3. α pontosan akkor V-ből alkotott szó, ha az első két szabály véges sokszori alkalmazásával hozható létre.
* **DEF**: Azt mondjuk, hogy két szó egyenlő, ha hosszuk megegyezik, s az első, második stb., utolsó betűje páronként egyenlők, azaz a1a2 … an = b1b2 .. bm Pontosan ekkor teljesül, ha n = m, és a1 = b1, a2 = b2, … an = bm
* **PÉLDA**: Tekintsük az alábbi két ábécét:

*V1* = {0, 1};

*V2* = {if, then, else, for, do, a, b, c}.

* Ekkor a *V1*-ből alkotott szavak lesznek pl. ,1101000 és 00; hosszuk rendre 0, 7, illetve 2.
* *V2*-ből alkotott szavakra pedig az alábbiak mutatnak példát: *if b then for c do if b then a else c, valamint if b then a else c* ; ezeknek a hossza 12, ill 6.
* A szavakkal műveleteket is végezhetünk.
* **DEF**: Az alpha és béta szavak szorzatán (egyesítésén, illesztés én) azt az új szót értjük, amely úgy jön létre, hogy bétát az alpha után írjuk. Az így kapott új szót alpha béta jelöli. Ha tehát alpha = a1a2 … ak és béta = b1b2 … bl, akkor ab = a1a2 … akb1b2 … bl.
  + Könnyen látható, hogy ha az alpha szót a V, bétát pedig a W ábécéből alkottuk, akkor alpha béta olyan szó lesz, amely a V U W ábécéből alkotható, és d(alpha béta) = d(alpha) + d(béta)
  + Legyenek alpha, béta és gamma tetszőleges szavak. Ekkor az alpha és béta gamma, illetve az alpha béta és gamma szavak szorzata megegyezik. Ezt másképpen úgy is megfogalmazhatjuk, hogy a szavak szorzása asszociatív művelet. Ezzel szemben a szorzás általában nem lesz kommutatív. Könnyen találhatunk ugyanis olyan alpha és béta szavakat, amelyekre alpha béta nem egyezik meg béta alphával.
  + Tetszőleges alpha szóra érvényes a következő azonosság: üresszó alpha = alpha üresszó = alpha. Ez azt jelenti, hogy az üres szó a szavak szorzási müveletére nézve egységelemként viselkedik.
  + A szorzási művelet segítségével definiálhatjuk szavak hatványait is.
* **DEF**: Ha alpha tetszőleges szó, akkor legyen: alpha 0 = üres szó, alpha 1 = alpha, illetve alpha n = alpha n-1 alpha, n = 2,3 … esetén. (Itt alpha n-1alpha az alpha n-1 és az alpha szavak szorzatát jelöli.)
  + Az alpha szó tükörképének azt az alpha -1-gyel jelölt szót nevezzük, amely az alpha betűit fordított sorrendben tartalmazza, azaz alpha = a1a2 … an esetén alpha -1 = an … a2a1. Emellett legyen üresszó -1 = üresszó.

## Halmaz jelölések

X = {0,1,a} 0 eleme X a eleme X b nem eleme X-nek.

Y = {1, a, 0} = {1, a, 0, 0} //Nem számít hányszor szerepel.

{} üres halmaz.

X Y = { x : x X or x Y}

X Y = { x : x X, x Y}

X \ Y (X - Y) = {x X| x Y}

Xc = \ X (Meg kell határozni egy univerzumot neki.)

X részhalmaza Y <-> ha x eleme X akkor x eleme Y-nak.

De morgan azonosság

* X Y = ( Xc Yc)c

N - Természetes számok

Z - Egész számok {..., -2, -1, 0, 1, 2, 3, …}

Q - Racionális számok

R - Irracionális számok

DEF: V az egy ábécé ha V egy véges, nem üres halmaz.

* Példa: V = { 0, 1 }
* Alpha = a1 a2 a3 … an egy string, ha ai eleme V-nek (i = 1, 2 … n)
* V szavak: 001101, 011
  + Ha egy szó hossza végtelen, akkor nem szó!

1. alpha (epszilon) = 0 (hossza)
2. ha alpha(alpha) = n és alpha eleme V akkor alpha (alpha a) = n + 1

Fibonacci sorozat:  
A következő elem, az előző két elem szumja:

1, 1, 2, 3, 5, 8, …

rekurzív sorozat.

alpha = a1, a2, … an

inverze alphának: an an-1 … a2 a1.

Példa: alpha cat

alpha inverze tac (45:00)

Műveletek:

szorzás

alpha = a1 a2 … an

béta = b1 b2 … bm

alpha \* béta = a1 a2 a3 … anb1 b2 b3 … bm

alpha = cica

béta = mica

alpha \* béta = cicamica

(NEM KOMMUTATÍV minden esetben)

ha minden alpha, béta : alpha \* béta = béta \* alpha then we talk about commutative

ellenkezője: létezik alpha, béta such that alpha \* béta != béta \* alpha (mica \* cica)

(UNIT ELEMENT)

alpha \* epszilon = epszilon \* alpha = alpha

Minden alpha, béta, gamma eleme V \* : (alpha \* béta) \* gamma = alpha \* (béta \* gamma)

(ASSZOCIATÍV)

V = {a,b}

V\* = {epszilon, a, b, aa, ab, ba, bb, aaa, aab, aba, abb, baa, bab, bba, bbb} (lexikográfiai sorrendben (ÁBÉCÉ sorrendben))

V ={ if, then, else, a, b, c, (,)}

alpha = if a then b | hossza 4 (mert if = 1 szó a V nyelvben)

DEF: Legyen V egy ábécé. Akkor bármelyik L részhalmaza V \* az egy formális nyelv.

Két halmaz egyenlő, ha az elemeik megegyeznek!

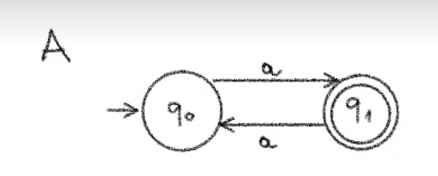
x részhalmaza y, y részhalamaza X <-> x = y

## Véges, determinisztikus automaták

### Fogalom

* Ezek a nyelv felismerő, véges, determinisztikus automaták, a chomsky hierarchia szerint kizárólag a reguláris nyelvek felismerésére alkalmasak.
  + **Determinisztikus előre meghatározott.**
* Minden állapothoz egyetlen lépést rendelünk hozzá.
* Akkor fejezi be a munkáját, ha a bemeneti szalagról minden írott betűt leolvasott.
* Egy véges, nem determinisztikus automata, a kezdőállapotból kiindulva, miután leolvasta a szalagra felirt szót, és az automata megjelölt végső állapotba került, akkor azt mondjuk, hogy felismerte, azaz elfogadta a szót.
  + automata által felismert nyelv
* A v.n.d automatákat olyan felismerőkként ábrázolhatjuk, amelyek nem rendelkeznek külső memóriával.

**Az A automata felismerte nyelve a következő:**



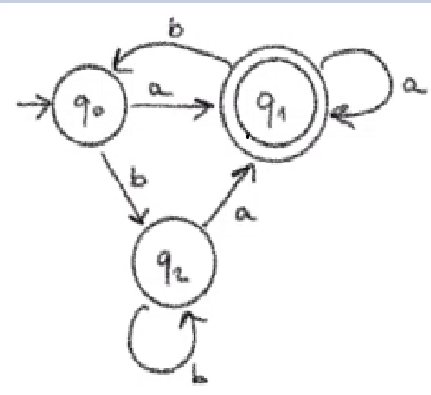
### Egy automata

* Egy V ábécével működő determinisztikus véges automatának nevezzük, ha olyan rendezett ötös, :
  + K a belső állapotok véges nem üres halmaza, az ún. állapothalmaz.
  + V a bemenő jelek véges halmaza, ez a bemenő ábécé
  + egy K-ba képező függvény, amelynek értelemzési tartománya -nek valamilyen része; ez az átmenet függvény (a nyílak)
  + kezdőállapot
  + a végállapotok halmaza

### Mikor egyenlő kettő automata?

* Két véges, determinisztikus automata egyenlő, ha ugyanazt a nyelvet ismerik fel.

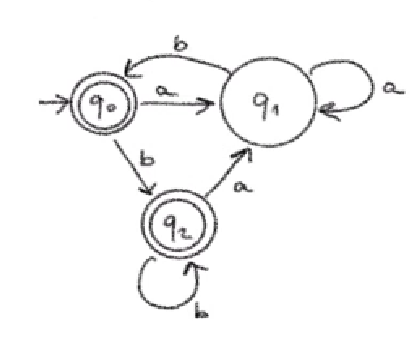
Bármilyen véges, determinisztikus automata esetében van egy A definiált A’, ahol T(A)=T(A’)



Független attól, hogy melyik fázisban van az automata, amikor ‘a’-t olvas, akkor átmegy a végső állásba.

Függetlenül attól, hogy melyik fázisban van az automata, amikor ‘b’-t olvas, akkor átmegy a vagy fázisba.

(A processzor egy véges automata)



Ez elöző automata inverze. Olyan szavakat fogad el, amit az elöző elutasitott.

Az elöző automata komplementere

# Vizsgatételek

## Első tétel

### Leírás

* **Definíciók**: ábécék, szavak, formális nyelvek, műveletek.
* **Indokolja** meg, hogy miért van az, hogy amikor egy determinisztikus automatából reguláris grammatikát készítünk, akkor az algoritmus alapján készült grammatika tényleg pont azokat a szavakat generálja, amelyeket az automata felismer!

### Tétel

* **Definíciók:**
  + **ÁBC** - Szimbólumok, tehát jelek tetszőleges nemüres, véges halmaza.
  + **Szavak -** A formális nyelvek esetében a szavak, tetszőleges ábécékből képzett véges hosszú jelsorozatok.
    - Beláthatjuk, hogy egy szó, ami , ahol
    - Egy szó inveze: , vagyis
    - Üres szót egy görög epszilon betűvel jelöljük, hossza 0.
  + **Formális nyelvek** - egy adott ábc-kből alkotott szavak tetszőleges halmazát, a formális nyelvek hívjuk.
    - Ilyen formális nyelv lehet:
  + **Műveletek** - Formális nyelvek műveletei (szorzás, hatványozás (tulajdonságok, iteráció,) ):
    - * *Két formális nyelv szorzata:*
      * *Formális nyelv hatványa:*
      * *Formális nyelv iterációja:*
        + Az összes véges szó, ami eleme a terminális nyelvek halmazának.
* **Formális nyelvek tulajdonsága:**
  + - *Asszociatívitás:*
    - ***Nem kommutatív:***
      * + Legyen és
    - ***Két formális nyelv szorzata disztributív az unióra tekintve***
    - Vegyünk három tetszőleges reguláris formális nyelvet.
      * és
      * **Bizonyítás**:
        + **Először** bizonyítjuk, hogy:

1. Vegyünk egy elemet:
2. A szót ketté tudjuk úgy választani, hogy ahol és
3. Belátjuk, hogy , ahol és
4. Ezáltal belátjuk, hogy

* **Másodszor** pedig:

1. Vegyünk ismét egy tetszőleges elemet:
2. Opcionális: vagy bizonyítsuk?
3. Felbonthatjuk alpha szót úgy, hogy: , ahol : és
4. Ha , akkor
5. Felbonthatjuk alpha szót úgy, hogy , ahol és
6. Ha , akkor

* **Mi az a grammatika?**
  + A grammtika az a formális nyelveknek véges típusú leirását adja. Vagyis nem csak generálja a formális nyelveket, de valamilyen struktúrát is csatol hozzá.
  + Egy G Grammatika a következő képpen épül fel:
    - * terminális jelek halmaza
      * nemterminális jelek halmaza
      * kezdő szimbólum
      * G grammatika szabályainak halmaza
* **Mitől reguláris egy grammatika?**
  + Egy grammatika akkor reguláris, ha regulár szó generálható, deriválható belőle. Valamint a szabályok úgy néznek ki, hogy , ahol
* **Mi az a reguláris grammatika?**
  + A chomsky Hierarchiához megfeleltetve, 3.tipusú nyelveket jellemzi. Grammatikában a generált szó, és a P szabályok alapján ismerhető fel.
* **Mi az a véges automata?**
  + Egy automatát, mint egy rendezett ötös tudjuk definiálni. Képes egy szalagábc-ről beolvasni, valamint felismerni bizonyos szavakat.
  + Úgy gondolunk egy véges automatára, mint egy külső memóriával rendelkező gépre.
  + Az ‘A’ véges, determinisztikus automata struktúrája a következőképpen épül fel:
    - * Az állapotok véges, nemüres halmaza
      * szalag ábc, amit lépésről-lépésre olvas be
      * átmenetfüggvény, ami voltaképpen állapotok közti nyilakat indikál. Minden szalagábc elem hatására, csak egyetlen következő állapothoz juthat.
      * kezdőállapot
      * végállapot, ami
      * Az ‘A’ automata által felismert szó.
  + Továbbá egy ‘B’ véges, nemdeterminisztikus automata struktúrája így néz ki:
    - * Az állapotok véges, nemüres halmaza
      * szalag ábc
      * átmenetfüggvény, ami voltaképpen állapotok közti nyilakat indikál.
      * kezdőállapot
      * végállapot, ami
      * Az ‘B’ automata által felismert szó
  + Egy automata akkor ismer fel egy nyelvet, ha véges sok lépés után, az utolsó szalagábc által beolvasott jel hatására – követvén az átmenetfüggvényt – a végállapotba kerül az utolsó jel beolvasását követően..
  + Elmondható automaták esetében, hogy két automata ekvivalens, ha ugyanazt a nyelvet ismerik fel.
  + Fontos megjegyezni, hogy minden állapothoz egyetlen lépést rendelünk hozzá.
  + Egy A automata teljesen definiált, ha az átmenet függvény az egész halmazon értelmezve van.
  + A véges determinisztikus, és véges nemdeterminisztikus automata reguláris nyelveket ismer fel.

***Hogyan készítsünk egy reguláris grammatikából egy véges, determinisztikus automatát?***

* Ismervén ezen axiómákat, belátható, hogy egy automatából készíthető egy generatív grammatika, amely ugyan azok szavakat készíti, mint amit a véges, determinisztikus automata felismer.
* Egy reguláris grammatikát – mivel reguláris nyelveket ismer fel egy véges, **determinisztikus**, valamint véges **nemdeterminisztikus** automata – felfoghatunk véges automataként. A nemterminális jelek halmaza a grammatikában, az automatában az állapotokat is jelezheti.
* Elsősorban egy grammatikában a terminális nyelvek halmaza lesz egy automatában a szalag ábc.
* Továbbá a nemterminális nyelvek halmaza lesz az állapotok, illetve mivel nemdeterminisztikus automatát készítünk az eljáráshoz vesszük, hogy hozzáteszünk egy új állapotot, ami végállapot lesz.
* Kezdőszimbólum és a kezdőállapot megeggyezik.
* Átmenetfüggvényeket pedig a szabályokhalmazából formáljuk a következő képpen:
  + Ahol a szabály úgy nézki, mint: az átmenetfüggvény úgyfog kinézni, hogy

## 

## Második tétel

### Leírás

* **Definíciók:** Generatív grammatikák, típusok, a Chomsky féle hierarchia.
* **Indokolja** meg, hogy a Chomsky hierarchia miért úgy épül fel, ahogy! Indokolja meg, hogy miért van szükség a Chomsky hierarchiában az „epszilonos” extra kitételre! Indokolja meg, hogy az „epszilonos” extra kitétel szűkíti-e a generálható nyelvek osztályát!

### Tétel

* **Definíciók**:
  + **Mi az a grammatika?**
    - A grammtika a formális nyelveknek véges típusú leirását adja. Tehát generáláson kívül struktúrát is készít a formális nyelvekhez.
    - Egy G Grammatika a következő képpen épül fel:
      * + terminális nyelvek halmaza
        + nemterminális nyelvek halmaza
        + kezdő szimbólum, ami
        + G grammatika szabályainak halmaza
  + **Chomsky hierarchia**
    - A Chomsky hierarchia a formális nyelveket kategorizálja be. El lehet képzelni mint egy halmazt, ahol a következő a struktúra:
      * 0.típusú az áltatlános grammatika , nincs formai megkötés
      * 1.típusú a környezetfüggő grammatika
      * 2.típusú a környezetfüggetlen grammatika
      * 3.típusú a reguláris grammatika
    - A általános típusú kategórián kívül az

**Chomsky Hierarhcia indoklása**

* **Mire hasznos a Chomsky hierarchia?**
  + A háttérben az ismert **szó probléma** áll, ami a következő:
  + ***“Írjunk algoritmust tetszőleges nyelvtanra, tetszőleges szóra ami eldönti az a szó generálható egy nyelvtannal, vagy sem!”***
  + Matematikai bizonyítás létezik arra, hogy **ez lehetetlen**, azonban Noam Chomsky törekedett arra, hogy kiküszöbölje ezt a problémát.
* ***Mik a következményei a Chomsky Hierarchiának?***
  + - Vegyük észre, hogy az szinteken a szabályok **baloldala sosem hosszabb, mint a jobboldal.**
      * ***Vegyünk egy környezetfüggetlen nyelv esetét:***

Ebben az esetben a baloldal hossza legalább , ezáltal, mivel , így a jobboldal is 1 lesz.

* + - * + , ahol
      * ***Reguláris nyelv esetén:***

Vagy ugyanolyan hosszú, vagy valamennyivel hosszabb a jobboldal.

* + ***Mi a lényege?***
    - **Tegyük fel, hogy igaz, hogy minden szabály esetében hosszabb a jobboldal, mint a baloldal.**
    - Vezessük fel a következő generálást:
      * + és ugyanitt:

ez pedig legalább

* + - * Vegyük észre, hogy az hosszabb, mint az S kezdőszimbólum.
      * Ebből következtethetjük azt, hogy ha szeretnénk generálni egy 4 hosszúságú szót, akkor azt1, 2 vagy három lépésben tehjetjük meg!
    - **Levonhatjuk azt, hogy egy adott szó generálást, csak véges sokféleképpen érdemes megpróbálni.**
      * Vegyünk egy 3 hosszúságú szót. Vegyük sorjában a generálási lépéseket:

1. Az első generálási lépést legfeljebb -beli szabályt alkalmazzuk, azaz az első generálási lépés legfeljebb -féle lehet.
2. A második generálási lépésben ismét valamelyik P-beli szabályt aslkalmazzuk, azaz egy két lépést igénylő generálást legfeljebb -féle képpen lehet.
3. A harmadik, és egyben utolsó generálásban ismét valamelyik P-beli szabályt alkalmazzuk, ami legfeljebb -féle képpen végezhető el.

* **Általában egy k lépéses generálása legfeljebb -féle képpen lehet.**
* Minden lépéshez egyetlen szabály tartozik, és legfeljebb annyi féle lépés lehet, ahány szabály van a szabályhalmazban.
* **Ha követjük ezeket a lépéseket, és véges sokszor, az algoritmussal próbálgatjuk a megadott szót készíteni vagy sikerül, vagy nem.**
  + Hi kijött, akkor az adott grammatikával lehet generálni, különben ha nem jött ki, akkor nem generálható a grammatikával.
* Tehát a Chomsky-hierarchia szintjein a szóprobléma eldönthetővé válik, tehát készíthető hozzá algoritmus.
* **Mi az bizonyos üresszó kitétel?**
  + A bizonyítás hasonló minden szinten.

1. Vegyünk egy tetszőleges formális nyelvet:
2. is reguláris, mivel egy reguláris grammatika, ami generálta létezik.
   1. Egy ilyen grammatika így néz ki:
   2. Ha benne van az üres szó, akkor is reguláris. Amit bebizonyítunk az órán tanult alogritmussal.
      1. Az üres szó nem szerepelhet a jobboldalon, ezért amikor belevesszük, kell venni egy új szimbólumot:
      2. Az új szimbólum mutasson az üresszóra, illetve arra a szimbólumra, amivel elkezdhetjük generálási lépéseket. Ez így néz ki: , ahol
   3. Ha nincs benne az üres szó, akkor belátható, hogy az is reguláris, mivel az üres szót egyszerűen letörölhetjük a szabályokból:

## 

## Harmadik tétel

### Leírás:

* **Definíció**: Reguláris nyelvek és tulajdonságaik (zártsági tételek; az automaták elméleténél tanult reguláris nyelvekre vonatkozó tételek is).
* **Indokolja** meg, hogy a reguláris nyelvek szorzatára vonatkozó tételnél a konstrukció miért helyes!

### Tétel

* **Mi az a grammatika?**
  + A grammtika a formális nyelveknek véges típusú leirását adja. Tehát generáláson kívül struktúrát is készít a formális nyelvekhez.
  + Egy G Grammatika a következő képpen épül fel:
    - * terminális nyelvek halmaza
      * nemterminális nyelvek halmaza
      * kezdő szimbólum
      * G grammatika szabályainak halmaza
* **Mitől reguláris egy grammatika?**
  + Egy grammatika akkor reguláris, ha regulár szó generálható, deriválható belőle. Valamint a szabályok úgy néznek ki, hogy , ahol
* **Mi az a reguláris grammatika?**
  + A chomsky Hierarchiához megfeleltetve, 3.tipusú nyelveket jellemzi. Grammatikában a generált szó, és a P szabályok alapján ismerhető fel.
* **Chomsky hierarchia**
  + A Chomsky hierarchia a formális nyelveket kategorizálja be. El lehet képzelni mint egy halmazt, ahol a következő a struktúra:
    - 0.típusú az áltatlános grammatika , nincs formai megkötés
    - 1.típusú a környezetfüggő grammatika
    - 2.típusú a környezetfüggetlen grammatika
    - 3.típusú a reguláris grammatika
  + A 0.dik típusú kategórián kívül az
  + **A Chomsky hierarchia célja kiküszöbölni az ismert szóproblémát.**

**Két reguláris nyelv uniója is reguláris (A konstrukció magáért beszél)**

* Legyen tetszőleges reguláris nyelvek.
* Ha és nyelvek regulárisak, akkor létezik őket generáló reguláris grammatikák!
  + Tekintsük ezek grammatikáit:
    - , ahol
    - , ahol
  + Készítsük el ezen grammatikák unióit:
    - , ahol M egy új szimbólum.
    - A P szabályok halmaza álljon azon szabályokból, melyek legalább benne voltak a vagy szabályaiban, KIVÉVE az és , már ha egyáltalán voltak ilyen szabályok.
    - Ezen kívül minden , illetve , , -beli szabályokhoz vegyünk be be a P-be egy-egy , szabályt.
    - Végül pedig, legyen egy , ha a vagy -beli szabályokban legalább az egyikben szerepel.
    - Lássuk be, hogy az így kapott grammatika reguláris lesz, mivel a P-beli szabályok formailag megegyeznek a és -beli szabályokkal, és a kezdőszimbólum nem fordul elő a szabályok jobb oldalán.
    - **Mutasd meg, hogy ! (mármint a konstrukciót)**

**Két reguláris nyelv szorzata is reguláris**

* Legyen tetszőleges reguláris nyelvek.
* [Ha és nyelvek regulárisak, akkor létezik őket generáló reguláris grammatikák!](#bookmark=id.4gc7sny2wyau)
  + [Tekintsük ezek grammatikáit:](#bookmark=id.4gc7sny2wyau)
    - , ahol
    - , ahol
  + Vizsgáljuk az első esetet, miszerint nyelv sem tartalmazzák az üres szót:
    - Készítsünk egy új grammatikát:
    - , ahol a kezdőszimbólum megegyezik az első grammatika kezdőszimbólumával.
    - A P tartalmazza a és szabályainak összes elemét, kivéve az típusúakat. Az utóbbiak mindegyikéhez készítsük el a megfelelő szabályt, amit bevesszük a P-be.
  + Aztán vizsgáljuk a második esetet, ahol az tartalmazzák az üres szót:
    - Készítsünk egy új grammatikát:
    - , ahol a kezdőszimbólum egy új eset, amit felvettünk a nemterminális szimbólumok halmazába.
    - A P szabályokhoz fel kell venni egy új szabályt, ahol a kezdőszimbólum mutat az üresszóra, illetve ez a kezdőszimbólum az első nyelvhez tartozó grammatika szabályának kezdő esetére mutat. Továbbá a P tartalmazza a és szabályainak összes elemét, kivéve az típusúakat. Az utóbbiak mindegyikéhez készítsük el a megfelelő szabályt, amit bevesszük a P-be.
  + Beláthatjuk, hogy a G grammatika reguláris lesz, mivel a szabályai nyílvánvalóan eleget tesznek a követelményeknek!
  + **Mutasd meg, hogy (mármint a konstrukciót)**

**Egy reguláris nyelv iterációja is reguláris**

* Legyen egy reguláris grammatika, ahol Képezzünk egy grammatikát, amelyben a P az összes G szabályát tartalmazza, kivéve az szabályt, ha szerepelt P-ben.
* Továbbá vegyünk minden szabályhoz vegyünk hozzá egy szabályt.
* Az így kapott új szabályok alakjából látszik, hogy a grammatika reguláris marad.
* Valahányszor a valamilyen típusú szabályának az alkalmazásakor a szót lezárhatnánk, az eredeti nyelvben is megtehetjük, azonban az szabállyal tovább folytathatjuk a levezetést.
* Ezáltal hozzá ragaszthatunk további szavakat a jobboldalhoz, amit akárhányszor elvégezhetünk.
* Bármilyen L véges szót elő tudunk állítani a grammatikával, ezáltal beláthatjuk, hogy mivel a szabályok formaisága megfelel, illetve a generált szó is reguláris, ezért bebizonyítottuk a tételt.

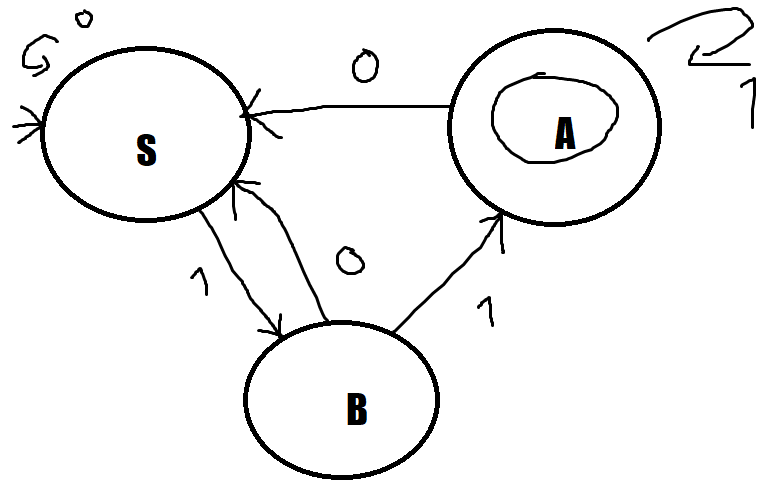
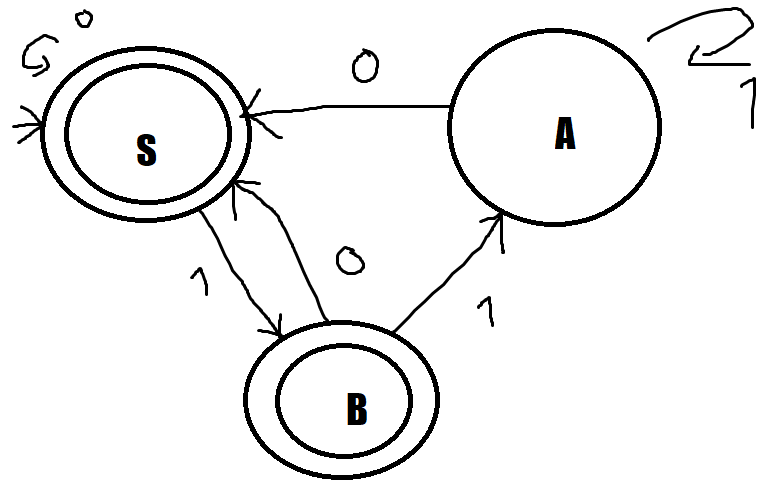
[**Két formális nyelv szorzata disztributív az unióra tekintve**](#bookmark=id.miq08e6wz60k)

* + - ***Két formális nyelv szorzata disztributív az unióra tekintve***
    - Vegyünk három tetszőleges reguláris formális nyelvet.
      * és
      * **Bizonyítás**:
        + Először bizonyítjuk, hogy:

1. Vegyünk egy elemet:
2. A szót ketté tudjuk úgy választani, hogy ahol és
3. Belátjuk, hogy , ahol és
4. Ezáltal belátjuk, hogy

* Másodszor pedig:
* Vegyünk ismét egy tetszőleges elemet:
* Opcionális: vagy bizonyítsuk?
* Felbonthatjuk alpha szót úgy, hogy: , ahol : és
* Ha , akkor
* Felbonthatjuk alpha szót úgy, hogy , ahol és
* Ha , akkor

**Két reguláris nyelv komplementere is reguláris**

* Vegyünk egy véges, determinisztikus, teljesen definiált automatát. 
* Ez az automata a következő szavakat fogja felismerni: (Ahol legalább 2 ‘1’szerepel egymás után és ‘1’-re végződik.
  + Egy véges determinisztikus automatához készíthető olyan generatív grammatika, amely pont azokat a szavakat generálja, amiket ez az automata felismer.
  + Készítsünk grammatikát.
    - Az automatából nézzük meg, hogy generálható a következő forma, legrövidebb szó esetében:
      * + A következő levezetést ZHban talán érdemes levezetni.
      * Tehát pont ezeket szavakat ismeri fel egy véges determinisztikus automata, illetve egy ebből készítet grammatika pontosan ezeket a szavakat készíti el.
  + Belátjuk tehát, hogy ez a grammatika reguláris grammika, mivel [megfelel a reguláris grammatika formaiságának.](#bookmark=id.pff4klw9hmc2) Továbbá, mivel ez a grammatika egy létező automatából lett elkészítve, ami teljesen definiálva van, ezért elmondható, hogy mivel egy véges automata reguláris nyelveket ismert fel, ezért a hozzá készített grammatika is reguláris.
* Vegyük az elöző automata komplementerét a következőképpen:
  + A végállapotok, legyenek nem-végállapotok
  + A nem-végállapotok legyenek végállapotok.
* 
* Ez az automata azon reguláris szavakat fogja felismerni, amely és
* Az ilyen szavak úgy néznek ki, mint: ahol legfeljebb 1 nulla szerepel egymás után.
* Készítsük el ehhez is a grammatikát, ugyanazzal az eljárással, mint elözőleg.
  + Tehát pont ezeket szavakat ismeri fel egy véges determinisztikus automata, illetve egy ebből készítet grammatika pontosan ezeket a szavakat készíti el.
* Belátjuk tehát, hogy ez a grammatika reguláris grammika, mivel [megfelel a reguláris grammatika formaiságának.](#bookmark=id.pff4klw9hmc2) Továbbá, mivel ez a grammatika egy létező automatából lett elkészítve, ami teljesen definiálva van, ezért elmondható, hogy mivel egy véges automata reguláris nyelveket ismert fel, ezért a hozzá készített grammatika is reguláris.

**Két reguláris formális nyelv metszete is reguláris**

* Mivel bebizonyítottuk azt, hogy két formális reguláris nyelv [komplementere](#bookmark=id.8al02kj119zw) is reguláris, ezért ez a bizonyítás triviális.
  + A **DE MORGAN** azonosságot alkalmazva a következő megállapítást tehetjük:
    - Mivel már bizonyítottuk, hogy két formális nyelv [komplementere](#bookmark=id.8al02kj119zw) és [uniója](#bookmark=id.tuzn0f8ssmdp) is reguláris, beláthatjuk, hogy ez a tétel igaz.

**Egy véges nyelv, reguláris nyelv**

* Ha egy tetszőleges formális nyelv véges, akkor belátható az órán tanult algoritmus alapján, hogy készíthető hozzá egy generatív grammatika, ami által készíthető egy reguláris nyelv.
* Ezzel belátható, hogy minden véges nyelv reguláris nyelv is, mivel készíthető hozzá egy grammatika.

## 

## Negyedik tétel

### Leírás

* **Definíció**: Determinisztikus véges automaták komponensei és működése.
* **Indokolja** meg, hogy miért igaz a Bar-Hillel lemma! Indokolja meg, hogy miért igazak a Bar-Hillel lemma általunk tanult következményei!

### Tétel

* **Mi az a véges automata?**
  + Egy automatát, mint egy rendezett ötös tudjuk definiálni. Képes egy szalag ábc-ről beolvasni szavakat, amit képes felismerni.
  + Az A automata struktúrája a következőképpen épül fel
    - * Az állapotok véges, nemüres halmaza
      * szalag ábc
      * átmenetfüggvény, ami voltaképpen állapotok közti nyilakat indikál.
      * kezdőállapot
      * végállapot, ami
      * Az A automata által felismert szó.
  + Egy automata akkor ismer fel egy nyelvet, ha elemei hatására, követvén az átmenetfüggvényt a végállapotba kerül az utolsó jel beolvasása után.
  + Elmondható automaták esetében, hogy két automata ekvivalens, ha ugyanazt a nyelvet ismerik fel.
  + Fontos megjegyezni, hogy minden állapothoz egyetlen lépést rendelünk hozzá.
  + Egy A automata teljesen definiált, ha az átmenet függvény az egész halmazon értelmezve van.
* **Mi az a véges Determinisztikus automata?**
  + A Determinisztikus véges automata esetében fontos megjegyezni, hogy minden egyes ütemben pontosan egy belső állapotot rendelhetünk hozzá.
  + Akkor fejezi be a munkáját, ha a bemeneti szalagra írott szó összes betűjét leolvasta, de képes lefagyni abban az esetben, ha egy jelhez nem tartozik pontosan meghatározott átmenetfüggvény.

**Bar-hillel lenma**

* A Bar-Hillel lenma alkalmas eldönteni, hogy egy adott formális nyelvet képes felismerni egy automata. (Algoritmizálható?)
* Próbálkozni -ig érdemes
  + Legyen , ahol ‘A’ egy véges determinisztikus automata és L ez az automata által felismert nyelv.
  + Ekkor van olyan küszöbszám, ahol, mint támpont feldarabolhatjuk három részre a szót.
    - (belső állapotok száma)
    - esetén
  + A ‘’ fontos kitétel, mivel hatványozhatjuk a v részletet, ha üres szó lenne, akkor ezt nem tehetnénk meg.
  + Ha az automata felismeri formális nyelvet, akkor a nyelvünk az automata által felismerhető lesz!
* Tekintsük a következő formális nyelvet példaként:
* Vajon az automata felismeri?
  + A Bar-Hillel szerint nem:
    - “Tegyük fel, hogy mégis felismerhető:”
    - Abban az esetben ha az ‘a’ egységekenk ismétlődének hossza megegyezik a küszöbszámmal, akkor igaznak kell lennie b-k esetével is, mivel egy ‘n’ kitevőn osztoznak.

## 

## Ötödik tétel

### Leírás:

* **Definíció**: Nemdeterminisztikus véges automaták. A determinisztikus véges automatákkal és a reguláris nyelvekkel való kapcsolatuk.
* **Indokolja** meg, hogy miért van az, hogy amikor egy reguláris grammatikából nem-determinisztikus automatát készítünk, akkor az algoritmus alapján készült automata tényleg pont azokat a szavakat ismeri fel, amelyek generálhatók a grammatikával!

### Tétel

* **Mi az a grammatika?**
  + A grammtika a formális nyelveknek véges típusú leirását adja. Tehát generáláson kívül struktúrát is készít a formális nyelvekhez.
  + Egy G Grammatika a következő képpen épül fel:
    - * terminális nyelvek halmaza
      * nemterminális nyelvek halmaza
      * kezdő szimbólum, ami
      * G grammatika szabályainak halmaza
* **Chomsky hierarchia**
  + A Chomsky hierarchia a formális nyelveket kategorizálja be. El lehet képzelni mint egy halmazt, ahol a következő a struktúra:
    - 0.típusú az áltatlános grammatika , nincs formai megkötés
    - 1.típusú a környezetfüggő grammatika
    - 2.típusú a környezetfüggetlen grammatika
    - 3.típusú a reguláris grammatika
  + A általános típusú kategórián kívül az
* **Mitől reguláris egy grammatika?**
  + Egy grammatika akkor reguláris, ha regulár szó generálható, deriválható belőle. Valamint a szabályok úgy néznek ki, hogy , ahol
* **Mi az a reguláris grammatika?**
  + A chomsky Hierarchiához megfeleltetve, 3.tipusú nyelveket jellemzi. Grammatikában a generált szó, és a P szabályok alapján ismerhető fel.
* **Mi az a véges automata?**
  + Egy automatát, mint egy rendezett ötös tudjuk definiálni. Képes egy szalagábc-ről beolvasni, valamint felismerni bizonyos szavakat.
  + Úgy gondolunk egy véges automatára, mint egy külső memóriával rendelkező gépre.
  + Az ‘A’ véges, determinisztikus automata struktúrája a következőképpen épül fel:
    - * Az állapotok véges, nemüres halmaza
      * szalag ábc, amit lépésről-lépésre olvas be
      * átmenetfüggvény, ami voltaképpen állapotok közti nyilakat indikál. Minden szalagábc elem hatására, csak egyetlen következő állapothoz juthat.
      * kezdőállapot
      * végállapot, ami
      * Az ‘A’ automata által felismert szó.
  + Továbbá egy ‘B’ véges, nemdeterminisztikus automata struktúrája így néz ki:
    - * Az állapotok véges, nemüres halmaza
      * szalag ábc
      * átmenetfüggvény, ami voltaképpen állapotok közti nyilakat indikál.
      * kezdőállapot
      * végállapot, ami
      * Az ‘B’ automata által felismert szó
  + Egy automata akkor ismer fel egy nyelvet, ha véges sok lépés után, az utolsó szalagábc által beolvasott jel hatására – követvén az átmenetfüggvényt – a végállapotba kerül az utolsó jel beolvasását követően..
  + Elmondható automaták esetében, hogy két automata ekvivalens, ha ugyanazt a nyelvet ismerik fel.
  + Fontos megjegyezni, hogy minden állapothoz egyetlen lépést rendelünk hozzá.
  + Egy A automata teljesen definiált, ha az átmenet függvény az egész halmazon értelmezve van.
  + A véges determinisztikus, és véges nemdeterminisztikus automata reguláris nyelveket ismer fel.

***Hogyan készítsünk egy reguláris grammatikából nemdeterminisztikus, véges automatát?***

* Ismervén ezen axiómákat, belátható, hogy egy automatából készíthető egy generatív grammatika, amely ugyan azok szavakat készíti, mint amit a véges, nemdeterminisztikus automata felismer.
* Egy reguláris grammatikát – mivel reguláris nyelveket ismer fel egy véges, determinisztikus, valamint véges nemdeterminisztikus automata – felfoghatunk véges automataként. A nemterminális jelek halmaza a grammatikában, az automatában az állapotokat is jelezheti.
* Elsősorban egy grammatikában a terminális nyelvek halmaza lesz egy automatában a szalag ábc.
* Továbbá a nemterminális nyelvek halmaza lesz az állapotok, illetve mivel nemdeterminisztikus automatát készítünk az eljáráshoz vesszük, hogy hozzáteszünk egy új állapotot, ami végállapot lesz.
* Kezdőszimbólum és a kezdőállapot megeggyezik.
* Átmenetfüggvényeket pedig a szabályokhalmazából formáljuk a következő képpen:
  + Ahol a szabály úgy nézki, mint: az átmenetfüggvény úgyfog kinézni, hogy

## Hatodik tétel

### Leírás:

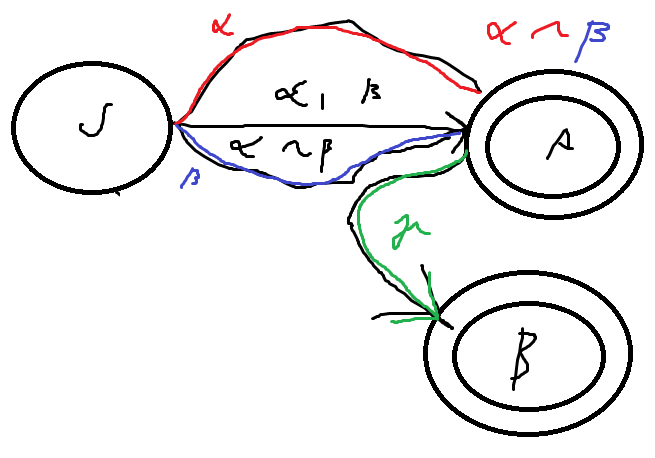
* Véges automaták minimalizálása.

### Tétel

* **Mi az a véges automata?**
  + Egy automatát, mint egy rendezett ötös tudjuk definiálni. Képes egy szalagábc-ről beolvasni, valamint felismerni bizonyos szavakat.
  + Úgy gondolunk egy véges automatára, mint egy külső memóriával rendelkező gépre.
  + Az ‘A’ véges, determinisztikus automata struktúrája a következőképpen épül fel:
    - * Az állapotok véges, nemüres halmaza
      * szalag ábc, amit lépésről-lépésre olvas be
      * átmenetfüggvény, ami voltaképpen állapotok közti nyilakat indikál. Minden szalagábc elem hatására, csak egyetlen következő állapothoz juthat.
      * kezdőállapot
      * végállapot, ami
      * Az ‘A’ automata által felismert szó.
  + Továbbá egy ‘B’ véges, nemdeterminisztikus automata struktúrája így néz ki:
    - * Az állapotok véges, nemüres halmaza
      * szalag ábc
      * átmenetfüggvény, ami voltaképpen állapotok közti nyilakat indikál.
      * kezdőállapot
      * végállapot, ami
      * Az ‘B’ automata által felismert szó
  + Egy automata akkor ismer fel egy nyelvet, ha véges sok lépés után, az utolsó szalagábc által beolvasott jel hatására – követvén az átmenetfüggvényt – a végállapotba kerül az utolsó jel beolvasását követően..
  + Elmondható automaták esetében, hogy két automata ekvivalens, ha ugyanazt a nyelvet ismerik fel.
  + Fontos megjegyezni, hogy minden állapothoz egyetlen lépést rendelünk hozzá.
  + Egy A automata teljesen definiált, ha az átmenet függvény az egész halmazon értelmezve van.
  + A véges determinisztikus, és véges nemdeterminisztikus automata reguláris nyelveket ismer fel.

**Minimalizáció:**

* Pontosan minden, véges, teljesen definiált determinisztikus automatának létezik egyetlen minimalizált változata.
* A minimalizációs eljárás egy – az órán tanult – algoritmus alapján elvégezhető.
  + A minimalizáció szükséges és elégséges feltétele, hogy teljesen definiált legyen egy automata.
  + Ha létezik egy izolált pont, akkor azt letörölhetjük.
* Első lépésként megvizsgáljuk az állapotok közötti hierarchiát. A hierarchia úgy épül fel, hogy vég és nemvég állapotok közötti hierarchia. Nekünk csak az fontos, hogy van-e különbség.
* **Két állapot akkor van egymással relációban, ha egy szó hatására ugyanoda jutnak el.**
  + Két nemekvivalens állapot között érvényesül az **ekvivalenciareláció**:
    - **Reflexív**
      * Minden szóra igaz, hogy önmagával relációban áll hiszen saját ugyanoda jut a szó hátására
    - **Szimmetrikus**
      * Mert, ha egy szó relációban van szóval, akkor ez fordítva is igaz.
    - **Tranzitív**
      * Ha egy szó kap egy szót és is ugyanezt a gamma szót kapja, akkor ugyanabba az állapotba fognak érkezni, és ez igaz, mivel és már a szó megkapása előtt relációban állnak.



* A toboz lenma szerint legfeljebb annyi ekvivalencia-reláció, ahány állapot van.
  + Ha véges reláció indexű (a tobozban legfeljebb véges reláció osztály van), akkor reguláris a nyelv.
* Minden ekvivalenciareláció diszjunkt osztályokra bontja az alaphalmazát.
* Minimalizációhoz érdemes készíteni egy négyzetmátrixot, aminek minden sora és oszlopa egyetlen állapotot jelöl

|  |  |  |  |
| --- | --- | --- | --- |
|  | a | b | c |
| a |  |  |  |
| b |  |  |  |
| c |  |  |  |

* Ha végigtekintjük az automatát, össze kell mérni a vég és nemvég állapotokat, és odacsillagozni a mátrixunkba.
  + A csillag jelöli azon állapotokat, melyeket nem vonhatunk össze.
  + Ahova nem került csillag, azokat külön vizsgálni kell.