Számításelmélet ZH

I. Elmélet

1. Nagyságrendek

Nagyságrendileg nem nagyobb

- "f nagyságrendileg nem nagyobb g-nél":

  $f \in O(g)$, $f(n) \in O(g(n))$, $f = O(g)$

- Definíció:

  $\exists c \in \mathbb{R}^+$, $N \in \mathbb{N}$ (küszöbindex) : $\forall n \geq N$ $|f(n)| \leq c \cdot |g(n)|$

- Példa:

f	g	
$n+2$	$5n$	$N=1$, $c=1$
$5n$	$n+2$	$N=1$, $c=5$
$5n$	n^2	$N=5$, $c=1$

Nagyságrendileg nem kisebb

- Jelölés:

  "f nagyságrendileg nem kisebb g-nél"

  $f \in \Omega(g)$

- Definíció:

  $\exists c \in \mathbb{R}^+$, $N \in \mathbb{N}$: $\forall n \geq N$: $|f(n)| \geq c \cdot |g(n)|$

Nagyságrendileg határozottan kisebb

- Jelölés:

  "f nagyságrendileg határozottan kisebb, mint g"

  $f \in o(g)$

- Definíció

  $f \in O(g) \wedge f \notin \Omega(g) \Rightarrow f \in o(g)$

Nagyságrendileg határozottan nagyobb

- Jelölés

$f \in \omega(g)$ (vagy: $f >> g$)

- Definíció

$f \in \Omega(g) \wedge f \notin O(g) \Longrightarrow f \in \omega(g)$

Nagyságrendileg egyenlő

- Jelölés:

$f \in \Theta(g)$

- Definíció:

$f \in \Theta(g) \Longleftrightarrow f \in O(g) \wedge f \in \Omega(g)$

azaz:

$\exists c_1, c_2 \in \mathbb{R}^+, N \in \mathbb{N}: \forall n \geq N: \quad c_1 \cdot |g(n)| \leq f(n) \leq c_2 \cdot |g(n)|$

Tulajdonságok

- Ezek a tulajdonságok tranzitívak (T)
- Konstans 0 fv. mindennél kisebb nagyságrendileg
  - $0 \ll c \quad (c \neq 0)$
  - jelölés: $\Theta(1)$
- Bármely fv-re: $f \in \Theta(a \cdot f + b) \quad (a \neq 0)$
- $f + g \in \Theta(f) \Longleftrightarrow g \in O(f)$
- logaritmus függvények nagyságrendileg határozottan kisebbek n-nél

  $\log(n) \ll n \quad (\log(n) \in o(n))$

Összefoglaló

- $0 \ll c \ll \lg(n) \ll n \ll n^k \ll n^{k+1} \ll c^n \quad (c \neq 0)$
- exponenciális fv. a legnagyobb, ha kitevője >1

1. Állítás: $x \in \mathbb{R}^+ : x \leq 2^x$

Bizonyítás:

- Segédtétel: $n \in \mathbb{N} : n+1 \leq 2^n$
  - Biz.: Teljes indukcióval
    1.) $0+1 = 1 \leq 2^0 = 1$ ✓
    2.) Tfh. $i \in \mathbb{N}$-ig igaz az állítás: $n = i$-ig teljesül
    3.) $n = i+1$-re:
    $$(i+1)+1 \leq 2 \cdot (i+1) \leq 2 \cdot 2^i = 2^{i+1}$$
- Tehát:
  $$\forall x \in \mathbb{R}^+ : \exists n \in \mathbb{N} : n \leq x \leq n+1$$
  $$x \leq n+1 \leq 2^n \leq 2^x \quad \blacksquare$$

2. Állítás: $\forall c > 1 : \exists N \in \mathbb{N} : \forall n \geq N : 2^* \leq c^n$

Bizonyítás:

$$\varepsilon := c - 1 > 0$$
$$c^n = (1+\varepsilon)^n = 1 + n \cdot \varepsilon + \underbrace{\sigma}_{\geq 0} \qquad N := \left\lceil \frac{1}{\varepsilon} \right\rceil \quad \blacksquare$$

MJ: Bármilyen x-et írhattunk volna 2^* helyére, az esetben:
$$N := \left\lceil \frac{x-1}{\varepsilon} \right\rceil$$

3. Állítás: $k \in \mathbb{N},\ c > 1 \Rightarrow n^k \in \sigma(c^n)$

Bizonyítás: 2. állításban tett. küszöbindex N^2

$$N^2 : 2 \leq c^n \ (n \geq N);$$

1. állítás felhasznál.

$$n \geq N \cdot k : n^k = (N \cdot k)^k \cdot \left(\frac{n}{N \cdot k}\right)^k \leq (n \cdot k)^k \cdot \left(2^{\frac{n}{N k}}\right)^k =$$
$$= (N \cdot k)^k \cdot 2^{\frac{n}{N}} \leq (N \cdot k)^k \cdot (c^n)^{\frac{n}{N}} = (N \cdot k)^k \cdot c^n \quad \blacksquare$$

4. Állítás: $k \in \mathbb{N},\ c > 1 : n^k \notin \Omega(c^n)$

Bizonyítás: Indirekt módon

Tfh. $n^k \in \Omega(c^n) : \exists d_1 \in \mathbb{R}^+,\ N_1 \in \mathbb{N} : \forall n \geq N_1 : n^k \geq d_1 \cdot c^n$

$\exists c_1 : 1 \leq c_1 \underset{\text{3. áll.}}{\leq} c : \exists d_2 \in \mathbb{R}^+,\ N_2 \in \mathbb{N} : \forall n \geq N_2 : n^k \leq d_2 \cdot c_1^n$

$$\forall n \geq \max(N_2, N_1) : d_1 \cdot c^n \leq n^k \leq d_2 \cdot c_1^n$$
$$\Rightarrow \left(\frac{c}{c_1}\right)^n \leq \frac{d_2}{d_1} \quad (2. \text{ állítás}) \quad \lightning \quad \blacksquare$$

2. Számosság — a) Véges és végtelen számosság

Halmazok számossága

- A és B halmazoknak **megegyezik a számossága**, ha létezik bijekció köztük. ($|A| = |B|$)
- A **számossága** legalább annyi, mint B számossága, ha van B-ből injekció A-ba. ($|A| \geq |B|$)
- A **számossága** nagyobb, mint B számossága, ha van B-ből injekció A-ba, de bijekció nincs. ($|A| > |B|$)

Cantor-Bernstein tétel

Ha A-ból B-be van injekció és B-ből A-ba is van, akkor A és B között bijekció is van, azaz ha $|A| \leq |B|$ és $|A| \geq |B|$, akkor $|A| = |B|$

Megszámlálhatóan véges számosság

- $\mathbb{N}$ számosságát megszámlálhatóan végtelennek nevezzük.
- Egy halmaz megszámlálható, ha véges / megszámlálhatóan végtelen.

Tétel

Megszámlálható sok megszámlálható halmaz uniója megszámlálható.

Példa: $|\mathbb{N}| = |\mathbb{Z}|$

Számosságuk azonos, mivel létezik bijekció köztük:

$$f: \mathbb{N} \to \mathbb{Z} : f(n) = \begin{cases} n/2 & \text{, ha } n \text{ páros} \\ -\frac{n+1}{2} & \text{, ha } n \text{ páratlan} \end{cases}$$

$$g: \mathbb{Z} \to \mathbb{N} : g(z) = \begin{cases} 2z, & \text{ha } z > 0 \\ -z \cdot 2 - 1 & \text{ha } z < 0 \end{cases}$$

Continuum számosság

$\mathbb{R}$ számosságát continuumnak nevezzük. ($|\mathbb{R}| > |\mathbb{N}|$)

Példa: $|(0,1)| = |\mathbb{R}|$

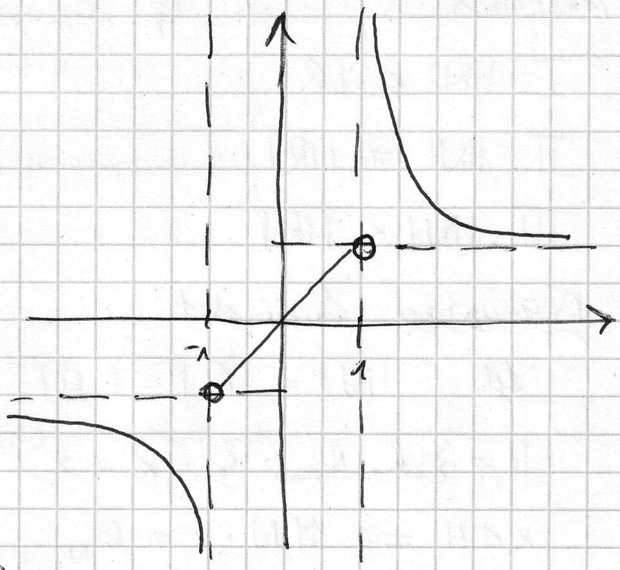
$\operatorname{tg}(\pi(x - \frac{1}{2}))\big|_{(0,1)} : (0,1) \to \mathbb{R}$ bijekció $(0,1)$ és $\mathbb{R}$ között.

<u>MJ</u>: $|\mathbb{R}| = |(a,b)| = |[c,d]|$ és $|\mathbb{R}| = |\mathbb{R}^n|$

Példa: $|(-2,2)| = |\mathbb{R}|$

$$f(x) = \begin{cases} \frac{1}{x+1} & \text{ha } x \in (-2,-1) \\ x & \text{ha } x \in (-1,1) \\ \frac{1}{x-1} & \text{ha } x \in (1,2) \end{cases}$$

$(-2,2) \to \mathbb{R}$



$$\mathbb{R} \to (-2,2): \quad g(r) = \begin{cases} \frac{1}{r}-1, & \text{ha } r \in (-1,-\infty) \\ r, & \text{ha } r \in (-1,1) \\ \frac{1}{r}+1, & \text{ha } r \in (1,+\infty) \end{cases}$$

b) Szavakkal kapcsolatos számosságok

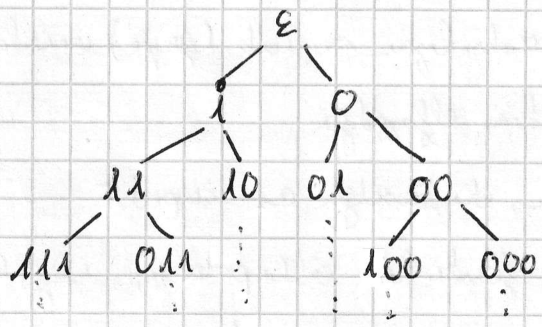
Példa: $|\{\text{véges hosszú bináris szavak}\}| = |\mathbb{N}|$

Mj: $\{\text{véges hosszú bináris szavak}\} = \{0,1\}^n \quad (\infty > n \in \mathbb{N})$

- Hossz-lexikografikus (shortlex) rendezés egy bijekció:

  $\varepsilon, 0, 1, 00, 01, 10, 11, 000, 001, 010, 011, \dots$

- Rendezés bináris fával:



Példa: $|\{\text{megszámlálhatóan végtelen hosszú bináris szavak}\}| = |\mathbb{R}|$

$$\underset{\text{tetsz.}}{\underline{x}}\,0\dot{1} = x\,1\dot{0}$$

Példa: $|\{\text{véges hosszú bin. szavakból álló nyelvek halmaza}\}| = |\mathbb{R}|$

- Egy nyelvet meghatároz az, hogy mely szavak vannak benne, melyek nem
  - ↳ ez egy végtelen hosszú bináris számmal "ábrázolható"
    - ↳ 1 az első bit, ha ε eleme a nyelvnek, 0, ha nem, így tovább...
  - ↳ $|\{\text{megszáml. } \infty \text{ hosszú bin. szavak}\}| = |\mathbb{R}|$

Állítás

I. $|\mathbb{N}| \leq |\mathbb{R}|$

II. $|\mathbb{N}| \neq |\mathbb{R}|$

III. $|\mathbb{N}| < |\mathbb{R}|$

Bizonyítás: Indirekt

Tfh. $|\mathbb{N}| = |\mathbb{R}| = |H|$

$H = \{h_0, h_1, \dots\}$ $\quad x := x^{(i)} = \overline{h_i^{(i)}}$

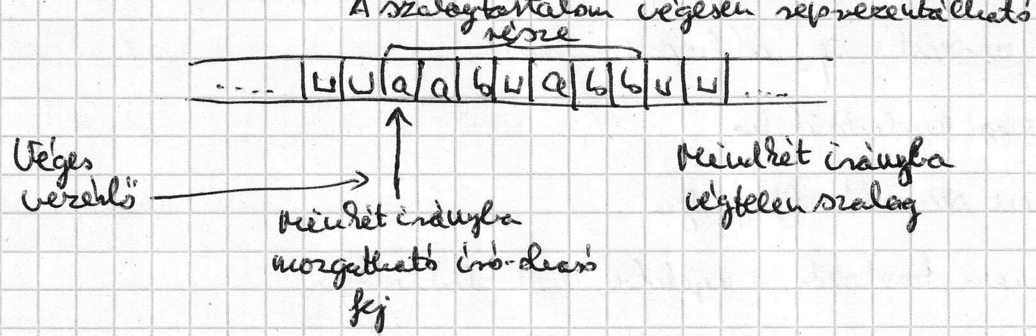
$x \in H \Rightarrow \exists N.\, x = h_N \quad h_N^{(N)} = x^{(N)} = \overline{h_N^{(N)}}$ ↯

3. Turing-gép a) Egyszalagos Turing gép

Bevezető

- a Turing gép (TG) az algoritmus egy lehetséges modellje
- a TG egyetlen programot hajt végre (bármely inputra!), azaz tekinthető egy célszámítógépnek
- működése
  - kezdetben egy input szó van a szalagon, a fej ennek első betűjétől indul
  - az indulást követően a fej a szabályai szerint (gépé) működik
  - ha eljut az elfogadó állapotába, elfogadja
  - ha eljut az elutasító állapotba, elutasítja az inputot
  - 3. lehetőség: nem jut el soha a fenti két állapotba, "végtelen ciklusba" kerül
- a gép determinisztikus
- a végtelen szalag potenciálisan végtelen tár
- egy P probléma példányait egy megfelelő ábécé felett elkódolva a probléma igen-példányai egy L_P formális nyelvet alkotnak.

  L_P (és így a probléma is) eldönthető, ha van olyan mindig termináló gép, mely pontosan L_P szavait fogadja el.



- a Church-Turing tézis értelmében úgy gondolhatjuk tehát, hogy éppen a TG-pel eldönthető problémák (nyelvek) az algoritmikusan eldönthető problémák.

Turing-gép

A Turing gép (TG) egy $M = \langle Q, \Sigma, \Gamma, \delta, q_0, q_i, q_n \rangle$ rendezett hetes, ahol

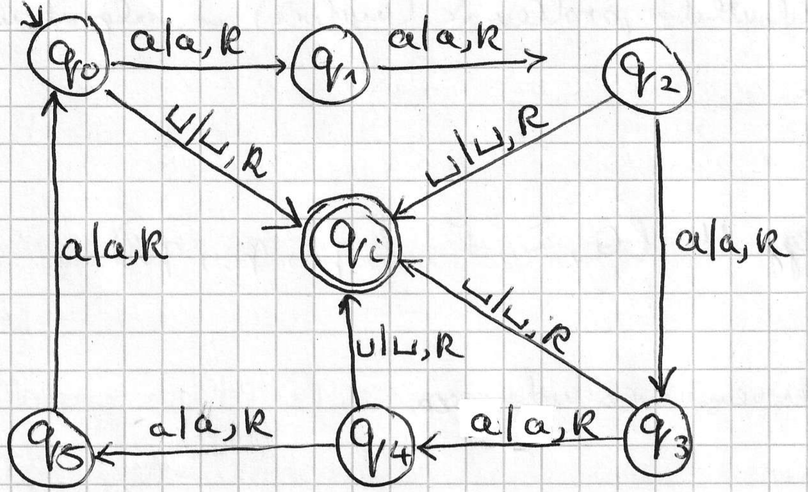
- Q: állapotok véges, nem üres halmaza
- $q_0, q_i, q_n \in Q$
  - q_0: kezdő állapot
  - q_i: elfogadó állapot
  - q_n: elutasító állapot
- Σ és Γ ábécék
  - Σ: bemenő jelek ábécéje
  - Γ: szalagszimbólumok ábécéje
  - $\Sigma \subseteq \Gamma$ és $\sqcup \in \Gamma \setminus \Sigma$
- δ átmenet fv.: $\delta : (Q \setminus \{q_i, q_n\}) \times \Gamma \to Q \times \Gamma \times \{L, S, R\}$
  - $\{L, S, R\}$: TG lépéseinek irányai
    - ↳ L: balra (left)
    - ↳ R: jobbra (right)
    - ↳ S: helyben marad (stand)

Működése

- az író-olvasó fej egy lépésben beolvassa a szalagi lécé megfelelő tagját, melyet írhat is, majd mozog (L/R/S)
- elfogadja a szót, ha q_i állapotba kerül

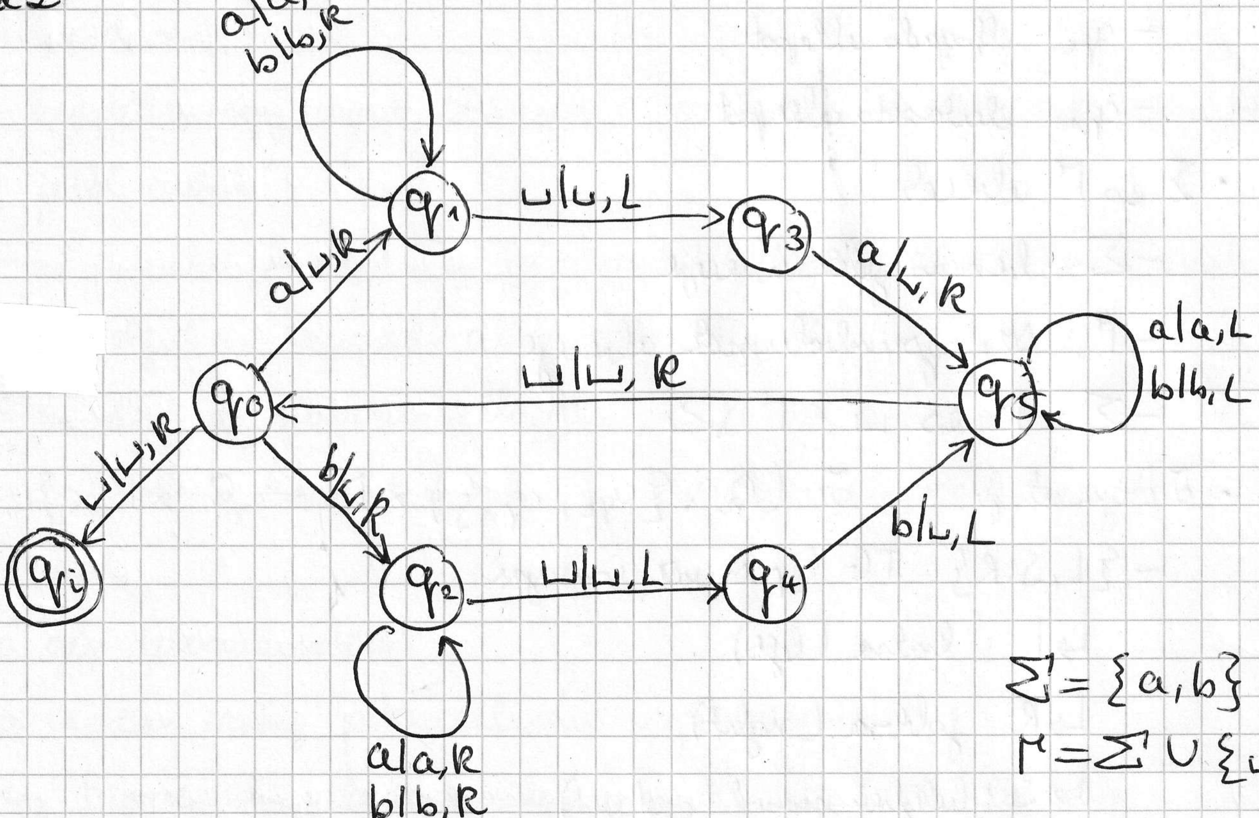
- elutasítja a szót: q_n állapotba kerül
- felismer egy nyelvtant, ha
  - minden szavát elfogadja
  - ami nem tartozik a nyelvbe, azt elutasítja

Példa 1



2 vagy 3-al osztható hosszúságú szavakat fogad el

Példa 2



$\Sigma = \{a, b\}$

$\Gamma = \Sigma \cup \{\sqcup\}$

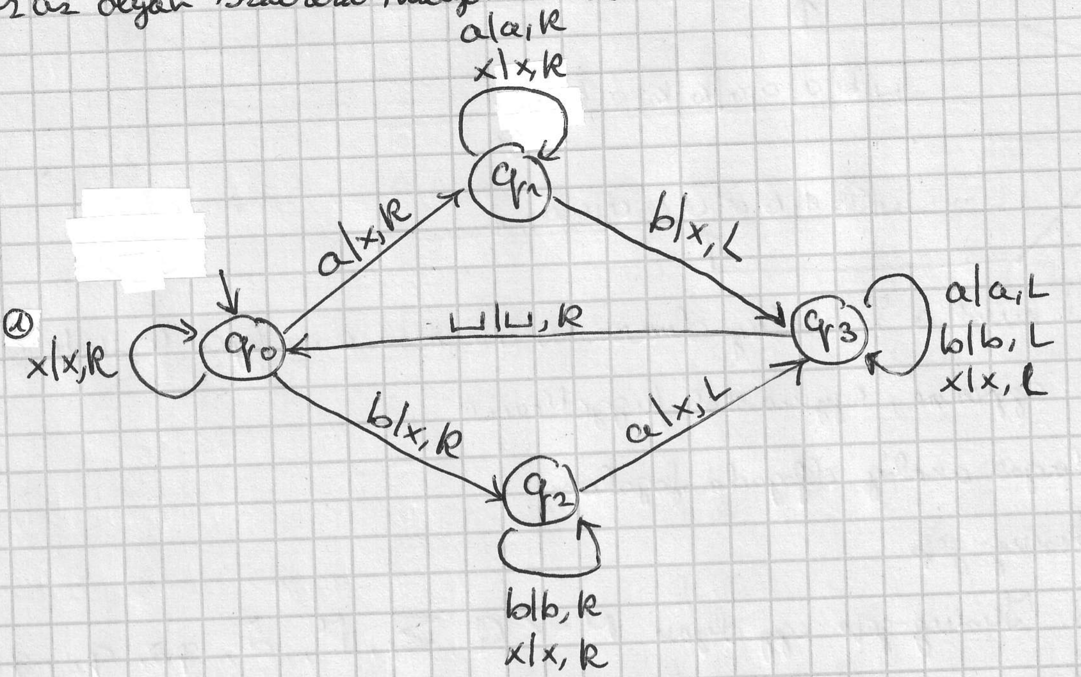
Csak a páros hosszúságú palindromokat fogadja el

- vagy töröl a szó elejéről egy "a"-t, majd a végére megy és ott is
- vagy töröl a szó elejéről egy "b"-t, majd a végére megy és ott is.
- a művelet után a fej visszamegy a szó elejére, s elfogadja ha üres a szalag

Feladat: adj meg olyan gépet, ami a következő szavakat fogadja el:

$\Sigma = \{a, b\}$; $\Gamma = \Sigma \cup \{\sqcup, x\}$; $\{u \in \{a, b\}^* \mid |u|_a = |u|_b\}$

(azaz olyan szavakat, melyekben az "a" és "b" karakterek száma azonos)



- x karakter azért szükséges, mivel ha a szó közepén törölnénk, nem tudnánk megállapítani, hol a szó vége (törölt karakter helyére "⊔" kerül, akár a szó elején és végén)

① itt optimalizálhatunk, ha a szó eleji x-eket töröljük, így a későbbiekben nem kell rajtuk átlépkedni

Műveletigény

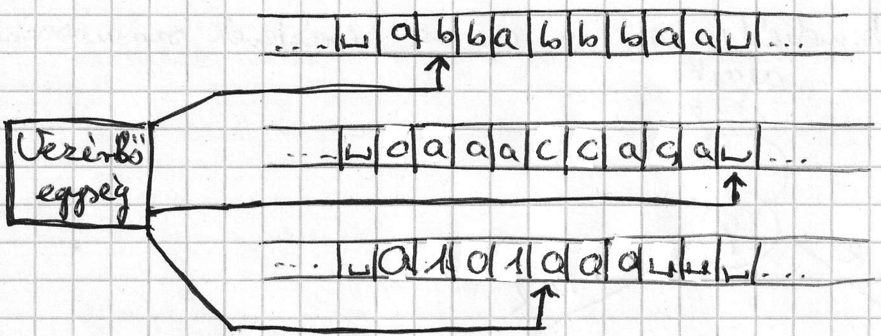
- $O(f(n))$ a műveletigény, ha legfeljebb $f(n)$ lépésben véget ér
- $\Theta(f(n))$ ha van olyan input, melyre nagyságrendileg $f(n)$ a lépésigény (pontosabb, mint $O(f(n))$)

Példák:

- $\Theta(n)$: lineáris műveletigény
- $\Theta(n^2)$: négyzetes nagyságrendű → Példa 2 műveletigénye (de ha nem palindromot kap, előbb leáll)

b) k-szalagos Turing-gép

Bevezetés



- Egy ütem: mind a k szalag olvasása, átírása és a fejek léptetése egyszerre, egymástól függetlenül
- Az egyszalagos analóg elfogadásfogalom

k-szalagos Turing-gép

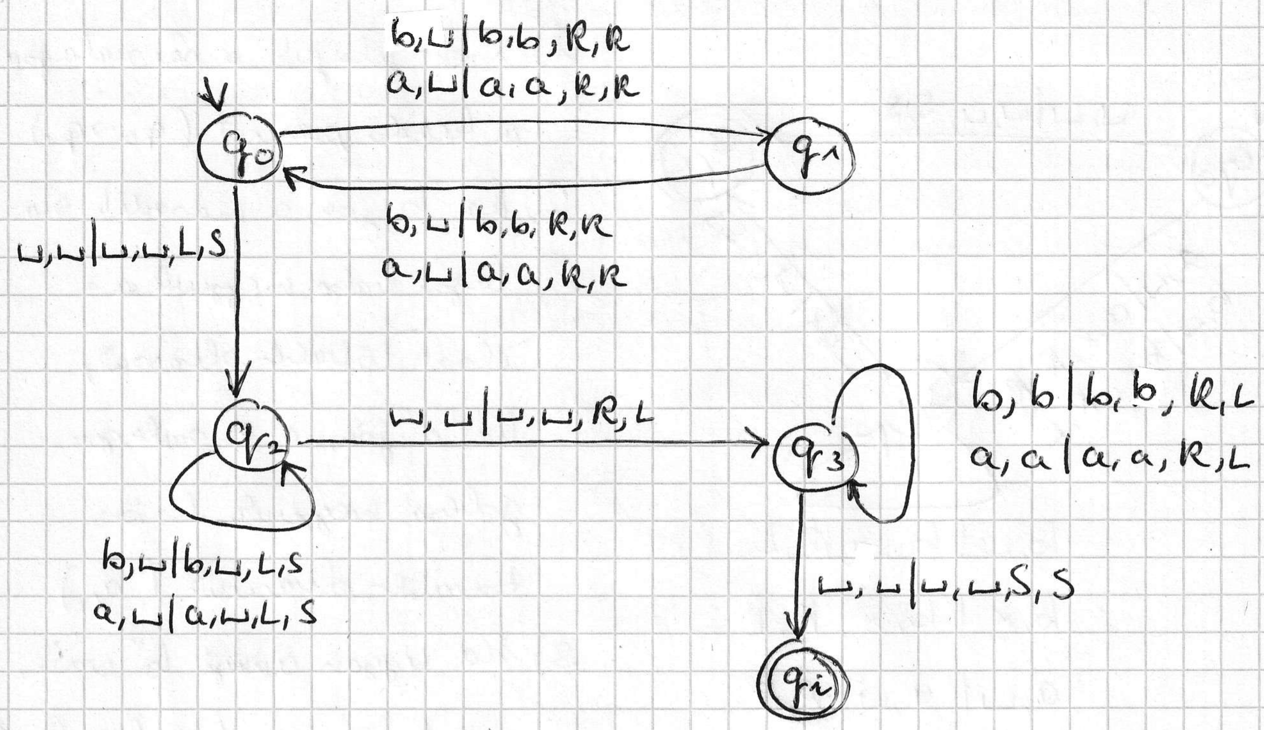
A k-szalagos Turing-gép egy olyan $M = \langle Q, \Sigma, \Gamma, \delta, q_0, q_i, q_n \rangle$ rendszer, ahol

- Q az állapotok véges, nem üres halmaza
- $q_0, q_i, q_n \in Q$
  - q_0: kezdő állapot
  - q_i: elfogadó állapot
  - q_n: elutasító állapot
- Σ és Γ ábécék
  - Σ: bemenő jelek ábécéje
  - Γ: szalagszimbólumok ábécéje
  - $\Sigma \subseteq \Gamma$ és $\sqcup \in \Gamma \setminus \Sigma$
- δ: átmenet fv.

Működése

- a fejek nem kell, hogy szinkronban lépjenek, egymástól függetlenül mozgathatók
- analóg az egyszalagossal, csak itt egy lépésben egyszerre több fej dolgozik
- műveletigény számításánál egy lépés (több fej egyszeri mozgatása) az egység
- amit 1 szalagossal meg lehet csinálni → többszalagossal is lehet
- amit több szalagossal meg lehet csinálni → 1 szalagossal is lehet
  ↳ (több szalagossal hatékonyabb)

Példa

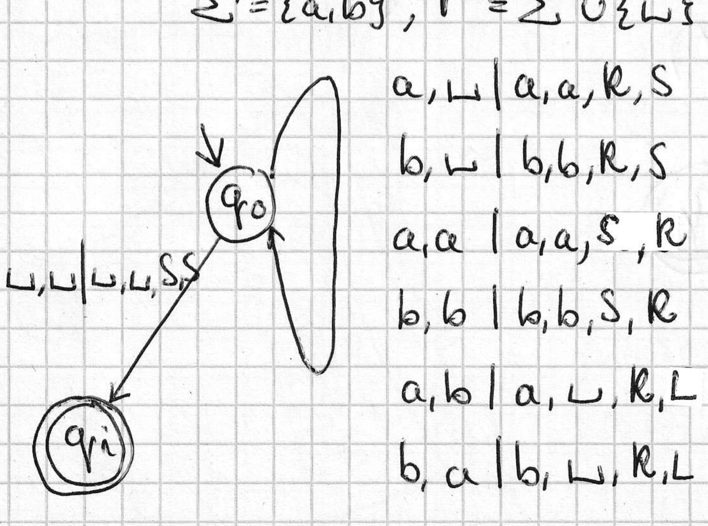


A gép a páros hosszú palindromokat fogadja el.

1) Az 1. szalag tartalmát átmásoljuk a 2. szalagra (q_1)
2) Ha a másolás kész, az 1. szalagon a fejet előre visszük (q_2)
   ↳ így az 1. fej a szó elején az 1. szalagon,
   a 2. fej a szó végén a 2. szalagon van
3) Az 1. szalagon előre, a 2. szalagon hátra halad a fej, amíg ugyan azt olvassák (q_3)
4.) Ha a fejek elakadnak valahol (kül. karaktert olvasnak) a gép elutasít, ha mind2 fej a szó elejére illetve végére ért, a gép elfogad (q_i)

Feladat: 2 szalaggal elfogadni ugyan annyi "a"-t mint "b"-t tart. szavakat

$\Sigma = \{a,b\}$, $\Gamma = \Sigma \cup \{\sqcup\}$

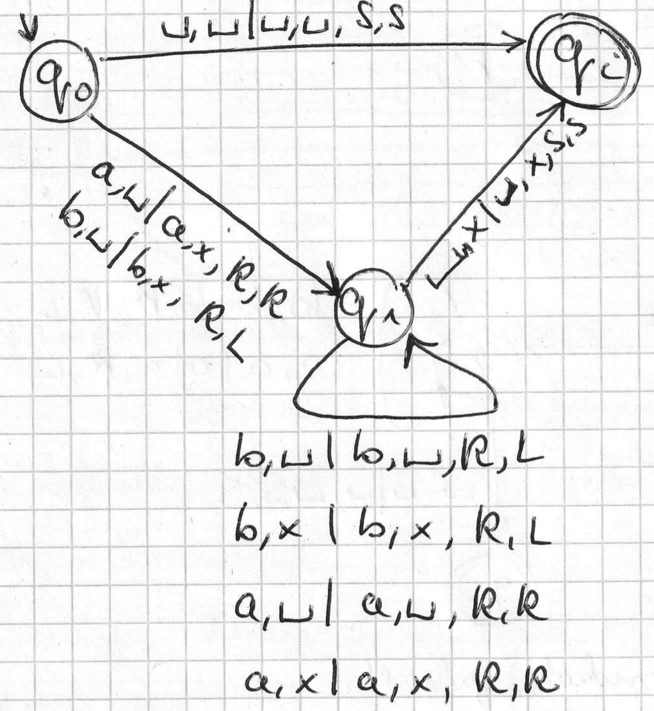


Veremautomatát szimuláljuk
(helyes zárójelezés)

(Második szalagon gyűjtjük a karaktereket; ha ellentétes karaktert olvasunk az elsőről, töröljük, ha azonosat, írjuk)

↳ A 2. szalag veremszerűen működik

Kétszalagos-megoldás:



1.) x-el jelöljük a 2. szalagon a kezdő pozíciót ($q_0 \to q_1$)

2.) ha "b" jön, a második szalagon balra megyünk, az elsőt tovább olvassuk, ha "a" jön, a 2. szalagon jobbra megyünk 1.-ön tovább olvasunk (q_1)

3.) Ha ugyan annyi "b" van, mint "a", az 1. szalag beolvasásának végére a 2. szalagon "x"-el jelölt kezdőpozícióba érünk: elfogadjuk ($q_1 \to q_i$)

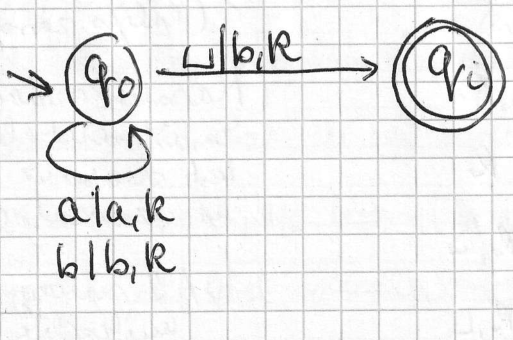
4. Függvények kiszámítása Turing-géppel

Szófüggvényt kiszámító TG

Azt mondjuk, hogy az $M = \langle Q, \Sigma', \Gamma, \delta, q_0, q_i, (q_n) \rangle$ TG kiszámítja az $f: \Sigma'^* \to \Delta^*$ szófüggvényt, ha minden $u \in \Sigma'^*$-beli szóra megáll, és akkor $f(u) \in \Delta^*$ olvasható az utolsó szalagján.

NJ: Nincs szükség q_i és q_n megkülönböztetésére, elég benne egyetlen megállási pont.

Pl.: $f(u) = ub$ $\quad (u \in \{a, b\}^*)$

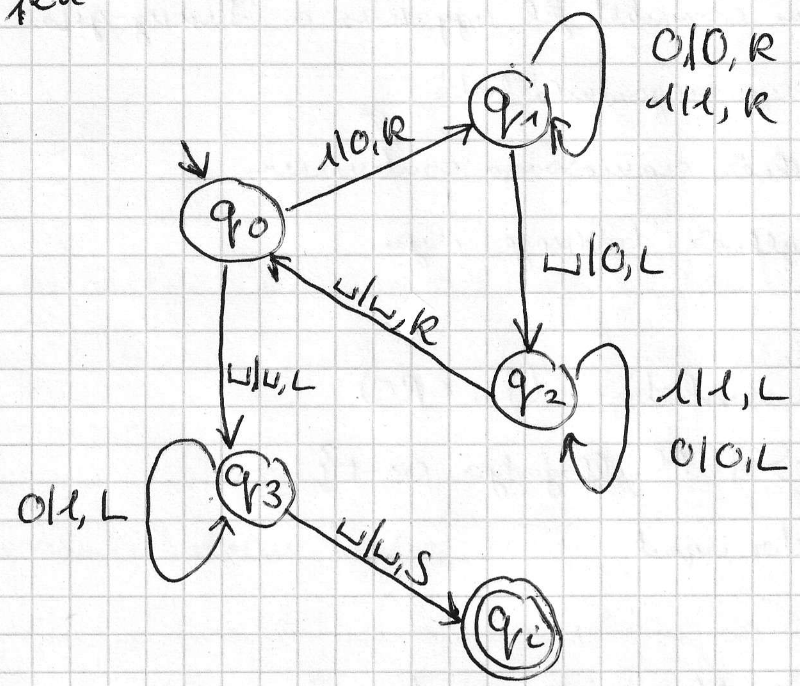


Példa

$f: \Sigma^* \to \Sigma^*$ $\Sigma = \{1\}$

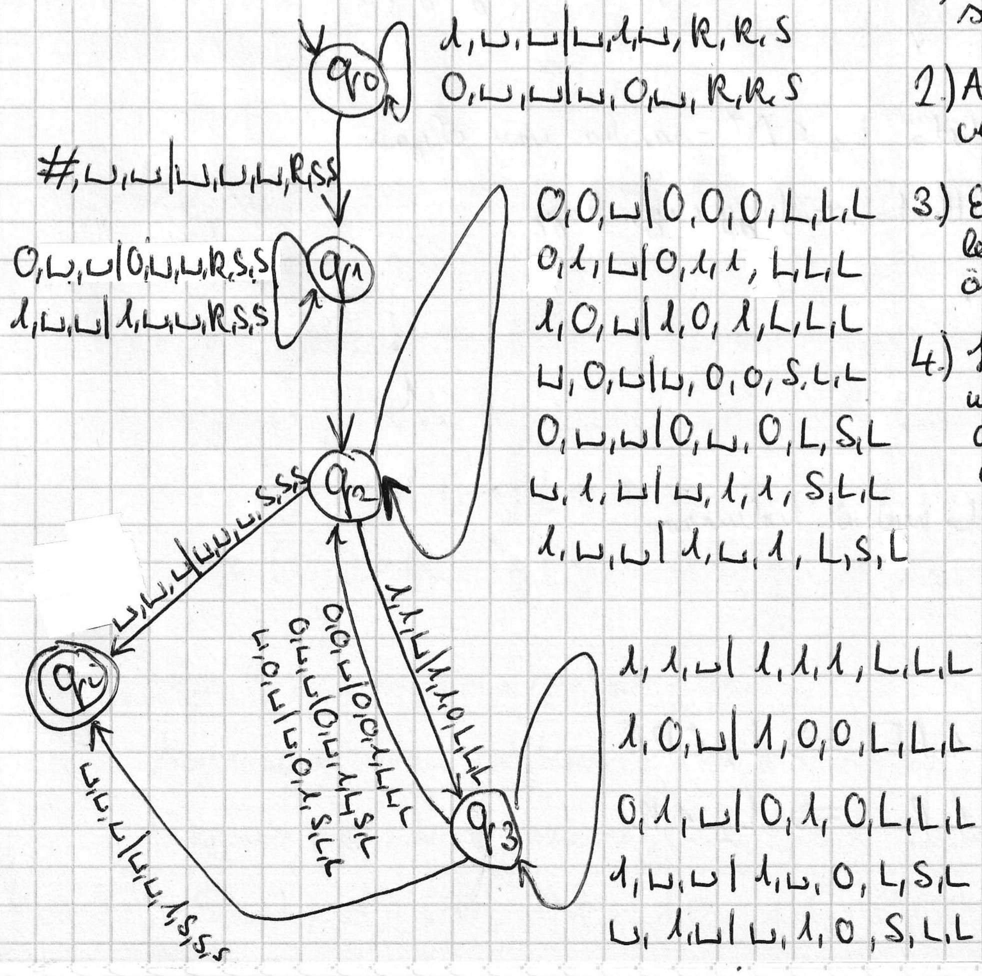
$w \in \Sigma^*$ $f(w)$

$f: 1^n \to 1^{2n}$



Feladat: Az első szalagon lévő #-el elválasztott u és v bináris szavak összegét állítsd elő többszalagos Turing géppel.

$f: u\#v \to u+v$



1) Átmásoljuk az 1. szót a 2. szalagra a "#"-ig (q_0)

2) Az 1. szalagon maradt szó végére megyünk (q_1)

3) Elkezdjük az 1. és 2. szalagon lévő bin. számokat hátulról összeadni (q_2)

4) Ha túlcsordulás van (azaz mind 2 szalagon "1" következik) átlépünk egy másik állapotba így jegyezve meg, hogy van 1 maradék (q_3)

5) A maradék figyelembe vételével folytatjuk az összeadást. (q_3)

6) Ha már nincs maradék visszatérünk q_2-be, vagy ha végére értünk a műveletnek elfogadó állapotba kerülünk.

5. Nem Turing-felismerhető nyelvek

Tétel: létezik nem Turing-felismerhető nyelv

Bizonyítás:

- Két kül. nyelvet nem ismerhet fel ugyanaz a Turing-gép.
- A TG-ek számossága megszámlálható
- A $\{0,1\}$ feletti nyelvek számossága continuum.
- Azaz: valójában a nyelvek "többsége" ilyen.

Példák:

- Univerzális TG felismerhető $(L_u \in RE)$

$$L_u = \{ \underset{\substack{\uparrow \\ TG}}{\langle M}, \underset{\substack{\uparrow \\ \text{tetsz. input}}}{w \rangle} \mid M \text{ elfogadja } w\text{-t} \}$$

- Megállási probléma

$$L_{HALT} = \{ \langle M, w \rangle \mid M \text{ megáll } w\text{-n} \}$$

6. Visszavezetés

Kiszámítható fv.

$f: \Sigma^* \to \Delta^*$ kiszámítható, ha van olyan Turing-gép, ami kiszámítja.

Visszavezetés

$L_1 \subseteq \Sigma^*$ visszavezethető $L_2 \subseteq \Delta^*$-ra, ha van olyan
$f: \Sigma^* \to \Delta^*$ kiszámítható szófüggvény, hogy
$w \in L_1 \iff f(w) \in L_2$.

Jel: $L_1 \leq L_2$

Jelölések

RE: Turing-felismerhető nyelvek halmaza

R: megáll

Tétel

- Ha $L_1 \leq L_2$ és $L_1 \notin RE \Rightarrow L_2 \notin RE$
- Ha $L_1 \leq L_2$ és $L_1 \notin R \Rightarrow L_2 \notin R$
- Ha $L \in RE \Rightarrow L \in R$

feloléses

- P: determinisztikus Turing-géppel polinom időben eldönthető probléma
- NP: nem P

II. Gyakorlat

1. Mi az alábbi halmazok számossága (megszámlálhatóan végtelen, vagy kontinuum sok)? Válaszodat indokold! (12p)

ⓐ $\mathbb{Z}$

Megszámlálhatóan végtelen.

Mivel $|\mathbb{Z}| = |\mathbb{N}|$ ($\mathbb{N}$ megszámlálhatóan végtelen)

Ugyanis: létezik bijekció $\mathbb{N}$ és $\mathbb{Z}$ között:

$$f: \mathbb{N} \to \mathbb{Z}: \quad f(n) = \begin{cases} \frac{n}{2}, & \text{ha } n \text{ páros} \\ \frac{-n+1}{2}, & \text{ha } n \text{ páratlan} \end{cases}$$

$$g: \mathbb{Z} \to \mathbb{N}: \quad g(z) = \begin{cases} 2 \cdot z, & \text{ha } z \geq 0 \\ -2 \cdot z - 1, & \text{ha } z < 0 \end{cases}$$

ⓑ $(0;1)$ nyílt intervallum pontjai

Kontinuum sok.

Mivel $|(0;1)| = |\mathbb{R}|$ ($\mathbb{R}$ számossága kontinuum)

Ugyanis: létezik bijekció $(0;1)$ és $\mathbb{R}$ között:

$$f: \mathbb{R} \to (0;1): f(x) = tg(\pi(x - \tfrac{1}{2}))$$

$$g: (0;1) \to \mathbb{R}: g(y) = \frac{arc\,tg(y)}{\pi} + \frac{1}{2}$$

ⓒ véges hosszú bináris szavak halmaza

Megszámlálhatóan végtelen, mivel felsorolható.

Ugyanis: pl. hossz szerint, azon belül lexikografikusan rendezve:

$\varepsilon, 0, 1, 00, 01, 10, 11, 000, 001, 010, 011, \ldots$

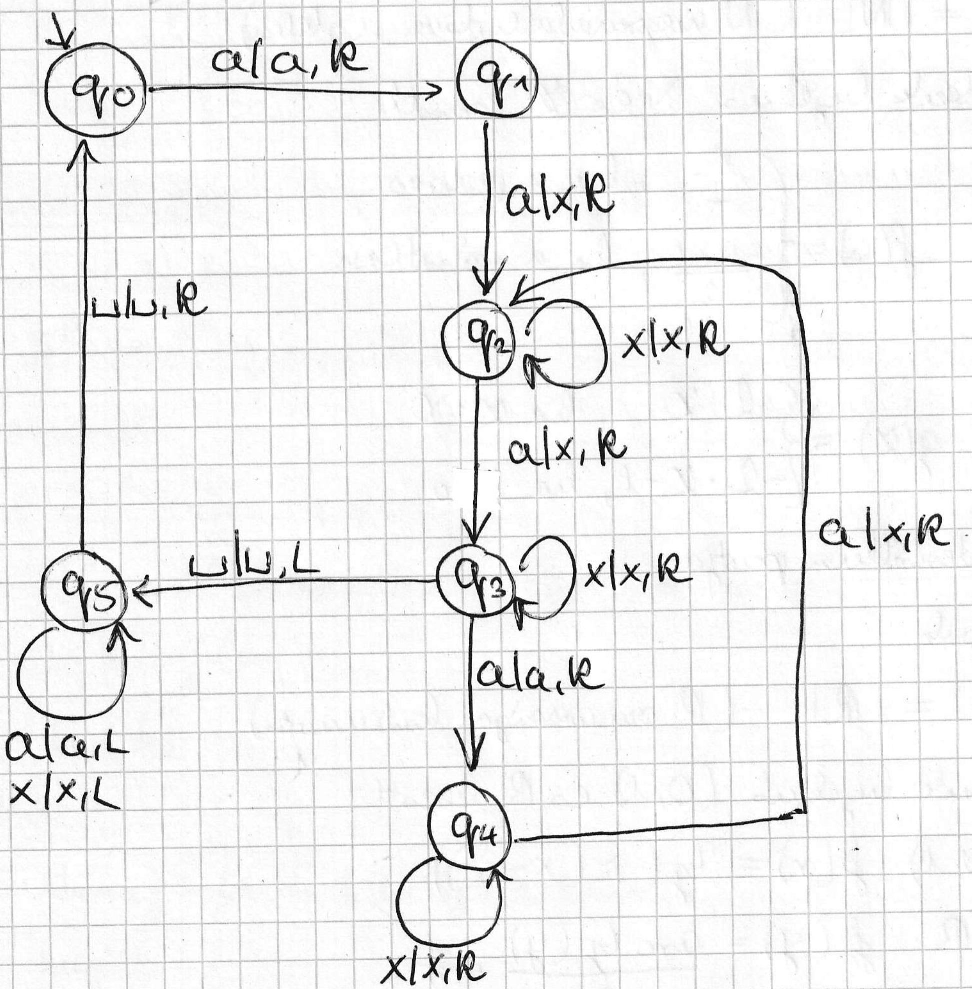
2. Milyen nyelvet ismer fel az alábbi Turing-gép? (10p)

$M = (Q, \Sigma, \Gamma, \delta, q_0, q_i, q_n)$ $Q = \{q_0, q_i, q_n, q_1, q_2, q_3, q_4, q_5\}$

$\Sigma = \{a\}$ $\Gamma = \Sigma \cup \{\sqcup, x\}$

$\delta =$

	a	x	⊔
q_0	(q_1, a, R)	(q_n, x, R)	$(q_n, \sqcup, R)$
q_1	(q_2, x, R)	(q_1, x, R)	$(q_i, \sqcup, R)$
q_2	(q_3, x, R)	(q_2, x, R)	$(q_n, \sqcup, R)$
q_3	(q_4, a, R)	(q_3, x, R)	$(q_5, \sqcup, L)$
q_4	(q_2, x, R)	(q_4, x, R)	$(q_n, \sqcup, R)$
q_5	(q_5, a, L)	(q_5, x, L)	$(q_0, \sqcup, R)$



A folyamat során minden 3. a-t hagyjuk csak meg (a többi helyére x-et írunk)
↳ tulajdonképpen valahányszor végigmegyünk a szón, osztjuk 3-al.

$L(M) = \{a^{3^n} \mid n \geq 0\}$

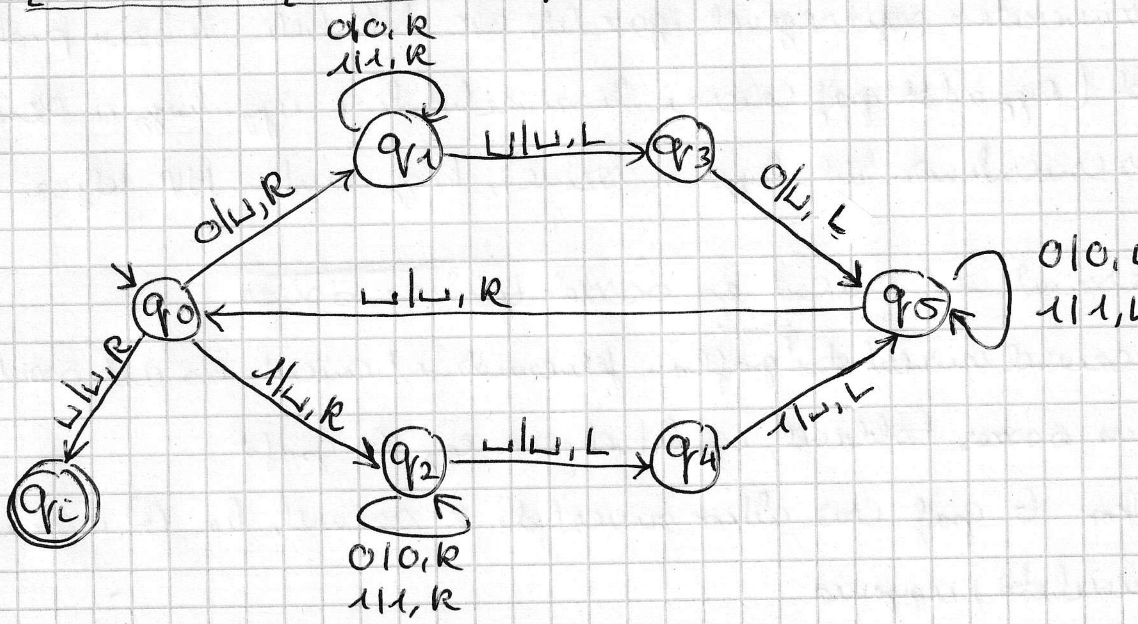
$\Sigma = \{a\}$

$\Gamma = \Sigma \cup \{\sqcup, x\}$

3

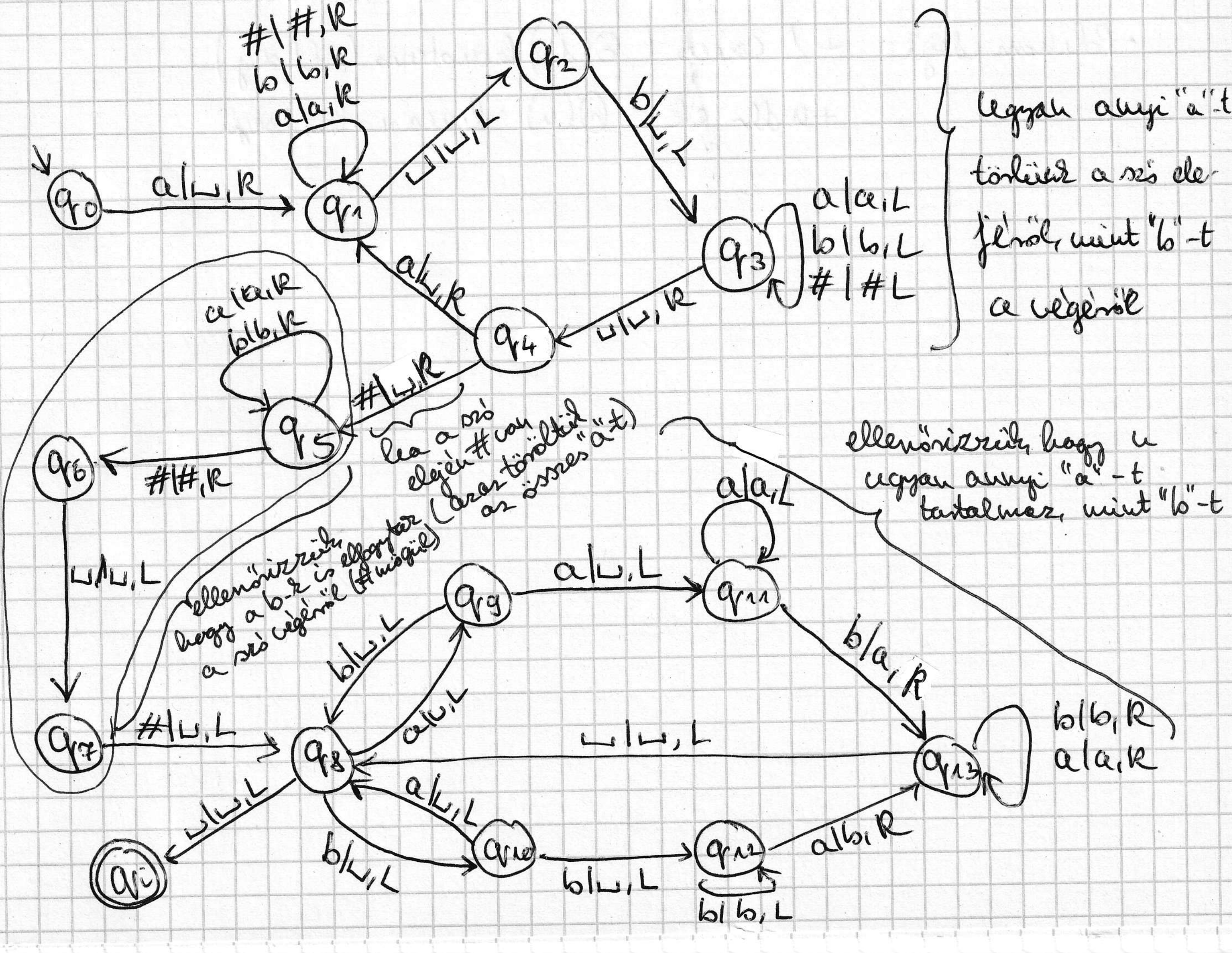
Adj meg olyan egyszalagos Turing-gépeket, amelyek az alábbi nyelvet ismerik fel, és add meg a gépek időigényét is: (18p)

a) $\{uu^{-1} \mid u \in \{0,1\}^*\}$ (palindrom)



Műveletigény: $\Theta(n^2)$

b) $\{a^n \# u \# b^n \mid n \geq 1, u \in \{a,b\}^*, |u|_a = |u|_b\}$



4. Mint ismeretes, a 3 Szín probléma (egy adott gráf csúcsai kiszínezhetők-e 3 színnel úgy, hogy szomszédos csúcsoknak különböző legyen a színe) NP-teljes.
Visszavezetés segítségével igazold az NP-beli 4 Szín problémáról (egy adott gráf csúcsai kiszínezhetők-e úgy, hogy a szomszédos csúcsoknak kül. legyen a színe), hogy szintén NP-teljes! (10p)

- A 3 színnel színezni kívánt (G) gráfban felveszünk +1 csúcsot és azt összekötjük az összes többivel, így létrehozva egy G' gráfot
  - Ekkor G gráf csak akkor színezhető 3 színnel, ha G' 4 színnel színezhető, ugyanis:
    - ↳ az izomorf rész G'-ben kiszínezhető 3 színnel ha G is
    - ↳ ekkor a hozzáadott csúcsot egy 4. színnel kiszínezve G'

      4 színnel színezhető, ha G 3 színnel színezhető
- Polinom idejű: +1 csúcs ($\Theta(1)$: konstans költség)

  +n db él : $\Theta(n)$: lineáris költségű