

Digitális Számításelmélet

Majnár László, Komporday András, Felföldi András és egy kiló koffein,
amit nem Kalcsevszki Regi hozott, de azért Ő is itt van

May 2016



Konzulens: Aladár, az akaratlan nőcsábász

Tartalomjegyzék

1. Első előadás:	4
1.1. Determinisztikus véges automaták:	4
1.2. Nyelvek reprezentációja:	4
1.3. Reguláris operátorok:	5
1.4. Operátorok nyelveken:	5
1.5. Reguláris kifejezések:	6
1.6. Nemdeterminisztikus véges automaták:	6
1.7. L_R és L_D ekvivalenciája	7
1.8. Véges átalakító:	8
2. Második előadás	9
2.1. Minimális DVA	9
2.2. Reguláris nyelvek jellemzése	10
3. Harmadik előadás	11
3.1. Generatív grammatikák	11
3.2. Rekurzív grammatikák	11
3.3. Környezetfüggetlen grammatikák	12
3.4. Chomsky-féle nyelvosztályok	13
4. Negyedik előadás	13
4.1. Verem automaták	13
4.2. Műveletek környezetfüggetlen nyelveken	14
4.3. Determinisztikus CF nyelvek	15
5. Ötödik előadás	15
5.1. R és CF nyelvek összehasonlítása	15
5.2. Turing gépek	16
5.3. Turing gép változatok: helyben maradó fej	17
5.4. Turing gép változatok: több szalag	18
5.5. Turing gép változatok: nem determinisztikus	18
6. Hatodik előadás:	19
6.1. Turing-gépek	19
6.2. Függvényszámítás Turing géppel:	20
7. Hetedik előadás:	21
7.1. Kongruens számok felismerése:	21
7.2. Egy nem Turing felismerhető nyelv:	21
7.3. Egy nem Turing felismerhető nyelv: L_d	21
7.4. Az univerzális Turing gép:	22
7.5. A megállási probléma:	22
7.6. A megállási probléma üres inputra:	23
7.7. Visszavezetések:	23
7.8. A megállási probléma eldönthetősége újra:	23

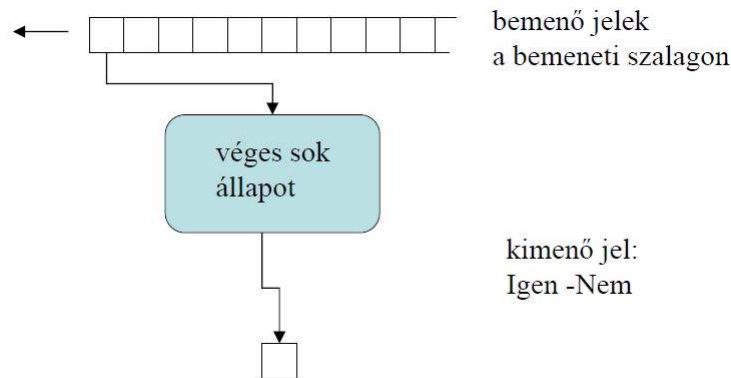
7.9. L_ε eldönthetlensége újra	24
8. Nyolcadik előadás	24
8.1. Rekurzíve felsorolható és nyelvtannal generálható nyelvek kapcsolata	24
8.2. Környezetfüggő nyelvtannal generálható nyelvek felismerése TG-pel	25
8.3. Numerikus függvények	25
9. Kilencedik előadás	26
9.1. Függvények asszimptotikus összehasonlítása	26
9.2. TG időkorlátja	27
9.3. A P osztály	27
9.4. TG-ek szimulációjának idő és tárigénye	28
9.5. Nyelvosztályok komplementere	28
9.6. Tár-idő tételek komplementens nyelvosztályokra	29
10. Tizedik előadás	30
10.1. Nemdeterminisztikus TG időkorlátja	30
10.2. Az NP nyelvosztály	30
10.3. NP-teljes problémák	31
10.4. Savitch tétele	32
10.5. Problémaosztályok tartalmazása	32

1. Első előadás:

1.1 Determinisztikus véges automaták:

1.  **Definíció. Determinisztikus véges automata** (DVA vagy VA): $M = (Q, \Sigma, \delta, q_0, F)$ ahol:

1. Q , egy véges halmaz az állapotok halmaza
2. Σ egy véges halmaz az abc
3. $\delta : Q \times \Sigma \rightarrow Q$ az állapotátmenet függvény
4. $q_0 \in Q$ a kezdeti állapot
5. $F \subseteq Q$ a végállapotok halmaza



2.  **Definíció. Felismert (elfogadott) jelsorozat.** $M = (Q, \Sigma, \delta, q_0, F)$ egy dva, $w = x_1x_2...x_n$ Σ -beli jelek sorozata. M felismert w -t, ha $\exists r_0r_1...r_n$ Q -beli állapotok sorozata, hogy:

1. $r_0 = q_0$
2. $\delta(r_i, x_{i+1}) = r_{i+1} \quad i = 1...n - 1$
3. $r_n \in F$

3.  **Definíció.** Az M dva **felismeri az L nyelvet**, ha $L = \{w \mid M \text{ felismeri } w\text{-t}\}$.

1.  **Jelölés.** Az M által felismert nyelv: $L(M)$

1.2 Nyelvek reprezentációja:

4.  **Definíció.** Σ tetszőleges jelek egy nemüres, véges halmaza, az **abc**. A Σ^* jelöli az összes, Σ -beli jelekből álló véges **sorozatokat (szavakat)** halmazát.

$$\Sigma^* = \{w = a_1...a_n \mid a_i \in \Sigma, i = 1...n, n \geq 0\}$$

$w = a_1 \dots a_n$ szó **hossza** n .

Speciálisan, ε jelöli a 0 hosszú sorozatot, az **üres szót**. A **bináris abc**, **I** a 0 és 1 jeleket tartalmazza.

5.  **Definíció.** L egy **formális nyelv** Σ felett, ha $L \subseteq \Sigma^*$.
Egy nyelv véges módon megadható:

1. Elemeit felsorolva - ha azok száma véges.
2. $L = \{w \in \Sigma^* \mid P \text{ tulajdonságú}\}$
3. $L = L(M)$ ahol M egy automata, pl. dva.
4. Operátorokkal
5. Generáló nyelvtannal

1.  **Megjegyzés.** Nem lehet az összes Σ feletti nyelvet végesen reprezentálni!

1.3 Reguláris operátorok:

6.  **Definíció.** Reguláris operátorok egy Σ abc feletti nyelveken.

- **Egyesítés (unió):** $A \cup B = \{x \mid x \in A \text{ vagy } x \in B\}$
- **Szorzat (illesztés, konkatenáció):** $A \circ B = \{xy \mid x \in A, y \in B\}$, ahol $x = x_1x_2\dots x_n$, $y = y_1y_2\dots y_m$ esetén $xy = x_1x_2\dots x_ny_1y_2\dots y_m$
- **Iterálás (Kleene csillag):** $A^* = \{w_1w_2\dots w_k \mid w_i \in A \text{ } i = 1\dots k, k \geq 0\}$
 1. megj. $\varepsilon \in A^*$
 2. megj. $L = \Sigma$ esetén $L^* = \Sigma^*$

1.4 Operátorok nyelveken:

7.  **Definíció.** További operátorok egy Σ abc feletti nyelveken.

- **Metszet:** $A \cap B = \{w \mid w \in A \text{ és } w \in B\}$
- **Különbség:** $A - B = \{w \mid w \in A \text{ és } w \notin B\}$, $A^C = \Sigma^* - A$
- **Tükörkép:** $A^R = \{x^R \mid x \in A\}$ ahol $x = x_1x_2\dots x_n$ esetén $x^R = x_nx_{n-1}\dots x_1$

1.5 Reguláris kifejezések:

8.  **Definíció.** Egy Σ abc feletti reguláris kifejezések:

1. Σ minden eleme
2. ε
3. $\emptyset$
4. Ha α és β reguláris kifejezések, akkor $(\alpha\beta)$ is az.
5. Ha α és β reguláris kifejezések, akkor $(\alpha \cup \beta)$ is az.
6. Ha α reguláris kifejezések, akkor α^* is az.

Minden reguláris kifejezés reprezentál egy nyelvet!

1.  **Tétel.** Legyen $\mathbf{L}$ azon nyelvek halmaza, melyek felismerhetők determinisztikus véges automatával. $\mathbf{L}$ zárt a reguláris operátorokra nézve. Azaz ha $A_1 \in \mathbf{L}$ és $A_2 \in \mathbf{L}$ akkor:

1. $A_1 \cup A_2 \in \mathbf{L}$
2. $A_1 \circ A_2 \in \mathbf{L}$
3. $A_1^* \in \mathbf{L}$

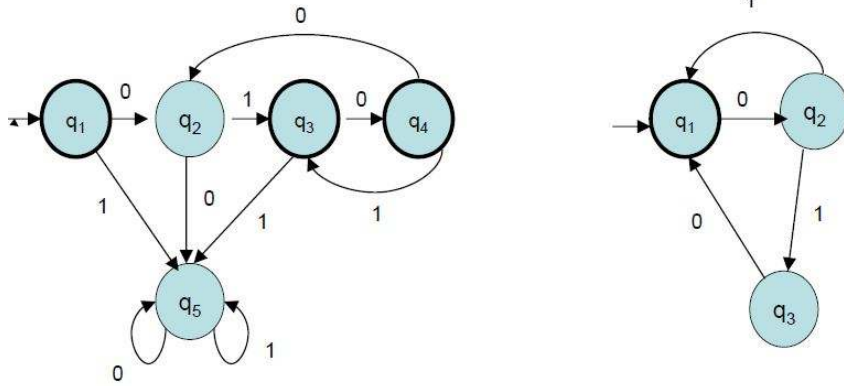
1.6 Nemdeterminisztikus véges automaták:

9.  **Definíció.** Nemdeterminisztikus véges automata (NVA): $N = (Q, \Sigma, \delta, q_0, F)$ ahol:

1. Q , egy véges halmaz az **állapotok** halmaza
2. Σ egy véges halmaz az **abc**
3. $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \rightarrow P(Q)$ az **állapotátmenet függvény**
4. $q_0 \in Q$ a **kezdeti állapot**
5. $F \subseteq Q$ a **végállapotok halmaza**

2.  **Jelölés.** $\Sigma_\varepsilon = \Sigma \cup \{\varepsilon\}$

$$L = (01 \cup 010)^*$$



10.  **Definíció. Felismert (elfogadott) jelsorozat.** $N = (Q, \Sigma, \delta, q_0, F)$ egy nva, $w = w_1w_2\dots w_n$ Σ -beli jelek sorozata. N felismert w -t, ha $\exists r_0r_1\dots r_n$ Q -beli állapotok sorozata, hogy:

1. $r_0 = q_0$
2. $r_{i+1} \in \delta(r_i, w_{i+1}) \quad i = 1 \dots n - 1$
3. $r_n \in F$

11.  **Definíció.** Az N nva **felismeri az L nyelvet**, ha $L = \{w \mid N \text{ felismeri } w - t\}$.

12.  **Definíció.** M_1 és M_2 **ekvivalensek**, ha $L(M_1) = L(M_2)$.

2.  **Tétel.** Minden $N = (Q, \Sigma, \delta, q_0, F)$ nva-hoz létezik vele ekvivalens M dva.

3.  **Tétel.** Legyen $\mathbf{L}$ azon nyelvek halmaza, melyek felismerhetők determinisztikus véges automatával. $\mathbf{L}$ zárt a reguláris operátorokra nézve. Azaz ha $A_1 \in \mathbf{L}$ és $A_2 \in \mathbf{L}$ akkor:

1. $A_1 \cup A_2 \in \mathbf{L}$
2. $A_1 \circ A_2 \in \mathbf{L}$
3. $A_1^* \in \mathbf{L}$

1.7 L_R és L_D ekvivalenciája

Következmény (L_R és L_D ekvivalenciája): Az alábbi, valamely adott Σ feletti nyelvosztályok megegyeznek:

1. L_R a reguláris nyelvek halmaza (reguláris műveletekkel / kifejezésekkel megadható nyelvek).
2. L_N az nva-k által felismert nyelvek.

3. L_D a dva által felismert nyelvek.

13.  **Definíció. Általánosított nemdeterminisztikus véges automata (ánva)**

$N = (Q, \Sigma, \delta, q_{kezdő}, q_{vég})$ ahol :

1. Q , egy véges halmaz az állapotok halmaza
2. Σ egy véges halmaz az abc
3. $\delta : (Q - \{q_{vég}\}) \times (Q - \{q_{kezdő}\}) \rightarrow$ az állapotátmenet fv, ahol R a Σ feletti reguláris kifejezések halmaza.
4. $q_{kezdő} \in Q$ **kezdeti állapot**
5. $q_{vég} \in Q$ **végállapot**

14.  **Definíció. Felismert (elfogadott) jelsorozat** $N = (Q, \Sigma, \delta, q_{kezdő}, q_{vég})$ egy ánvá. N felismeri a $w \in \Sigma^*$ sorozatot, ha $w = w_1 w_2 \dots w_k$ ahol $w_i \in \Sigma^*$ $i = 1 \dots k$ és $\exists r_0 r_1 \dots r_k$ Q -beli állapotok sorozata, hogy:

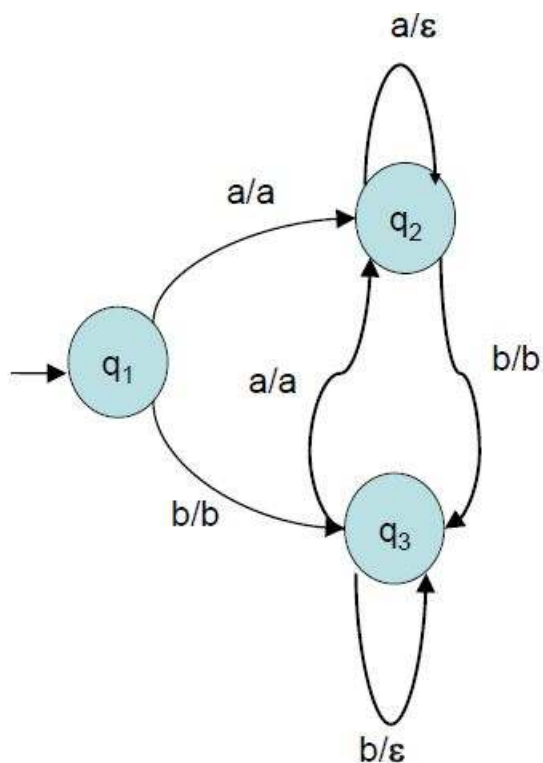
1. $r_0 = q_{kezdő}$
2. $r_k = q_{vég}$
3. $w_i \in L(R_i)$, ahol $R_i = \delta(r_{i-1}, r_i)$ $i = 1 \dots k$

15.  **Definíció.** N ánvá **felismeri az L nyelvet**, ha $L = \{w \mid N \text{ felismeri } w - t\}$.

1.8 Véges átalakító:

16.  **Definíció. Véges átalakító (fordító)** $M = (Q, \Sigma, \Gamma, \delta, q_0)$ ahol :

1. Q , egy véges halmaz az állapotok halmaza
2. Σ egy véges halmaz, a bemeneti abc
3. Γ egy véges halmaz, a kimeneti abc
4. $\delta : Q \times \Sigma \rightarrow Q \times \Gamma$ az állapotátmenet függvény
5. $q_0 \in Q$ a kezdeti állapot



aaaaa	a
aba	aba
aaaabbaba	ababa

2. Második előadás

2.1 Minimális DVA

17. **Definíció.** Legyen $L \subseteq \Sigma^*$ egy nyelv. $x, y \in \Sigma^*$ **L-re nézve ekvivalensek**, ha $\forall z \in \Sigma^*$ esetén igaz, hogy: $xz \in L$ akkor és csak akkor, ha $yz \in L$. Ezt a relációt $\approx_L$ -vel jelöljük.

2. **Megjegyzés.** $\approx_L$ ekvivalencia reláció Σ^* -on, mely ekvivalencia osztályokra bontja Σ^* -t. Egy $x \in \Sigma^*$ ekvivalencia osztályát $[x]$ jelöli.

18. **Definíció.** Legyen $M = (Q, \Sigma, \delta, q_0, F)$ dva. $(r, x) \in Q \times \Sigma^*$, **M egy konfigurációja**.

19. **Definíció.** Legyen $M = (Q, \Sigma, \delta, q_0, F)$ dva, (r, x) egy konfiguráció. (r, x) **rákövetkezője** az (r', x') konfiguráció, ha:

- $ax' = x$, ahol $a \in \Sigma$,
- $\delta(r, a) = r'$.

3. **Jelölés.** $(r, x) \vdash_M (r', x')$.

20.  **Definíció.** $(r, x) \vdash_M^* (r', x')$, **(r, x)-ből (r', x') elérhető**, ha $(r, x) \vdash_M (r_1, y_1) \vdash_M \dots (r_k, y_k) \vdash_M (r', x')$ valamely $k \geq 0$.

21.  **Definíció.** Legyen $M = (Q, \Sigma, \delta, q_0, F)$ dva. $x, y \in \Sigma^*$ **M-re nézve ekvivalensek**, ha $\exists q \in Q$ hogy $(q_0, x) \vdash_M^* (q,)$ és $(q_0, y) \vdash_M^* (q,)$. Ezt a relációt $\sim_M$ -vel jelöljük.

3.  **Megjegyzés.** $\sim_M$ **ekvivalencia reláció** Σ^* -on.

4.  **Jelölés.** Minden q elérhető állapothoz egy-egy Σ^* -részhalmaza ekvivalencia osztály tartozik, ezt $[q]$ jelöli.

4.  **Tétel.** Legyen $M = (Q, \Sigma, \delta, q_0, F)$ dva. Ekkor $\forall x, y \in \Sigma^*$ esetén: $x \sim_M y \Rightarrow x \approx_{L(M)} y$

5.  **Tétel.** Ha $L \subseteq \Sigma^*$ egy reguláris nyelv, akkor $\exists M = (Q, \Sigma, \delta, q_0, F)$ dva, hogy $L(M) = L$ és állapotainak száma, n éppen az $\approx_L$ reláció által meghatározott ekvivalencia osztályok száma. M az állapotainak számától eltekintve egyértelmű.

1.  **Következmény.** $L \subseteq \Sigma^*$ reguláris akkor és csak akkor, ha $\approx_L$ reláció által meghatározott ekvivalencia osztályok száma véges.

2.2 Reguláris nyelvek jellemzése

6.  **Tétel. Pumpáló lemma, Bar-Hillel lemma:** Ha $L \subseteq \Sigma^*$ egy reguláris nyelv, akkor $\exists p \in \mathbb{N}$ szám, hogy $\forall s \in L$ esetén ha s hossza legalább p , akkor s felbontható $s = xyz$ alakba úgy, hogy:

1. $xy^n z \in L$ minden $n \geq 0$ esetén,
2. $|y| > 0$,
3. $|xy| \leq p$.

1.  **Bizonyítás.** • Legyen $p = |Q|$.

- Legyen $s \in L$, $s = s_1 s_2 \dots s_n$, $p \leq n$. Az s -et elfogadó állapotok sorozata legyen $r = r_1, r_2 \dots r_{n+1}$, ekkor $p < n + 1$. Tehát r tagjai között kell lennie legalább két azonosnak, a **skatulya elv** miatt. Jelölje ezeket r_j és r_k . Feltehetjük, hogy $k \leq p + 1$.

- $x = s_1 s_2 \dots s_{j-1}$, $y = s_j s_{j+1} \dots s_{k-1}$, $z = s_k s_{k+1} \dots s_n$.

1. $xy^n z \in L$ minden $n \geq 0$ esetén valóban,
2. $|y| > 0$, mert $j \neq k$,
3. $|xy| = k - 1 \leq p$, mert $k \leq p + 1$.

3. Harmadik előadás

3.1 Generatív grammatikák

22.  **Definíció.** Egy $G=(V, \Sigma, R, S)$ négyes generatív grammatika, ahol:
- V egy véges halmaz, a **grammatikai változók, vagy nemterminálisok** halmaza.
 - Σ egy véges halmaz, diszjunkt V -vel, a **terminálisok halmaza**,
 - R egy véges halmaz, a **levezetési szabályok halmaza**, ahol egy levezetési szabály alakja $\alpha \rightarrow \beta$, ahol α és β terminálisok és nemterminálisok tetszőleges sorozata, de α -ban kell legyen egy nemterminális.
 - $S \in V$ a **kezdő- vagy mondatzimbólum**.

23.  **Definíció.** Egy $G=(V, \Sigma, R, S)$ grammatika alapján a $\Gamma \in (V \cup \Sigma)^*$ szó, ha léteznek olyan $\gamma, \delta \in (V \cup \Sigma)^*$ szavak és $\alpha \rightarrow \beta$ szabály R -ben, hogy:

$$\Gamma = \gamma\alpha\delta, \mu = \gamma\beta\delta$$

Jelölés: $\Gamma \Rightarrow_G \mu$ vagy $\Gamma \Rightarrow \mu$

24.  **Definíció.** Tetszőleges u és $v \in (V \cup \Sigma)^*$ -beli szavak esetén, azt mondjuk, hogy **v levezethető az u -ból** a $G=(V, \Sigma, R, S)$ grammatika alapján, ha $u=v$, vagy léteznek $n \geq 0$ és $u = w_1, \dots, w_n = v \in (V \cup \Sigma)^*$ -beli szavak úgy, hogy

$$w_1 \Rightarrow_G w_2 \Rightarrow_G \dots \Rightarrow_G w_n$$

Jelölés: $u \Rightarrow^*_G v$ vagy $u \Rightarrow^* v$

25.  **Definíció.** A $G = (V, \Sigma, R, S)$ grammatika generálja az $L \subseteq \Sigma^*$ nyelvet, ha $L=L(G)$, ahol

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

a G grammatika által generált nyelv.

26.  **Definíció.** A G és G' grammatikák ekvivalensek, ha $L(G)=L(G')$

3.2 Rekurzív grammatikák

27.  **Definíció.** A $G=(V, \Sigma, R, S)$ grammatika rekurzív, ha van olyan $A \in V$, és $\gamma, \delta \in (V \cup \Sigma)^*$ szavak, vagy $A \Rightarrow^* \sigma A \delta$

Ekkor az A egy **rekurzív nemterminális** a grammatikának. *Megjegyzés:* Nemrekurzív grammatikák véges nyelvet generálnak.

28.  **Definíció.** $G=(V, \Sigma, R, S)$ grammatika **környezetfüggetlen**, ha a levezetési szabályainak bal oldalán egyetlen nemterminális szerepel, azaz a szabályok $A \rightarrow B$ alakúak, ahol $A, B \in V, \beta \in (V \cup \Sigma)^*$

3.3 Környezetfüggetlen grammatikák

7.  **Tétel.** Egy reguláris nyelv környezetfüggetlen.

29.  **Definíció.** Egy CF G grammatika szerinti levezetés olyan fával ábrázolható, melynek gyökere a kezdőszimbólum, levelei balról jobbra a levezetett szó betűi, további csomópontjai nemterminálisok nemterminálisok, és az elágazások egy-egy alkalmazott szabálynak felelnek meg. Ez a **levezetési fa**.

30.  **Definíció.** Egy környezetfüggetlen grammatika szerinti levezetés baloldali, ha a levezetés egyes lépéseiben a legbaloldalibb nemterminális helyettesítjük be.

31.  **Definíció.** Egy nyelvtan egyértelmű, ha minden generált szavának egyetlen baloldali levezetése van. Különben többértelmű.

Megjegyzés:

Minden baloldali levezetés meghatároz egy levezetési fát. Tehát egyértelmű nyelvtan esetén minden generált szóhoz egyetlen levezetési fa létezik.

32.  **Definíció.** Egy CF nyelv eredendően többértelmű, ha nem generálható egyértelmű grammatikával.

33.  **Definíció.** A $G=(V, \Sigma, R, S)$ CF grammatika **Chomsky-féle normál alakban** adott, ha a szabályok az alábbi alakúak:

- $A \rightarrow BC$
- $A \rightarrow a$
- $S \rightarrow \varepsilon$

8.  **Tétel.** Tetszőleges CF nyelvhez megadható őr generáló grammatika Chomsky-féle normál alakban.

9.  **Tétel. Pumpáló lemme, Bar-Hillel lemma:** Ha $L \subseteq \Sigma^*$ egy CFL, akkor $\exists p \in \mathbb{N}$ szám, hogy $\forall s=uvxyz$ alakba úgy, hogy:

1. $uv^nx y^n z \in L$ minden $n \geq 0$ esetén,
2. $|vy| > 0$,
3. $|vxy| \leq p$

3.4 Chomsky-féle nyelvosztályok

34.  **Definíció. Chomsky-féle nyelvosztályok** A nyelvet generáló grammatika szabályaira tett megkötések alapján:

- **3-as nyelvosztály: Reguláris (R)** $A \rightarrow a, A \rightarrow aB$ alakú osztályok.
- **2-es nyelvosztály: Környezetfüggetlen (CF)** $A \rightarrow a$ alakú szabályok.
- **1-es nyelvosztály:**
 - $\gamma \rightarrow \beta\alpha\gamma$, alakú szabályok,
 - $\alpha \rightarrow \beta$ alakú szabályok, ahol $|\alpha| \leq |\beta|$ nem csökkenő.
 - **Megjegyzés:** ezek nem ekvivalensek!
- **0-ás nyelvosztály:** tetszőleges alakú szabályok

4.  **Megjegyzés.** egyre bővebb nyelvosztályok

5.  **Megjegyzés.** nyelv jellemzése az őt generáló grammatika alapján.

4. Negyedik előadás

4.1 Verem automaták

35.  **Definíció. Verem automata:** $M = (Q, \Sigma, \Gamma, \delta, q_0, F)$ ahol:

1. Q , egy véges halmaz az állapotokhalmaza
2. Σ egy véges halmaz a bemenetiabc
3. Γ egy véges halmaz a veremabc
4. $\delta : Q \times \Sigma_\epsilon \times \Gamma_\epsilon \rightarrow P(Q \times \Gamma_\epsilon)$ az állapotátmeneti függvény
5. $q_0 \in Q$ a kezdeti állapot
6. $F \subseteq Q$ a végállapotok halmaza

5.  **Jelölés.** $\Sigma_\epsilon = \Sigma \cup \epsilon, \Gamma_\epsilon = \Gamma \cup \epsilon$ ('nem olvasunk-írunk')

36.  **Definíció. Felismert (elfogadott) jelsorozat** $M = (Q, \Sigma, \Gamma, \delta, q_0, F)$ egy verem automata, $w = w_1w_2...w_n$ Σ_ϵ -beli jelek sorozata. M felismeri w -t ha $\exists$:

1. $r_0r_1...r_n$ Q -beli állapotok sorozata, és
2. $s_0s_1...s_n\Gamma_\epsilon$ - feletti szavak sorozata hogy:

3. $r_0 = q_0$ és $s_0 = \varepsilon$
4. $(r_{i+1}, b) \in \delta(r_i, w_{i+1}, a) \quad i = 0 \dots n-1$ ahol :
5. $s_i = at$ és $s_{i+1} = bt$ valamely $a, b \in \Gamma_\varepsilon$ és $t \in \Gamma^*$
6. $r_n \in F$

37.  **Definíció.** M verem automata **felismeri az A nyelvet**, ha $A = \{w \mid M \text{ felismeri } w-t\}$.

10.  **Tétel.** Minden L CF nyelvhez létezik őt felismerő verem automata.

2.  **Bizonyítás.** Legyen egy, az L-et generáló CF grammatika G. Megkonstruálunk egy M verem automatát, amely minden bemeneti jelsorozatról eldönti, hogy levezethető-e a G grammatika szerint.

11.  **Tétel.** Minden M verem automatához létezik G CF grammatika, mely éppen az M által felismert nyelvet generálja.

1.  **Lemma.** Ha x levezethető A_{pq} -ből, akkor M a p állapotból üres veremmel indulva, az x-et beolvasva el tud jutni a q állapotba úgy, hogy a verem ismét üres.

2.  **Lemma.** Ha M valamely p állapotából és üres veremből x-et beolvasva, van olyan állapotátmenetek sorozata, hogy az automata q állapotba kerül, és a verem ismét üres, akkor x levezethető a grammatika A_{pq} nemterminálisából.

4.2 Műveletek környezetfüggetlen nyelveken

12.  **Tétel.** Legyen $\mathbf{L}$ a CF nyelvek halmaza. $\mathbf{L}$ zárt a reguláris operátorokra nézve. Azaz ha $A_1 \in \mathbf{L}$ és $A_2 \in \mathbf{L}$, akkor :

1. $A_1 \cup A_2 \in \mathbf{L}$.
2. $A_1 \circ A_2 \in \mathbf{L}$.
3. $A_1^* \in \mathbf{L}$.

13.  **Tétel.** Legyen $\mathbf{L}$ a CF nyelvek halmaza. $\mathbf{L}$ nem zárt a metszetre nézve. $\mathbf{L}$ nem zárt a komplementerre nézve.

3.  **Bizonyítás.** $A_1 = a^n b^n c^m \mid n, m \geq 0$ mindkettő CF nyelvek. $A_1 \cap A_2 = a^n b^n c^n \mid n, m \geq 0$ viszont nem CF (beláttuk). A DeMorgan azonosság szerint: $A_1 \cap A_2 = (A_1^C \cup A_2^C)^C$, így a komplementerre zártság maga után vonná $\cap$ -re zártságot.

14.  **Tétel.** Ha A_1 CF nyelv, A_2 reguláris nyelv, akkor $A_1 \cap A_2$ CF nyelv.

4.3 Determinisztikus CF nyelvek

38.  **Definíció.** Az $M = (Q, \Sigma, \Gamma, \delta, q, F)$ verem automata determinisztikus, ha a $\delta : Q \times \Sigma_\epsilon \times \Gamma_\epsilon \rightarrow P(Q \times \Gamma_\epsilon)$ állapotátmenet fv olyan, hogy minden q, a, b esetén $\delta(q, a, b)$ maximum egyelemű halmaz.

39.  **Definíció.** Az $L \subseteq \Sigma^*$ nyelv determinisztikus CF, ha az $L\{\$$ nyelvhez van őtgeneráló determinisztikus verem automata, ahol $\$$ nem Σ -beli szimbólum, mely a nyelv szavainak végét jelöli.

15.  **Tétel.** A determinisztikus CF nyelvek osztálya zárt a reguláris operátorokra, valamint a komplementerképzésre, és a metszetképzésre.

16.  **Tétel.** A determinisztikus CF nyelvek osztálya szűkebb, mint CF nyelvek osztálya, azaz van CF nyelv, ami nem determinisztikus CF nyelv.

4.  **Bizonyítás.** A fenti tételt alkalmazzuk és keresünk egy alkalmas reguláris nyelvet.

5. Ötödik előadás

5.1 R és CF nyelvek összehasonlítása

	Reguláris nyelvek	Környezetfüggetlen nyelvek
Megadása	• reguláris kifejezéssel • grammatikával $A \rightarrow aB \mid a$	• $A \rightarrow \beta, A \in V, \beta \in (V \cup \Sigma)^*$ • Chomsky-féle normál alak
Példák	$\{1^n k \mid n \geq 0\}$, szó felismerés	$\{0^i 1^i \mid i > 0\}$, zárójelezés, progr. nyelvek (BNF)
Ellenpéldák	$\{w \mid \text{azonos számú } 0 \text{ és } 1\}$	$\{a^n b^n c^n \mid 0 \leq n\}$
Pumpáló l.	$s = xyz, xy^n z \in L, y > 0, xy \leq p$	$s = uvxyz, uv^n xy^n z \in L, vy > 0, vxy \leq p$
Nemdet. gép	nemdet. véges automata	(nemdet) verem automata
Det. gép	ekvivalens a nemdet. va	gyengébb mint a nemdet. változat
Minimális gép	egyértelmű minden nyelvhez	?
Operátorokra zárt	reguláris operátorok $A \cap B, A-B, A^C = \Sigma^* - A, A^R$	reguláris operátorok $A \cap B, A-B$ csak ha A reguláris!

5.2 Turing gépek

40.  **Definíció. Turing gépek:**

- idealizált számítógép
- a legáltalánosabb számítási modell: minden algoritmikusan megoldható probléma megoldható Turing-géppel is
- de: nem minden probléma oldható meg Turing-géppel!
- kiszámíthatóság elmélete, algoritmikusan eldönthetetlen feladatok

41.  **Definíció. Turing gép:** $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{elfogad}, q_{elutasit})$ ahol:

1. Q , egy véges halmaz, az **állapotok halmaza**
2. Σ egy véges halmaz a bemeneti ábécé, mely nem tartalmazza a $_$ jelet
3. Γ , egy véges halmaz a szalag ábécé, ahol $_ \in \Gamma$ és $\Sigma \subseteq \Gamma$
4. $\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{J, B\}$ az **állapotátmenet függvény**
5. $q_0 \in Q$ a kezdő állapot
6. $q_{elfogad} \in Q$ az elfogadó állapot
7. $q_{elutasit} \in Q$ az elutasító állapot

6.  **Megjegyzés. Turing gép működése:** Kezdőhelyzetben az input sorozat a szalag első n mezőjében szerepel, a többi mezőben a $jeláll$. *Agépvagy megálllel fogadó vagy elutasító állapotban, vagy nem áll mindenütt értelmezett!*

7.  **Megjegyzés.** Egy pillanatnyi konfiguráció leírható a szalag tartalmával, a fej pozíciójával és a pillanatnyi állapottal. Megadása: $u q v$, ahol q egy állapot, uv a szalagon lévő sorozat a végtelen sok jel előtt, a fej vel $ső$ elemén áll.

42.  **Definíció. Turing gép egy lépése:** $a, b, c \in \Gamma$ szalagjelek, $u, v \in \Gamma^*, p, q \in$ állapotok. Az $ua p bv$ konfiguráció rákövetkezője az $u q acv$, ha $\delta(p, b) = (q, c, B)$ (a fej balra lépett) Az $ua p bv$ konfiguráció rákövetkezője az $uac q v$, ha $\delta(p, b) = (q, c, J)$ (a fej jobbra lépett)

8.  **Megjegyzés.** $p bv$ rákövetkezője balra lépés esetén $p cv$ (nem esik le a szalagról)

43.  **Definíció.** $u, v \in \Gamma^*, p \in Q$ (u, p, v) egy **konfiguráció**, elfogadó/elutasító, ha $p = q_{elfogad} / q_{elutasit}$ kezdeti: (ε, q_0, w) ahol w a szalagra írt bemenő jelsorozat

44.  **Definíció.** M Turing gép elfogad egy w sorozatot, ha $\exists C_1 C_2 \dots C_n$ konfigurációk sorozata, hogy:

1. C_1 a kiindulási konfiguráció,
2. C_i rákövetkezője C_{i+1} , $i = 1, \dots, n-1$,
3. C_n elfogadó konfiguráció.

6.  **Jelölés.** $L(M)$ jelöli az M Turing gép által elfogadott sorozatok halmazát. Ez a **gép által felismert** nyelv.

45.  **Definíció.** Egy $L \subseteq \Sigma^*$ nyelv **Turing felismerhető (rekurzíve felsorolható, parciálisan rekurzív, félig eldönthető)**, ha van olyan M Turing gép, melyre $L(M) = L$.

9.  **Megjegyzés.** $s \in \Sigma^*$ -L esetén nem biztos, hogy megáll a gép, így csak az L elemeire kapunk igenlő választ.

46.  **Definíció.** Egy $s \in \Sigma^*$ nyelv **Turing eldönthető** vagy rekurzív, ha van olyan M Turing gép, melyre $L(M) = L$ és M minden $s \in \Sigma^*$ szóramegáll.

17.  **Tétel.** Ha L rekurzív, akkor L^C is az.

5.  **Bizonyítás.** Ha $L(M) = L$, akkor M -ben az elfogadó és elutasító állapotot felcserélve, az így kapott M' TG-re $L(M') = L^C$.

47.  **Definíció.** Definíció: M és S Turing gép változatok ekvivalensek, ha $L(M) = L(S)$.

10.  **Megjegyzés.** Ekvivalencia bizonyítására minden esetben megmutatjuk, hogy egy M TG változat szimulálható egy alkalmas S alap TG-pel.

5.3 Turing gép változatok: helyben maradó fej

48.  **Definíció.** Turing gép helybenmaradó fejű (HF TG): $M = (Q, \Sigma, \Gamma, q_0, q_{elfogad}, q_{elutasit})$ ahol:

1. Q , egy véges halmaz, az **állapotok halmaza**
2. Σ egy véges halmaz a bemeneti ábécé, mely nem tartalmazza a $_$ jelet
3. Γ , egy véges halmaz a szalag ábécé, ahol $_ \in \Gamma$ és $\Sigma \subseteq \Gamma$
4. $\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{J, B, H\}$ az **állapotátmenet függvény**
5. $q_0 \in Q$ a kezdő állapot
6. $q_{elfogad} \in Q$ az elfogadó állapot
7. $q_{elutasit} \in Q$ az elutasító állapot

18.  **Tétel.** A HF TG-ek ekvivalensek a TG-ekkel.

5.4 Turing gép változatok: több szalag

49.  **Definíció.** Turing gép több szalaggal (TSZ TG): $M = (Q, \Sigma\Gamma_0, q_{elfogad}, q_{elutasit})$ ahol:

1. Q , egy véged halmaz, az **állapotok halmaza**
2. Σ egy véges halmaz a bemeneti ábécé, mely nem tartalmazza a $_$ jelet
3. Γ , egy véges halmaz a szalag ábécé, ahol $_ \in \Gamma$ és $\Sigma \subseteq \Gamma$
4. $\delta : Q \times \Gamma^k \rightarrow Q \times \Gamma^k \times \{J, B\}^k$ az **állapotátmenet függvény**
5. $q_0 \in Q$ a kezdő állapot
6. $q_{elfogad} \in Q$ az elfogadó állapot
7. $q_{elutasit} \in Q$ az elutasító állapot

11.  **Megjegyzés.** A helyben maradást itt is megengedhetjük.

19.  **Tétel.** A TSZ TG-ek ekvivalensek a TG-ekkel.

6.  **Bizonyítás.** Legyen M egy k -szalagos TG. Megadunk egy S alap TG-et, mely szimulálja M -et.

1. Γ' tartalmazza Γ -t, minden Γ -beli elem megjelölt változatát, és a speciális elválasztó jelet. Az M gép k szalagjának tartalma egymás után, $_$ -val szeparálva szerepel S szalagján. A k fej pozícióján megjelölt szimbólum szerepel.
2. S megkeresi a szalagján az első és utolsó $(k+1)$ -dik $_$ jelet, és leolvassa a k db megjelölt szimbólumot, majd M állapotátmenete szerint elvégzi a megjelölt szimbólumok felülírását.
3. Ha egy $_$ jelre kellett lépni valamely fejnek, akkor oda egy megjelölt üres szimbólum kerül, miután a szalag tartalma az utolsó $_$ jelig eggyel jobbra csúsztatódott.

5.5 Turing gép változatok: nem determinisztikus

50.  **Definíció.** Nemdeterminisztikus Turing gép (ND TG): $M = (Q, \Sigma\Gamma_0, q_{elfogad}, q_{elutasit})$ ahol:

1. Q , egy véged halmaz, az **állapotok halmaza**
2. Σ egy véges halmaz a bemeneti ábécé, mely nem tartalmazza a $_$ jelet
3. Γ , egy véges halmaz a szalag ábécé, ahol $_ \in \Gamma$ és $\Sigma \subseteq \Gamma$
4. $\delta : Q \times \Gamma \rightarrow P = (Q \times \Gamma \times \{J, B\})$ az **állapotátmenet függvény**
5. $q_0 \in Q$ a kezdő állapot
6. $q_{elfogad} \in Q$ az elfogadó állapot

7. $q_{elutasít} \in Q$ az elutasító állapot

12.  **Megjegyzés.** A ND TG működését fa írja le. Ha egy ágon az elért állapot elfogadó, akkor a ND TG elfogadta a kezdeti sorozatot.

20.  **Tétel.** Minden M ND TG-hez van vele ekvivalens TG.

7.  **Bizonyítás.** m az M állapotátmeneteinek legnagyobb száma
S egy 3 szalagos TG lesz (mely, láttuk, ekvivalens valamely TG-pel).

- input szalag
- szimulációs szalag,
- cím szalag, az 1, 2, ... m számokból álló sorozatot tartalmaz, az elemeket , választja el.
pl: 1,11,23,45

S működése:

1. Kezdetkor az input szalagon szerepel az M-nek adott sorozat, a másik két szalag üres.
2. Az input szalagot a szimulációs szalagra másol.
3. A szimulációs szalagot használva S szimulálja M egy-egy ágának működését. Az csomópontokat szélességi kereséssel járja be. Az állapot átmeneteket a címszalagról olvassa le. Ha
 - (a) itt nincs több szimbólum, vagy
 - (b) a következő szimbólumnak megfelelő állapotátmenet nem megengedett, vagy
 - (c) az állapot elutasít, akkor 4-re lép.

Ha az állapot elfogad, akkor elfogadja a sorozatot.

4. A címszalagon levő címet felülírja a lexikografikusan következővel, és 2-re lép.

6. Hatodik előadás:

6.1 Turing-gépek

51.  **Definíció. Felsoroló Turing gép:** $M=(Q, \Sigma, \Gamma, \delta, q_0, q_{elfogad}, q_{elutasít})$ ahol:

1. Q , egy véges halmaz, az **állapotok halmaza**
2. nincsenek bemeneti sorozatok, csak munkaszalag
3. Γ , egy véges halmaz a szalag ábécé, ahol $_ \in \Gamma$ és $\Sigma \subseteq \Gamma$
4. $\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{J, B\} \times \Sigma^*$ az **állapotátmenet függvény**, ahol a Σ^* -beli sorozatok a **felsorolt(kinyomtatott) sorozatok**.
5. $q_0 \in Q$ a kezdő állapot

6. $q_{elfogad} \in Q$ az elfogadó állapot

7. $q_{elutasít} \in Q$ az elutasító állapot

Megjegyzés: Az összes felsorolt elemek (egyesekek esetleg többször is) alkotják a TG által felsorolt nyelvet.

21.  **Tétel.** Egy L nyelvet felsoroló E TG-hez létezik egy N TG, mely éppen L-et ismeri fel, és viszont.

Turing gép variációk: felsoroló TG:

M-hez E:

Legyen A az S által felismert nyelv. Legyen $s_1 s_2 \dots s_n \dots$ az összes lehetséges Σ^* -beli sorozat. E a következő képen működik:

1. $i = 1, 2, 3, \dots$ – *ra ismétli* :
2. Szimulálja M-et i lépésig, az $s_1, s_2, \dots, s_i$ szavakon.
3. Ha elfogadó állapotba ér M, kinyomtatja a megfelelő s_j szót.

6.2 Függvényszámítás Turing géppel:

52.  **Definíció.** $x \in \Sigma^*$ kezdeti bemeneti sorozatra az M Turing gép **outputja** az $y \in \Sigma^*$ sorozat, ha M megáll az x inputra, és megállásakor a szalagon éppen y szerepel (az üres szimbólumok előtt), valamint a fej ismét az első pozíción áll. Ezt a tényt $y=M(x)$ jelöli.

53.  **Definíció.** Az M által **kiszámított fv** az a $f_M : \Sigma^* \rightarrow \Sigma^*$ parciális függvény **parciálisan rekurzív**, ha van M TG, hogy $f = f_M$. Ha ezen felül f minden Σ^* -ra értelmezve van, akkor f **rekurzív**.

54.  **Definíció.** $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{elfogad}, q_{elutasít})$ TG, $0, 1, : \in \Sigma$ M **kiszámítja** az $f : \mathcal{B}^k \rightarrow \mathcal{B}$ függvényt, ha tetszőleges $w_1, \dots, w_k \in \mathcal{B}$ esetén a $w = w_1, \dots, w_k$ sorozatra $M(w)=u$, ahol $u = f(w_1, \dots, w_k)$.

22.  **Tétel. Church-Turing tézis** Ami algoritmussal kiszámítható/eldönthető, az Turing kiszámítható/eldönthető. Nevezetesen:

1. Egy $f : \Sigma^* \rightarrow \Sigma^*$ parciális fv algoritmussal kiszámítható $\Leftrightarrow$ f parciálisan rekurzív.
2. Egy $f : \Sigma^* \rightarrow \Sigma^*$ teljes fv algoritmussal kiszámítható $\Leftrightarrow$ f rekurzív.
3. L nyelvre a nyelvbe való tartozás problémája algoritmussal eldönthető $\Leftrightarrow$ a nyelv rekurzív.

23.  **Tétel. Hilbert 10 problémája** Diofantikus egyenletek általános megoldhatósága: adjunk algoritmust, mely tetszőleges diofantikus egyenletre eldönti, hogy az megoldható-e. $f(x_1, \dots, x_n) = 0$ egész megoldását keressük, ahol $f(x_1, \dots, x_n)$ egész együtthatós, n változós polinom.

7. Hetedik előadás:

7.1 Kongruens számok felismerése:

24.  **Tétel.** Egy m pozitív szám **kongruens**, ha van olyan derékszögű háromszög, melynek területe m , és oldalai racionális számok.

13.  **Megjegyzés.** K rekurzíve felsorolható: Mivel a rac. számok megszámlálhatók, az azokból alkotott párok is azok. Tehát ha van olyan p, q hogy $pq = 2m$ akkor az kiderül véges sok lépés után. Majd egy ilyen párra az is véges sok lépésben kiderül, ha az a Pythagorasz tételnek eleget tesz valamely r rac. számot választva.

25.  **Tétel.** Van olyan $L' \subseteq \Sigma^* = \{0, 1\}^*$ nyelv, amely nem Turing felismerhető, azaz nem rekurzíve felsorolható.

7.2 Egy nem Turing felismerhető nyelv:

14.  **Megjegyzés.** a $\{0, 1\}^*$ -beli szavak felsorolhatók. Pl. a kanonikusan: Az üres szót, majd az 1, 2, ... k , ... hosszúságúakat soroljuk fel. Az azonos hosszú sorozatokat lexikografikusan soroljuk fel, így: $\varepsilon, 0, 1, 00, 01, 10, 11, 000, \dots$

7.  **Jelölés.** M_w a w által kódolt Turing-gépet jelöli.

7.3 Egy nem Turing felismerhető nyelv: L_d

55.  **Definíció.** Diagonális nyelv: azon TG kódja, mely TG-ek nem fogadják el a saját kódjukat $L_d = \{w \in \{0, 1\}^*, \text{ az } M_w \text{ gép létezik, és } w \notin L_{M_w}\}$

26.  **Tétel.** L_d nem rekurzíve felsorolható nyelv.

8.  **Bizonyítás.** Indirekten, t. fel, hogy L_d rekurzíve felsorolható. Azaz van N TG, mely éppen L_d szavait ismeri fel, $L_d = L_N$. Azaz N felismeri w -t, ha w egy olyan M_w TG-t kódol, mely nem fogadja el a saját kódját.

Mi a helyzet N -nel? Legyen N kódja s . Felismeri-e N s -et, azaz $s \in L_d$ teljesül-e?

- Ha feltesszük hogy igen, akkor a def. szerint $s \notin L_d$, ELLENTMONDÁS!!
- Ha feltesszük hogy nem, akkor a def. szerint $s \in L_d$, ELLENTMONDÁS!!

Tehát mindkét esetben képtelenségre jutottunk, így az indirekt feltevésünk hamis.

15.  **Megjegyzés.** L_d : nem Turing felismerhető nyelv

7.4 Az univerzális Turing gép:

27.  **Tétel.** Van olyan U 3 szalagos TG, melyre teljesül, hogy ha $w, s \in \{0, 1\}^*$, és M_w létezik, akkor az U gép pontosan akkor fogadja el (utasítja el, kerül végtelen ciklusba) az ws bemenetet, amikor M_w az s bemenetet elfogadja (elutasítja, ciklusba kerül).

9.  **Bizonyítás.** A 3 szalag szerepe:

w#s	nem változik, M_w működését adja meg
s	M_w szalagjának a szimulálása
0	M_w pillanatnyi állapota

U működése:

1. Ellenőrzi, hogy w TG-t ír-e le, ha nem, elutasít. Ha igen, beírja s -ta 2., és 0-t (a kezdőállapot kódját) a 3. szalagra.
2. Az első szalag alapján meghatározza, hogy M_w mit lépne, ha a 3. szalagon megadott állapotban lenne, és az olvasófej alatt a 2. szalagon szereplő jelet látná. U megteszi M_w lépését, és megfelelően módosítja a 2. és 3. szalag tartalmát. U megáll elfogadó ill. elutasító állapotban, ha az M_w szerinti új állapot éppen elfogadó ill. elutasító. Ha nem, akkor az 1. lépést ismétli.

16.  **Megjegyzés.** U - univerzális Turing gép

56.  **Definíció.** Univerzális nyelv: $L_u = \{w\#s \in \{0, 1\}^*, M_w \text{ létezik, és } s \in L_{M_w}\}$

17.  **Megjegyzés.** L_u éppen az univerzális TG nyelve, azaz $L_u = L_U$, ahol U egy univ. TG. Tehát L_u rekurzíve felsorolható.

28.  **Tétel.** (Turing, 1936) L_u nem rekurzív nyelv.

7.5 A megállási probléma:

57.  **Definíció.** A megállási problémához (halting problem) tartozó L_h nyelv $L_h = \{w\#s \in \{0, 1\}^*, M_w \text{ létezik, és } s \text{ bemenettel elindítva megáll.}\}$.

29.  **Tétel.** L_h rekurzíve felsorolható, de nem rekurzív nyelv.

10.  **Bizonyítás.** Legyen U egy univ. TG. Ez L_h szavaira meg fog állni. Legyen U' az a TG, mely azonos U -val, kivéve hogy U elutasít állapotát elfogadra változtattuk. Ekkor $L_h = L_{U'}$, azaz L_h rek. felsorolható. Azt indirekte mutatjuk meg, hogy L_h nem rekurzív. Ind. felt: L_h rekurzív, azaz van olyan M TG, mely mindig megáll, és $L_h = L_M$. Legyen M' az a TG, mely először M -et futattja, majd ha M elfogadó állapotban végzett, U -t. M' mindig megáll az ind. felt. miatt, és M' éppen L_u -t ismeri fel. De L_u -ról láttuk, hogy nem rekurzív. Az ellentmondás oka az ind. feltevés, az hamis tehát, azaz a tétel második részének állítása is igaz.

7.6 A megállási probléma üres inputra:

58.  **Definíció.** $L_\varepsilon = \{w \in \{0,1\}^*, M_w \text{ létezik, és az } \varepsilon \text{ (üres) bementtel indítva megáll}\}$.

30.  **Tétel.** L_ε rekurzíve felsorolható.

11.  **Bizonyítás.** A rek. felsorolhatóságot bizonyítja, hogy egy $w \in L_\varepsilon$ esetén az L_h megállási nyelvet felismerő automatát a $w\varepsilon = w$ inputtal futtatva, ez a gép éppen L_ε -t ismeri fel.

31.  **Tétel.** L_ε nem eldönthető.

59.  **Definíció.** $L_1 \subseteq \Sigma^*$ visszavezethető az $L_2 \subseteq \Delta^*$ nyelvre, ha van olyan Turing-géppel kiszámítható $f : \Sigma^* \rightarrow \Delta^*$ függvény, hogy minden $w \in \Sigma^*$ szóra teljesül, hogy $w \in L_1 \Leftrightarrow f(w) \in L_2$.

7.7 Visszavezetések:

32.  **Tétel.** Ha $L_1 \leq L_2$ és

- L_2 eldönthető (rekurzív), akkor L_1 is eldönthető.
- L_2 Turing-felismerhető (rekurzívan felsorolható), akkor L_1 is Turing-felismerhető.

12.  **Bizonyítás.** pl. a visszavezetést kiszámoló és az L_2 -t eldöntő Turing-gép „kompozíciója” az L_1 nyelvet dönti el.

7.8 A megállási probléma eldönthetősége újra:

33.  **Tétel.** L_h eldönthetetlen.

13.  **Bizonyítás.** Megmutatjuk, hogy $L_u \geq L_h$. Tetszőleges M Turing-géphez és s bemenethez konstruáljuk meg az M' Turing-gépet a következő módon. M' annyiban különbözik M -től, hogy amikor M elutasító állapotba lép, akkor M' végtelen ciklusba lép. Minden további esetben az M' másolja M működését. Legyen w és w' rendre az M és az M' kódja. Nyilvánvaló, hogy az $f(w)=w'$ fgv. kiszámítható egy Turing-géppel.

7.9 L_ε eldönthetlensége újra

34.  **Tétel.** L_ε eldönthetetlen.

14.  **Bizonyítás.** Megmutatjuk, hogy $L_h \geq L_\varepsilon$. Adott M Turing-géphez és s bemenethez legyen M' a következő Turing-gép. M' tetszőleges u bemeneten a következőt teszi. M' megvizsgálja u-t, és ha u nem az üres szó, akkor M' elfogadja a bemenetet. Ha u az üres szó, akkor M felmásolja a szalagjára az s szót és elkezdi szimulálni az M-et az s szón. Legyen rendre w és w' az M és az M' kódja. Az $f(w)=w'$ fgv. kiszámítható egy Turing-géppel. Továbbá: $w\#s \in L_u \Leftrightarrow w' \in L_h$

8. Nyolcadik előadás

8.1 Rekurzíve felsorolható és nyelvtannal generálható nyelvek kapcsolata

35.  **Tétel.** A nyelvtannal generálható nyelvek osztálya megegyezik a rekurzíve felsorolható nyelvek osztályával. Azaz L nyelv akkor és csak akkor generálható grammatikával, ha van a nyelvet felismerő Turing -gép.

15.  **Bizonyítás.** Megmutatjuk, hogy ha L grammatikával generálható, akkor van M TG, hogy $L(M)=L$.
Legyen $G = (V, \Sigma, R, S)$ grammatika mely L-et generálja. A grammatika szerinti levezetés egy-egy lépését meghatározza, hogy mely szabályt alkalmazzuk, és annak bal oldalát az éppen aktuális mondatszerű forma hányadik elemére illesztjük.

36.  **Tétel. M működése:**

- Átmásolja a 3. szalagot a 4-re, majd törli 3 tartalmát.
- A 4. szalagon szereplő minden egyes mondatszerű formára:
 - Minden egyes $\alpha \rightarrow \beta$ szabályra megkeresi mindazon helyeket a mondatszerű formában, ahol a előfordul.
 - Minden egyes ilyen esetben a mondatszerű formát úgy toldja hozzá a 3. szalagon szereplő sorozatokhoz, hogy elválasztójel után a mondatszerű forma megfelelő helyén α helyett β szerepel.
- Megvizsgálja, hogy a 4. szalagon szerepel-e w-vel azonos mondatszerű forma. Ha igen, elfogadó állapotban megáll, ha nem, 1-re lép.

M éppen a G grammatika által generált szavakat fogja elfogadni.

8.2 Környezetfüggő nyelvtannal generálható nyelvek felismerése TG-pel

60.  **Definíció.** Lineárisan korlátozott TG egy olyan, speciális TG, mely a szalagnak csak a bemeneti által elfoglalt részét használja.

37.  **Tétel.** A környezetfüggő nyelvtannal generálható nyelvek osztálya (CS) megegyezik a lineárisan korlátozott TG-pel felismerhető nyelvek osztályával.

38.  **Tétel.** A környezetfüggő nyelvtannal generálható nyelvek osztálya (CS) valódi részhalmaza a rekurzív nyelvek osztályának.

8.3 Numerikus függvények

61.  **Definíció.** Az **alap függvények** a következők: $N \rightarrow N$ rákövetkező fv: $\text{succ}(n) = n+1$. $N^k \rightarrow N$ ($k \rightarrow 0$) k -változós zero fv: $\text{zerok}(n_1, \dots, n_k) = 0$. $N^k \rightarrow N$ ($k \geq 0$) k -változós j -dik identitás fv: $\text{idk}_j(n_1, \dots, n_k) = n_j$.

Bonyolultabb függvényeket a következő módokon kombinálunk:

62.  **Definíció.** Legyen $k, m \geq 0$, és adottak a $g : N^k \rightarrow N$ illetve $h_1, \dots, h_k : N^m \rightarrow N$ függvények. Ekkor g kompozíciója a $h_1, \dots, h_k$ függvényekkel azaz $f : N^m \rightarrow N$ fv, melyre:

$$f(n_1, \dots, n_m) = g(h_1(n_1, \dots, n_m), \dots, h_k(n_1, \dots, n_m)).$$

63.  **Definíció.** Legyen $k \geq 0$, és adottak a $g : N^k \rightarrow N$ illetve $h : N^{k+2} \rightarrow N$ függvények. Ekkor g és h által rekurzívan definiált függvény az az $f : N^{k+1} \rightarrow N$ fv, melyre:

$$f(n_1, \dots, n_k, 0) = g(n_1, \dots, n_k)$$

$$f(n_1, \dots, n_k, m+1) = h(n_1, \dots, n_k, m, f(n_1, \dots, n_k, m)).$$

64.  **Definíció.** A primitív rekurzív függvények az alap függvények, és a belőlük tetszőleges, véges számú kompozícióval és rekurzióval származtatott függvények.

39.  **Tétel.** A primitív rekurzív függvények számossága megszámlálhatóan végtelen.

16.  **Bizonyítás.** A primitív rekurzív függvények megadhatók az alap függvények, a kompozíció és rekurzió, valamint zárójelek és term. számok szimbólumaiból álló véges sorozatokkal, ilyenből megsz. végtelen van.

40.  **Tétel.** Nem minden kiszámítható fv. primitív rekurzív.

17.  **Bizonyítás.** Cantor-féle átlós módszerrel, unáris fv-ekre Soroljuk fel az összes primitív rekurzív unáris függvényt, $f_1, f_2, \dots, f_n, \dots$. Legyen g az az unáris függvény, melyre $g(n) = f_n(n) + 1$

65.  **Definíció.** Legyen $k \geq 0$. A $g: N^{k+1} \rightarrow N$ függvény minimalizációja az az $f: N^k \rightarrow N$ függvény, melyet a következő módon értelmezünk:

- $f(n_1, \dots, n_k) = a$ legkisebb m , melyre $g(n_1, \dots, n_k, m) = 1$, ha van ilyen m
- $f(n_1, \dots, n_k) = 0$ különben.

66.  **Definíció.** : Legyen $k \geq 0$. A $g: N^{k+1} \rightarrow N$ függvény minimalizálható, ha a fenti eljárás mindig megáll, azaz minden $n_1, \dots, n_k \rightarrow N$ esetén van olyan $m \in N$, hogy $g(n_1, \dots, n_k, m) = 1$.

67.  **Definíció.** Egy f fv μ -rekurzív, ha előállítható az alap függvényekből a kompozíció, a rekurzió és a minimalizáció operátorokkal, az utóbbit minimalizálható függvényre alkalmazva.

41.  **Tétel.** Az $f: N^k \rightarrow N$ függvény μ -rekurzív akkor és csak akkor, ha rekurzív, azaz kiszámítható Turing géppel.

9. Kilencedik előadás

9.1 Függvények asszimptotikus összehasonlítása

Hogyan értelmezzük algoritmusok hatékonyságát?

- Eldönthető nyelv (feladat) felismerése van-e praktikus használható algoritmus?
Milyen fajta feladatok vannak?
Eldöntendő kérdés pl. prímszám-e az adott szám? - nyelvbe tartozás
Hozzárendelés pl. lnko, Hamilton-kör keresés - fv kiszámítás
- Egy problémára (pl. szám prím mivoltának eldöntése, utazó ügynök) adott algoritmusokat hogyan értékeljük, hasonlítsunk össze?

Algoritmusok összehasonlítása:

- Időigény (lépésszám) illetve tárigény szerinti összehasonlítás
- A feladat nagyságának (a megadás hosszának) függvényében, a nagy feladatokra koncentrálva (asszimptotikus viselkedés)
- A legrosszabb esetet tekintve (worst case)
- Azonos fajta TG-pel és be és kimeneti reprezentációval megvalósított algoritmusok (ezt majd még pontosítjuk)

68.  **Definíció.** 'Nagy ordó': $f, g: N \rightarrow R^+$ függvények. $f(n) = O(g(n))$, ha találhatók c és n_0 pozitív. Azt mondjuk, hogy $g(n)$ **aszimptotikus felső korlátja** $f(n)$ -nek.

9.2 TG időkorlátja

8.  **Jelölés.** Egy adott M TG esetén:

- $T_M(n)$ jelöli M maximális lépésszámát (számolási idejét) az n hosszúságú bemenetek esetén.
- $S_M(n)$ jelöli az M által maximálisan elolvasott munkaszalag jelek számát n hosszúságú bemenetek esetén.

18.  **Megjegyzés.** Ha M nem áll meg valamely n hosszú bemenetre, akkor $T_M(n)$ végtelen. Ekkor $S_M(n)$ -t is végtelennek tekintjük.

69.  **Definíció.** Legyen $t : N \rightarrow N$ függvény olyan hogy $t(n) \geq n$ minden n -re. Az M TG **$t(n)$ időkorlátos**, ha az n hosszú bemenetek esetén legfeljebb $t(n)$ lépést tesz meg, azaz $T_M(n) \geq t(n)$.

70.  **Definíció.** $TIME(t(n)) = \{L | L \text{ felismerhető egy } O(t(n)) \text{ időkorlátos M TG-pel.}\}$.

71.  **Definíció.** Legyen $s : N \rightarrow N$ függvény olyan, hogy $s(n) \log_2 n$ minden n -re. Az M TG **$s(n)$ tárkorlátos**, ha az n hosszú bemenetek esetén legfeljebb $s(n)$ db cellát használ a munkaszalagon, azaz $S_M(n) \geq s(n)$.

72.  **Definíció.** $SPACE(s(n)) = \{L | L \text{ felismerhető } O(s(n)) \text{ tárkorlátos M TG-pel.}\}$.

9.3 A P osztály

73.  **Definíció.** $P = \cup_{k \leq 1} TIME(n^k)$ a polinom időben egyszalagos TG-pel eldönthető nyelvek osztálya.

19.  **Megjegyzés.** Valamely, az alappal ekvivalens speciális TG modellt használva, ha az polinom időben szimulálható az alap TG-pel, akkor P invariáns arra nézve, hogy az alap vagy a speciális TG-et használjuk modellként.

20.  **Megjegyzés.** P a praktikus számítógéppel megoldható problémáknak felel meg. Általában találunk P-beli problémákra gyors (négyzetes, köbös, ...) algoritmust.

Példák P-beli problémákra Ismert polinomiális algoritmus az alábbi problémákra:

- PATH = $\langle G, s, t \rangle$ — G irányított gráf, és van benne irányított út s -ből t -be.
- RELPRIME = $\langle x, y \rangle$ — x és y relatív prímek
- GRAPH-2-COLOR = G — G gráf csúcsai kiszínezhetők 2 színnel
- EULER-CYCLE = G — G gráfban van EULER kör

- $\text{PRIME} = x \rightarrow x \text{ prím}$

Példák mindmáig nem P-beli problémákra Nem sikerült eddig polinomiális algorimust találni pl az alábbi problémákra:

- $\text{HAMILTON-CYCLE} = G \rightarrow G \text{ gráfban van Hamilton kör}$
- $\text{GRAPH-3-COLOR} = G \rightarrow G \text{ gráf csúcsai kiszínezhetők 3 színnel}$

9.4 TG-ek szimulációjának idő és tárigénye

Láttuk, hogy speciális TG-ek (többszalagos, RAM, ... nem determinisztikus) szimulálhatók alap TG-pel. Ehhez több idő illetve tár kell, mennyivel? Mi az egyszerűbb gépeken a hatékonyságromlás mértéke?

42.  Tétel. Az N k -szalagos TG-hez van olyan egyszalagos M TG, hogy $L(M)=L(N)$, és ha N $f(n)$ időigényű, akkor M $O(f(n)^2)$ időigényű.

43.  Tétel. Az N TG-hez van olyan többszalagos M TG, hogy egyalkalmas n_0 számon túl:

1. tetszőleges $\varepsilon > 0$ esetén $T_M(n) \geq n(1 + \varepsilon)$, ha $T_N(n) = O(n)$
2. tetszőleges $c > 0$ esetén $T_M(n) \geq cT_N(n)$, ha $\lim T_N(n)/n = \infty$

21.  Megjegyzés. A fenti tétel magyarázza, hogy miért tekinthettünk el a konstans szorzótól az időigény definiálásakor.

44.  Tétel. Tár-idő tétel Ha $L \in \text{SPACE}(s(n))$, akkor van olyan L -től függő c konstans, hogy $L \in \text{TIME}(2^{cs(n)})$.

18.  Bizonyítás. vázlat Legyen M egy L -et $O(s(n))$ tárkorlással eldöntő Turing-gép és legyen w az M egy tetszőleges n hosszú bemenete. Ekkor a kezdőkonfigurációból elérhető konfigurációk száma $cn2^{O(s(n))} = 2^{O(s(n))}$, ahol c egy a szalagábécétől függő konstans.

Mivel M $2^{O(s(n))}$ lépése után konfiguráció ismétlődés áll fenn, ügyes konstrukcióval elérhető, hogy az M -et szimuláló N Turing-gép $2^{O(s(n))}$ lépés után észrevegye ha végtelen ciklusba esik. Következik, hogy N az L -et ismeri fel és $2^{O(s(n))}$ időigényű, ami bizonyítja a tétel állítását.

9.5 Nyelvosztályok komplementere

9.  Jelölés. L I feletti nyelv esetén $L^C = I * /L$ az L komplementere

74.  Definíció. L nyelvosztály esetén $\text{co}L = \{L | L^C \in L\}$ az L nyelvosztály komplementere.

45.  Tétel. 1. $\text{co}(\text{co}(L)) = L$

2. Ha $\mathcal{A} \subseteq \mathcal{B}$ nyelvosztályok, $\text{co } \mathcal{A} \subseteq \text{co } \mathcal{B}$

19.  **Bizonyítás.** 1. $(L^C)^C = L$ alapján

2. $L \in \text{co } \mathcal{A} \rightarrow L^C \in \mathcal{A} \rightarrow L^C \in \mathcal{B} \rightarrow L \in \text{co } \mathcal{B}$

9.6 Tár-idő tételek komplement nyelvosztályokra

46.  **Tétel.** $\text{TIME}(t(n)) = \text{coTIME}(t(n))$, azaz $L \in \text{TIME}(t(n))$ akkor és csak akkor, ha $L^C \in \text{TIME}(t(n))$.

20.  **Bizonyítás.** Az L -et eldöntő M TG-ben az elfogadó és elutasító állapotokat felcserélve, egy azonos futásidejű, L^C -t eldöntő TG-et kapunk, tehát $L^C \in \text{TIME}(t(n))$, azaz $\text{coTIME}(t(n)) \subseteq \text{coTIME}(t(n))$.

Másrészt $\text{TIME}(t(n)) = \text{coTIME}(t(n))$.

Tehát $\text{TIME}(t(n)) = \text{coTIME}(t(n))$.

47.  **Tétel.** $\text{SPACE}(s(n)) = \text{coSPACE}(s(n))$, azaz $L \in \text{SPACE}(s(n))$ akkor és csak akkor, ha $L^C \in \text{SPACE}(s(n))$.

21.  **Bizonyítás.** Legyen $L \in \text{SPACE}(s(n))$. A tár-idő tételbeli N gép minden inputra megáll, $O(s(n))$ tárkorlátos. Ebben felcserélve az elfogadó és elutasító állapotokat, adódik, hogy $L^C \in \text{SPACE}(s(n))$. A más irány mint előbb.

75.  **Definíció.** $\text{EXPTIME} = \bigcup_{k \geq 1} \text{TIME}(2^{n^k})$ az exponenciális időben felismerhető nyelvek osztálya.

76.  **Definíció.** $\text{SPACE} = \bigcup_{k \geq 1} \text{SPACE}(2^{n^k})$ az exponenciális időben felismerhető nyelvek osztálya.

48.  **Tétel.** $\mathbf{P} \subseteq \mathbf{PSPACE} \subseteq \mathbf{EXPTIME}$

22.  **Bizonyítás.** Ha M TG $t(n)$ időkorlátos, akkor $t(n)$ tárkorlátos is (lépésenként max egy jelet ír a munkaszalagra). Tehát $\text{TIME}(n^k) \subseteq \text{SPACE}(n^k)$, és így $\mathbf{P} = \bigcup_{k \geq 1} \text{SPACE}(n^k) = \mathbf{PSPACE}$

Ha $L \in \mathbf{PSPACE}$, akkor valamely k -ra $L \in \text{SPACE}(n^k)$. A tár-idő tétel szerint ekkor hogy $L \in \text{TIME}(2^{n^k}) \subseteq \text{EXPTIME}$.

49.  **Tétel.** $\text{EXPTIME} \subset \mathbf{R}$ ($=$ Rekurzívnyelvekosztálya)

23.  **Bizonyítás.** $\subseteq$ következik EXPTIME időkorlátosságából. A valódi tartamazáshoz mutatunk egy L nyelvet, mely rekurzív, de nem EXPTIME -beli, $L = w \in 0,1^*$, az M_v gép létezik, és legfeljebb $2^{|w|}$ lépésben elutasítja w -t, ahol $|w|$ a szó hossza.

1. L -nek végtelen sok eleme van, mivel egy M_w -t kibővíthetünk a kiinduló helyzetből soha nem elérhető állapotokkal. Egy ilyen y leírás esetén M_y éppen úgy fog működni mint M_w , tehát az időkorlátja is azonos.
2. Megmutatjuk, hogy L semmilyen n -re nincs benne $\text{TIME}(2^{2^{n-1}})$ -ban. Ez elég is, mivel $\text{EXPTIME} \subseteq \text{TIME}(2^{2^{n-1}})$.

Indirekte feltesszük, hogy L felismerhető $c2^{2^{n-1}}$ időkorlátos M TG-pel. Legyen n_0 olyan szám, hogy nn_0 esetén $c2^{2^{n-1}} < 2^{2^n}$. Legyen w egy n_0 -nál hosszabb szó, melyre M_w létezik és úgy viselkedik mint M . Most megmutatjuk, hogy mind $w \in L$, mind a $w \notin L$ feltételezés ellentmondásra vezet. Ha $w \in L$, akkor M_w elfogadja w -t $c2^{2^{n-1}} < 2^{2^n}$ lépésben, de ekkor L def. szerint $w \notin L$. Ha $w \notin L$, akkor M_w nem utasítja el w -t $c2^{2^{n-1}} < 2^{2^n}$ lépésben, de megáll, ezért elfogadja w -t, amiből $w \in L$ következne.

10. Tizedik előadás

10.1 Nemdeterminisztikus TG időkorlátja

77.  **Definíció.** Legyen $t: \mathbb{N} \rightarrow \mathbb{N}$ függvény olyan, hogy $t(n) \geq n$ minden n -re. Az M NDTG $t(n)$ időkorlátos, ha az n hosszú bemenetek esetén M minden számítási út mentén legfeljebb $t(n)$ lépést megtéve megáll. Azaz n hosszú bemenetek esetén legfeljebb $t(n)+1$ mély a számítási fa.

22.  **Megjegyzés.** Nem ismert olyan fizikai megvalósítás, mely $t(n)$ -nel arányos időben szimulálná egy $t(n)$ időkorlátos NDTG működését.

78.  **Definíció.** $\text{NTIME}(t(n)) = L \subseteq I^* \mid L \text{ felismerhető egy } O(t(n)) \text{ időkorlátos } M \text{ NDTG-pelel..}$

10.2 Az NP nyelvosztály

79.  **Definíció.** $\text{NP} = \cup_{k \geq 1} \text{NTIME}(n^k)$ a polinom időben egyszalagos NDTG-peleldönthető nyelvek osztálya.

23.  **Megjegyzés.** $P \subseteq \text{NP}$ mivel egy $t(n)$ időkorlátos TG tekinthető egy speciális, $t(n)$ időkorlátos NDTG-nek, egy $\text{TIME}(n^k) \subseteq \text{NTIME}(n^k)$.

50.  **Tétel.** $P \subseteq \text{NP} \cap \text{coNP}$

24.  **Bizonyítás.** $P \subseteq \text{NP}$ láttuk. De co operátor tartalmazásmegőrző volta miatt ekkor $\text{coP} \subseteq \text{coNP}$.
Láttuk, hogy $\text{coP} = P$, tehát P is teljesül.

80.  **Definíció.** Legyen L egy Σ feletti nyelv. L polinomiálisan bizonyítható, ha van olyan V algoritmus, hogy

$$L = \{w \mid \exists V \text{ elfogadja } \langle w, c \rangle \text{ valamely } \Sigma^* \text{ -beli } c \text{ -re,}$$

és V polinomidejű w hosszát tekintve.

51.  **Tétel.** NP azokat a nyelveket tartalmazza, amelyek polinomiálisan bizonyíthatók.

81.  **Definíció.** A és B Σ^* feletti nyelvek. A polinomidőben visszavezethető B -re, ha van

$$f : \Sigma^* \rightarrow \Sigma^*$$

Polinomiális időben kiszámítható fv, hogy $w \in A \Leftrightarrow f(w) \in B$.

52.  **Tétel.** Ha $A \leq_p B$ és $B \in P$, akkor $A \in P$.

25.  **Bizonyítás.** Legyen M az a TG, mely polinom időben felismeri B -t. Ekkor a következő N TG A -t ismeri fel polinom időben:
w input esetén N :

- Kiszámítja $f(w)$ -t,
- M -et futtatja $f(w)$ -n, és outputként M outputját adja.

10.3 NP-teljes problémák

82.  **Definíció.** L NP-teljes, ha

- L NP-beli, és
- tetszőleges A NP-beli probléma polinom időben visszavezethető L -re.

83.  **Definíció.** Bool változók igaz (1) vagy hamis (0) értéket vehetnek fel.

53.  **Tétel.** Ha B NP-teljes és $B \in P$, akkor $NP = P$.

54.  **Tétel.** Ha B NP-teljes, C NP-beli, és $B \leq_p C$, akkor C is NP-teljes.

55.  **Tétel.** (Cook-Levin) $SAT \in P$ akkor és csak akkor, ha $NP = P$. Másképp fogalmazva, SAT NP-teljes.

84.  **Definíció.** $3SAT = \{\Phi \mid \Phi = (a_1 \vee b_1 \vee c_1)(a_2 \vee b_2 \vee c_2) \dots (a_k \vee b_k \vee c_k)\}$ valamely k -ra, ahol tetszőleges a_i, b_i, c_i egy Boole-változó vagy annak negáltja Φ kielégíthető Boole-kifejezés.

56.  **Tétel.** VERTEX COVER (lefogható-e G minden éle legfeljebb k ponttal) NP-teljes.
57.  **Tétel.** k -COLORABILITY (kiszínezhető-e G minden éle legfeljebb k színnel) NP-teljes $k \geq 3$ esetén.
58.  **Tétel.** HAMPATH NP-teljes.
26.  **Bizonyítás.** 3SAT polinom időben visszavezethető HAMPATH-ra.
85.  **Definíció.** $NSPACE = \bigcup_{k \geq 1} NSPACE(n^k)$, ahol $NSPACE(S(n)) = L$ — L felismerhető egy $O(s(n))$ tárkorlátos M nemdeterminisztikus TG-pel.

10.4 Savitch tétele

59.  **Tétel.** Ha $f(n) \geq O(\log_2(n))$, akkor $NSPACE(F(n)) \subseteq DSPACE((f(n))^2)$
27.  **Bizonyítás.** Egyrészt, Savitch megmutatta, hogy a $PATH \in TIME((\log_2(n))^2)$. Másrészt, legyen N egy $f(n)$ tárigényű nemdeterminisztikus TG. Az N konfigurációs gráfja egy n hosszú bemeneten $2^{O(f(n))}$, és ebben a gráfban kell egy determinisztikus TG-nek egy utat keresni a kezdőkonfigurációból egy elfogadó konfigurációba. Savitch tétele alapján ez elvégezhető $(\log_2(2^{O(f(n))}))^2$ azaz $O((f(n))^2)$ tárral, ami bizonyítja az állítást.
2.  **Következmény.** $PSPACE = NSPACE$

10.5 Problémaosztályok tartalmazása

60.  **Tétel.** $PSPACE \subseteq EXPTIME$
28.  **Bizonyítás.** Legyen $L \in PSPACE$, és N egy L -et $f(n)$ tárral felismerő TG. Korábban láttuk, hogy N konfigurációs gráfja egy n hosszú bemeneten $2^{O(f(n))}$ méretű. Ebben a gráfban kell egy M determinisztikus TG-nek egy elfogadó konfigurációt keresni. Mivel a PATH probléma P-beli, ez megoldható $2^{O(f(n))}$ időben. Mivel $f(n)$ egy polinom függvény kapjuk, hogy $L \in EXPTIME$.
86.  **Definíció.** Egy L nyelv $PSPACE$ -teljes, ha $PSPACE$ -beli és minden további $PSPACE$ -beli nyelv polinomidőben visszavezethető L -re.
87.  **Definíció.** QBF (True quantified Boolean formula) $= (\phi) : \phi$ igaz prenex alakú zárt Boole-formula
61.  **Tétel.** QBF $PSPACE$ teljes.

alma ♡