

A nemdeterminisztikus Turing-gép

A két irányban végtelen szalaggal rendelkező, többszalagos Turing-gép **általánosítása**

- Az átmeneti függvény:

$$\delta: (Q - \{q_i, q_n\}) \times \Gamma^k \rightarrow P(Q \times \Gamma^k \times \{L, R\}^k)$$

- Azaz: $\delta(q, a_1, \dots, a_k) \subseteq Q \times \Gamma^k \times \{L, R\}^k$
- A további komponensek mint az egyszalagos esetben

Konfiguráció: ugyanaz, mint a determinisztikus esetben

Konfiguráció átmenet: a determinisztikus esethez hasonlóan

- **Például**, legyen M egy nemdeterminisztikus **egyszalagos** Turing-gép és C_1, C_2 az M két konfigurációja. C_2 **közvetlenül elérhető** C_1 -ből, jele $C_1 \Rightarrow C_2$, ha az alábbiak egyike teljesül:
 - $C_1 = uqav$, $C_2 = ubpv$ és $(p, b, R) \in \delta(q, a)$, ahol $a, b \in \Gamma, u \in \Gamma^*, v \in \Gamma^+$
 - $C_1 = ucqav$, $C_2 = upcbv$ és $(p, b, L) \in \delta(q, a)$, ahol $a, b, c \in \Gamma, u, v \in \Gamma^*$

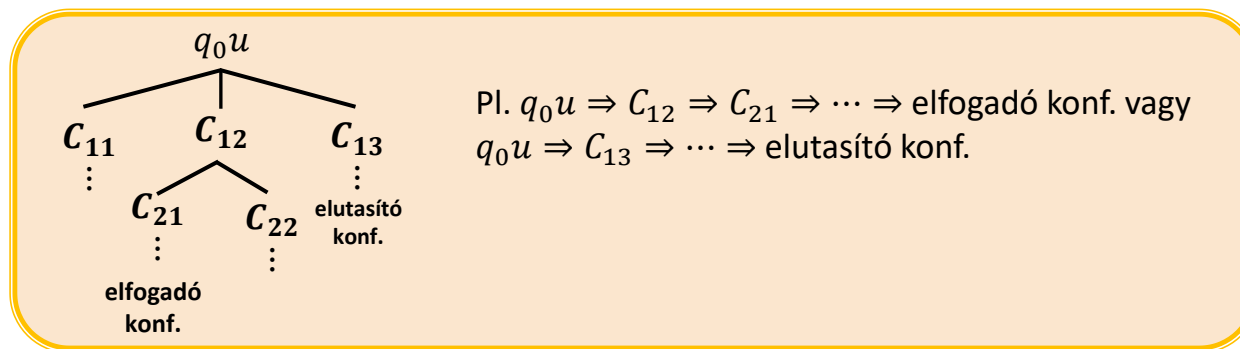
Csak itt különbözik a determinisztikus definíciótól

A nemdeterminisztikus Turing-gép

A **többlépéses konfiguráció átmenet** és a **felismert nyelv** a determinisztikus esethez hasonlóan definiálható

Egy M nemdeterminisztikus Turing-gép összes számítása egy u szón egy ún. **számítási fával** szemléltethető:

- A gyökere az M **kezdőkonfigurációja** az u -n
- A **csúcsai** M konfigurációi
- Két szomszédos csúcs (azaz egy éllel összekötött csúcspár) megfelel M egy lehetséges **konfigurációátmenetének**
- Minden csúcsnak annyi gyermeke van ahány **nemdeterminisztikus választás** létezik az adott konfigurációban



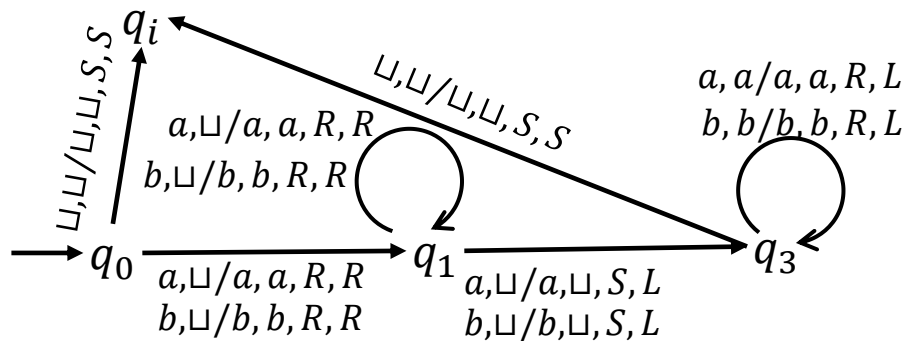
M elfogadja az u -t ha a számítási fa legalább egy levele elfogadó konfiguráció

Nemdeterminisztikus Turing-gép

M eldönti az $L \subseteq \Sigma^*$ nyelvet, ha felismeri és minden $u \in \Sigma^*$ szóra az M számítási fája véges és minden levele elfogadó vagy elutasító konfiguráció

M $f(n)$ időigényű, ha minden $u \in \Sigma^*$ n hosszú szóra a számítási fa legfeljebb $f(n)$ magas

Az M_{pal3} nemdeterminisztikus, kétszalagos Turing-gép:



A felismert nyelv: $L(M_{pal3}) = \{uu^{-1} \mid u \in \{a, b\}^*\}$

Időigény: $O(n)$

M_{pal3} működése:

- Ha üres a bemenet, akkor elfogad, egyébként átmásol 1 betűt a 2-es szalagra és q_1 -be lép
- q_1 -ben **szétosztja** a számítást:
 - Az egyik továbbra is másol q_1 -ben
 - A másik abbahagyja és q_3 -ba lép
- q_3 -ban az első szalagon jobbra, a másodikon balra lépve összehasonlítja a két szalagon lévő szót
- Ha megegyeznek, akkor elfogad, egyébként elutasít

Nemdeterminisztikus Turing-gép

Minden $M = (Q, \Sigma, \Gamma, \delta, q_0, q_i, q_n)$ $f(n)$ idejű nemdeterminisztikus Turing-géphez megadható egy ekvivalens, $2^{O(f(n))}$ idejű M' (det.) Turing-gép

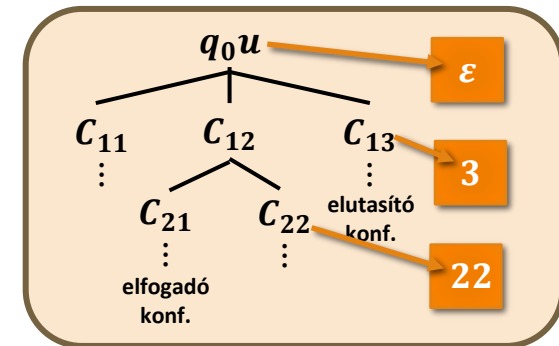
Bizonyítás (vázlat)

Legyen $u \in \Sigma^*$ és tekintsük az M t számítási fáját az u -n

A t minden csúcsához egyértelműen hozzárendelhető egy $\{1, \dots, d\}^*$ -beli szó, ahol d a t kifokainak maximuma

M' egy ciklusban az egyik szalagján lexikografikusan felsorolja a $\{1, \dots, d\}^*$ -beli szavakat egészen addig, amíg nem talál t -ben elfogadó konfigurációt:

- legyen az aktuális szó w
- M' szimulálja a $q_0 u \Rightarrow^* C$ számítást, ahol C a w -vel címkézett konfiguráció t -ben
- Ha C elfogadó, akkor M' is elfogadó állapotba lép
- Egyébként átírja w az azt lexikografikusan követő szóra



Nemdeterminisztikus Turing-gép

Következmény: Ha egy L nyelv eldönthető nemdeterminisztikus Turing-géppel, akkor eldönthető determinisztikussal is

- Az előző szimulációban a determinisztikus Turing-gép megadható úgy, hogy csak $O(f(n))$ hosszú számításokat szimuláljon, és ha ezek között nincs elfogadó, akkor elutasítja a bemenetet

Az a sejtés (még nincsen bebizonyítva), hogy nem lehet a nemdeterminisztikus Turing-gépet az időigény drasztikus romlása nélkül determinisztikus Turing-géppel szimulálni

Turing-gép - bonyolult vagy egyszerű

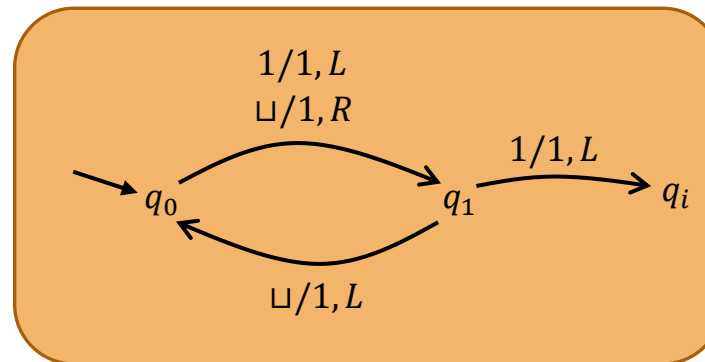
Bár a Turing-gép egy nagyon egyszerű algoritmus modell, már a vele kapcsolatos legegyszerűbb kérdések megválaszolása is kihívás lehet

Ezt demonstrálja a „**Szorgos Hód**” játék:

- Adott $n \geq 2$ számhoz keressük azt az n állapotú Turing-gépet (n állapot plusz egy amiben megáll), ami bizonyíthatóan a legtöbbet lépi, mielőtt megáll
- A gépet üres bemenettel kell indítani
- Szalagszimbólumként csak \sqcup és 1 használható

A 2 állapotú „szorgos hód”

- Vajon mennyit lép és mennyi 1-est ír a szalagra, amíg megáll:

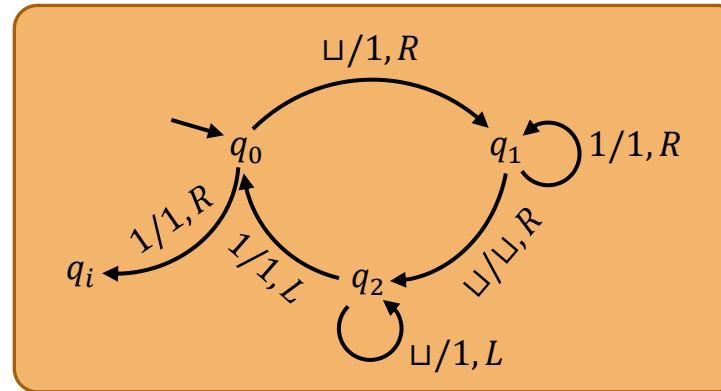


- Válasz: 4 darab 1-es 6 lépés után

Turing-gép - bonyolult vagy egyszerű

A 3 állapotú „szorgos hód”

- Vajon mennyit lép és mennyi 1-est ír a szalagra, amíg megáll:



- Válasz: 6 darab 1-es 14 lépés után

A 4 állapotú „Szorgos Hód”

13 1-est ír a szalagra 107 lépés után

A legjobb 5 állapotú jelölt:

4098 1-est ír a szalagra 47.176.870 lépés után

A legjobb 6 állapotú jelölt:

$\approx 3.5 \cdot 10^{18267}$ 1-est ír a szalagra $\approx 7.4 \cdot 10^{36534}$ lépés után

Eldöntési probléma mint formális nyelv

Jelölés: Tetszőleges D objektumra $\langle D \rangle$ jelöli a D **egy tömör elkódolását** egy megfelelő szóban (D lehet formula, gráf, Turing-gép – lásd később, vagy bármilyen végesen reprezentálható dolog)

Tömör elkódolás: például ha egy számot **binárisan** adunk meg (unárisan nem tömör)

Egy **P eldöntési problémának** megfeleltethető egy

$$L_P := \{ \langle I \rangle \mid I \text{ a } P \text{ pozitív bemenete} \} \text{ formális nyelv}$$

Például

A **SAT problémának** megfeleltethető a

$\{ \langle \varphi \rangle \mid \varphi \text{ egy kielégíthető ítéletkalkulusbeli konjunktív nf.} \}$ nyelv

$L_u = \{ \langle M, w \rangle \mid w \in L(M) \}$ (M egy Turing-gép w pedig ennek egy bemenete) megfelel azon problémának, ahol a feladat annak eldöntése, hogy egy M TG elfogadja-e a w bemenetet

- Ezt szokás **univerzális nyelvnek** hívni

$L_{\text{átló}} = \{ \langle M \rangle \mid \langle M \rangle \notin L(M) \}$ (M egy Turing-gép) megfelel azon problémának, ahol a feladat annak eldöntése, hogy egy M Turing-gép elfogadja-e saját maga kódját bemenetként

SAT probléma:

- Input: konjunktív normálformájú (ítéletkalkulus-beli) formula.
- Output: kielégíthető-e?

A Turing-gépek elkódolásáról

Megjegyzés: Ebben a részben csak $\{0,1\}$ bemenő ábécével rendelkező Turing-gépeket vizsgálunk

Az $\langle M, w \rangle$ és $\langle M \rangle$ definíciója

Legyen $M = (Q, \{0,1\}, \Gamma, \delta, q_0, q_i, q_n)$ egy Turing-gép

- Q és Γ minden eleméhez rendelhetünk egy sorszámot
- a fej L és R irányait tekinthetjük rendre az 1-es és 2-es irányoknak

M minden $\delta(q, a) = (p, b, D)$ átmenete **egyértelműen elkódolható** a következő szóval:

$$\underbrace{0 \dots 0}_q 1 \underbrace{0 \dots 0}_a 1 0 \dots 0 1 0 \dots 0 1 0 \dots 0 \underbrace{0 \dots 0}_{\text{az irány sorszámába db 0}}$$

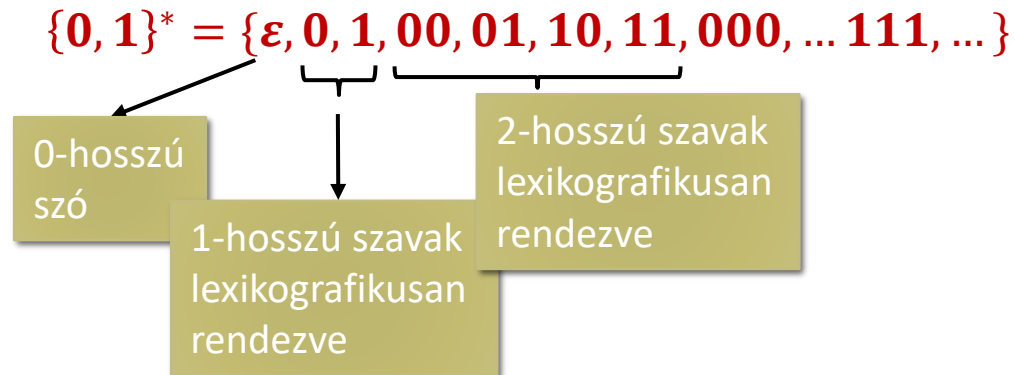
 q sorszáma db 0 a sorszáma db 0 ... az irány sorszáma db 0

$\langle M \rangle$ kódja: $\langle \delta_1 \rangle 11 \langle \delta_2 \rangle 11 \dots 11 \langle \delta_l \rangle$, ahol $\delta_1, \dots, \delta_l$ az M összes átmenete, $\langle \delta_i \rangle$ az i -ik átmenet kódja

$\langle M, w \rangle$ kódja: $\langle M \rangle 111w$

A Turing-gépek elkódolásáról

A $\{0,1\}^*$ elemei felsorolhatóak, pl. lexikografikusan:



Jelölés: Minden $i \geq 1$ -re,

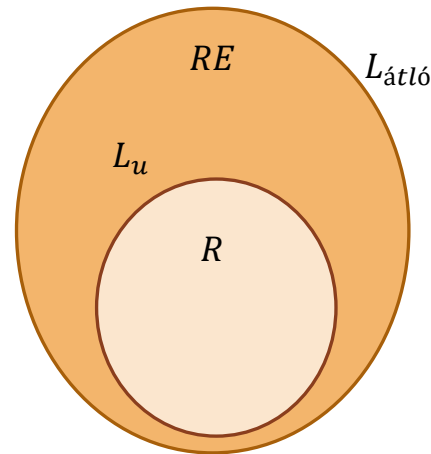
- w_i jelöli a $\{0,1\}^*$ halmaz i -ik elemét
- M_i jelöli a w_i által kódolt Turing-gépet (ha w_i nem kódol Turing-gépet, akkor M_i egy olyan tetszőleges Turing-gép, ami nem fogad el semmit)

Az RE szerkezete

Emlékeztető:

- R az eldönthető, RE pedig a Turing-felismerhető nyelvek osztálya
- $R \subseteq RE$

A célunk megmutatni, hogy :



$$L_u = \{\langle M, w \rangle \mid w \in L(M)\}$$

$$L_{\text{átló}} = \{\langle M \rangle \mid \langle M \rangle \notin L(M)\}$$

Eldönthetetlen problémák

$L_{\text{átló}}$ nem ismerhető fel Turing-géppel

Bizonyítás

Tekintsük azt a T táblázatot, mely i -ik sorának j -ik oszlopa ($i, j \geq 1$), azaz $T(i, j)$, akkor és csak akkor 1, ha $w_j \in L(M_i)$

Legyen d a T **átlójában** olvasható végtelen hosszú bitsztring, és legyen \bar{d} a d bitenkénti **komplementere**

Ekkor igazak az alábbi megjegyzések:

- Minden $i \geq 1$ -re, T i -ik sora az $L(M_i)$ nyelv karakterisztikus függvénye
- \bar{d} az $L_{\text{átló}}$ karakterisztikus függvénye
- Minden Turing-géppel felismerhető, azaz RE -beli nyelv karakterisztikus függvénye megegyezik T valamelyik sorával
- \bar{d} különbözik T minden sorától

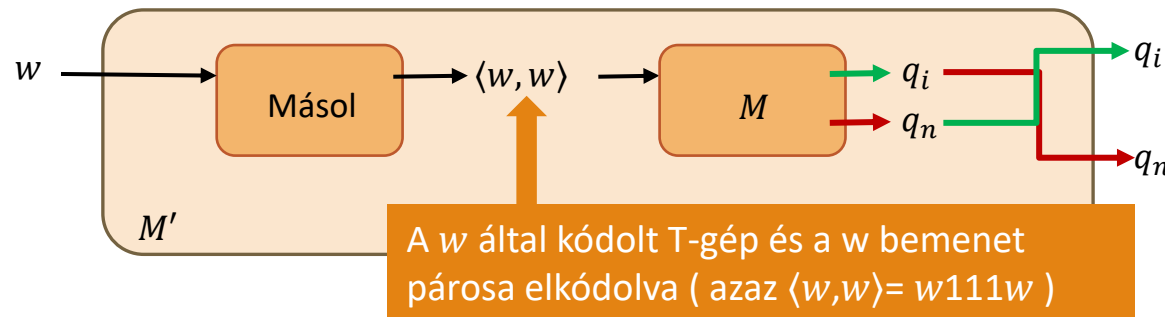
Ezek alapján $L_{\text{átló}}$ különbözik az összes RE -beli nyelvtől

Eldönthetetlen problémák

Az L_u nyelv eldönthetetlen

Bizonyítás

- Indirekt módon tfh. $L_u \in R$ és legyen M egy L_u -t eldöntő Turing-gép
- Konstruáljuk meg M' -t:



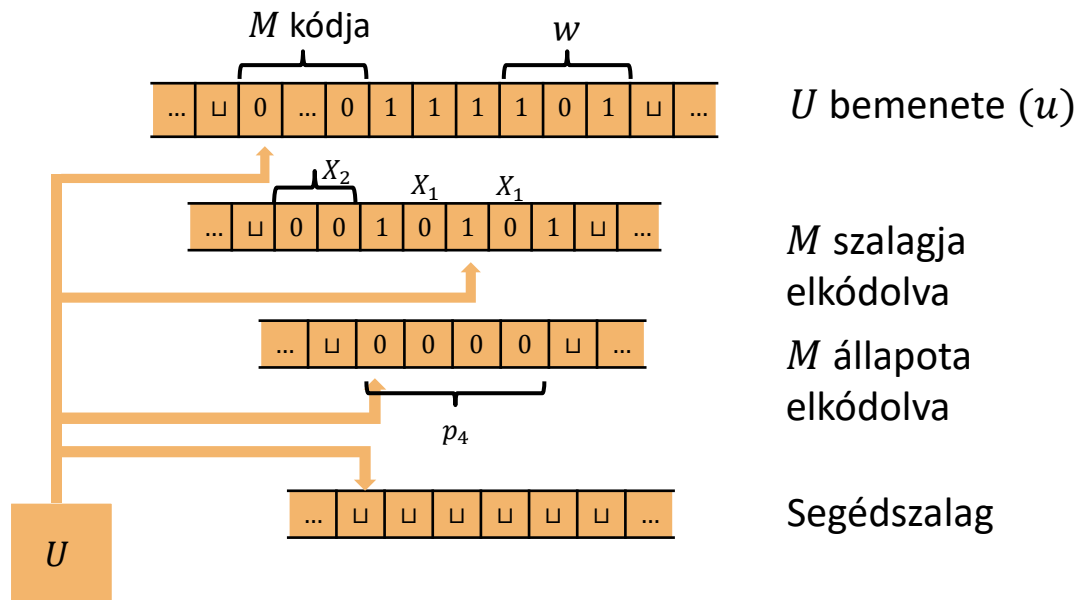
- Ekkor M' elfogadja w -t $\Leftrightarrow M$ nem fogadja el a $\langle w, w \rangle$ szót \Leftrightarrow A w által kódolt Turing-gép nem fogadja el w -t
- Azaz $w \in L(M') \Leftrightarrow w \in L_{\text{átló}}$
- Tehát $L(M') = L_{\text{átló}}$, ami ellentmondás, mert $L_{\text{átló}}$ -t nem lehet felismerni
- Következik, hogy L_u -t nem lehet eldönteni

Eldönthetetlen problémák

L_u felismerhető Turing-géppel

Bizonyítás

L_u -t egy ún. **Univerzális Turing-gép** ismeri fel:



U működése vázlatosan:

- Létrehozza M kezdőkonfigurációját elkódolva a 2-es (M szalagja) és 3-as (M állapota) szalagon
- Szimulálja M egy lépését:
 - Leolvassa a második szalagról M aktuálisan olvasott szalagszimbólumát
 - Megnézi, hogy a bemenetén szereplő u szó kódol-e Turing-gépet; ha nem elutasítja a bemenetet
 - Leolvassa a harmadik szalagról M aktuális állapotát
 - Szimulálja M egy lépését (M ott van az 1-es szalagon)
- Ha M aktuális állapota elfogadó vagy elutasító, akkor U is belép a saját elfogadó vagy elutasító állapotába

Megjegyzés: ha M nem áll meg w -n, akkor U sem áll meg $\langle M, w \rangle$ -n

Egy eldöntési probléma komplementere

Jelölés:

- Ha $L \subseteq \Sigma^*$, akkor $\bar{L} := \{u \in \Sigma^* \mid u \notin L\}$ az L **komplementere**
- Tetszőleges C problémaosztályra **coC** jelöli a C -beli problémák komplementereinek az osztályát

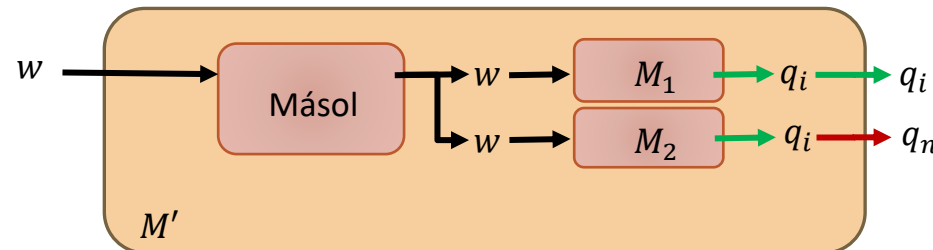
Tehát egy P eldöntési probléma komplementere az a \bar{P} probléma aminek pozitív bemenetei a P negatív bemenetei, negatív bemenetei pedig az P pozitív bemenetei

Ha L és \bar{L} is felismerhető Turing-géppel, akkor L eldönthető

Bizonyítás

Legyen M_1 és M_2 rendre az L -t és \bar{L} -t felismerő Turing-gép

Konstruáljuk meg az M' kétszalagos Turing-gépet:



M' lemásolja w -t a második szalagjára, majd **felváltva szimulálja M_1 és M_2 egy-egy lépését** addig, amíg valamelyik elfogadó állapotba lép

Így M' az L -et ismeri fel, és **minden bemeneten meg is áll**, azaz M el is dönti L -et

RE nem zárt a komplementerképzésre

Következmény: $RE \neq coRE$

Bizonyítás

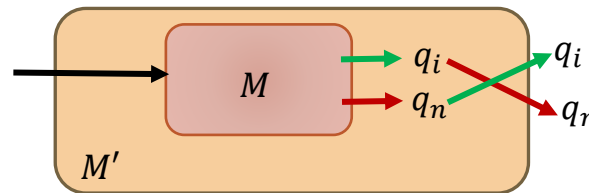
- Tudjuk, hogy $L_u \in RE$, ezért elég belátni, hogy $\overline{L_u} \notin RE$
- Indirekt módon tegyük fel, hogy $\overline{L_u} \in RE$
- Akkor a korábbi állításunk alapján $L_u \in R$
- De azt tanultuk, hogy $L_u \notin R$, ellentmondás, tehát $\overline{L_u} \notin RE$

$$R = coR$$

Bizonyítás

Legyen $L \in R$ és M egy Turing-gép, ami az L -t dönti el_w

Akkor a következő M' \overline{L} -t dönti el:



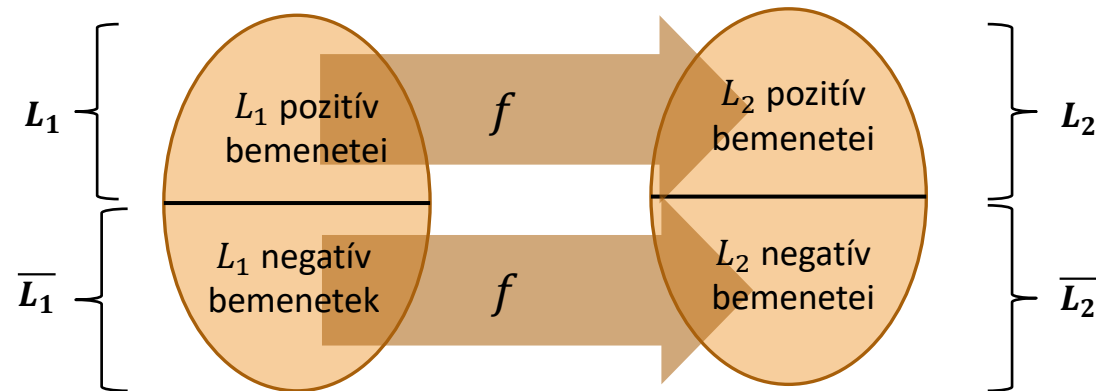
- Hasonlóan igaz, hogy $P = coP$
- Ugyanakkor az a sejtés, hogy $NP \neq coNP$

Visszavezetések

Egy $f: \Sigma^* \rightarrow \Delta^*$ függvény **kiszámítható** ha van olyan Turing-gép, ami egy $u \in \Sigma^*$ szóval az első szalagján elindítva úgy áll meg, hogy az utolsó szalagján az $f(u)$ szó van

Egy $L_1 \subseteq \Sigma^*$ nyelv visszavezethető egy $L_2 \subseteq \Delta^*$ nyelvre (jele: $L_1 \leq L_2$), ha van olyan $f: \Sigma^* \rightarrow \Delta^*$ kiszámítható függvény, hogy minden $u \in \Sigma^*$ szóra,

ha $u \in L_1$, akkor $f(u) \in L_2$ és ha $u \in \bar{L}_1$, akkor $f(u) \in \bar{L}_2$



Mivel tetszőleges L nyelvre $\bar{\bar{L}} = L$, az L_1 és \bar{L}_1 illetve L_2 és \bar{L}_2 szerepének felcserélésével kapjuk, hogy $\bar{L}_1 \leq \bar{L}_2$ is teljesül

Visszavezetések

Legyen L_1 és L_2 két nyelv, és tegyük fel, hogy $L_1 \leq L_2$

Ha $L_2 \in RE$ akkor $L_1 \in RE$, továbbá ha L_2 eldönthető, akkor L_1 is az

Következmény:

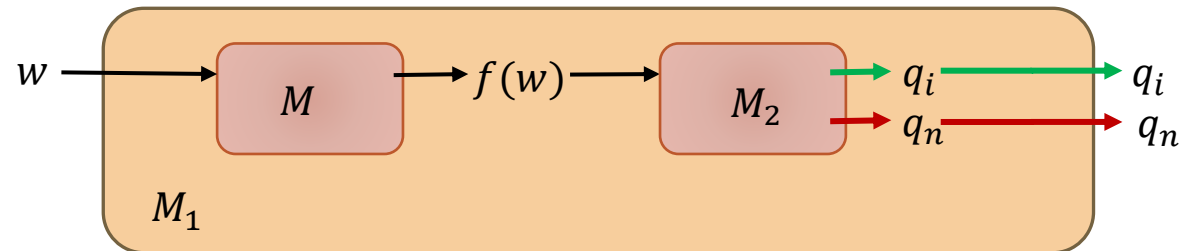
Ha L_1 eldönthetetlen, akkor L_2 is az

Bizonyítás

Legyen $L_2 \in RE$

Legyen M_2 az L_2 -t felismerő, M pedig a visszavezetést kiszámító Turing-gép

Konstruáljuk meg M_1 -et:



Világos, hogy M_1 az L_1 -et ismeri fel, azaz $L_1 \in RE$

Ha $L_2 \in R$, akkor M_2 választható olyannak, hogy minden bemeneten megáll

Akkor M_1 is minden bemeneten megáll, tehát **eldönti** L_1 -et

A Turing-gépek megállási problémája

Az alábbi nyelvet a **Turing-gépek megállási problémájának** nevezzük:

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

Eldöntési problémaként megfogalmazva:

- Adott egy M Turing-gép és M egy w bemenete
- **Kérdés:** Megáll-e M a w -n?

Fontos: egy Turing-gép csak a q_i vagy q_n állapotok valamelyikében tud megállni

L_h eldönthetetlen

Bizonyítás

Korábbi tétel alapján elég megmutatni, hogy $L_u \leq L_h$

Tetszőleges M Turing-gépre, legyen M' az alábbi Turing-gép

M' tetszőleges u bemeneten a következőket teszi:

1. Futtatja M -et u -n (ahogyan azt az Univerzális Turing-gép is tesz)
2. Ha M q_i -be lép, akkor M' is q_i -be lép
3. Ha M q_n -be lép, akkor M' egy **végtelen ciklusba** lép

A megállási probléma

Bizonyítás (folyt.)

- Belátható, hogy
 1. Az $f: \langle M \rangle \rightarrow \langle M' \rangle$ leképezés egy kiszámítható függvény
 2. Tetszőleges M, w Turing-gép — bemenet párosra:
$$\langle M, w \rangle \in L_u \Leftrightarrow M \text{ megáll } w\text{-n } q_i\text{-ben} \Leftrightarrow \text{Az } M' \text{ megáll } w\text{-n} \Leftrightarrow \langle M', w \rangle \in L_h$$
- Tehát az M' konstrukciója az L_u visszavezetése L_h -ra
- Mivel L_u eldönthetetlen, következik, hogy L_h is az

Visszavezetések megadásakor jellemzően csak azon szavakat vizsgáljuk, amelyek kódolnak valamilyen objektumot

Pl. a fenti esetben nem foglalkoztunk azzal, hogy f mit rendeljen egy olyan szóhoz, ami nem kódol Turing-gépet (ami egyébként egy könnyen kezelhető eset)