

DIGITÁLIS
SZÁMÍTÁS
ELMÉLETE

2008

■

ELMÉLET

összeállította:
Esztergár-Kiss Domokos

Determinisztikus Végtes Automaták (=DVA):

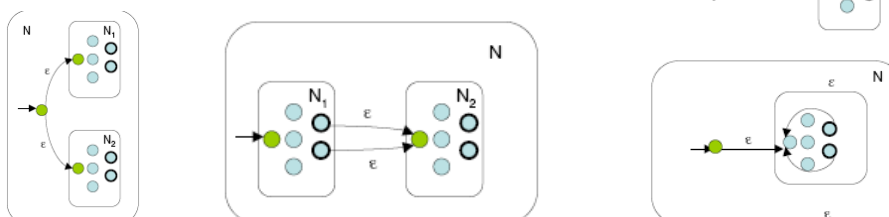
- $M = (Q, \Sigma, \delta, q_0, F)$, ahol:
 - Q ..véges, állapotok halmaza,
 - Σ ..véges halmaz, abc,
 - $\delta: Q \times \Sigma \rightarrow Q$..állapotátmenet fv,
 - $q_0 \in Q$..kezdeti áll,
 - $F \subseteq Q$..végáll;;
- Felismert (=elfogadott) jelsorozat: $M = (Q, \Sigma, \delta, q_0, F)$ egy DVA, $w = x_1 x_2 \dots x_n \in \Sigma$ jelek sorozata, M felismeri w -t, ha $\exists r_0 r_1, \dots, r_n \in Q$ állapotok sorozata, hogy: $r_0 = q_0$, $\delta(r_i, x_{i+1}) = r_{i+1}$, $r_n \in F$;
- M DVA felismeri A nyelvet, ha $A = \{w \mid M \text{ felismeri } w\}$, jele: $A = L(M)$;;
- összes Σ –beli jelekből álló végtes sorozatok (=szavak) halmaza, jele: $\Sigma^* = \{w = a_1 \dots a_n \mid a_i \in \Sigma, i=1, \dots, n, n \geq 0\}$, szó hossza: n , ε ..0 hosszú, üres szó; L ..bináris abc ($0 \vee 1$);;
- L formális nyelv: Σ felett, ha $L \subseteq \Sigma^*$;
- nyelv végtes módon megadható (nem minden Σ feletti nyelvet lehet végtesen reprezentálni):
 - elemek felsorolása (ha számuk végtes),
 - $L = \{w \in \Sigma^* \mid w \text{ P tulajdonságú}\}$,
 - $L = L(M)$, ahol: M egy automata,
 - operátorokkal,
 - generáló nyelvtannal;;
- Reguláris operátorok: Σ abc feletti nyelvekben:
 - egyesítés (=unió): $A \cup B = \{x \mid x \in A \text{ vagy } x \in B\}$;
 - szorzat (=illesztés, konkatenáció): $A \circ B = \{xy \mid x \in A, y \in B\}$;
 - iterálás (=Kleene csillag): $A^* = \{w_1 \dots w_k \mid w_i \in A, i=1, \dots, k, k \geq 0\}$, $\varepsilon \in A^*$, ha $L = \Sigma$, akkor $L^* = \Sigma^*$; további operátorok:
 - metset: $A \cap B = \{w \mid w \in A \text{ és } w \in B\}$;
 - különbség: $A - B = \{w \mid w \in A \text{ és } w \notin B\}$, $A^C = \Sigma^* - A$;
 - tükörkép: $A^R = \{x^R \mid x \in A\}$;;
- Reguláris kifejezések: Σ abc felett reg kif:
 - Σ minden eleme,
 - ε ,
 - üres,
 - ha A és B reguláris, akkor A^* , $A \circ B$, $A \cup B$ is; minden reg kif reprezentál egy nyelvet;;

Nemdeterminisztikus véges automata (=NVA):

- $N = (Q, \Sigma, \delta, q_0, F)$, ahol:
 - Q ..véges, állapotok halmaza,
 - Σ ..véges halmaz, abc,
 - $\delta: Q \times (\Sigma \cup \{\varepsilon\}) \rightarrow P(Q)$..állapotátmenet fv,
 - $q_0 \in Q$..kezdeti áll,
 - $F \subseteq Q$..végáll;;
- Felismert (=elfogadott) jelsorozat: $N = (Q, \Sigma, \delta, q_0, F)$ egy NVA, $w = x_1 x_2 \dots x_n \in \Sigma^*$ jelek sorozata, N felismeri w -t, ha $\exists r_0 r_1, \dots, r_n \in Q$ állapotok sorozata, hogy: $r_0 = q_0, r_{i+1} \in \delta(r_i, x_{i+1}), r_n \in F$;
- N NVA felismeri A nyelvet, ha $A = \{w \mid N \text{ felismeri } w\}$, jele: $A = L(N)$;;
- M_1 és M_2 ekvivalens: ha $L(M_1) = L(M_2)$, ua nyelvet definiálják;
- T: Minden NVA-hoz létezik vele ekvivalens DVA;
 - Biz: ha N nem tartalmaz ε -t: megkonstruáljuk $M = (Q', \Sigma', \delta', q_0', F')$ DVA-t, amivel NVA-t szimuláljuk:
 - $Q' = P(Q)$,
 - $\delta'(R, a) = \{q \in Q \mid \exists r \in R, \text{ hogy } q \in \delta(r, a)\} = \bigcup_{r \in R} \delta(r, a)$,
 - $q_0' = \{q_0\}$,
 - $F' = \{R \in Q' \mid \exists r \in R, \text{ hogy } r \in F\}$;
 - ha N tartalmaz ε -t: $E(R) = \{q \in Q \mid q \text{ elérhető } R\text{-ből } 0 \text{ v több } \varepsilon \text{ él mentén}\}$, $R \subseteq Q$;
 - $Q' = P(Q)$,
 - $\delta'(R, a) = \{q \in Q \mid \exists r \in R, \text{ hogy } q \in E(\delta(r, a))\} = \bigcup_{r \in R} E(\delta(r, a))$,
 - $q_0' = E(\{q_0\})$,
 - $F' = \{R \in Q' \mid \exists r \in R, \text{ hogy } r \in F\}$;;
- T: legyen L azon nyelvek halmaza, melyek felismerhetően DVA-val, L zárt a reg op-ra nézve, azaz, ha A_1 és $A_2 \in L$, akkor:
 - $A_1 \cup A_2 \in L$,
 - $A_1 \circ A_2 \in L$,
 - $A_1^* \in L$;
- Biz1: Legyen $A_1 = L(M_1)$, $M_1 = (Q_1, \Sigma_1, \delta_1, q_1, F_1)$, $A_2 = L(M_2)$, $M_2 = (Q_2, \Sigma_2, \delta_2, q_2, F_2)$,
 - megkonstruáljuk: $M = (Q, \Sigma, \delta, q_0, F)$ -t, hogy $L(M) = A_1 \cup A_2$,
 - $\Sigma = \Sigma_1 = \Sigma_2$,
 - $Q = Q_1 \times Q_2 = \{(r_1, r_2) \mid r_1 \in Q_1 \text{ és } r_2 \in Q_2\}$,
 - $\delta((r_1, r_2), a) = (\delta_1(r_1, a), \delta_2(r_2, a))$,
 - $q_0 = (q_1, q_2)$,
 - $F = \{(r_1, r_2) \mid r_1 \in F_1 \text{ vagy } r_2 \in F_2\} = (F_1 \times Q_2) \cup (Q_1 \times F_2)$;
- Biz1 NVA-val: $L(N_1) = A_1$, $L(N_2) = A_2$, $L(N) = A$, 2 db ε átmenet köti össze N_1 és N_2 -t, új kezdőáll;
- Biz 2: N_1 végállapotaiból ε éllel N_2 -be;
- Biz3: kezdőáll-ból ε hatására tovább;;;

Bizonyítás:

3. $A = A_1^*$



Általánosított NVA (=ÁNVA):

- $N = (Q, \Sigma, \delta, q_k, q_v)$, ahol:
 - Q ..véges, állapotok halmaza,
 - Σ ..véges halmaz, abc,
 - $\delta: (Q - \{q_v\}) \times (Q - \{q_k\}) \rightarrow R$..állapotátmenet fv., ahol R Σ feletti reg kif-ek halmaza,
 - $q_k \in Q$..kezdeti áll,
 - $q_v \in Q$..végáll;;
 - élek reg kif-el címkézettek;
 - kezdőáll-ból minden más állapotba megy él, de bele nem;
 - egyetlen a kezdőáll-tól kül végáll;
 - közbülső áll mindegyikéből megy egyetlen él minden közbülső állapotba, saját magába is;;
- Felismert (=elfogadott) jelsorozat: N egy ÁNVA, $w = x_1 x_2 \dots x_n \in \Sigma^*$ jelek sorozata, N felismeri w -t, ha $\exists r_0, r_1, \dots, r_n \in Q$ állapotok sorozata, hogy: $r_0 = q_k$, $x_i \in L(R_i)$. ahol: $R_i = \delta(r_{i-1}, r_i)$, azokból szavakból áll, amit felismer, $r_k = q_v$;
- N ÁNVA felismeri A nyelvet, ha $A = \{w \mid N \text{ felismeri } w\}$, jele: $A = L(N)$;;
- Ekvivalencia: adott Σ feletti nyelvek azonos halmazok:
 - L_R ..reguláris nyelvek halmaza,
 - L_N ..NVA által felismert nyelvek,
 - L_D ..DVA által felismert nyelvek;
 tudjuk: $L_R \subseteq L_N$ és $L_N = L_D \Rightarrow L_R \subseteq L_D$; megmutatjuk: $L_D \subseteq L_R$;;
- Biz: megkonstruáljuk ÁNVÁ-kat:
 - M ..kiindulási k állapotú DVA,
 - A_{k+2} .. M -el ekv $k+2$ állapotú ÁNVA,
 - A_{k+1} .. A_{k+2} -vel ekv $k+1$ állapotú ÁNVA, ...,
 - A_2 .. A_m -el ekv 2 állapotú ÁNVA, az A_2 egyetlen élén szereplő reg kif-t keressük (ami leírja azt a nyelvet, ami M -et határozza meg);
- A_{k+2} -t így kapjuk: hozzáadunk Q -hoz q_k/q_v -t, melyből/be minden F -beli áll-ból ε címkéjű él vezet, többszörös éleket újjal helyettesítjük, melyre kihagyott éleken szimbólumok unióját írjuk;
- konvertál(G) nevű rek eljárást A_{k+2} -re alkalmazva előállítunk reg kif-t:
 - ha $k=2$ (áll-ok száma), akkor visszaadjuk egyetlen élen lévő reg kif-t;
 - ha $k>2$: legyen $q' \in Q - \{q_k, q_v\}$ tetsz, $G' = (Q', \Sigma, \delta', q_k, q_v)$, ahol:
 - $Q' = Q - \{q'\}$ és $\delta'(q_i, q_j) = R_1 R_2^* R_3 \cup R_4$, ahol: $R_1 = \delta(q_i, q')$, $R_2 = \delta(q', q')$, $R_3 = \delta(q', q_j)$, $R_4 = \delta(q_i, q_j)$;
- konvertál(G) ekv G -vel: Biz: TI: $k = |Q|$,
 - $k=2$ -re igaz,
 - tfh $k-1$ -re igaz, megmutatjuk, hogy k -ra is: $L(G) = L(G')$, ehhez biz:
 - 1) $L(G) \subseteq L(G')$: legyen $w \in L(G)$ és $q_k, q_1, q_2, \dots, q_v$ felismerhető áll-ok sorozata,
 - ha q' nem szerepel, akkor ez sorozat Q' -beli is, így $w \in L(G')$;
 - ha $\dots, q_i, q', \dots, q', q_j, \dots$ szerepel, akkor: $w = \dots w_i, w_1', \dots, w_s', w_j$ és $w_i \in \delta(q_i, q') = R_1$, $w_t' \in \delta(q', q') = R_2$, $t=1, \dots, s$, $w_j \in \delta(q', q_j) = R_3$, tehát $\dots, q_i, q_j, \dots$ elfogadó áll-ok sorozata Q' -ben, mert $\delta'(q_i, q_j) = R_1 R_2^* R_3 \cup R_4$, így $w \in L(G')$;
 - 2) $L(G') \subseteq L(G)$: legyen $w \in L(G')$ és $q_k, q_1, q_2, \dots, q_v$ felismerhető áll-ok sorozata, fentiekhez hasonlóan belátható, hogy: $w \in L(G)$;;

Véges átalakítók (=fordítók):

- $M = (Q, \Sigma, \Gamma, \delta, q_0)$, ahol:
 - Q ..véges, állapotok halmaza,
 - Σ ..véges halmaz, abc,
 - Γ ..véges kimeneti abc,
 - $\delta: \Sigma \times Q \rightarrow Q \times \Gamma$..állapotátmenet fv,
 - $q_0 \in Q$..kezdeti áll;;
- véges sok bemenő jel bemeneti szalagról- véges sok áll- véges sok kimenő jel kimeneti szalagra;;;

Minimális DVA:

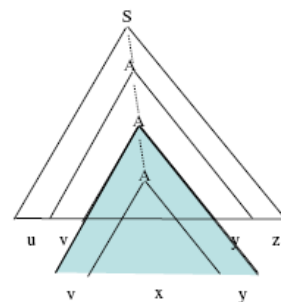
- $x, y \in \Sigma^*$ L-re nézve ekvivalens: $L \subseteq \Sigma^*$ nyelv,
 - ha minden $z \in \Sigma^*$ esetén: $xz \in L$ csakkor $yz \in L$,
 - *jele:* $\approx_L$ (L nyelv szerinti reláció minden stringek halmazán);
 - $\approx_L$ ekvivalencia reláció Σ^* -on (refl, szimm, tranz), mely ekv osztályokra bontja Σ^* -t (*jele:* $[x]$, $x \in \Sigma^*$, különböznek és együtt kiadják Σ^* -t);;
- M egy konfigurációja: $(r, x) \in Q \times \Sigma^*$;
- (r, x) rákövetkezője (r', x') : , ha $x' = ax$, $\delta(r, a) = r'$, ahol: $a \in \Sigma$, *jele:* $(r, x) \vdash_M (r', x')$;
- (r', x') elérhető (r, x) -ből: ha $(r, x) \vdash_M (r_1, y_1) \vdash_M \dots \vdash_M (r_k, y_k) \vdash_M (r', x')$, $k \geq 0$, *jele:* $(r, x) \vdash^*_M (r', x')$;;
- $x, y \in \Sigma^*$ M-re nézve ekvivalens: M DVA, ha $\exists q \in Q$, hogy:
 - $(q_0, x) \vdash^*_M (q, \varepsilon)$ és $(q_0, y) \vdash^*_M (q, \varepsilon)$, (ua áll-ba jutnak számítási folyamat során),
 - *jele:* $\sim_M$ (2 string egy automatára nézve ekv);
 - $\sim_M$ ekvivalencia reláció Σ^* -on, minden q elérhető állapothoz tartozik ekv osztály (*jele:* $[q]$; azon stringek, amik kezdeti áll-ból q-ba jutnak);;
- T: legyen $M = (Q, \Sigma, \delta, q_0, F)$ DVA, ekkor: minden $x, y \in \Sigma^*$ esetén $x \sim_M y \rightarrow x \approx_{L(M)} y$,
- Biz: $x \in \Sigma^*$ esetén legyen $q(x)$ állapot, melyre:
 - $(q_0, x) \vdash^*_M (q(x), \varepsilon)$, így $x, z \in \Sigma^*$ esetén: $xz \in L(M)$ csakkor $(q(x), z) \vdash^*_M (q', \varepsilon)$, ahol: $q' \in F$;
 - ha $x \sim_M y$, akkor $q(x) = q(y)$, tehát: $xz \in L(M)$ csakkor $yz \in L(M)$ minden $z \in \Sigma^*$ esetén, azaz $x \approx_{L(M)} y$;;
 - $\sim_M$ finomítása $\approx_{L(M)}$ -nek (ha automata szerint ua osztályba esik, akkor nyelv szerint is);;
- T: ha $L \subseteq \Sigma^*$ reg nyelv, akkor $\exists M = (Q, \Sigma, \delta, q_0, F)$ DVA, hogy $L(M) = L$ és állapotainak száma n ($\approx_L$ reláció által meghatározott ekv oszt száma), M egyértelmű;
- Biz: legyen
 - 1) $Q = \{[x] \mid x \in \Sigma^*\}$ véges, mivel L reg, $\exists M'$ DVA, hogy $L = L(M')$, de tudjuk, hogy $\sim_{M'}$ finomítása $\approx_L$ -nek, így $\approx_L$ -nek max annyi ekv osztálya van, mint $\sim_{M'}$ -nek, ami pedig véges;
 - 2) $\delta([x], a) = [xa]$ jól definiált, azaz ha $x' \in [x]$, akkor $[xa] = [x'a]$, de tudjuk: $xa \approx_L x'a$ csakkor $x \approx_L x'$;
 - 3) $q_0 = [\varepsilon]$, $F = \{[x] \mid x \in L\}$;

- 4) megmutatjuk, hogy $L=L(M)$: $y \in \Sigma^*$ esetén $y \in L$ csak akkor $([\varepsilon], y) \vdash_M^* (q, \varepsilon)$, ahol: $q \in F$, és igaz: $q = [\varepsilon y] = [y]$, ezért: $y \in L$ csak akkor $y \in L(M)$;
- 5) $([x], y) \vdash_M^* ([xy], \varepsilon)$, y hossza szerinti TI:
 - $y = \varepsilon$ esetén triviális,
 - tfh igaz n hosszú y -ra, ekkor $y = y'a$, ind feltevést 2x alkalmazva:
 $([x], y) \vdash_M^* ([xa], y') \vdash_M^* ([xy], \varepsilon)$;
- Köv: $L \subseteq \Sigma^*$ reg csak akkor $\approx_L$ rel által meghatározott ekv osztályok száma véges;;
- Pumpáló lemma: ha $L \subseteq \Sigma^*$ reg nyelv, akkor $\exists p \in \mathbb{N}$ szám, hogy minden $s \in L$ esetén, ha s hossza $\geq p$, akkor s felbontható: $s=xyz$ alakra, hogy:
 - $xy^nz \in L$ minden $n \geq 0$ esetén,
 - $|y| > 0$,
 - $|xy| \leq p$;
- Biz: legyen $p = |Q|$, $s = s_1s_2..s_n \in L$, $p \leq n$,
 - az s -et elfogadó állapotok sorozata: $r = r_1r_2..r_{n+1}$, ekkor: $p < n+1$, tehát r tagjai közül kell lennie 2 azonosnak skatulya elv miatt, ezek: r_j és r_k ;
 - $x = s_1s_2..s_{j-1}$, $y = s_js_{j+1}..s_{k-1}$, $z = s_ks_{k+1}..s_n$, így:
 - $xy^nz \in L$ igaz,
 - $|y| > 0$, mert $j \neq k$,
 - $|xy| = k-j \leq p$;
 - ha pumpálható, attól még nem reguláris;;;

Grammatikák:

- $G = (V, \Sigma, R, S)$ generatív grammatika, ahol:
 - V ..véges, nemterminálisok halmaza,
 - Σ ..véges, diszjunkt V -vel, terminálisok halmaza (nyelv abc-je),
 - R ..véges, levezetési szabályok halmaza (rek alkalmazva minden szó megkapható, alakja: $\alpha \rightarrow \beta$, ahol α és β term és nemterm sorozata),
 - $S \in V$..kezdőszimbólum (kitüntetett nemterminális jel);
- $k \in (Vu\Sigma^*)$ szóból közvetlenül levezethető $\mu \in (Vu\Sigma^*)$: ha léteznek olyan $\gamma, \delta \in (Vu\Sigma^*)$ szavak és $\alpha \rightarrow \beta$ szabály R -ben, hogy: $k = \gamma\alpha\delta$, $\mu = \gamma\beta\delta$, *jele:* $k \Rightarrow_G \mu$;
- w_n -nek w_1 -ből levezetése: $w_1..w_n \in (Vu\Sigma^*)$ szavak sorozata G grammatika alapján, ha $w_1 \Rightarrow_G w_2 \Rightarrow_G \dots \Rightarrow_G w_n$, *jele:* $u \Rightarrow^*_G v$ vagy $u \Rightarrow^* v$;;
- G generálja L nyelvet: $L \subseteq \Sigma^*$, $L = \{w \in \Sigma^* \mid S \Rightarrow^* w\}$ (minden lehetséges szabály alkalmazásával ilyen szavakat tudunk levezetni), *jele:* $L(G)$;;
- Grammatikák ekvivalenciája: G és G' ekvivalensek, ha $L(G) = L(G')$;;
- G rekurzív: ha $\exists A \in V$ és $\gamma, \delta \in (Vu\Sigma^*)$ szavak, hogy: $A \Rightarrow^* \gamma A \delta$, ekkor A egy rek nemterminálisa grammatikának; (nemrek gr véges nyelvet generálnak);
- G környezetfüggetlen (CF= Context Free): ha levezetési szabályainak bal oldalán egyetlen nemterm szerepel, azaz $A \rightarrow \beta$, ahol: $A \in V$, $\beta \in (Vu\Sigma^*)$;;
- T : reg nyelv mindig CF;
- Biz : ha L reg, akkor $\exists M$: $L = L(M)$, azaz őt felismerő automata, készítünk egy éppen L -et generáló CF grammatikát:
 - M minden q_i állapotának megfeleltetünk egy R_i nemterm szimbólumot, kezdőszimb: R_0 ;
 - $R_i \rightarrow aR_k$ szabály, ha $\delta(r_i, a) = r_k$, $R_i \rightarrow \epsilon$, ha r_i végáll M -ben;;
- T : CF grammatika szerinti levezetés nem mindig egyértelmű, mivel a szabályok alkalmazásának sorrendje tetsz;;
- Levezetési fa: CF grammatika szerinti levezetés ábrázolható fával, melynek:
 - gyökere= kezdőszimbólum,
 - levelei= levezetett szó betűi,
 - csomópontjai= nemterm,
 - elágazások= alkalmazott szabályok;;
- CF gr szerinti levezetés baloldali: ha levezetés során mindig legbaloldalibb nemterm-t helyettesítjük be (CF gr szerinti baloldali levezetés sem mindig egyértelmű);
- Egyértelmű: ha minden generált szavának egyetlen baloldali levezetése van (egyetlen levezetési fa létezik hozzá);
- Eredendően többértelmű: ha nem generálható egyértelmű gr-val;;

- **Chomsky-féle Normál Alak (=CNF):** $G = (V, \Sigma, R, S)$ CF gr CNF, ha:
 - $A \rightarrow BC$,
 - $A \rightarrow a$,
 - $S \rightarrow \varepsilon$, ahol: $A, B, C \in V$, $a \in \Sigma$, B, C nem kezdőszimb;;
- **T:** tetsz CF nyelvhez megadható öt generáló gr CNF-ben;
- **Biz:**
 - 1) $S_0 \rightarrow S$ új kezdőszimb;
 - 2) $A \rightarrow \varepsilon$ szabályra:
 - töröljük, ha A nem kezdőszimb,
 - $B \rightarrow \alpha A \beta$ -hez A -t törölve hozzáadunk újakat minden lehetséges módon (2^{k-1} új szabály) ,
 - $B \rightarrow A$ esetén $B \rightarrow \varepsilon$ -t beiktatunk, ha korábban nem töröltünk ilyen szabályt;
 - 3) $A \rightarrow C$ szabályra:
 - töröljük,
 - de ha van $C \rightarrow \alpha$, akkor beiktatjuk $A \rightarrow \alpha$ -t, kivéve ha töröltük korábban;
 - 4) végül konvertáljuk szabályokat max 2 hosszú jobboldalas alakra:
 - $A \rightarrow \alpha_1 \alpha_2 \dots \alpha_k$ -ből lesz: $A \rightarrow \alpha_1 A_1$, $A_1 \rightarrow \alpha_2 A_2$, ... $A_{k-2} \rightarrow \alpha_{k-1} \alpha_k$;
 - minden α_i terminálist(!) U_i nemterm-al helyettesítünk és beiktatjuk: $U_i \rightarrow \alpha_i$ szabályt;;
- **Bar-Hillel (pupáló) lemma:** ha $L \subseteq \Sigma^*$ CF nyelv, akkor $\exists p \in \mathbb{N}$ szám, hogy minden $s \in L$ esetén, ha s hossza la p , akkor s felbontható: $s = uvxyz$ alakra, hogy:
 - $uv^nxy^n z \in L$ minden $n \geq 0$ esetén,
 - $|vy| > 0$,
 - $|vxy| \leq p$;
- **Biz:** s jó hosszú L -beli sorozat, így levezetésében $2x$ szerepel A ;
 - legyen G adott L -t generáló CF gr, b..szabályok jobb oldalán szereplő szimb max száma ($b \geq 2$), tetsz m mélységű fában max b^m levél szerepel;
 - legyen $p = b^{|V|+2}$, ahol: $|V|$.. G -beli változók száma, $p > b^{|V|+1}$, (mivel $b \geq 2$) ezért ha s (L -beli) la p hosszú sorozat, akkor minden levezetési fája la $|V|+2$ mély;
 - legyen τ s egy lk csomópontú levezetési fája, τ la $|V|+2$ mély, így egy út la $|V|+1$ változót tartalmaz, tehát van olyan, ami $2x$ szerepel- jelöljük A -val;
 - $s = uvxyz$, $uv^nxy^n z \in L$ igaz,
 - v és y közül egyik nem ε (különben létezne kevesebb csomópontú levezetési fa s -hez),
 - vxy max p hosszú, mivel A -ból max $|V|+2$ mélységű fa vezet le, aminek csak $b^{|V|+2} = p$ levele lehet;;
- **Chomsky-féle nyelvosztályok:**
 - 3-as ($R = \text{reguláris}$): $A \rightarrow a$, $A \rightarrow aB$;
 - 2-es (CF): $A \rightarrow \alpha$;
 - 1-es (CS): $\beta A \gamma \rightarrow \beta \alpha \gamma$, $\alpha \rightarrow \beta$, ahol: $|\alpha| \leq |\beta|$, nem csökkenő;
 - 0-s: tetsz alakú szabályok;;;



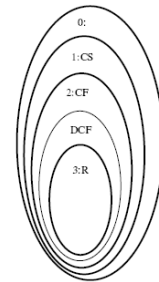
Verem automaták:

- $M = (Q, \Sigma, \Gamma, \delta, q_0, F)$, ahol:
 - Q ..véges, állapotok halmaza,
 - Σ ..véges halmaz, abc,
 - Γ ..véges halmaz, veremabc,
 - $\delta: Q \times \Sigma \times \Gamma \rightarrow P(Q \times \Gamma)$..állapotátmenet fv,
 - $q_0 \in Q$..kezdeti áll,
 - $F \subseteq Q$..végáll;
- Bemenő jelek bemeneti szalagok- véges sok áll- verem memória (LIFO); CF nyelvek felismeréséhez;;
- Felismert (=elfogadott) jelsorozat: $M = (Q, \Sigma, \Gamma, \delta, q_0, F)$ egy VA, $w = x_1 x_2 \dots x_n \in \Sigma^*$ jelek sorozata, M felismeri w -t, ha $\exists r_0 r_1, \dots, r_n \in Q$ állapotok sorozata és $s_0 s_1 \dots s_n \in \Gamma^*$, hogy: $r_0 = q_0$ és $s_0 = \epsilon$, $(r_{i+1}, b) \in \delta(r_i, x_{i+1}, a)$, $i = 0..n-1$, ahol: $s_i = at$ és $s_{i+1} = bt$ valamely $a, b \in \Gamma$ és $t \in \Gamma^*$, $r_n \in F$;
- M VA felismeri A nyelvet, ha $A = \{w \mid M \text{ felismeri } w\}$;;
- T: minden CF nyelvhez létezik öt felismerő veremautomata;
- Biz: legyen G CF gr, ami L -et generálja, megkonstruáljuk M veremautomatát, amely minden bemeneti jelsorozatról eldönti, hogy levezethető-e G gr szerint;
 - 1) beírjuk verembe egyszerre: q, r állapotok, $a \in \Sigma$, $s \in \Gamma$ (automata q -ból r -be megy, ha 'a' beolvasott jel hatására veremből s -t olvassa ki, verembe beteszi $u = u_1 \dots u_k$ -t egyszerre);
 - beiktatunk új, közbülső állapotokat: $q_1, q_2, \dots, q_{k-1}$, melyekre: $\delta(q, a, s) = \{(q_1, u_k)\}$, $\delta(q_1, \epsilon, \epsilon) = \{(q_2, u_{k-1})\}, \dots, \delta(q_{k-1}, \epsilon, \epsilon) = \{(r, u_1)\}$;
 - 2) M állapotainak halmaza: $Q = \{q_k, q_v, q_{ciklus}\} \cup E$, ahol: E ..új közbülső állapotok;
 - inicializálás: $\delta(q_k, \epsilon, \epsilon) = \{(q_{ciklus}, S\$)\}$,
 - ha verem legfelső eleme nemterm: $\delta(q_{ciklus}, \epsilon, A) = \{(q_{ciklus}, w) \mid A \rightarrow w \text{ szabálya } G\text{-nek}\}$,
 - ha verem legfelső eleme term: $\delta(q_{ciklus}, a, a) = \{(q_{ciklus}, \epsilon)\}$,
 - verem legfelső eleme $\$$: $\delta(q_{ciklus}, \epsilon, \$) = \{(q_v, \epsilon)\}$;;
- T: minden M VA-hoz létezik G CF grammatika, mely éppen M által felismert nyelvtant generálja;
- Biz: tfh M -re igaz: egyetlen végállapota van(q_v), ezt üres veremmel éri el, minden állátmenet betesz v kivesz szimbólumot verembe/ből; egy adott M módosítható úgy, hogy feltételek teljesüljenek;;
- T: L CF nyelvek halmaza, L zárt reg op-ra nézve, azaz ha A_1 és $A_2 \in L$,akkor:
 - $A_1 \cup A_2 \in L$,
 - $A_1 \circ A_2 \in L$,
 - $A_1^* \in L$;
- Biz: legyen $G_1 = (V_1, \Sigma_1, R_1, S_1)$, $G_2 = (V_2, \Sigma_2, R_2, S_2)$ A_1 ill A_2 L -beli nyelveket generáló gr, tfh V_1 és V_2 diszjunktak;
- 1) $G = (V_1 \cup V_2 \cup \{S\}, \Sigma_1 \cup \Sigma_2, R, S)$, ahol: $R = R_1 \cup R_2 \cup \{S \rightarrow S_1, S \rightarrow S_2\}$, ekkor: $L(G) = L(G_1) \cup L(G_2)$;
- 2) $G = (V_1 \cup V_2 \cup \{S\}, \Sigma_1 \cup \Sigma_2, R, S)$, ahol: $R = R_1 \cup R_2 \cup \{S \rightarrow S_1 S_2\}$, ekkor: $L(G) = L(G_1) \circ L(G_2)$;
- 3) $G = (V_1 \cup \{S\}, \Sigma_1, R, S)$, ahol: $R = R_1 \cup \{S \rightarrow \epsilon, S \rightarrow S S_1\}$, ekkor: $L(G) = L(G_1)^*$;;

- T: $\mathbb{L}$ CF nyelvek halmaza nem zárt metszetre és komplementerre nézve;
- Biz: $A_1 = \{a^n b^n c^m \mid n, m \geq 0\}$, $A_2 = \{a^n b^m c^m \mid n, m \geq 0\}$ CF nyelvek, $A_1 \cap A_2 = \{a^n b^n c^n \mid n \geq 0\}$ nem CF (pumpálási lemmával belátható);
- $A_1 \cap A_2 = (A_1^C \cup A_2^C)^C$, tfh zárt és CF: unióra zárt és CF, ezért egész zárt és CF, de tudjuk: A_1 és A_2 nem CF! (maga után vonná metszetre zártságot);;
- T: ha A_1 CF nyelv és A_2 reg nyelv, akkor $A_1 \cap A_2$ CF nyelv;
- Biz: legyen $M_1 = (Q_1, \Sigma, \Gamma, \delta_1, q_1, F_1)$ VA, hogy $A_1 = L(M_1)$ és $M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ DVA, hogy $A_2 = L(M_2)$, $M = (Q_1 \times Q_2, \Sigma, \Gamma, \delta, (q_1, q_2), F_1 \times F_2)$ VA, ahol: $\delta: Q_1 \times Q_2 \times \Sigma \times \Gamma \rightarrow P(Q_1 \times Q_2 \times \Gamma)$; $\delta(p, q, a, b) = \{(\delta_1(q, a), r, s) \mid (r, s) \in \delta_2(p, a, b)\}$;;
- M VA determinisztikus: ha $\delta: Q \times \Sigma \times \Gamma \rightarrow P(Q \times \Gamma)$ olyan, hogy minden q, a, b esetén $\delta(q, a, b)$ max egyelemű halmaz;
- L nyelv determinisztikus CF: $L \subseteq \Sigma^*$, ha L nyelvhez van azt generáló det VA, ahol $\$ \in \Sigma$ (nyelv elemeinek végét jelöli);
- T: Determinisztikus CF nyelvek osztálya zárt reg op-ra, komplementerre, metszetre;
- T: det CF nyelvek osztálya szűkebb, mint CF (van CF, ami nem det CF);;

Turing gép:

- $M = (Q, \Sigma, \Gamma, \delta, q_0, q_i, q_n)$, ahol:
 - Q ..véges, állapotok halmaza,
 - Σ ..véges halmaz, abc (nincs benne: $_$),
 - Γ ..véges halmaz, szalagabc ($_$ eleme és $\Sigma \subseteq \Gamma$),
 - $\delta: Q \times \Gamma \rightarrow P(Q \times \Gamma \times \{J, B\})$..állapotátmenet fv,
 - $q_0 \in Q$..kezdeti áll, $q_i \in Q$..elfogadó áll, $q_n \in Q$..elutasító áll;;
- Idealizált szgép, legáltalánosabb számítási modell; minden algoritmikusan megoldható probléma megoldható TG-vel is (de nem minden probléma); író-olvasó fej, véges sok állapot; ír-olvas szalagról/ra, fej J és B mozog, szalag végtelen, gép nem áll meg feltétlenül;;
- Működése: kezdőhelyzet: input a szalag első n mezőjében, többi mezőben: $_$; a gép megáll elfogadó áll-ban vagy elutasítóban v nem áll meg;
- rákövetkezője: $(u a p b v)$ ráköv: $(u q a c v)$, ha balra lépett, azaz: $\delta(p, b) = (q, c, B)$ és $(u a c q v)$, ha jobbra, ahol: $a, b, c \in \Gamma$ szalagjelek, $u, v \in \Gamma^*$, $p, q \in Q$ állapotok;
- konfiguráció: (u, q, v) , ahol: $u, v \in \Gamma^*$, $q \in Q$ (szalag tartalma, fej pozíciója, pillnyi áll);
- elfogadó-elutasító konfiguráció: ha q állapot elf-elut;
- kezdeti konfiguráció: (ε, q_0, w) , ahol: w a szalagra írt bemenő jelsorozat;;
- M TG elfogad w sorozatot: ha $\exists C_1 C_2 \dots C_n$ konfigurációk sorozata, hogy: C_1 ..kiindulási konfiguráció, C_i rákövetkezője: C_{i+1} , $i=1..n-1$, C_n elfogadó konfiguráció;
- M TG által felismert nyelv: gép által elfogadott sorozatok halmaza, $L(M)$;;
- L nyelv Turing felismerhető (=rekurzíve felsorolható= RE): $L \subseteq \Sigma^*$, ha van olyan M TG, melyre: $L(M) = L$ ($s \in \Sigma^* - L$ nem nyelvbeli szó esetén nem biztos, hogy megáll);
- L nyelv Turing eldönthető (=rekurzív= R): $L \subseteq \Sigma^*$, ha van olyan M TG, melyre: $L(M) = L$ és M minden $s \in \Sigma^*$ szóra megáll; minden rek nyelv egyben rek felsorolható is;;
- T: ha L rek, akkor L^C is;



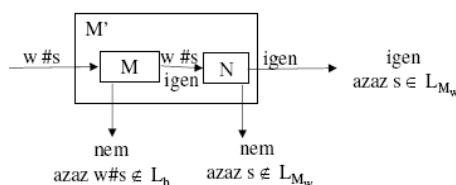
- Biz: ha $L(M)=L$, akkor M -ben felcseréljük elfogadó és elutasító állapotokat, így M'
TG: $L(M')=L^C$;;
- M és S TG változatok: ekvivalensek, ha $L(M)=L(S)$, ua nyelveket ismerik fel;
- Helyben maradó fejjel (=HF TG): $M=(Q,\Sigma,\Gamma,\delta,q_0,q_i,q_n)$, ahol:
 - Q ..véges, állapotok halmaza,
 - Σ ..véges halmaz, abc (nincs benne: $_$),
 - Γ ..véges halmaz, szalagabc ($_$ eleme és $\Sigma \subseteq \Gamma$),
 - $\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{J,B,H\}$..állapotátmenet fv,
 - $q_0 \in Q$..kezdeti áll, $q_i \in Q$..elfogadó áll, $q_n \in Q$..elutasító áll;;
- Több szalaggal (=TSZ TG): M , ahol: ua, $\delta: Q \times \Gamma^k \rightarrow Q \times \Gamma^k \times \{J,B\}^k$;
- T: TSZ TG ekv TG-vel;
- Biz: legyen M egy k szalagú TG és S alap TG, mely szimulálja M -et;
 - 1) Γ' tartalmazza minden Γ -beli elem megjelölt változatát és $\#$..elválasztó jelet; M gép k szalagjának tartalma egymás után $\#$ -al elválasztva S -en, k fej pozícióján megjelölt elem;
 - 2) S megkeresi szalagján első és utolsó $(k+1.)$ $\#$ jelet, és leolvassa k db megjelölt szimbólumot, majd M állátmenete szerint elvégzi felülírást;
 - 3) ha $\#$ jelre kellene lépni, akkor megjelölt üres szimbólum kerül oda, miután szalag tartalma utolsó $\#$ jelig eggyel jobbra csúszott;;
- Nemdeterminisztikus (=ND TG): M , ahol: ua, $\delta: Q \times \Gamma \rightarrow P(Q \times \Gamma \times \{J,B\})$; működését fa írja le (ha egy ágon az elért állapot elfogadó, akkor ND TG elfogadta sorozatot);
- T: M NDTG ekv S TG-vel;
- Biz: S 3 szalagos TG (input, szimulációs, cím szalag $(1,2,...m)$ számokból álló sorozat, ahol: $m = M$ állapotátmeneteinek $\ln$ száma), működése:
 - 1) kezdetben input szalagon M -nek adott sorozata, másikkal üresek, majd input szalagot szimulációsra másolja;
 - 3) szimulációs szalagon S szimulálja M egy-egy ágának működését-csomópontokat szélességi kereséssel járja be, állapotátmeneteket címszalagról olvassa le, ha nincs több szimbólum vagy nem megengedett vagy elutasít, akkor címszalagon lévő címet felülírja következővel, és vissza: input szalagot szimulációsra másolja, de ha áll elfogad, akkor elfogadja sorozatot;;
- Felsoroló TG (=enumerator): M , ahol: ua, nincs bemeneti sorozat, csak munkaszalag, $\delta: Q \times \Gamma \rightarrow (Q \times \Gamma \times \{J,B\} \times \Sigma^*)$, Σ^* ..felsorolt (=kinyomtatott) sorozatok; összes felsorolt elemek alkotják TG által felsorolt nyelveket;
- T: L nyelvet felsoroló E TG-hez létezik M TG, mely épp L -t ismeri fel, és vissza;
- Biz E-hez M: legyen N 2 szalagos TG (1: bemeneti sorozat, 2: E által felsorolt új sorozat), N elfogadja w inputot, ha egy 2. szalagra írt sorozattal azonos (véges időn belül megtaláljuk);
- Biz M-hez E: legyen A S által felismert nyelv, $s_1s_2...s_n$ összes lehetséges Σ^* -beli sorozat (lexikografikusan felsorol minden lehetséges Σ -beli sorozatot), működés: $i=1,2,3,...$ -ra ismétli: M fut i . lépésig egymás után $s_1...s_i$ inputtal, ha elfogadó áll-ba ér, akkor kinyomtatja megfelelő s_j input sorozatot;;
- Köv: L nyelv Turing felismerhető csakkor van ND TG, mely felismeri;
- T: L nyelv Turing eldönthető csakkor van ND TG, mely eldönti; (ND TG eldöntő, ha minden ágon megáll);;;

Fvszámítás TG-el:

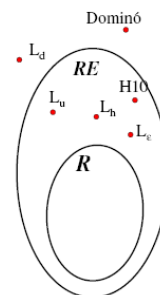
- Output: $x \in \Sigma^*$ kezdeti bemeneti sorozat M TG outputja $y \in \Sigma^*$ sorozat, ha M megáll x inputra és szalagon éppen w szerepel és fej 1. pozícióban áll, *jele:* $y=M(x)$;;
- M által kiszámított fv: $f_M: \Sigma^* \rightarrow \Sigma^*$ parciálisan értelmezett fv, mely minden x-en értelmezett, melyre van y, hogy: $y=M(x)$;
- fv parciálisan rekurzív: $f: \Sigma^* \rightarrow \Sigma^*$, ha van M TG, hogy: $f=f_M$;
- fv rekurzív: fenti és minden Σ^* -ra értelmezve;;
- $B=0u1(0u1)^*$, $\text{bin}(n)=w$, azaz tetsz n term szám egyértelműen felírható; term számon értelmezett fv: $f: B \rightarrow B$, minden k dimenziós $N^k \rightarrow N$ fv tekinthető $f: B^k \rightarrow B$ fv-nek;
- M kiszámítja f-t: $f: B^k \rightarrow B$, ha tetsz $w_1, \dots, w_k \in B$ esetén $w=w_1..w_k$ sorozatra $M(w)=u$, ahol: $u=f(w_1, \dots, w_k)$;;
- Algoritmus: véges sok lépéssel kiszámít fv-t, eldönt egy kérdést;
- Church-Turing tézis: ami algoritmussal kiszámítható/eldönthető, az Turing kiszámítható/eldönthető;
 - $f: \Sigma^* \rightarrow \Sigma^*$ parciális fv algoritmussal kiszámítható $\Leftrightarrow$ f parciálisan rekurzív;
 - $f: \Sigma^* \rightarrow \Sigma^*$ teljes fv alg kisz $\Leftrightarrow$ f rekurzív;
 - L nyelvre nyelvbe való tartozás alg-al eldönthető $\Leftrightarrow$ nyelv rekurzív;;
- Hilbert 10. problémája: diofantikus egyenletek általános megoldhatósága: $f(x_1, \dots, x_n)=0$ egész megoldását keressük, ahol f egészeühatós n változós polinom (nincs algoritmus, de T-felismerhető, azaz ha van mo, akkor véges sok lépésben kiderül, de ha nincs az nem);;
- Dominóprobléma: $D= \{F \in \{0,1,*,\#\}^* \mid \text{sík lefedhető az F féle dominóval}\}$, $F=a_1*b_1*c_1*d_1* \dots d_k\#$ féle dominóval lefedhető-e sík, mindegyik fajtából végtelen sok van (nem eldönthető, nem ismerhető fel);;
- Kongruens számok felismerése: m poz szám kongruens, ha van olyan derékszögű háromszög, melynek területe m és oldalai rac számok: $p^2+q^2=r^2$, $pq=2m$ (rek felsorolható: rac számok megszámlálhatók- véges sok lépésben kiderül, de nem rek);;
- T: van olyan $L' \subseteq \Sigma^* = \{0,1\}^*$ nyelv, amely nem T-felismerhető;
- Biz: TG leírható véges jelsorozattal, összes TG felsorolható, mivel k hosszú jelsorozattal megadott TG véges sok van;
 - legyen $M_1, M_2, \dots, M_n, \dots$ egy felsorolásuk; M_i TG egyértelműen meghatározza általa felismert L_i nyelvet, így T-felismerhető nyelvek is felsorolhatóak: $L_1, L_2, \dots, L_n, \dots$, azaz megszámlálhatóan sokan vannak;
 - az összes Σ feletti nyelv számossága $\{0,1\}^*$ lehetséges részhalmazainak számossága, ami több, mint megszámlálható;
 - *Biz:* Cantor-módszer: Σ^* elemei kanonikusan felsorolhatóak (üres szó, 1,2,...k hosszú szavak), azonos hosszú szavakat lexikografikusan soroljuk el (0,1,00,01,10,11,000...)- ezt felsorolást $w_1, w_2, \dots, w_n$ -el jelöljük; megadjuk L' nyelvet Cantor-féle átlós módszerrel: $w_i \in L'$, ha $w_i \notin L_i$;;
- TG megadása 0,1 sorozattal: $M=(Q, \Sigma, \Gamma, \delta, q_0, q_i, q_n)$, tfh: $Q=\{0,1, \dots, q\}$, 0..kezdő, q..elfogadó, q-1..elutasító áll, $\Sigma=\{0,1\}$ bemeneti abc, $\Gamma=\{0,1, \dots, s\}$ szalag abc, s..üres jel „_”, $J=1$, $B=0$, ekkor: M kódja: M egyértelműen megadható egy $\{0,1\}^*$ -beli w szóval, ill: egy $\{0,1\}^*$ -beli szóhoz legfeljebb egy M_w tartozik; *pl:* $q\#s\#q_0\#x_0\#q_0'\#x_0'\#m_0' \dots$, helyettesítés: $\#=11$, $1=01$, $0=00s$;;

Univerzális TG:

- képes szimulálni tetsz, a w kódjával adott M_w TG-et;
- T: van olyan U 3 szalagos TG, melyre: ha $w, s \in \{0,1\}^*$ és M_w létezik, akkor U gép pontosan akkor fogadja el/utasítja el a $w\#s$ bemenetet amikor M_w elfogadja/elutasítja;
- Biz:
 - 1.szalag: $w\#s$, nem változik, M_w működését adja meg,
 - 2.szalag: s , szalag tartalma, M_w állapota szerint,
 - 3.szalag: M_w pillanatnyi állapota;
- működése:
 - 0) ellenőrzi, hogy w TG-t ír-e le, ha nem, akkor elutasít, ha igen, akkor: beírja s -t a 2. és 0-t(kezdőáll kódja) a 3.szalagra;
 - 1) 1.szalag alapján meghatározza, hogy M_w mit lépne, ha 3. szalagon megadott áll-ban lenne és olvasófej 2. szalagon lévő jelet látná; U megteszi M_w lépést és módosítja 2. és 3. szalag tartalmát; U megáll elfogadó ill elutasító áll-ban, ha M_w szerinti új állapot elf vagy elut, ha nem, akkor újra 1) lépés;;
- Diagonális nyelv: azon TG kódja, mely nem fogadja el saját kódját, $L_d = \{w \in \{0,1\}^*, M_w \text{ gép létezik és } w \notin L_{M_w}\}$;
- T: L_d nem rekurzíve felsorolható nyelv;
- Biz: indirekt, van N TG, mely épp 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 saját kódját; N kódja s :
 - $t\text{fh}$ N felismeri s -t, azaz $s \in L_d$, akkor def szerint $s \notin L_d$;
 - $t\text{fh}$ $s \notin L_d$, akkor def szerint $s \in L_d \rightarrow$ ellentmondás;;
- Univerzális nyelv: $L_u = \{w\#s \in \{0,1\}^*, M_w \text{ létezik és } s \in L_{M_w}\}$; L_u az U univerzális TG nyelve, azaz $L_u = L_U$, tehát L_u rekurzíve felsorolható;;
- T: L_u nem rekurzív nyelv;
- Biz: indirekt, ekkor van N TG, mely eldönti L_u -t (minden inputra megáll) és $L_u = L_N$; legyen N' TG:
 - ha $w \in \{0,1\}^*$ bemenet nem TG kódja, akkor elutasítja,
 - ha w TG kódja, akkor: elindítja N -t $w\#w$ bemenettel, ha ez szimuláció elfogadó áll-ban megáll, akkor N' elutasít és fordítva; N' elfogadja w -t, ha M_w létezik és $w \in L_{M_w}$; tehát: $L_N = L_d$, amiről tudjuk, hogy nem rek felsorolható-ellentmondás;;
- Megállási probléma: eldönthető-e, hogy adott M TG megáll-e minden bemenetre? $L_h = \{w\#s \in \{0,1\}^*, M_w \text{ létezik és } s \text{ bemenettel elindítva megáll}\}$;
- T: L_h rek felsorolható, de nem rekurzív;
- Biz: legyen U univerzális TG, L_h szavaira megáll; U' TG, azonos U -val, de U elutasító állapota itt elfogadó, ekkor: $L_h = L_{U'}$, azaz L_h rek felsorolható;
- Biz: indirekt, L_h rek, azaz van olyan M TG, mely mindig megáll és $L_h = L_M$; legyen M' TG, mely M -t futtatja, ha elfogadó állapot, akkor futtatja U -t; M' mindig megáll és L_u -t ismeri fel, ami nem rek- ellentmondás;;

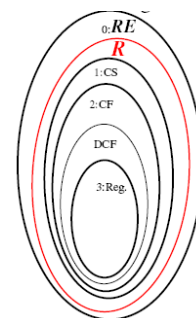


- Megállási probléma üres inputra: eldönthető-e, hogy adott M TG megáll-e üres bemenetre? $L_\varepsilon = \{w \in \{0,1\}^* \mid M_w \text{ létezik és } \varepsilon \text{ bemenettel elindítva megáll}\}$;
- T: L_ε rek felsorolható, de nem rekurzív;
- Biz: $w \in L_\varepsilon$ esetén L_h megállási nyelvet felismerő automatát $w\# \varepsilon = w\#$ inputtal futtatva éppen L_ε -t ismeri fel;
- Biz: megállási probléma visszavezethető erre; $\exists N$ TG, amely eldönti, M TG s bemenetre való viselkedését szimuláljuk egy M_s TG ε bemenetre való viselkedésével:
 - M_s üres input esetén felírja s-t szalagjára, majd úgy működik, mint M működne s inputra;
 - Futtatjuk N-et M_s TG-en:
 - igen, ha M_s megáll üres stringen $\Leftrightarrow M$ megáll s-en,
 - nem, ha M_s nem áll meg üres stringen $\Leftrightarrow M$ nem áll meg s-en;
 - legyen M_s kódja w_s , nyilván: $w_s \in L_\varepsilon$, így ha L_ε eldönthető lenne, akkor ált megállási probléma is az lenne;;
- M TG-hez létezik-e olyan input, amin megáll? Tfh $\exists N$ TG, amely eldönti M -ről, hogy létezik-e; visszavezetés: el tudnánk dönteni, hogy adott M TG megáll-e s stringen; tetsz M_s bemenet, megkonstruáljuk M' TG-et, tetsz x bemenetre:
 - ha $x \neq s$, akkor kerüljön N végtelen ciklusba;
 - ha $x = s$, akkor szimulálja M -et: N-t lefuttatjuk M' -n:
 - igen, ha M' -höz létezik olyan string, amin megáll $\Leftrightarrow M$ megáll s-en,
 - nem, ha M' -höz nincs olyan string $\Leftrightarrow M$ nem áll meg s-en;;
- $L = \{w \in \{0,1\}^* \mid M_w \text{ létezik és elfogad minden bemenetet}\}$: legyen M , w tetsz, konstruáljuk meg M' TG-et: M' tetsz u bemenetre:
 - ha $u \neq w$, akkor M' elfogadja u-t;
 - ha $u = w$, akkor M' szimulálja M működését w-n:
 - ha M elfogadja w-t, akkor M' is elfogadja,
 - ha M elutasítja w-t, akkor M' is elutasítja u-t;
 - tehát: ha L eldönthető, akkor L_u is – ellentmondás;;
- $L = \{w \in \{0,1\}^* \mid M_w \text{ létezik és } L(M_w) = \text{üres, azaz nem fogad el semmit}\}$: M , w tetsz, megkonstruáljuk M' TG-et: M' tetsz u bemenetre:
 - ha $u \neq w$, akkor M' elutasítja u-t;
 - ha $u = w$, akkor M' szimulálja M működését w-n:
 - ha M elfogadja w-t, akkor M' is elfogadja,
 - ha M elutasítja, akkor M' is;;
- $L = \{w\#s \mid M_w \text{ nem áll meg s-en}\}$: megmutatni: $L \notin RE$; $L \notin L_h$ komplementere ($L_h = \{w\#s \mid M_w \text{ megáll s-en}\}$), tudjuk, hogy $L_h \in RE$; tudjuk, hogy ha egy nyelv és annak komplementere is RE-beli, akkor nyelv R-beli;
 - Biz: M_1, M_2 párhuzamosan működő, M működteti M_1 -et 1. szalagján, M_2 -t 2.szalagján, eldöntő algoritmus van;
 - Biz: indirekt, $L \in RE$ és $L_h \in RE$, akkor $L \in R$ is teljesül, de akkor L_h is eleme lenne R-nek, de ez ellentmondás;;



- M TG minden $x \in \Sigma^*$ bemenetet elfogadja/elutasítja? Tfh $\exists N$ TG, amely eldönti problémát, visszavezetés: M' TG elfogad-e s inputot; definiálunk M'' TG-et:
 - ha $x \neq s$, akkor elfogadja/elutasítja inputot;
 - ha $x = s$, akkor szimulálja M' működését s-en: N -et lefuttatjuk M'' -n:
 - M'' minden inputot elfogad/elutasít $\Leftrightarrow M'$ elfogadja s-et;
 - M'' nem minden inputot fogad el/utasít el $\Leftrightarrow M'$ nem fogadja el s-t;;
- M TG által felismert nyelv reguláris? Tfh $\exists N$ TG, mely eldönti, visszavez: M' -ről eldönthető, hogy elfogad-e s inputot; definiálunk M'' -t:
 - ha $x = 0^n 1^n$ alakú szavak, akkor fogadja el;
 - ha $x \neq 0^n 1^n$, akkor szimulálja M' működését; N -t lefuttatjuk M'' -n:
 - M'' elfogad reg nyelvet $\Leftrightarrow M'$ elfogadja s-t;
 - M'' nem reg nyelvet fogad el $\Leftrightarrow M'$ nem fogadja el s-t;;
- $L = \{w \in \{0,1\}^* \mid M_w\text{-ben van felesleges állapot}\}$? Tetsz M, S párhoz legyen M' a köv: M' egy w bemeneten
 - ha $w \neq s$, akkor M' elutasítja w -t,
 - ha $w = s$, akkor M' futtatja M -et s-en; tfh M' -nek van egy q^* állapota, amibe nem lép,
 - ha M elfogadja s-t, akkor M' először egy ciklusban végigmegy összes állapotán, utána elfogadja bemenetet,
 - ha M elutasítja s-t, akkor M' elutasítja w -t;
 - M' kódja e $L \Leftrightarrow s \in L(M) \Leftrightarrow w$ kódja $\#s \in L_u$;;
- M_1 és M_2 ua nyelvet fogadják el? Tfh $\exists N$ TG, amely eldönti, visszavez: M' TG, ami összes nyelvet elfogadja-e; definiálunk M'' TG-et: N -t lefuttatom M', M'' páron: $L(M'') = \text{üres}$;
 - ha M' és M'' ua nyelvet fogadják el: $L(M') = \text{üres}$, akkor $L(M'') = \text{üres}$;
 - Ha M' nem ua nyelvet fogadja el, mint M'' , akkor $L(M'') \neq \text{üres}$;;
- M_1 legalább annyi stringen áll meg, mint M_2 ? Tfh $\exists N$ TG, amely eldönti, visszavez: L_u/L_h ; M_1 TG minden bemenetet elutasít, M_2 tetsz; N -t lefuttatjuk s inputtal M_1, M_2 -n: N megmondja, hogy az egyik legalább annyi szót ismer-e fel, mint a másik (hiszen M_1 mindent elutasít, M_2 -ről pedig nem tudtuk ezt előre)- futás mindig megáll- M_2 -ről meg tudja mondani, hogy elfogadja-e s bemenetet/minden bemenetre megáll-e;;
- $L_i = \{w_1 \# w_2 \mid L(M_{w_1}) \cap L(M_{w_2}) = \emptyset\}$ nincs közös szava, megmutatni: $L \in RE$;
- Biz: indirekt, tfh $L_i \in RE$, megmutatjuk, hogy akkor $L_e = \{w \mid L(M_w) = \emptyset\} \in RE$, ami ellentmondás, mivel $L_e \notin RE$;
 - legyen M_i TG, amire $L(M_i) = L_i = \text{ha } L_i \in RE$, akkor $\exists$ hozzá M_i TG);
 - legyen M TG: tetsz w bemenetre futtatja M_i -t $w \# u$ bemeneten, ahol: u azon TG kódja, ami elfogad minden bemenetet;
 - ha M_i elfogadja $w \# u$ -t, akkor M elfogadja w -t,
 - ha M_i elutasítja, akkor M is; így nem jó(?)(!) cél: $L(M) = L_e$, de $L_e \notin RE$;;

- PMP (=Post Megfeleltetési Probléma): PCP =(P Correspondence P): adott véges dominókészlet: $\{u_1/v_1, \dots, u_n/v_n\}$, lehet-e egymás után rakni készletből dominókat, hogy alul és felül összeolvasva szavakat ua szót kapjuk? L_u visszavezethető PMP-re: $L_u \rightarrow \text{PMP}$, tetsz TG-hez dominókészletet megadni (elfogadja $\Leftrightarrow$ dominónál létezik mo); *def:* CF nyelvtan egyértelmű, ha minden $u \in L(G)$, azaz pontosan egy baloldali levezetése van; *T:* nem lehet algoritmikusan eldönteni, hogy egy CF nyelvtan egyértelmű-e; PMP-t visszavezetjük erre: $\text{PMP} \rightarrow$ egyértelmű; adott dominókészlethez (D) megadunk G_D CF nyelvtant úgy, hogy G_D pontosan akkor nem egyértelmű, ha D-nek van megoldása; $S \rightarrow A|B$, $A \rightarrow u_1 A a_1 | \dots | u_n A a_n$, $B \rightarrow v_1 B a_1 | \dots | v_n B a_n$, ahol: $a_1 \dots a_n$ új terminálisok; két kül bo levezetés így nézne ki: $u = u_{i1} u_{i2} \dots u_{il} a_{i1} \dots a_{il}$, $v = v_{i1} v_{i2} \dots v_{il} a_{i1} \dots a_{il}$ (ua jobboldalakat kell alkalmazni levezetésben, így ua dominók);;
- Eldönthető-e, hogy reg kif generál-e w stringet? Igen, mert reg nyelvhez automata felírható. TG el tudja dönteni, hogy generál-e w-t;;
- Eldönthető-e, hogy G CF gr generálja-e w szót? Igen, n hosszú w szó: n-1 szabály, n term-ből áll a szó, így összesen $2n-1$ lépés, ami véges;;
- Eldönthető-e, hogy DVA elfogad-e egyáltalán bármilyen stringet (= van-e string, amit elfogad)? Igen, elfogad, ha van ir út, ami elfogadó áll-ba vezet;;
- Eldönthető-e, hogy NVA elfogad-e w szót? Igen, szimulálható DVA-val, n hosszú szót n időben szimulálja;;
- Eldönthető-e, hogy 2 automata ua nyelvet fogadja-e el? $L(A) = ? L(B)$, Igen, $L(A) \Delta L(B) = 0$..szimm differencia, ref nyelvek zártak- op megkonstruálható- szimm diff-t megadja op- eldönthető, hogy mit fogad el;;
- Eldönthető-e, hogy egy P veremautomata elfogad-e w szót? Igen, P-hez van ekvivalens G CF nyelvtan, G-hez ekv CNF-ben lévő G' , minden $w \in L(G')$ -re az $s \Rightarrow w$ levezetés hossza: $2n-1$, ahol: $n = |w|$;;
- T: nyelvtannal generálható nyelvek osztálya megegyezik RE nyelvek osztályával, azaz L nyelv csakkor generálható gr-val, ha van nyelvet felismerő TG;
- T: CS nyelvtannal generálható nyelvek osztálya megegyezik lin korlátozott TG-el (szalagnak csak bemeneti által elfoglalt részét használja) felismerhető nyelvek osztályával;
- T: CS nyelvtannal generálható nyelvek osztálya valódi részhalmaza R nyelvek osztályának;;;



Problémák összehasonlítása:

- Milyen fajta problémák vannak? Problémáfajták:
 - eldöntendő kérdés- nyelvbe tartozás (pl: prímszám-e, van-e?);
 - hozzárendelés- fv kiszámítás (pl: lnko, rendezések, H-kör keresés);
 - optimalizálás- hozzárendelések sorozata (pl: utazó ügynök);;
- Egy problémára adott algoritmusokat hogyan hasonlítsunk össze? Algoritmusok összehasonlítása:
 - Időigény (=lépésszám), tárigény szerint;
 - Feladat nagyságának (megoldás hossza) függvényében, nagy feladatokra koncentrálva (aszimptotikus viselkedés);
 - Legrosszabb esetet tekintve (worst case);
- Nagy ordó: fv aszimptotikus viselkedését jelöli; $f, g: \mathbb{N} \rightarrow \mathbb{R}^+$, $f(n) = O(g(n))$, ha vannak c és n_0 poz egészek, hogy $f(n) \leq c \cdot g(n)$, ha $n \geq n_0$; ekkor $g(n)$ *aszimptotikus felső korlátja* $f(n)$ -nek; pl: $f(n) = 6n^3 + 3000n^2 + 10^{100}$, $g(n) = n^3$; $f(n) = n^{100} + 2^n$, $g(n) = 2^n$; $f(n) = \log n + n$, $g(n) = n$; $f(n) = \log_2 n + \log_3 n$, $g(n) = \log n$ (nem számít az alap);;
- Pl: $n^2 - \log n \cdot n^2 - n^3 - n^{\log_2 n} = 2^{(\log_2 n)^2} - 2^n - n^n - 2^{2^n} - 2^{2^{(n+1)}}$;
- TG időkorlátja: adott M TG esetén $T_M(n)$.. M max lépésszáma (számolási ideje) n hosszú bemenetek esetén; $S_M(n)$.. M által max elolvasott munkaszalag jelek számát n hosszú bemenetek esetén; ha M nem áll meg bemenetre, akkor $T_M(n)$ végtelen, ekkor $S_M(n)$ is az;
- Időkorlátos: $t: \mathbb{N} \rightarrow \mathbb{N}$, $t(n) \geq n$, minden n -re; M TG $t(n)$ időkorlátos, ha n hosszú bemenetek esetén lf $t(n)$ lépést tesz meg, azaz $T_M(n) \leq t(n)$;
- $\text{TIME}(t(n)) = \{L \subseteq \Sigma^* \mid L \text{ felismerhető egy } O(t(n)) \text{ időkorlátos } M \text{ TG-el}\}$, ez egy nyelvosztály;;
- Tárkorlátos: $s: \mathbb{N} \rightarrow \mathbb{N}$, $s(n) \geq \log_2 n$, minden n -re; M TG $s(n)$ tárkorlátos, ha n hosszú bemenetek esetén lf $s(n)$ db cellát használ a munkaszalagon, azaz $S_M(n) \leq s(n)$;
- $\text{SPACE}(s(n)) = \{L \subseteq \Sigma^* \mid L \text{ felismerhető egy } O(s(n)) \text{ tárkorlátos } M \text{ TG-el}\}$;;;

P osztály:

- P osztály: $P = \bigcup_{k \geq 1} \text{TIME}(n^k)$. polinom időben egyszalagos TG-vel eldönthető nyelvek osztálya; P invariáns, hogy alap v spec TG-et használunk (ha spec polinom időben szimulálható alap TG-el); praktikus szgéppel megoldható problémák (ált található gyors algoritmus, pl: négyzetes, köbös);
- P-beli problémák:
- Térfogatszámítás: leszorítani testet egy négyzetbe, rákérdezni pontra, benne van-e, pontokat véletlenszerűen választani, igenek arányából kijön térfogat;
- path = $\{ \langle G, s, t \rangle \mid G \text{ ir gráf és van benne ir út } s\text{-ből } t\text{-be} \}$ – szélességi keresés; relprime = $\{ \langle x, y \rangle \mid x \text{ és } y \text{ rel prímek} \}$ – lnko;
- graph-2-color = $\{ G \mid G \text{ gráf csúcsai kiszínezhetőek 2 színnel} \}$ – van-e benne ptlan hosszú kör (nincs- akkor jó) $\Leftrightarrow$ páros gráf, szélességi keresés: $A_0 \cup A_2 \cup A_4$, ill $A_1 \cup A_3 \cup A_5$;
- euler-cycle = $\{ G \mid G \text{ gráfban van Euler-kör} \}$ – minden élen csak egyszer $\Leftrightarrow$ minden csúcs fokszáma páros;
- prime = $\{ x \mid x \text{ prím} \}$;
- gráf összefüggő-e? tetsz pont- mélységi v szélességi keresés (ha öf, akkor minden pont megvan);
- G-ben van-e háromszög? Minden ponthármaszt végignézni (n alatt 3) $\sim n^3/6$;;
- elérhetőségi probléma: adott $G=(U,E)$ ir gráf és $1, n \in V = \{1, \dots, n\}$ csúcsok, van-e ir út 1.csúcsból utolsóba? $S_0 = 0, S_1 = \{1\}$.. megjelöljük csúcsokat, kiveszünk egy csúcsot, amibe megyél és még nincs benne, akkor bele vesszük (ha n benne van, akkor jó); legyen $S := \{1\}$, és jelöljük meg egy csúcsot, amíg S nem üres: vegyünk ki egy i csúcsot S -ből, az összes $j \in V$ -re (ha j nem megjelölt és $(i,j) \in E$, akkor megjelöljük j -t és felvesszük S -be), ha n megjelölt, akkor kimenet: igen, egyébként: nem (implementálás: FIFO- szélességi, FILO- mélységi), időigénye: ciklus max n -szer fut le, ezen belül max $(n-1)$ összehasonlítás- $O(n^2)$;;
- 2SAT: megmutatjuk: $2SAT \leq_p$ elérhetőség; legyen Φ egy olyan formula, melyben minden tag pontosan 2 literált tartalmaz, megadunk G gráfot: G csúcsai Φ változói és azok tagadása; ha van Φ -ben egy $(A \vee B)$ tag, akkor vegyük fel G -ben: $\neg A$ -val ekv csúcsból B -be, valamint $\neg B$ -ből A -ba egy élt; látható, hogy Φ pontosan akkor kielégíthető, ha nincs olyan változó Φ -ben, hogy G -ben elérhető x -ből $\neg x$ és $\neg x$ -ből x – ez összesen polinom*polinom idejű, tehát polinom;;
- Boole hálózatok: ir, körmentes gráf, csúcsok befoka lehet 0 (bemeneti kapu, ígérték és változók), 1 (negált), 2 (és, vagy); kimeneti kapu (belőle nem megyél);
- hálózatkiértékelés (=HKS): P-teljes! adott változómentes C hálózat, kiértékelve hálózatot igazat kapunk v sem; megmutatjuk: elérhetőség $\leq_p$ HKS; G -hez megadunk egy változómentes C -t úgy, hogy G -ben pontosan akkor érthető el 1-ből n , amikor C igaz: vagy kapu: $g_{i,j,k}$ - ($h_{i,j,k-1}$ és $h_{i,k,k}$, és kapu: $h_{i,j,k}$ - ($g_{i,k,k-1}$ és $g_{k,j,k-1}$); leveleken: g_{ij0} kapu, ami igaz, ha $i=j$ v van i -ből j -be él; belátható, hogy C -ben kimeneti kapu értéke pontosan akkor igaz, amikor G -ben elérhető 1-ből n , mivel hálózat mérete polinomiális (n fv-ében), így HKS P-beli;;
- Monoton hálózatkiértékelés (=MHKS): P-teljes! adott C monoton hálózat, amiben nincs $\neg$ kapu, igaz-e C értéke; megmutatjuk: $HKS \leq_p$ MHKS (fordítva?); tetsz C változómentes hálózathoz meg lehet adni ekv monoton C' hálózatot, ötlet: minden kapunak elkészítjük komplementerét felhasználva De Morgan azonosságokat- ez polinom idejű;

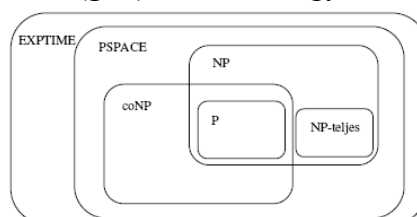
- nem P-beli problémák: eddig nem sikerült polinomiális algoritmust találni, nem feltételezünk benne, pl:
 - hamilton-cycle= $\{G \mid G \text{ gráfban van Hamilton-kör}\}$ – minden csúcson egyszer megy át, zárt út, ami minden pontot végigmegy, $(n-1)!$ sorrend;
 - graph-3-color= $\{G \mid G \text{ gráf csúcsai kiszínezhetők 3 színnel}\}$ – 3 partícióba szétosztom, n pont esetén 3^n kül színezés;;
- T: ha L nyelv $n^{\log n}$ -nél rövidebb időben nem ismerhető fel, akkor $L \notin P$;
- Biz: indirekt, tff $L \in P$, ekkor van olyan $k > 0$, hogy $L \in \text{TIME}(n^k)$, tehát találhatók olyan c és n_0 konstansok, hogy $T(n) \leq cn^k$, ha $n \geq n_0$, ebből következik, hogy: $n^{\log n} \leq cn^k$, ami viszont nem igaz;;
- TG szimulációjának idő és tárigénye: N k -szalagos TG-hez van olyan egyszalagos M TG, hogy $L(M) = L(N)$ és $T_M(n) \leq 2T_N(n)^2$, $S_M(n) \leq S_N(n) + n$, tehát P invariáns TG-re;;
- T: N TG-hez van olyan többszalagos M TG, hogy egy alkalmas n_0 számon túl: tetsz $\varepsilon > 0$ esetén: $T_M(n) \leq n(1 + \varepsilon)$, ha $T_N = O(n)$ (lin időkorlátos), tetsz $c > 0$ esetén: $T_M(n) \leq cT_N(n)$, ha $\lim T_N(n)/n = \text{Inf}$ (tetsz konstanssal leoszoríthatjuk futási időt, ezért tekinthetünk el konstans szorzótól időigény def-nál);;
- Tár-idő tétel: ha $L \in \text{SPACE}(s(n))$, akkor van olyan L -től függő c konstans, hogy: $L \in \text{TIME}(c^{s(n)})$; ((ha $L \in \text{TIME}(t(n))$, akkor $L \in \text{SPACE}(t(n))$));
- Biz: legyen M egy $b^*s(n)$ tárkorlátos L -et felismerő TG, mely külön munkaszalagon dolgozik; M egy K konfigurációjára:
 - mi van épp munkaszalagon ($\max S(n)$ cella) és hol áll a fej,
 - mi van input szalagon (összesen n jel) és hol áll a fej,
 - mi a gép pillanatnyi állapota;
- felső becslés- konfigurációk száma: $|Q||\Gamma|^{S(n)}(n+1)S(n)$ (áll, ennyi db cellát használunk, ennyiféle string input szalagon, fej ennyi cellán állhat), mivel $S(n) \geq \log_2 n$, $d = 2^2|\Gamma|$ választással: $\#K \leq c_1 d^{S(n)} = t$; ha t -nél több ideig fut M , akkor biztosan végtelen ciklusba kerül;
- megadunk N gépet: N tartalmazza M 2 példányát (M_1, M_2), x input; működése:
 - 1) M_1 -et elindítjuk x inputtal, minden lépés után felfüggesztjük futását (épp k . lépésnél tart), ha elfogadó v elutasító, akkor N megáll és elfogadja ill elutasítja x -et, különben 2-re lépünk;
 - 2) M_2 -t elindítjuk x inputtal és futtatjuk lf k . lépésig, ha egy j (ahol: $j < k$) lépés során M_2 konfigurációja megegyezik M_1 konfigurációjával, akkor végtelen ciklus áll ell, ekkor N megáll és elutasítja x -et, különben M_1 -et továbbléptetve 1-re ugrik; – TG $2x$ belép ua konfigurációba, ekkor nem áll meg soha; N
- erőforrásának becslése: $k \leq t$, $T_N(n) \leq O(t^2) = O((d^{S(n)})^2) = O((d^{b^*s(n)})^2) = O((d^{2b})^{s(n)})$, $c = d^{2b}$ választással teljesül tétel;;
- Nyelvosztályok komplementere: L nyelv esetén komplementere: $L^C = I^* \setminus L$; L nyelvosztály esetén komplementere: $\text{co}L = \{L \mid L^C \in L\}$ (nem halmazelméleti kompl);;
- T: $\text{co}(\text{co}L) = L$, Biz: $(L^C)^C = L$ alapján;;
- T: ha $A \subseteq B$ nyelvosztályok, akkor $\text{co}A \subseteq \text{co}B$, Biz: $L \in \text{co}A \rightarrow L^C \in A \rightarrow L^C \in B \rightarrow L \in \text{co}B$;;

- Tár-idő tételek: $\text{TIME}(t(n)) = \text{coTIME}(t(n))$, azaz $L \in \text{TIME}(t(n))$ csak akkor $L^C \in \text{TIME}(t(n))$;
- Biz: L -et eldöntő M TG-ben felcseréljük elfogadó és elutasító állapotokat, akkor egy azonos futásidejű L^C -t eldöntő TG-et kapunk, tehát $L^C \in \text{TIME}(t(n))$, azaz $\text{coTIME}(t(n)) \subseteq \text{TIME}(t(n))$, másrészt: $\text{TIME}(t(n)) = \text{co}(\text{coTIME}(t(n))) \subseteq \text{coTIME}(t(n))$, tehát: $\text{TIME}(t(n)) = \text{coTIME}(t(n))$;;
- T: $\text{SPACE}(s(n)) = \text{coSPACE}(s(n))$, azaz $L \in \text{SPACE}(s(n))$ csak akkor $L^C \in \text{SPACE}(s(n))$;
- Biz: legyen $L \in \text{SPACE}(s(n))$, tár-idő tételbeli N gép minden inputra megáll $O(s(n))$ tárkorlátos, felcserélve elfogadó és elutasító állapotokat: $L^C \in \text{SPACE}(s(n))$, azaz $\text{coSPACE}(s(n)) \subseteq \text{SPACE}(s(n))$, másrészt: $\text{SPACE}(s(n)) = \text{co}(\text{coSPACE}(s(n))) \subseteq \text{coSPACE}(s(n))$, így: $\text{SPACE}(s(n)) = \text{coSPACE}(s(n))$;;
- Nevezetes nyelvosztályok: $\text{EXPTIME} = \bigcup_{k \geq 1} \text{TIME}(2^{n^k})$..exp időben felismerhető nyelvek osztálya; $\text{PSPACE} = \bigcup_{k \geq 1} \text{SPACE}(n^k)$..polinom tárban felismerhető nyelvek osztálya;
- T: $P \subseteq \text{PSPACE} \subseteq \text{EXPTIME}$ (nem tudunk ilyet mondani: $L \notin P$ és $L \in \text{PSPACE}$, de olyat tudunk: $L \notin P$ és $L \in \text{EXPTIME}$, tehát valahol van valódi tartalmazás);
- Biz: ha M TG $t(n)$ időkorlátos, akkor $s(n)$ tárkorlátos is, tehát $\text{TIME}(n^k) \subseteq \text{SPACE}(n^k)$, és így: $P = \bigcup_{k \geq 1} \text{TIME}(n^k) \subseteq \bigcup_{k \geq 1} \text{SPACE}(n^k) = \text{PSPACE}$; ha $L \in \text{PSPACE}$, akkor valamely k -ra (létezik k) $L \in \text{SPACE}(n^k)$, tár-idő tétel szerint ekkor van alkalmas $c > 0$, hogy: $L \in \text{TIME}(c^{n^k}) \subseteq \text{TIME}(2^{\lg c \cdot n^k})$ ($\lg c$ konst) $\subseteq \text{TIME}(2^{n^{k+1}}) \subseteq \text{EXPTIME}$;;
- T: $\text{EXPTIME} \subsetneq \mathbb{R}$ (valódi!);
- Biz: $\subseteq$ következik EXPTIME időkorlátosságából; valódi tartalmazás: mutatunk L nyelvet, mely rekurzív, de nem EXPTIME -beli: $L = \{w \in \{0,1\}^*, M_w \text{ létezik és lf } 2^{2^{|w|}} \text{ lépésben elutasítja } w\text{-t}\}$;
 - L -nek végtelen sok eleme van (M_w -t kibővíthetünk soha el nem érhető állapotokkal, egy ilyen y leírás esetén M_y éppen úgy működik, mint M_w (időkorlátjuk azonos);
 - megmutatjuk, hogy L semmilyen n -re nincs benne $\text{TIME}(2^{2^{n-1}})$ -ben, ez elég is, mivel: $\text{EXPTIME} \subseteq \text{TIME}(2^{2^{n-1}})$;
 - biz: indirekt, tñh L felismerhető $c \cdot 2^{2^{n-1}}$ időkorlátos M TG-el, legyen n_0 olyan szám, hogy $c \cdot 2^{2^{n-1}} < 2^{2^n}$, ha $n \geq n_0$;
 - legyen w (n_0 -nál hosszabb szó), melyre M_w létezik és úgy viselkedik, mint M ; megmutatjuk:
 - ha $w \in L$, akkor M_w elfogadja w -t $c \cdot 2^{2^{n-1}} < 2^{2^n}$ lépésben – de ekkor L def szerint $w \notin L$;
 - ha $w \notin L$, akkor M_w nem utasítja el w -t $c \cdot 2^{2^{n-1}} < 2^{2^n}$ lépésben, de megáll, ezért elfogadja w -t, amiből $w \in L$ következne;;

NP osztály:

- NDTG időkorlátja: legyen $t: \mathbb{N} \rightarrow \mathbb{N}$, $t(n) \geq n$, minden n -re; M NDTG $t(n)$ időkorlátos, ha n hosszú bemenetek esetén M minden számítási út mentén $lf\ t(n)$ lépést megtéve megáll, azaz $lf\ t(n)+1$ mély számítási fa; (dupla max: minden stringre max, és adott stringre is leghosszabb számítási útvonal); nem ismert olyan fizikai megvalósítás, mely $t(n)$ -el arányos időben szimulálna egy $t(n)$ időkorlátos NDTG működését;
- $NTIME(t(n)) = \{L \subseteq \Sigma^* \mid L \text{ felismerhető egy } O(t(n)) \text{ időkorlátos } M \text{ NDTG-el}\};$
- NP nyelvosztály: $NP = \bigcup_{k \geq 1} NTIME(n^k)$. polinom időben egyszalagos NDTG-el eldönthető nyelvek osztálya; $P \subseteq NP$, mivel egy $t(n)$ időkorlátos TG tekinthető egy spec $t(n)$ időkorlátos NDTG-nek, így $TIME(n^k) \subseteq NTIME(n^k)$, de vajon: $P \subset NP$ vagy $P=NP$, ezt nem tudni;;
- T: $P \subseteq (NP \cap coNP)$;
- Biz: $P \subseteq NP$, ekkor $coP \subseteq coNP$, tudjuk, hogy: $coP = P$ (ha megcseréljük elfogadó és elutasító állapotokat), így: $P \subseteq coNP$ is teljesül;;
- NP-be tartozás megmutatása bizonyítékkal: legyen L egy Σ feletti nyelv, L polinomiálisan bizonyítható, ha van olyan V algoritmus, hogy: $L = \{w \mid V \text{ elfogadja } \langle w, c \rangle\}$ valamely Σ^* -beli c -re} és V polinom idejű w hosszát tekintve;
- T: NP azokat a nyelveket tartalmazza, amelyek polinomiálisan bizonyíthatóak;
- Biz1: $pol\ biz \rightarrow L \in NP$; legyen M a bizonyító V algoritmust megvalósító TG, időigénye: n^k (alkalmas k -ra), ekkor: c hossza $lf\ n^k$ lehet;
 - tekintsük N NDTG-t, mely w inputra (n hosszú) előállítja összes lehetséges n^k hosszú c sorozatot és minden egyes ilyen sorozatra lefuttatja M -et $\langle w, c \rangle$ inputtal,
 - N elfogadja w -t, ha M elfogadja $\langle w, c \rangle$ -t valamely c -re, különben elutasítja; $N \in NTIME(n^k)$, mivel M időigénye: n^k ;
- Biz2: $L \in NP \rightarrow pol\ biz$; legyen N L -et pol időben felismerő NDTG, ekkor N működését $lf\ m=n^k$ mélységű fa adja meg (ha bemenet n hosszú);
 - fa egy-egy ága N állátmeneteinek egy lehetséges sorozatát adja meg, egyes szinteken állátmenet indexe: $i_1, i_2, \dots, i_m$ sorozat, ahol i_j .j.szinten történt állátmenet indexe; minden $i_j \in [1..g]$, ahol: N állátmenetfv értéke $lf\ g$ állapotból álló halmaz;
 - tekintsük összes $i_1, i_2, \dots, i_m$ sorozatot és kódoljuk ezeket, mint Σ^* -beli sorozatokat- a tanúk ezek a c sorozatok lesznek;
 - a bizonyító M algoritmus így működik: n hosszú w input és n^k hosszú c tanú: szimulálja N működését a c -ben kódolt állátmenetekkel, ha N elfogadja w -t ezen az ágon, akkor M elfogadja $\langle w, c \rangle$ -t, különben elutasítja; M n^k időben lefut egy-egy c esetén;;
- állítás: $x \in L$, tanú, bizonyíték: y (nem hosszú, hatékonyan ellenőrizhető); Merlin azonnal kiszámolja és polinom időben bizonyítja, így meg tudja győzni Arthurt; sűgás: milyen számítási útvonal mentén lesz elfogadva szó;;

- NP-beli problémák:
- lovagok és udvarhölgyek: összeházasíthatóak-e úgy, hogy mindenkinek legyen párja $\Leftrightarrow$ van-e teljes párosítás (annyi fglen él, ahány pont bo b jo), tanú: egy párosítás;
- egymillió jegyű szám összetett-e: Merlin: igen, tanú: osztót megadni;
- gráf kiszínezhető-e 3 színnel? Igen, tanú: kiszínezi;
- van-e H-kör? I, tanú: egy H-kör (Arthur: egymás utáni pontok között valóban van-e él);
- gráf síkbarajzolható-e? (élek nem metszik egymást, nem tartalmazza K_5 v $K_{3,3}$ topológiai felbontást) igen, egy síkban lerajzolt gráf;
- 3SAT? Merlin megmutatja változók olyan értékeit, ami kielégíti;
- két gráf izomorf-e? (egy-egy értelmű leképezés csúcsok között), Merlin leképezést ad pontok között, Arthur ellenőrzi, hogy tényleg izomorf-e (minden pontpárra: (n alatt 2) $\sim n^2$ lépés);
- összetett számok? Minden osztóját megmondani (de csak hogy összetett-e, az P-beli!);
- NP-beli nyelv komplementere nem mindig NP-beli: nem ismert, hogy $coNP=NP$ teljesül-e;
- Merlin lovagai és udvarhölgyei nem házasíthatóak össze $\Leftrightarrow$ nincs teljes párosítás, tanú adható: (Hall tétel: G páros gráfban létezik A minden pontját lefedő párosítás $\Leftrightarrow$ bármely $s \subseteq A$, $|s| \leq |N(s)|$, s-ből kevesebb pontba képezünk(?));
- nincs H-kör gráfban: tanú nem adható;
- G gráf nem síkbarajzolható: tanú adható: valamelyik Kuratowski-gráf G részgráfja;
- 2 gráf nem izomorf? Nem egyértelmű;
- Visszavezethető: A,B Σ^* feletti nyelvek, A pol időben visszavezethető B-re, ha van $f: \Sigma^* \rightarrow \Sigma^*$ pol időben kiszámítható fv, hogy: $w \in A \Leftrightarrow f(w) \in B$, jele: $A \leq_p B$ (ha B NP-beli, akkor A is az, ha A nem NP-beli, akkor B sem az); ($A \leq B$: ha B eldönthető, akkor A is, ha A nem eldönthető, akkor B sem);;
- T: ha $A \leq_p B$ és $B \in P$, akkor $A \in P$;
- Biz: legyen M TG, mely polinom időben felismeri B-t, ekkor N TG A-t ismeri fel polinom időben: w input esetén kiszámítja $f(w)$ -t, M-et futtatja $f(w)$, és outputként M outputját adja;;
- NP-teljes: L NP-beli nyelv NP-teljes, ha abból a feltételezésből, hogy L P-beli is következik, hogy összes NP-beli nyelv P-beli, azaz $NP=P$; *másik def:* L NP-beli és tetsz A NP-beli probléma pol időben visszavezethető L-re (adható hozzá pol időben tanú);; legnehezebb, ezt használva minden NP problémát polinom időben meg lehetne oldani; nem érdemes rá polinom idejű algoritmust keresni (mivel felt: $P \subseteq NP$);;
- T: $NP \subseteq PSPACE$;
- Biz: legyen $L \in NP$, N L-et $t(n)$ időben felismerő NDTG, tekintsük azt M TG-et, mely N-et szimulálja: M N lehetséges állapotait veszi sorra (mélység szerint), ha N-ben max állátmenet-halmaz g elemű, akkor N fájában k. szinten $lf g^k$ áll szerepel, tehát összes állapot megvizsgálásához: $g^0 + g^1 + \dots + g^{t(n)} \leq (g+1)^{t(n)}$ idő kell; így alkalmas m-re: $t(n) \leq c \cdot 2^m$, tehát L e EXPTIME;
- T: ha B NP-teljes és $B \in P$, akkor $NP=P$;
- T: ha B NP-teljes és C NP-beli és $B \leq_p C$, akkor: C is NP-teljes;;



- NP-teljes problémák: $SAT = \{\Phi \mid \Phi \text{ kielégíthető Bool-kifejezés}\}$ (létezik olyan kombinációja változóknak, hogy 1-et ad) $\Phi = (\neg x \vee y) \vee (x \vee \neg z)$; $3SAT = \{\Phi \mid \Phi = (a_1 \vee b_1 \vee c_1) \wedge (a_2 \vee b_2 \vee c_2) \wedge \dots \wedge (a_k \vee b_k \vee c_k), \text{ valamely } k\text{-ra, } a_i, b_i, c_i \text{ tetsz Bool változó v negáltja, } \Phi \text{ kielégíthető Bool kif}\}$;
- Cook-Levin T: $SAT \in P$ csak akkor $NP=P$, azaz SAT NP-teljes;;
- ((hálózat-kielégíthetőség (=HKG)): adott C hálózat, lehet-e változóknak értéket adni úgy, hogy egyik kitüntetett kimeneti kapu értéke igaz legyen;)) CSAT: adott egy Φ KNF, kielégíthető-e Φ ? NP-beli: igen, pol visszavez: $HKG \leq_p CSAT$, C hálózathoz megadunk Φ KNF-et úgy, hogy C pontosan akkor kielégíthető, amikor Φ kielégíthető – ez polinom idejű konstrukció;;
- 3SAT: adott egy Φ KNF, melyben minden tag pontosan 3 literált tartalmaz, kielégíthető-e Φ ? SAT pol időben visszavezethető 3SAT-ra (át lehet írni 3 KNF alakra), vegyük $HKG \leq_p CSAT$ visszavezetést, az ott megkonstruált Φ helyett vegyük Φ' -t: minden olyan Φ -beli tag helyett, ami 3-nál kevesebb literált tartalmaz, vegyünk egy olyan tagot, melyben az eredeti tag egy literálja többször szerepel (pl: $g_1 \vee g_2$ helyett $g_1 \vee g_2 \vee g_2$); látható, hogy kapott Φ' ekv eredeti Φ -vel, vagyis a konstrukció HKG visszavezetése 3SAT-ra;;
- Nem mind egyenlő SAT (=NMESAT): adott 3SAT egy Φ példánya (=bemenete), kielégíthető-e Φ úgy, hogy tagokban nem minden literál értéke ugyanaz; pol visszavez: vegyük $HKG \leq_p CSAT$ visszavezetést, az ott megadott Φ helyett vegyük köv Φ' -t: azokhoz Φ -beli tagokhoz, amelyek kevés literált tartalmaznak, vagyoljunk hozzá egy új z változót (pl: g helyett $g \vee z \vee z$); belátható, hogy C pontosan akkor kielégíthető, ha $\Phi' \in NMESAT$;;
- vertex cover: lefogható-e G minden éle lf k ponttal;;
- k-colorability: kiszínezhető-e G minden éle lf k színnel (NP-teljes $k \geq 3$ esetén);;
- Halmazrendszer lefogható-e k ponttal? $A_1, \dots, A_l$, ahol: $A_i \in [n]$, $\exists? X_1, \dots, X_k \in [n]$, hogy minden A_i -ben benne van valamelyik x_j ; NP-beli: k pontú lefogást mutat; visszavezetjük NP-teljes problémát polinomiálisan erre: $3SAT \leq_p$ lefogás; $3SAT$: $(x_i \vee x_j \vee x_k) \wedge \dots \wedge \wedge$ db változó: $x_1, \dots, x_n$, adott 3SAT-hoz definiálunk halmazrendszert: alaphalmaz: $\{x_1, \dots, x_n, x_{l_1}, \dots, x_{l_n}\}$, hozzárendelünk 3 elemű halmazt F : $\{x_i, x_j, x_k\}$ és $\{x_i, x_{l_i}\}$ (ahány tagja KNF-nek és ahány változó van); B kielégíthető $\Leftrightarrow F$ lefogható n db ponttal;
 - Biz1: $\{x_i, x_{l_i}\}$ diszjunktak- n db ilyen pár van- úgy lefogni, hogy minden párból egyet veszünk, akkor maradék 3 elemű halmazok is lefoghatók (pl: lefogjuk x_j -vel, ekkor: $\{x_i, x_j, x_{l_k}\}$ is kielégíthető;
 - Biz2: vegyünk egy jó kielégítést, ami kijelöli minden x_i értékét (0 v 1), ami benne van megfelelő 3 elemű halmazban, ha $x_i=1$ - lefogó halmazrendszerbe bele vesszük x_i -t- párokat is lefogjuk, mivel $x_i \vee x_{l_i}$ mindenképp benne van;;
- H-path: NP-beli: TG megsejti gráf csúcsainak egy permutációját; $3SAT$ polinom időben visszavezethető H-path-ra: $3SAT \leq_p$ H-path; legyen Φ egy 3SAT alakú formula, minden x Φ -beli változóhoz legyen G-ben az alábbi részgráf (s-t, $x_1: d_1 \dots d_k, \dots x_l: d_1 \dots d_k$), ha változó poz-an szerepel klózban, akkor rombuszban él d_i -hez és vissza él csúcshoz, ha neg, akkor jo-ból indulunk d_i -hez és baloldaliba térünk vissza; Φ kielégíthető $\Leftrightarrow$ G-ben van H-út s-ből t-be (ha x_1 hamis, akkor jobbra, ha igaz, akkor balra);;

- $H\text{-path}_{u,v}$? gráfban van-e olyan H-út, ami u-ból v-be megy? *NP-beli*: megmutatjuk H-utat (2 egymás utáni pont valóban össze van-e kötve); ((H-kör $\leq_p$ H-path $_{u,v}$, Biz: $u=v$, $u \neq 0$)) Biz: $H\text{-path}_{u,v} \leq_p H\text{-path}(u_1, v_1)$ új csúcsok, akkor van H-út, amikor másik gráfban van u és v között út);;
- Gráf 17 színnel kiszínezhető? *NP-beli*: mutatunk jó színezést; *pol visszavez*: $3\text{szín} \leq_p 17\text{szín}$, G gráf és K_{14} (14 pontú teljes gráf, azaz minden pont minden másikkal össze van kötve, így minden pontot külön színnel kell színezni), tehát G-ben másik 3 színt kell használni;;
- Van-e legalább $n/2$ hosszú út? Adott n pontú gráf, *NP-teljes*: mutatunk la $n/2$ hosszú utat; *pol visszavez*: H-utat visszavezetjük erre, ugyanennyi üres pontot hozzáveszünk, ha eredeti gráfnak van H-útja, akkor új felének van;;
- G-ben van-e 5 pontú teljes részgráf (5 klikk)? *NP-beli*: mutatunk 5-t, ahol bármely két pont össze van kötve; *pol visszavez*: végignézzünk minden ötöst (n alatt 5) $\sim n^5$ lépésben ellenőrizni tudni, de $n/2$ teljes részgráf a kérdés, akkor: (n alatt $n/2$) $\sim \exp$ (NP-tejes);;
- Független csúcshalmaz (=FCS): adott G irlan gráf és k szám, van-e G-ben k elemű fglen csúcshalmaz? (egyik sincs összekötve másikkal); Biz: $3\text{SAT} \leq_p \text{FCS}$, legyen Φ a 3SAT egy bemenete, konstruáljuk meg G-t: Φ minden tagja meghatároz 3 csúcsot (háromszöget) G-ben, G-ben behúzzuk a háromszögekben az éleket és az ell-es literálpárok közti éleket; minden részgráfból kell 1-1 elemet kiválasztani, hogy ne legyenek összekötve- ezeknek csúcsoknak adhatok egymástól fglenül igaz értéket; belátható, hogy Φ pontosan akkor kielégíthető, amikor G-ben van k elemű FCS;;
- Klikk: adott G, k, van-e G-ben k elemű klikk (teljes részgráf)? Biz: $\text{FCS} \leq_p \text{klikk}$; egy G gráfban pontosan akkor van k elemű FCS, amikor G_l -ben van k elemű klikk; visszavez pol idejű;;
- Csúcslefedés (=CSF): adott G, k, van-e G-ben k db olyan csúcs, hogy G összes élének legalább egyik végpontja rajta van ezen csúcsok valamelyikén? Biz: $\text{FCS} \leq_p \text{CSF}$; egy n csúcsú G gráfban pontosan akkor van k csúcsú fglen csúcshalmaz, amikor G-ben van n-k csúcsú csúcslefedés;;;