Bevezetés a bonyolultságelméletbe

Dr. Ésik Zoltán

SZTE,

Informatikai Tanszékcsoport

A kiszámíthatóság elméletének kialakulása I.

- **1900: Hilbert 10. problémája**
  Adott $f(x_1, \ldots, x_n) = g(x_1, \ldots x_n)$ diofantikus egyenlet, ahol f és g egész együtthatós polinomok, létezik-e (egész értékű) megoldása.
- **1971: Matijasevič**
  Hilbert 10. problémája algoritmikusan megoldhatatlan
- **1928: Hilbert**
  Találjunk olyan algoritmust, amellyel a predikátumkalkulus (függvénykalkulus) tetszőleges kijelentéséről eldönthetjük, hogy érvénye-e.
- **1930-as évek közepe: Church, Turing**
  Ilyen algoritmus nem létezik.

A kiszámíthatóság elméletének kialakulása II.

- **1931: Gödel**
  primitív rekurzív függvények
- **1934: Gödel**
  általános rekurzív függvények
- **1930-as évek eleje: Church, Kleene, Rosser (Princeton)**
  λ-definiálhatóság
- **1935, AMS New-York-i összejövetele**
  Church tézise: Az általános rekurzív függvények (matematikai fogalom) megfelelnek az algoritmikusan kiszámítható függvény fogalmának (intuitív fogalom).
- **1936: Kleene**
  Az általános rekurzív függvények megegyeznek a λ-definiálható függvényekkel.

A kiszámíthatóság elméletének kialakulása III.

- **1936: Turing**
  Bebizonyítja, hogy a Turing-gép ekvivalens a λ-definiálhatósággal, és megfogalmazza azt a tézist, hogy a **Turing-gép megfelel az algoritmussal kiszámítható függvényeknek.**

  A **Church-Turing tézist** tapasztalati tények és matematikai eredmények támasztják alá:
- Minden intuitív értelemben megoldható problémáról sikerült kimutatni, hogy azok megoldhatók a matematikai modellekben is.
- A matematikai modellek ekvivalensek.

Kiszámíthatóság és bonyolultság

- **Kiszámíthatóság**: Melyek az algoritmikusan megoldható problémák?
- **Bonyolultságelmélet**: Melyek a gyakorlatilag megoldható problémák, erőforrás igény.
- **1971: Cook**
  P és **NP** osztályok, **NP**-teljesség, SAT**NP**-teljes
- **1972: Karp**
  Rámutat az **NP**-teljes problémák változatosságára.
- **1973: Levin**
  Több kombinatorikus probléma „univerzális a kimerítő keresésre”

További bonyolultsági osztályok

(és számítási módok)

- **L**, **NL**, **PSPACE**, **EXP**, . . .
- Valószínűségi modellek
- Párhuzamos számítás, stb.

Polinom időben megoldható problémák

ELÉRHETŐSÉG:

Adott: Egy $G = (V, E)$ irányított gráf, feltehetjük hogy $V = \{1, 2, \ldots, n\}$.

Kérdés: Létezik-e 1-ből n-be vezető út?

Módszer:

- $S := \{1\}$ és "megjelöljük" az 1 csúcsot.
- Amíg S ki nem ürül:

- Választunk egy $i \in S$ csúcsot, $S := S - \{i\}$.

- $j = 1, \ldots, n$:
  $(i, j) \in E$, j nem megjelölt => $S := S \cup \{j\}$ és megjelöljük j-t!

- "Igen" választ adunk, ha n megjelölt , különben "nem" választ.

Néhány kimaradt részlet

- bemenet megadása – pld. szomszédsági mátrix
  (Függ a modelltől, de lényeges szerepet nem játszik.)
- S reprezentálása
  sor: szélességi keresés
  verem: egyfajta mélységi keresés

Hatékonyság

- A mátrix minden elemét egyszer használjuk fel.
- Amennyiben az egyszerű műveletek (S egy elemének kiválasztása, megjelölése, stb.) konstans időigényűek: $\mathcal{O}(n^2)$.

Az ELÉRHETŐSÉG **eldöntési probléma.**

MAXIMÁLIS FOLYAM

Adott: (V, E, s, t, c) **hálózat,** ahol

- (V, E) irányított gráf
- $s \in V$ **forrás,** $t \in V$ **nyelő**
  (ahol t elérhető s-ből)
- $(i, j) \in E \Rightarrow c(i, j) \in \mathbb{N}_+$ **kapacitás**

Feltesszük, hogy nem léteznek ellentétes élpárok és hurokélek, továbbá s-be nem vezet és t-ből nem indul él.

Kérdés: Mekkora a legnagyobb értékű folyam?

A folyamok definíciója

Folyam: $f : E \to \mathbb{N}$

- $(i,j) \in E \Rightarrow 0 \leq f(i,j) \leq c(i,j)$
- $$\sum_{(i,j)\in E} f(i,j) = \sum_{(j,k)\in E} f(j,k) \quad \forall j \in V \setminus \{s,t\}$$

Az f folyam **értéke**:

$$\sum_{(s,k)\in E} f(s,k)$$

Áll. Akkor és csak akkor létezik az f folyamnál nagyobb értékű f' folyam, ha az alábbi $N(f)$ hálózatban t elérhető s-ből.

$$
\begin{aligned}
N(f) &= (V, E', s, t, c') \\
E' &= (E \setminus \{(i,j) \ : \ f(i,j) = c(i,j)\} \cup \\
&\quad \{(i,j) : (j,i) \in E \quad f(j,i) > 0\}) \\
c'(i,j) &= \begin{cases} c(i,j) - f(i,j) & \text{ha } (i,j) \in E \cap E' \\ f(j,i) & \text{különben.} \end{cases}
\end{aligned}
$$

($N(f)$ tartalmazhat ellentétes élpárokat.)

Tegyük fel, hogy $N(f)$-ben adott egy s-ből t-be vezető út. Legyen d az úton előforduló minimális élkapacitás.

Minden (i,j) élre legyen:

$$f'(i,j) = \begin{cases} f(i,j) + d & \text{ha } (i,j) \in E \cap E' \text{ előfordul az úton} \\ f(i,j) - d & \text{ha } (j,i) \notin E \text{ előfordul az úton} \\ f(i,j) & \text{különben.} \end{cases}$$

Ekkor f' az f-nél nagyobb értékű folyam.

Módszer

- Kezdetben f az azonosan nulla folyam.
- Adott f esetén készítsük el $N(f)$-et.
- Ha $N(f)$-ben t nem érhető el s-ből, akkor f maximális.
- Különben egy s-ből t-be vezető úthoz és annak minimális kapacitású éléhez készítsük el az f' folyamot, majd az $f := f'$ folyammal ismételjük meg az eljárást.

Hatékonyság:

- $C := \max\{c(i,j) \mid (i,j) \in E\}$
- $\mathcal{O}(nC)$ lépés,lépésenként $\mathcal{O}(n^2)$ idő: $\mathcal{O}(n^3C)$
  NEM POLINOMIÁLIS. (C miatt.)
- Az útkeresés szélességi változatával (legrövidebb út): $\mathcal{O}(n^5)$

A MAXIMÁLIS FOLYAM optimalizálási probléma.
Eldöntési változat MAXIMÁLIS FOLYAM (E).

Az utazóügynök probléma

TSP

Adott: Az $1, \ldots, n$ városok és bármely két i, j város távolsága: $d_{i,j}$.

Kérdés: Találjunk az összes várost pontosan egyszer érintő legrövidebb körutat.

Eldöntési változat: TSP **(E)**

Triviális módszer: Az összes $\frac{1}{2}(n-1)!$ lehetséges körút megvizsgálásával.

Nem ismert, hogy TSP megoldható-e polinom idejű algoritmussal. (NP-teljes probléma.)

A **P** osztály

Def. Legyenek $f, g : \mathbb{N} \to \mathbb{N}$ függvények.

- $f(n) = \mathcal{O}(g(n))$, ha létezik olyan $c > 0$, hogy $f(n) \leq c\, g(n)$ majdnem minden n-re.
- $g(n) = \Omega(f(n))$, ha $f(n) = \mathcal{O}(g(n))$
- $f(n) = \Theta(g(n))$, ha $f(n) = \mathcal{O}(g(n))$ és $g(n) = \mathcal{O}(f(n))$.

Példa.

$$\begin{aligned} p(n) &= a_0 n^k + a_1 n^{k-1} + \ldots + a_k \\ q(n) &= b_0 n^l + b_1 n^{l-1} + \ldots + b_l \end{aligned}$$

ahol $a_i, b_j \in \mathbb{N}$, $a_0, b_0 > 0$ Ekkor:

- $p(n) = \mathcal{O}(q(n)) \Leftrightarrow k \leq l$,
- $p(n) = \Theta(q(n)) \Leftrightarrow k = l$.

Legyen $a \in \mathbb{N}$, $a > 1$. Ekkor

- $p(n) = \mathcal{O}(a^n)$, de $a^n \neq \mathcal{O}(p(n))$.

Def. Egy probléma a **P** osztályba esik, ha a probléma megoldható polinom időigényű (vagy $\mathcal{O}(n^k)$ időigényű) algoritmussal.

A polinomiális tézis

- Egy algoritmust akkor ad gyakorlatilag kielégítő megoldást egy problémára, ha időigénye polinomiális.
- Egy problémát akkor tekintünk gyakorlatilag megoldhatónak, ha a **P** osztályba esik.

- **Néhány ellenvetés**
- Elfogadható-e a gyakorlatban egy $\mathcal{O}(n^{100})$ időigényű algoritmus?
- A konstansok nagysága.
- Kisméretű adatokon egy exponenciális algoritmus is jobban viselkedhet, mint egy polinomiális (de csak véges sok adaton).
- Legrosszabb eset – várható viselkedés.
- Hatékony párhuzamosítás.
- **Ugyanakkor**
- A gyakorlatban előforduló **P**-beli problémák rendje kicsi.
- A **P** osztály robosztus, és elegáns matematikai elmélethez vezet.
- Fontos információkat szolgáltat a hatékonyan megoldható problémák szerkezetéről.

Turing-gépek definíciója

Def. Turing-gép: $M = (K, \Sigma, \delta, s)$

- K : állapotok véges, nemüres halmaza
- Σ : betűk (szimbólumok) véges, nemüres halmaza
- δ: átmenetfüggvény
  $K \times \Sigma \to (K \cup \{h, \text{„igen”}, \text{„nem”}\}) \times \Sigma \times \{\leftarrow, -, \rightarrow\}$
- s: kezdőállapot

$\triangleright, \sqcup \in \Sigma, \quad \Sigma - \{\triangleright, \sqcup\} \neq \emptyset$

$\delta(q, \triangleright) = (p, \rho, D) \Rightarrow \rho = \triangleright, D = \rightarrow$ (legalább is $D \neq \leftarrow$)

$p \in K$	$\sigma \in \Sigma$	$\delta(p,\sigma)$		
s	0	$(s, 0, \rightarrow)$	s	$\underline{\triangleright}010$
s	1	$(s, 1, \rightarrow)$	s	$\triangleright\underline{0}10$
s	$\sqcup$	$(q, \sqcup, \leftarrow)$	s	$\triangleright0\underline{1}0$
s	$\triangleright$	$(s, \triangleright, \rightarrow)$	s	$\triangleright01\underline{0}$
q	0	$(q_0, \sqcup, \rightarrow)$	s	$\triangleright010\underline{\sqcup}$
q	1	$(q_1, \sqcup, \rightarrow)$	q	$\triangleright01\underline{0}\sqcup$
q	$\sqcup$	$(q, \sqcup, -)$	q_0	$\triangleright01\sqcup\underline{\sqcup}$
q	$\triangleright$	$(h, \triangleright, \rightarrow)$	s	$\triangleright01\underline{\sqcup}0$
q_0	0	$(s, 0, \leftarrow)$	q	$\triangleright0\underline{1}\sqcup 0$
q_0	1	$(s, 0, \leftarrow)$	q_1	$\triangleright0\sqcup\underline{\sqcup}0$
q_0	$\sqcup$	$(s, 0, \leftarrow)$	s	$\triangleright0\underline{\sqcup}10$
q_0	$\triangleright$	$(h, \triangleright, \rightarrow)$	q	$\triangleright\underline{0}\sqcup 10$
q_1	0	$(s, 1, \leftarrow)$	q_0	$\triangleright\sqcup\underline{\sqcup}10$
q_1	1	$(s, 1, \leftarrow)$	s	$\triangleright\underline{\sqcup}010$
q_1	$\sqcup$	$(s, 1, \leftarrow)$	q	$\underline{\triangleright}\sqcup 010$
q_1	$\triangleright$	$(h, \triangleright, \rightarrow)$	h	$\triangleright\underline{\sqcup}010$

Konfigurációk

Def. A számítási folyamat adott pillanatának teljes leírását **konfigurációnak** nevezzük.

Formálisan:

$$(q, w, u)$$

- $q \in K \cup \{h, \text{„igen”}, \text{„nem”}\}$,
- $w \in \Sigma^+$, a mutatótól balra eső szó, beleértve azt a betűt, amelyet a mutató kijelöl,
- $u \in \Sigma^*$, a mutatótól jobbra lévő, esetleg üres szó.

Átmenetreláció

Def. $(q, w\sigma, u) \xrightarrow{M} (q', w', u')$

- $q \notin \{h, \text{„igen”}, \text{„nem”}\}, \delta(q, \sigma) = (q', \rho, D)$, és
- $D = \rightarrow$ esetén:

$$w' = \begin{cases} w\rho\tau & \text{ha } u = \tau u_1 \\ w\rho\sqcup & \text{ha } u = \varepsilon \end{cases} \qquad u' = \begin{cases} u_1 & \text{ha } u = \tau u_1 \\ \varepsilon & \text{ha } u = \varepsilon \end{cases}$$

- $D = -$ esetén: $w' = w\rho$ és $u' = u$
- $D = \leftarrow$ esetén: $w' = w$ és $u' = \rho u$

Def. $(q, w, u) \xrightarrow{M^k} (q', w', u')$

$(q, w, u) \xrightarrow{M^*} (q', w', u')$

A Turing-gép által kiszámított függvény

Def. Legyen $x \in (\Sigma \setminus \{\triangleright, \sqcup\})^*$. **M(x)** $=$

„igen” ha $(s, \triangleright, x) \xrightarrow{M^*} (\text{„igen”}, w, u)$ valamely w, u szavakra;

„nem” ha $(s, \triangleright, x) \xrightarrow{M^*} (\text{„nem”}, w, u)$ valamely w, u szavakra;

y ha $(s, \triangleright, x) \xrightarrow{M^*} (h, w, u)$ valamely w, u szavakra úgy, hogy $wu = \triangleright y \sqcup^k$ alakú és $y = \varepsilon$ vagy y utolsó betűje nem $\sqcup$;

$\nearrow$ különben, azaz ha M nem áll meg az $(s, \triangleright, x)$ kezdőkonfigurációból indítva.

Bináris növelés

$p \in K$	$\sigma \in \Sigma$	$\delta(p,\sigma)$
s	0	$(s,0,\rightarrow)$
s	1	$(s,1,\rightarrow)$
s	$\sqcup$	$(q,\sqcup,\leftarrow)$
s	$\triangleright$	$(s,\triangleright,\rightarrow)$
q	0	$(h,1,-)$
q	1	$(q,0,\leftarrow)$
q	$\triangleright$	$(h,\triangleright,\rightarrow)$

$(s,\triangleright,11011)$
$(s,\triangleright 1,1011)$
$(s,\triangleright 11,011)$
$(s,\triangleright 110,11)$
$(s,\triangleright 1101,1)$
$(s,\triangleright 11011,\varepsilon)$
$(s,\triangleright 11011\sqcup,\varepsilon)$
$(q,\triangleright 11011,\sqcup)$
$(q,\triangleright 1101,0\sqcup)$
$(q,\triangleright 110,00\sqcup)$
$(h,\triangleright 111,00\sqcup)$

$(s,\triangleright,111)$
$(s,\triangleright 1,11)$
$(s,\triangleright 11,1)$
$(s,\triangleright 111,\varepsilon)$
$(s,\triangleright 111\sqcup,\varepsilon)$
$(q,\triangleright 111,\sqcup)$
$(q,\triangleright 11,0\sqcup)$
$(q,\triangleright 1,00\sqcup)$
$(q,\triangleright,000\sqcup)$
$(h,\triangleright 0,00\sqcup)$

A palindromák felismerése

$p \in K$	$\sigma \in \Sigma$	$\delta(p,\sigma)$
s	0	$(q_0, \triangleright, \rightarrow)$
s	1	$(q_1, \triangleright, \rightarrow)$
s	$\triangleright$	$(s, \triangleright, \rightarrow)$
s	$\sqcup$	(„igen”, $\sqcup, -$)
q_0	0	$(q_0, 0, \rightarrow)$
q_0	1	$(q_0, 1, \rightarrow)$
q_0	$\sqcup$	$(q_0', \sqcup, \leftarrow)$
q_1	0	$(q_1, 0, \rightarrow)$
q_1	1	$(q_1, 1, \rightarrow)$
q_1	$\sqcup$	$(q_1', \sqcup, \leftarrow)$

$p \in K$	$\sigma \in \Sigma$	$\delta(p,\sigma)$
q_0'	0	$(q, \sqcup, \leftarrow)$
q_0'	1	(„nem”, $1, -$)
q_0'	$\triangleright$	(„igen”, $\triangleright, \rightarrow$)
q_1'	0	(„nem”, $0, -$)
q_1'	1	$(q, \sqcup, \leftarrow)$
q_1'	$\triangleright$	(„igen”, $\triangleright, \rightarrow$)
q	0	$(q, 0, \leftarrow)$
q	1	$(q, 1, \leftarrow)$
q	$\triangleright$	$(s, \triangleright, \rightarrow)$

Turing-gépek mint algoritmusok

Nyelvek eldöntése Szófüggvények kiszámítása

Def. Azt mondjuk, hogy az M Turing-gép **eldönti** az $L \subseteq (\Sigma - \{\triangleright, \sqcup\})^*$ nyelvet, ha bármely $x \in (\Sigma - \{\triangleright, \sqcup\})^*$ szóra:

$x \in L \Rightarrow M(x) =$ „igen”

$x \notin L \Rightarrow M(x) =$ „nem”

Def. Azt mondjuk, hogy M **felismeri** az $L \subseteq (\Sigma - \{\triangleright, \sqcup\})^*$ nyelvet, ha bármely $x \in (\Sigma - \{\triangleright, \sqcup\})^*$ szóra:

$x \in L \Rightarrow M(x) =$ „igen”

$x \notin L \Rightarrow M(x) = \nearrow$

$[x \notin L \Rightarrow M(x) \neq$ „igen”]

Def. Egy $L \subseteq (\Sigma - \{\triangleright, \sqcup\})^*$ nyelvet **rekurzívnak** vagy **eldönthetőnek** nevezünk, ha létezik olyan M Turing-gép, mely eldönti az L nyelvet.

Rekurzívan felsorolhatónak nevezzük az L nyelvet, ha létezik olyan Turing-gép, amely felismeri L-et.

Áll. Minden rekurzív nyelv rekurzívan felsorolható.

(A fordított irányú implikáció nem igaz.)

Def. Legyen $f : (\Sigma - \{\triangleright, \sqcup\})^* \to \Sigma^*$. Azt mondjuk, hogy f **rekurzív függvény**, ha létezik olyan M Turing-gép, hogy bármely $x \in (\Sigma - \{\triangleright, \sqcup\})^*$ szóra

$$M(x) = f(x).$$

Példák eldönthető nyelvekre

- palindromák a $\{0,1\}$ halmaz felett.
- $\{0^n c 0^m c 0^{n+m} \ : \ n,m \geq 0\}$
- $\{0^n c 0^m c 0^{nm} \ : \ n,m \geq 0\}$
- $\{0^n c 0^{2^n} \ : \ n \geq 0\}$
- $\{\overline{x_1} c \overline{x_2} c \overline{x_1 + x_2} \ : \ x_1, x_2 \geq 0\}$ ahol $\overline{x}$ jelenti az x szám bináris alakját.
- $\{\overline{x} \ : \ x$ prímszám $\}$

Példák rekurzív függvényekre

- $x \mapsto \sqcup x \qquad x \in \{0,1\}^*$
- $\overline{x} \mapsto \overline{x+1} \qquad x \geq 0$
- $\overline{x} c \overline{y} \mapsto \overline{x+y} \quad x,y \geq 0$
- $\overline{n} \mapsto \overline{p_n} \qquad n \geq 0$

$p_n := n$-dik prímszám.

- Nyelvek – eldöntési problémák
- Rekurzív nyelvek – algoritmikusan megoldható eldöntési problémák
- Szófüggvények – optimalizálási problémák
- Rekurzív függvények – algoritmikusan megoldható optimalizálási problémák

Bármely véges matematikai objektum (algoritmikusan) **kódolható** szavakkal. Bármely két „elfogadható” kódolás polinomiális kapcsolatban áll egymással.

A kiszámíthatóság és bonyolultság elmélete **független a kódolástól**.

Többszavas Turing-gépek

Def. Egy k szavas Turing-gépen egy $M = (K, \Sigma, \delta, s)$ rendszert értünk, ahol K, Σ, s ugyanazok, mint közönséges Turing-gép esetén, δ pedig

$$K \times \Sigma^k \to (K \cup \{\text{h}, \text{„igen”}, \text{„nem”}\}) \times (\Sigma \times \{\leftarrow, -, \rightarrow\})^k$$

leképezés.

Kikötés: $\triangleright$ nem írható át és „nem léphető át".

Def. Konfiguráció:

$$(q, w_1, u_1, \ldots, w_k, u_k), \quad q \in K \cup \{\text{h}, \text{„igen”}, \text{„nem”}\}, w_i \in \Sigma^+, u_i \in \Sigma^*$$

Az $\xrightarrow{M}, \xrightarrow{M^t}, \xrightarrow{M^*}$ relációk értelem szerint definiáltak.

Palindromák eldöntése 2 szalaggal

$p \in K$	$\sigma_1 \in \Sigma$	$\sigma_2 \in \Sigma$	$\delta(p, \sigma_1, \sigma_2)$
s	0	$\sqcup$	$(s, 0, \rightarrow, 0, \rightarrow)$
s	1	$\sqcup$	$(s, 1, \rightarrow, 1, \rightarrow)$
s	$\triangleright$	$\triangleright$	$(s, \triangleright, \rightarrow, \triangleright, \rightarrow)$
s	$\sqcup$	$\sqcup$	$(q, \sqcup, \leftarrow, \sqcup, -)$
q	0	$\sqcup$	$(q, 0, \leftarrow, \sqcup, -)$
q	1	$\sqcup$	$(q, 1, \leftarrow, \sqcup, -)$
q	$\triangleright$	$\sqcup$	$(p, \triangleright, \rightarrow, \sqcup, \leftarrow)$
p	1	1	$(p, 1, \rightarrow, \sqcup, \leftarrow)$
p	0	0	$(p, 0, \rightarrow, \sqcup, \leftarrow)$
p	0	1	(„nem”, $0, -, 1, -$)
p	1	0	(„nem”, $1, -, 0, -$)
p	$\sqcup$	$\triangleright$	(„igen”, $\sqcup, -, \triangleright, -$)

$(s, \underline{\triangleright}, 010, \underline{\triangleright}, \varepsilon)$

$(s, \triangleright 010\underline{\sqcup}, \varepsilon, \triangleright 010\underline{\sqcup}, \varepsilon)$

$(q, \triangleright 01\underline{0}, \sqcup, \triangleright 010\underline{\sqcup}, \varepsilon)$

$(q, \underline{\triangleright}, 010\sqcup, \triangleright 010\underline{\sqcup}, \varepsilon)$

$(p, \triangleright\underline{0}, 10\sqcup, \triangleright 01\underline{0}, \sqcup)$

$(p, \triangleright 0\underline{1}, 0\sqcup, \triangleright 0\underline{1}, \sqcup\sqcup)$

$(p, \triangleright 010\underline{\sqcup}, \varepsilon, \underline{\triangleright}, \sqcup\sqcup\sqcup\sqcup)$

(„igen”, $\triangleright 010\underline{\sqcup}, \varepsilon, \underline{\triangleright}, \sqcup\sqcup\sqcup\sqcup$)

Def. Legyen $x \in (\Sigma - \{\sqcup, \triangleright\})^*$.

$M(x) =$ „igen"

$$(s, \triangleright, x, \triangleright, \varepsilon, \ldots, \triangleright, \varepsilon) \xrightarrow{M^*} (\text{„igen”}, w_1, u_1, \ldots, w_k, u_k)$$

$M(x) =$ „nem"

$$(s, \triangleright, x, \triangleright, \varepsilon, \ldots, \triangleright, \varepsilon) \xrightarrow{M^*} (\text{„nem”}, w_1, u_1, \ldots, w_k, u_k)$$

$M(x) = y$

$$(s, \triangleright, x, \triangleright, \varepsilon, \ldots, \triangleright, \varepsilon) \xrightarrow{M^*} (h, w_1, u_1, \ldots, w_k, u_k)$$

és y a $w_k u_k$ elején álló $\triangleright$ és végén esetleg előforduló $\sqcup$ jelek elhagyásával előálló szó.

$M(x) = \nearrow$ különben.

Eldöntés, felismerés stb. közönséges Turing-géphez hasonlóan.

Def. Legyen $f : \mathbb{N} \to \mathbb{N}$, M pedig többszavas Turing-gép. M $f(n)$ **időkorlátos Turing gép**, ha M minden x bemeneten legfeljebb $f(|x|)$ lépésben megáll. ($|x|$ a szó hosszát jelöli.) Más elnevezések: M **időigénye** (legfeljebb) $f(n)$.

Def. $\mathrm{TIME}(f(n))= \{L : \exists M$ $f(n)$ időkorlátos Turing gép, mely eldönti L-et $\}$

Példa.

$L \subseteq \{0,1\}^*$, palindrómák.

$L \in \mathrm{TIME}(\frac{(n+1)(n+2)}{2})$

$L \in \mathrm{TIME}(3n+3)$

Időkorlát esetén mindig kikötjük, hogy $f(n) > n$.

Tétel. Bármely k-szavas $f(n) > n$ időkorlátos M Turing-géphez megadható olyan $\mathcal{O}((f(n))^2)$ időkorlátos, egyszavas M' Turing-gép, hogy $M(x) = M'(x)$ teljesül minden x bemenő szóra.

Biz.

- M' egyetlen szóban tárolja M k szavának konkatenációját.
- Minden egyes szóban egy betű „alá van húzva".
- Egy lépés szimulálásához M' végigolvassa az egész szót.
- Esetleg jobbra lépteti a szót.
- Időigény lépésenként $\mathcal{O}(f(n))$.
- Teljes időigény: $\mathcal{O}((f(n))^2)$.

Lineáris felgyorsítás

Tétel. Tegyük fel, hogy $L \in \mathrm{TIME}(f(n))$. Ekkor bármely $\varepsilon > 0$ valós számra $L \in \mathrm{TIME}(f'(n))$ is teljesül, ahol $f'(n) = \varepsilon f(n) + n + 2$.

Biz. Legyen M k-szavas, $f(n)$ időkorlátos Turing-gép.

$$k' = \begin{cases} k & \text{ha } k > 1 \\ 2 & \text{ha } k = 1 \end{cases}$$

Belátjuk, hogy M szimulálható k' szavas, $f'(n)$ időkorlátos Turing-géppel (M'-vel).

- M' az M szavait m hosszú blokkokra bontja, és minden blokkot egyetlen betűként tárol.
- Első menet: tömörítés. M' második szavában elkészíti az első szó (bemenet) tömörített változatát: $n+2$ lépés.
- A második szó elejére mozgatja a mutatót $\lceil \frac{n}{m} \rceil$ lépés.
- M' állapotai: $(q, i_1, \ldots, i_k) \quad 1 \leq i_j \leq m$.
- M' az M gép m egymást követő lépését 6 lépésben szimulálja: $6 \left\lceil \frac{f(n)}{m} \right\rceil$ lépés.
- Teljes időigény:
  $n + 2 + \lceil \frac{n}{m} \rceil + 6 \left\lceil \frac{f(n)}{m} \right\rceil \leq n + 2 + 7 \left\lceil \frac{f(n)}{m} \right\rceil$
- Ha m elég nagy, akkor ez $\leq \varepsilon f(n) + n + 2$.

Köv. Ha $L \in \mathrm{TIME}(f(n))$, ahol $f(n) = an + b$, $a, b \in \mathbb{N}$, $a > 0$, akkor n együtthatóját tetszőlegesen közel vihetjük 1-hez: $L \in \mathrm{TIME}(f'(n))$, ahol $f'(n) = (1 + \varepsilon)n + c$, valamely tetszőlegesen kicsi pozitív ε-ra és valamely c-re.

Ha $L \in \mathrm{TIME}(f(n))$, ahol $f(n)$ másodfokú polinom, akkor a másodfokú tag együtthatóját tetszőlegesen közel vihetjük 0-hoz.

Def.

$$\mathbf{P} = \mathrm{TIME}(n^k), \text{azaz} \quad \mathbf{P} = \bigcup_{i=1}^{\infty} \mathrm{TIME}(n^i)$$

Tehát $L \in \mathbf{P} \iff \exists p(n)$ polinom, hogy $L \in \mathrm{TIME}(p(n))$.

Tárkorlátok

Def. Legyen $k > 2$. Egy k-szavas **lyukszalagos Turing-gép** olyan k szavas Turing-gép, amely

- első szalagját csak olvassa,
- utolsó szalagjára csak ír.

(Itt a szalag a szó szinonímája.)

$\delta(q, \sigma_1, \ldots, \sigma_k) = (p, \rho_1, D_1, \ldots, \rho_k, D_k) \Rightarrow \rho_1 = \sigma_1$, és $D_k \neq \leftarrow$.

Továbbá, ha $\sigma_1 = \sqcup$, akkor $D_1 = \leftarrow$.

Áll. Bármely k-szavas $f(n)$ időkorlátos M Turing-géphez létezik vele ekvivalens, $\mathcal{O}(f(n))$ időkorlátos $(k+2)$-szavas M' lyukszalagos Turing-gép.

Ekvivalens: $M(x) = M'(x)$ minden x bemenetre.

Legyen M egy k-szavas Turing-gép és x egy bemenő szó. Tegyük fel, hogy

$$(s, \triangleright, x, \triangleright, \varepsilon, \ldots, \triangleright, \varepsilon) \xrightarrow{M^*} (p, w_1, u_1, \ldots, w_k, u_k),$$

ahol $p \in \{$h, „igen”, „nem”$\}$.

M **tárigénye** x-en: $\sum_{i=1}^{k} |w_i u_i|$.

Kivétel: M (hangsúlyozottan) lyukszalagos.

Ekkor a tárigény x-en: $\sum_{i=2}^{k-1} |w_i u_i|$.

Def. Azt mondjuk, hogy M $f(n)$ **tárigényű Turing-gép**, ha M minden x bemeneten megáll, és tárigénye minden x bemeneten legfeljebb $f(n)$.

Def. Azt mondjuk, hogy az L nyelv a $\mathrm{SPACE}(f(n))$ bonyolultsági osztályban van, ha létezik az L nyelvet eldöntő $f(n)$ tárkorlátos lyukszalagos Turing-gép.

Jelölés. $\mathbf{L} = \mathrm{SPACE}(\log n)$

Példa. A palindromák nyelve az $\mathbf{L}$ osztályba esik.

- Egy lyukszalagos Turing-gép második szavában egy $1 \leq i \leq n$ szám bináris alakját tárolja.
- Az i-dik menetben megnézi, hogy a bemenet i-dik betűje megegyezik-e a hátulról i-dikkel. Ehhez egy harmadik munkaszót használ a számláláshoz.

Tétel. $L \in \mathrm{SPACE}(f(n)) \Rightarrow \forall \varepsilon > 0 : L \in \mathrm{SPACE}(\varepsilon f(n))$.

Biz. Legyen M az L-et eldöntő $f(n)$ tárkorlátos lyukszalagos Turing-gép. Legyen $m > 0$. Készítünk egy M-et szimuláló M' lyukszalagos Turing-gépet, mely M munkaszalagjait m-es blokkokban tárolja egyetlen munkaszalagján.

M' tárigénye: $\left\lceil \frac{f(n)}{m} \right\rceil$.

De

$$\left\lceil \frac{f(n)}{m} \right\rceil \leq \lceil \varepsilon f(n) \rceil, \quad \text{ha} \quad m \geq \left\lceil \frac{1}{\varepsilon} \right\rceil.$$

Nemdeterminisztikus Turing-gép

Def. k-szavas **nemdeterminisztikus Turing-gép** egy $N = (K, \Sigma, \Delta, s)$ rendszer, ahol K, Σ, s ugyanazok, mint közönséges k-szavas Turing-gép esetén,

$$\Delta \subseteq \Big[K \times \Sigma^k\Big] \times \Big[(K \cup \{\mathsf{h}, \text{„igen”}, \text{„nem”}\}) \times (\Sigma \times \{\leftarrow, -, \rightarrow\})^k\Big]$$

A konfiguráció, közvetlen átmenet ($\xrightarrow{N}$), közvetett átmenet ($\xrightarrow{N^*}$), t lépéses átmenet ($\xrightarrow{N^t}$) relációkat az előzőeknek megfelelően definiáljuk.

$$(p', w_1', u_1', \ldots, w_k', u_k')$$

$$\nearrow$$

$$(q, w_1, u_1, \ldots, w_k, u_k)$$

$$\searrow$$

$$(p'', w_1'', u_1'', \ldots, w_k'', u_k'')$$

Def. Legyen $L \subseteq (\Sigma - \{\triangleright, \sqcup\})^*$. Azt mondjuk, hogy N **eldönti** az L nyelvet, ha minden $x \in (\Sigma - \{\triangleright, \sqcup\})^*$ szóra:

$$x \in L \iff \exists\, w_1, u_1, \ldots, w_k, u_k :$$
$$(s, \triangleright, x, \triangleright, \varepsilon, \ldots, \triangleright, \varepsilon) \xrightarrow{N^*} (\text{„igen”}, w_1, u_1, \ldots, w_k, u_k)$$

Tehát nem követeljük meg, hogy N minden bemeneten megálljon, és létezhetnek végtelen számítási sorozatok is!

Def. Legyen $L \subseteq (\Sigma - \{\triangleright, \sqcup\})^*$, $f : \mathbb{N} \to \mathbb{N}$. Azt mondjuk, hogy N $f(n)$ **időben eldönti** az L nyelvet, ha eldönti, és valahányszor egy x szóra, $(q, w_1, u_1, \ldots, w_k, u_k)$ konfigurációra és $t \geq 0$ számra:

$$(s, \triangleright, x, \triangleright, \varepsilon, \ldots, \triangleright, \varepsilon) \xrightarrow{N^t} (q, w_1, u_1, \ldots, w_k, u_k)$$

mindig fennáll, hogy $t \leq f(|x|)$.

Tehát minden számítási sorozat véges.

Def. $\mathrm{NTIME}(f(n)) = \{\ L$: létezik L-et $f(n)$ időben eldöntő nemdeterminiztikus Turing-gép $\}$.

Itt is feltesszük, hogy $f(n) > n$.

Áll. $\mathrm{TIME}(f(n)) \subseteq \mathrm{NTIME}(f(n))$.

Tétel. Minden $\varepsilon > 0$ valós számra:

$$\mathrm{NTIME}(f(n)) \subseteq \mathrm{NTIME}(\varepsilon f(n) + n + 2).$$

Def. $\mathbf{NP} = \mathrm{NTIME}(n^k) = \bigcup_{i=1}^{\infty} \mathrm{NTIME}(n^i)$

Példa. $TSP(E) \in \mathbf{NP}$

Nem ismert (de nagyon valószínűtlen), hogy létezik-e „hatékony” módszer nemdeterminisztikus Turing-gépek szimulálására.

$$\mathbf{P} \stackrel{?}{=} \mathbf{NP}$$

Tétel. $\mathrm{NTIME}(f(n)) \subseteq \bigcup_{c>1} \mathrm{TIME}(c^{f(n)})$.

Biz. Legyen N egy $f(n)$ időkorlátos nemdeterminisztikus Turing-gép. Belátjuk, hogy N szimulálható egy 3-szavas, $c^{f(n)}$ időkorlátos M Turing-géppel valamely $c > 1$ számra.

Legyen $N = (K, \Sigma, \Delta, s)$. Minden $(q, \sigma) \in K \times \Sigma$ rendezett párra legyen

$$C_{q,\sigma} = \{(q', \sigma', D) : ((q, \sigma), (q', \sigma', D)) \in \Delta\},$$
$$d = \max_{q,\sigma} |C_{q,\sigma}|.$$

Feltehető, hogy $d > 1$.

Minden t lépésből
álló nemdeterminisztikus választási sorozat reprezentálható egy

$$c_1 \dots c_t, \qquad c_i \in \{0, \dots, d-1\}$$

sorozattal. A sorozatokat rendezzük lexikografikusan.

M az egyik munkaszalagján sorra előállítja a $c_1 \ldots c_t$ választási sorozatokat, a másikon az adott sorozatra szimulálja N működését.

M akkor áll meg „igen” állapotban, ha valamely választási sorozatra N ugyanezt teszi.

M akkor áll meg „nem” állapotban, ha valamely t esetén N az összes t hosszú választási sorozatra egy „igen”-től különböző állapotban áll meg.

Időigény:

Választások száma: $\sum_{t=0}^{f(n)} d^t = \frac{d^{f(n)+1}-1}{d-1} = \mathcal{O}(d^{f(n)+1}) = \mathcal{O}(d^{f(n)})$

Sorozatonként: $\mathcal{O}(f(n))$

Összesen: $\mathcal{O}(f(n)d^{f(n)}) = \mathcal{O}(d'^{f(n)})$

Def. Azt mondjuk, hogy a k-szavas $N = (K, \Sigma, \Delta, s)$ nemdeterminisztikus lyukszalagos Turing-gép $f(n)$ **tárral** (vagy tárkorláttal) **eldönti** az $L \subseteq (\Sigma - \{\triangleright, \sqcup\})^*$ nyelvet, ha eldönti és valahányszor

$$(s, \triangleright, x, \triangleright, \varepsilon, \ldots, \triangleright, \varepsilon) \xrightarrow{N^*} (q, w_1, u_1, \ldots, w_k, u_k)$$

ahol $x \in (\Sigma - \{\triangleright, \sqcup\})^*$, $q \in K \cup \{\text{„igen”}, \text{„nem”}, \text{h}\}$, $w_i, u_i \in \Sigma^*$, fennáll, hogy

$$\sum_{i=2}^{k-1} |w_i u_i| \leq f(|x|).$$

Def. $\mathrm{NSPACE}(f(n)) = \{L$: létezik olyan nemdeterminisztikus Turing-gép , mely $f(n)$ tárral eldönti L-et $\}$.

$$\mathbf{NL} = \mathrm{NSPACE}(\log n)$$

A Turing-gépnek létezhet végtelen számítási sorozata is!

Áll. $\mathrm{SPACE}(f(n)) \subseteq \mathrm{NSPACE}(f(n))$

Példa. ELÉRHETŐSÉG $\in \mathbf{NL}$

Biz. 4-szavas lyukszalagos nemdeterminisztikus Turing-gépet tervezünk.

- Egyik munkaszalag: i aktuális csúcs bináris alakja (kezdetben 1).
- Másik munkaszalag: nemdeterminisztikusan generál a gép egy j csúcsot (binárisan).
- Ellenőrzi, hogy (i, j) él-e.
  - Ha nem, akkor h állapotban megáll,
  - Ha igen, akkor
    $j = n$ esetén „igen” állapotban megáll;
    $j \neq n$ esetén kicseréli i-t j-vel és folytatja az eljárást.
- Az „igen” választ az output szalagra kiírja.

Később azt is belátjuk, hogy

$$\text{ELÉRHETŐSÉG} \in \mathrm{SPACE}(\log^2 n)$$

Tétel. $\forall \varepsilon > 0 : \ \mathrm{NSPACE}(f(n)) \subseteq \mathrm{NSPACE}(\varepsilon f(n)))$.

Közvetlen hozzáférésű gépek (RAM)

A **Church-Turing tézis** legfőbb bizonyítéka:
Az algoritmus fogalom bármely két matematikai modellje ekvivalens.

Erősebb tézis: „ Az algoritmusok és időigényük matematikai eszközökkel való modellezésére tett bármely észszerű kísérlet szükségképpen olyan modellhez és időigény-fogalomhoz vezet, mely **polinomiálisan ekvivalens** a Turing-géppel.”

Ezt demonstráljuk RAM esetén.

RAM

- adatszerkezet: **regiszterek**, melyek 0-tól kezdve sorszámozottak. Minden regiszter tetszőleges pozitív vagy negatív egész számot tárolhat. A 0. regiszter: **akkumulátor.**
- **címzési módok:**
  direkt j
  indirekt $\uparrow j$
  közvetlen $= j$
- **utasításszámláló:** κ
- **bemenet:** $(i_0, i_1, \ldots, i_t)$ egész számok sorozata

A j-dik regiszter tartalmát r_j jelöli.

Utasítások:

READ j	$r_0 := i_j$
READ $\uparrow j$	$r_0 := i_{r_j}$
STORE j	$r_j := r_0$
STORE $\uparrow j$	$r_{r_j} := r_0$

$x \in \{j, \uparrow j, = j\}$ esetén

LOAD x	$r_0 := x$
ADD x	$r_0 := r_0 + x$
SUB x	$r_0 := r_0 - x$
HALF	$r_{r_j} := \lfloor r_0/2 \rfloor$
JUMP j	$\kappa := j$
JZERO j	ha $r_0 = 0$ akkor $\kappa := j$
JPOS j	ha $r_0 > 0$ akkor $\kappa := j$
JNEG j	ha $r_0 < 0$ akkor $\kappa := j$
HALT	

Program: Utasítások véges sorozata. A program szemantikáját természetes módon definiáljuk. Hibás utasítás a program terminálását eredményezi.

Időigény számítása: minden utasítás egységnyi
(Túlzottan liberális – nem)

Bemenet mérete: $I = (i_0, \ldots i_t)$ mérete: $l(I) = \sum_{j=0}^{t} l(i_j)$

$l(i_j)$ az i_j bináris reprezentációjának hossza.

Tehát $\mathcal{O}(n)$ időigény azt jelenti, hogy a lépésszám arányos a bemeneten adott számok logaritmusával.

Tétel. A Turing-gép polinomiálisan ekvivalens a RAM-mal. Pontosabban:

A Legyen $L \subseteq (\Sigma - \{\triangleright, \sqcup\})^*$ a T $f(n)$ időkorlátos Turing-géppel eldöntött nyelv. Legyen $\Sigma = \{\sigma_1, \ldots, \sigma_k\}$,

$$D_\Sigma = \{(i_1, \ldots, i_n, 0) : \ n \geq 0, \ 1 \leq i_j \leq k\}$$

(Tehát D_Σ a Σ^*-nak felel meg.)

Legyen

$$\varphi_L : D_\Sigma \to \{0, 1\} \quad \varphi_L(i_1, \ldots, i_n, 0) = 1 \iff \sigma_{i_1} \ldots \sigma_{i_n} \in L.$$

Ekkor létezik olyan RAM-program, mely $\mathcal{O}(f(n))$ időben kiszámítja φ_L-et.

B Legyen D egész számok véges sorozatainak halmaza φ pedig D-t az egész számokba képező függvény. Adott i egészre jelölje $b(i)$ az i bináris reprezentációját, $I = (i_0, \ldots, i_t)$ esetén legyen $b(I) = b(i_0);\ldots;b(i_n)$.

Ha φ „kiszámítható RAM programmal”, akkor létezik olyan M Turing-gép, mely kiszámítja φ-t a következő értelemben. Tetszőleges $I \in D$ esetén

$$M(b(I)) = b(\varphi(I)).$$

Továbbá, ha φ $f(n)$ lépésben kiszámítható RAM programmal, akkor kiszámítható $\mathcal{O}(\ (f(n))^3\ )$ időkorlátos Turing-géppel.

Univerzális Turing-gép

Tétel. Létezik olyan U Turing-gép, amelynek ha adott egy tetszőleges M Turing-gép és az M egy x bemenete, úgy viselkedik, mint M az x-en, azaz $U(M;x) = M(x)$.

Természetesen M és x kódolva adottak U számára, és U az $M(x)$-et ezen kódolás szerint számítja ki.

A kódolás

$M = (K, \Sigma, \delta, s)$. Feltehető, hogy

$$\begin{aligned} \Sigma &= \{1, 2, \ldots, |\Sigma|\} \\ K &= \{|\Sigma| + 1, \ldots, |\Sigma| + |K|\} \\ s &= |\Sigma| + 1 \end{aligned}$$

Így M megadható a $|K|$, $|\Sigma|$ számok bináris alakjával, melyet δ leírása követ, amely $((q,\sigma),(p,\rho,D))$ alakú párok sorozata. A q, σ, p, ρ, D mindegyike bináris számmal megadható. A(,) stb. karakterek is kódolhatók binárisan.

$\leftarrow$,$-$,$\rightarrow$, „igen” , „nem”, h feleljen meg a $|\Sigma|+|K|+1,\ldots,|\Sigma|+|K|+6$ számoknak.

Az x bemenet szintén megadható bináris számok sorozatával.

U működése

Kétszavas Turing-gép.

1. szó: M;x bemenet
2. szó: M aktuális konfigurációjának kódja

U az M minden lépését egy menetben szimulálja, melyben

- Végigolvassa a 2. szót
- Meghatározza a 2. szón elvégzendő transzformációt, és azt elvégzi

Ha M megáll, U is azt teszi.

Amennyiben az U bemenete nem M;x alakú, akkor (mondjuk) végtelen ciklusba esik.

A megállási probléma

Def. Legyen $H = \{M;x \ : \ M(x) \neq \nearrow\}$

Tétel. H rekurzívan felsorolható, de **nem rekurzív**.

Biz.

- Rek. felsorolhatóság: az univerzális Turing-gép létezéséből
- H nem rekurzív: indirekt bizonyítás.

Tfh, H rekurzív, azaz eldönthető egy M_H Turing-géppel.

Legyen D olyan Turing-gép, melyre:

$$D(M) = \ \textbf{if } M_H(M;M) \textbf{ then } \nearrow \textbf{ else halt}$$

Ekkor:

$$D(D) = \nearrow \iff M_H(D;D) = \text{„igen”} \iff D(D) \neq \nearrow$$

Ellentmondás.

MEGÁLLÁS

Adott: M és x

Kérdés: Megáll-e M az x-en?

MEGÁLLÁS eldönthetetlen probléma.

Megj. H **teljes** a rekurzívan felsorolható nyelvek között.

- H rekurzívan felsorolható
- Minden rekurzívan felsorolható L nyelv (rekurzívan) **visszavezethető** H-ra
  - Legyen L az M Turing-gép által felismert nyelv.
  - $x \in L \iff M;x \in H$
  - Tehát az $\overset{?}{\in}$ L kérdés eldöntését visszavezettük az $M;x\overset{?}{\in}H$ kérdésre.

Áll. Az alábbi probléma algoritmikusan eldönthetetlen.

Adott: M Turing-gép

Kérdés: M megáll-e minden bemenetén?

Biz. Adott M és x esetén legyen M' olyan Turing-gép, hogy az M' minden y bemenő szavára:

$$M'(y) = M(x)$$

Így

$$M(x) \neq \nearrow \iff M' \text{ megáll minden bemenetén.}$$

Ha tehát az adott probléma eldönthető lenne, akkor MEGÁLLÁS is az lenne.

Tétel. (Rice) A rekurzívan felsorolható nyelvek egyetlen nemtriviális tulajdonsága sem dönthető el algoritmikusan.

Áll. Ha L rekurzív, akkor $\bar{L}$ is az.

Biz. Az „igen” és „nem” állapotok felcserélésével.

Áll. Egy L nyelv akkor és csak akkor rekurzív, ha L és $\bar{L}$ rekurzívan felsorolhatóak,

Biz.
L rekurzív $\Rightarrow$ L rek. felsorolható
$\bar{L}$ rekurzív $\Rightarrow$ $\bar{L}$ rek. felsorolható

Tfh. L és $\bar{L}$ rek. felsorolhatóak. Ekkor L felismerhető egy M, $\bar{L}$ pedig egy $\bar{M}$ Turing-géppel.

Szimuláljuk M-et és $\bar{M}$-t párhuzamosan!

Def. Legyen $\mathcal{C}$ nyelvek egy osztálya. Ekkor

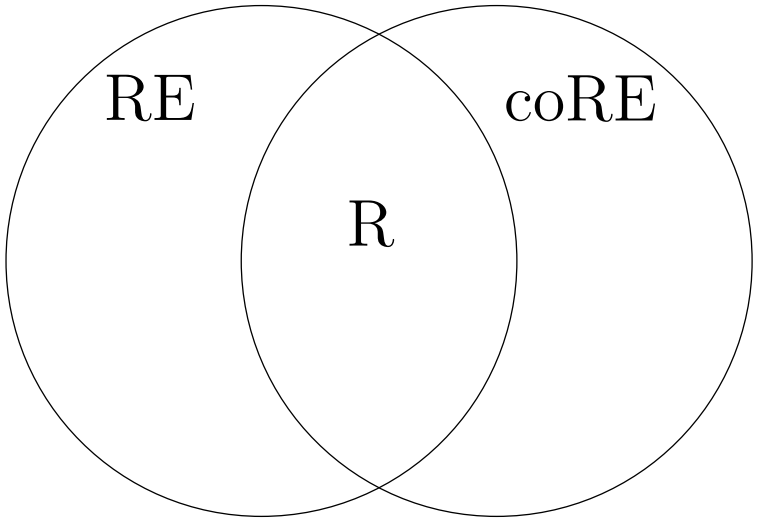
$$co\mathcal{C} = \{\bar{L} \ : \ L \subseteq \Sigma^*, \ L \in \mathcal{C}\}$$

Jelölés.

$$\begin{aligned} \text{RE} &= \{L \ : \ L \text{ rek. felsorolható}\} \\ \text{R} &= \{L \ : \ L \text{ rekurzív}\} \end{aligned}$$

Köv.

$$\begin{aligned} \mathrm{R} &= \mathrm{coR} \\ \mathrm{RE} &\neq \mathrm{coRE} \end{aligned}$$



HILBERT 10. PROBLÉMÁJA

Adott: $f(x_1, \ldots, x_n) = g(x_1, \ldots x_n)$ diofantikus egyenlet, ahol f és g egész együtthatós polinomok.

Kérdés: Létezik-e egész értékű megoldás?

Tétel. (Matijasevič) Hilbert 10. problémája algoritmikusan megoldhatatlan.

POST MEGFELELKEZÉSI PROBLÉMÁJA

Adott: $u_1, v_1, \ldots, u_n, v_n \in \Sigma^*$

Kérdés: Létezik-e olyan $i_1, \ldots, i_k$ ($k > 0$, $1 \leq i_j \leq n$) sorozat, hogy

$$u_{i_1} \ldots u_{i_k} = v_{i_1} \ldots v_{i_k}.$$

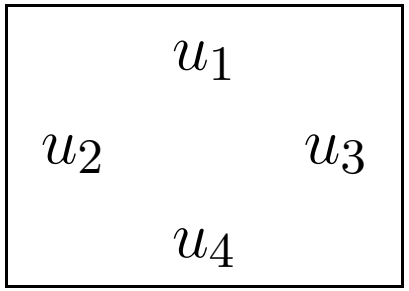
Tétel. (Post) A fenti probléma algoritmikusan megoldhatatlan.

Egy dominó probléma

Adott: Dominó típusok véges halmaza.

Kérdés: Kirakható-e dominókkal az egész sík hézagmentesen?

típus:



$u_i \in \Sigma^*$.

A dominók nem forgathatóak.

Tétel. A dominó probléma algoritmikusan megoldhatatlan.

Egy megoldatlan probléma

Adott: m pozitív egész szám.

Kérdés: Kongruens-e m?
Tehát azt kérdezzük, hogy létezik-e olyan derékszögű $\triangle$, mely oldalai racionális hosszúságúak és amely területe m.

Példa. $1, 2, 3, 4$ nem kongruensek

5 kongruens: (Fibonacci)

$a = 3/2 \quad b = 20/3 \quad c = 41/6$.

Nem ismert, hogy a probléma algoritmikusan eldönthető-e.

Bonyolultsági osztályok

Bonyolultsági osztály megadása

- számítási modell (általában nem túl fontos)
- számítási mód
- korlátozandó erőforrás
- korlát: $f : \mathbb{N} \to \mathbb{N}$

Def. Az $f(n)$ függvény **megengedett bonyolultsági függvény**, ha monoton nemcsökkenő, és létezik olyan M_f k-szavas lyukszalagos Turing-gép, mely $\mathcal{O}(\, n + f(n)\,)$ időben kiszámítja $f(n)$-et az alábbi értelemben: Minden n hosszú x bemenetre

$$(s, \triangleright, x, \triangleright, \varepsilon, \ldots, \triangleright, \varepsilon) \xrightarrow{M_f^t} (h, \triangleright, x, \triangleright, \sqcup^{j_1}, \ldots, \triangleright, \sqcup^{j_{k-1}}, \triangleright \sqcap^{f(n)}, \varepsilon)$$

ahol a $t = \mathcal{O}(\, n + f(n)\,)$ és $j_i = \mathcal{O}(\, f(n)\,)$ csak n-től függnek.

Idő esetén $f(n) > n$ is kikötés!

Példák.

- Minden konstans függvény.
- $\lceil \log n \rceil, 2^n, n$
- $f, g \Rightarrow f + g, f \cdot g, f^g$
- Tehát minden polinom.

Def. Egy M (lyukszalagos vagy nem, determinisztikus vagy nemdeterminisztikus) Turing-gép **pontos**, ha léteznek olyan f és g függvények, hogy M bármely n hosszú x bemenetéhez tartozó számítási sorozata pontosan $f(n)$ hosszú, és a megállás pillanatában M minden szava (lyukszalagos gép esetén az első és az utolsó kivételével) pontosan $g(n)$ hosszú.

Áll. Tegyük fel, hogy az M (determinisztikus vagy nemdeterminisztikus) Turing-gép $f(n)$ időben (vagy tárral) eldönti az L nyelvet, ahol f megengedett. Ekkor létezik olyan M' pontos Turing-gép, amely $\mathcal{O}(f(n))$ időben (vagy tárral) eldönti L-et. (Tehát M' még nemdeterminisztikus esetben is mindig megáll!)

- ***Biz.*** M' egy adott n-hosszú x bemeneten az „$f(n)$-et kiszámító” Turing-gép szimulálásával kezdi működését, melynek végén egy munkaszalagon előáll a $\sqcap^{f(n)}$ szó.

  Feltehető, hogy minden további munkaszalag hossza is ennyi az első fázis után.

  Ezután idő esetén az $\sqcap^{f(n)}$ szót mint órát használva pontosan $f(n)$ lépésig szimulálja M-et (esetleges üres lépésekkel).

  Tár esetén valamely alkalmas c konstansra $c^{f(n)}$ lépésig szimulálja M-et.

Jelölések.

$\mathrm{TIME}(\, f(n)\,)$,	$\mathrm{NTIME}(\, f(n)\,)$
$\mathrm{SPACE}(\, f(n)\,)$,	$\mathrm{NSPACE}(\, f(n)\,)$
$\mathbf{P} = \mathrm{TIME}(n^k)$,	$\mathbf{NP} = \mathrm{NTIME}(n^k)$
$\mathbf{PSPACE} = \mathrm{SPACE}(n^k)$,	$\mathbf{NPSPACE} = \mathrm{NSPACE}(n^k)$
$\mathbf{L} = \mathrm{SPACE}(\log n)$,	$\mathbf{NL} = \mathrm{NSPACE}(\log n)$
$\mathbf{EXP} = \mathrm{TIME}(2^{n^k})$	

A hierarchia tételek

Tétel. Ha $f(n) > n$ megengedett bonyolultsági függvény, akkor

$$\mathrm{TIME}(f(n)) \subsetneq \mathrm{TIME}(\ (f(2n+1))^3\ )$$

Biz.
$H_f = \{M;x\ :\ M$ legfeljebb $f(|x|)$ lépésben elfogadja x-et $\}$
ahol M tetszőleges többszavas Turing-gép.

Áll. $H_f \in \mathrm{TIME}(\ (f(n))^3\ )$.

Áll. $H_f \notin \mathrm{TIME}(\ f(\lfloor n/2 \rfloor)\ )$.

Tétel. Ha $f(n) > n$ megengedett bonyolultsági függvény, akkor

$$\mathrm{TIME}(f(n)) \subsetneq \mathrm{TIME}(\ f(n)\log^2 f(n)\ )$$

Példa.

$$\mathbf{P} \subseteq \mathrm{TIME}(2^n) \subsetneq \mathrm{TIME}(\ (2^{2n+1})^3\ ) \subseteq \mathrm{TIME}(2^{n^2}) \subseteq \mathbf{EXP}$$

Tehát $\mathbf{P} \subsetneq \mathbf{EXP}$.

Tétel. Ha $f(n)$ megengedett, akkor

$$\mathrm{SPACE}(\ f(n)\ ) \subsetneq \mathrm{SPACE}(\ f(n) \log f(n)\ )$$

Példa.

$$\mathbf{L} \subseteq \mathrm{SPACE}(n) \subsetneq \mathrm{SPACE}(n^2) \subseteq \mathbf{PSPACE}$$

Néhány alapvető összefüggés bonyolultsági osztályokra

Tétel.

(a) $\mathrm{TIME}(\, f(n)\,) \subseteq \mathrm{NTIME}(\, f(n)\,)$ és hasonlóan tárra

(b) $\mathrm{NTIME}(\, f(n)\,) \subseteq \mathrm{SPACE}(\, f(n)\,)$

(c) $\mathrm{NSPACE}(\, f(n)\,) \subseteq \mathrm{TIME}(\, k^{\log n + f(n)}\,)$

Biz.

(a) triviális

(b) Legyen M $f(n)$ időkorlátos nemdeterminisztikus Turing-gép. Feltehető, hogy minden számítási sorozat $f(n)$ lépésből áll n hosszú x bemeneten, és minden nem megállási konfigurációnak $d > 0$ „leszármazottja” van. Így **a nemdeterminisztikus választási sorozatokat reprezentálhatjuk az** $\{1, \ldots, d\}^{f(n)}$ **halmaz elemeivel.** M' ezeket generálja lexikografikusan, és a „tárat újra felhasználva” szimulálja minden sorozatra M működését.

M' megáll „igen” állapotban, ha M „igen” állapotban áll meg. M' akkor áll meg „nem” állapotban, ha M egyetlen választási sorozat esetén sem állt meg „igen” állapotban.

Mivel $f(n)$ megengedett, az első választási sorozatot generálni tudjuk ($1^{f(n)}$).

Tárigény: $f(n)$

(c) M legyen az L nyelvet $f(n)$ tárral eldöntő nemdeterminisztikus Turing-gép. Mivel M lyukszalagos, az utolsó szalag tartalmára nincs szükség.

Adott x, n-hosszú bemenethez tartozó konfiguráció:

$$(q, i, w_2, u_2, \ldots, w_{k-1}, u_{k-1})$$

Ezek száma: $nc_1^{f(n)} = c_1^{\log n + f(n)}$

Kérdés: Elérhető-e az x-hez tartozó konfigurációból egy („igen”,...) alakú konfiguráció?

Ez az ELÉRHETŐSÉG probléma, mely lineáris időben megoldható. Tehát létezik olyan c konstans, hogy L eldönthető Turing-géppel $c^{\log n + f(n)}$ időben.

A konfigurációs gráf elkészíthető „előre” is, de szebb megoldás az, hogy szükség esetén az x bemenet és M gép ismeretében eldöntjük minden esetben, hogy két konfiguráció között van-e él.

Köv. $\mathbf{L} \subseteq \mathbf{NL} \subseteq \mathbf{P} \subseteq \mathbf{NP} \subseteq \mathbf{PSPACE} \subseteq \mathbf{EXP}$

$\mathbf{L} \subseteq \mathbf{NL}$ (a)

$\mathbf{NL} = \mathrm{NSPACE}(\log n) \subseteq \mathrm{TIME}(k^{\log n}) \subseteq \mathbf{P}$ (c)

$\mathbf{P} \subseteq \mathbf{NP}$ (a)

$\mathbf{NP} = \mathrm{NTIME}(n^k) \subseteq \mathrm{SPACE}(n^k) = \mathbf{PSPACE}$ (b)

$\mathbf{PSPACE} = \mathrm{SPACE}(n^k) \subseteq \mathrm{TIME}(2^{n^k}) = \mathbf{EXP}$ (c)

Mivel $\mathbf{P} \subsetneq \mathbf{EXP}$, a $\mathbf{P} \subseteq \mathbf{NP}$, $\mathbf{NP} \subseteq \mathbf{PSPACE}$, $\mathbf{PSPACE} \subseteq \mathbf{EXP}$ valamelyike biztosan valódi.

Mivel $\mathbf{L} \subsetneq \mathbf{PSPACE}$, az $\mathbf{L} \subseteq \mathbf{NL}$, $\mathbf{NL} \subseteq \mathbf{P}$, $\mathbf{P} \subseteq \mathbf{NP}$, $\mathbf{NP} \subseteq \mathbf{PSPACE}$ valamelyike biztosan valódi.

Sejtés. Mindegyik az.

Savitch tétele

Tétel. ELÉRHETŐSÉG $\in \mathrm{SPACE}(\log^2 n)$

Biz. (Itt n akár a csúcsok számát, akár a szomszédsági mátrix tárolásához szükséges bitek számát is jelölheti.)

Legyen G egy n csúcsú gráf, azt kérdezzük, létezik-e 1-ből n-be vezető út, vagy általánosabban x_0-ból y_0-ba vezető út.

ÚT(x, y, i) $\iff$ **létezik x-ből y-ba vezető legfeljebb 2^i hosszúságú út.**

Világos, hogy akkor és csak is akkor érhető el y_0 az x_0-ból, ha ÚT$(x_0, y_0, \lceil \log n \rceil)$.

$i = 0$:

$$\text{ÚT}(x, y, i) = \begin{cases} \text{„igen”} & \text{ha } x = y \text{ vagy } (x, y) \text{ él} \\ \text{„nem”} & \text{különben} \end{cases}$$

$i > 0$:

Ellenőrizzük minden z-re, hogy **ÚT**$(x, z, i-1)$ **és ÚT**$(z, y, i-1)$ igaz-e. **Ha létezik ilyen** z, akkor ÚT$(x, y, i) =$ „igen”, különben ÚT$(x, y, i) =$ „nem”.

Készíthetünk olyan 3-szavas lyukszalagos Turing-gépet, mely ezt az algoritmust megvalósítja.

1. szalag: G, x_0, y_0 – bemenet

2. szalag: verem, rekurzív hívási lánc, kezdetben $(x_0, y_0, \lceil \log n \rceil)$

3. szalag: válasz

Hívási lánc hossza: $\lceil \log n \rceil$

Egy elem hossza: $\mathcal{O}(\log n)$

Összes tárigény: $\mathcal{O}(\log^2 n)$

Köv. Ha $f(n) \geq \log n$ megengedett bonyolultsági függvény, akkor

$$\mathrm{NSPACE}(f(n)) \subseteq \mathrm{SPACE}(\ (f(n))^2\ )$$

Speciálisan:

$$\mathbf{PSPACE} = \mathbf{NPSPACE}$$

Biz. Legyen N nemdeterminisztikus, $f(n)$ tárkorlátos lyukszalagos Turing-gép. Az N egy x bemeneten való szimulálásához **futtassuk le az előző algoritmust az N konfigurációs gráfján.** (Feltehető, hogy N mindig megáll.) A gráf mérete: $c^{f(n)}$. Így adódik a $\log^2 c^{f(n)} = \mathcal{O}(\ (f(n))^2\ )$ tárkorlát. (A konfigurációs gráfot nem készítjük el előre.)

Az Immerman – Szelepcsényi tétel

Tétel. Létezik olyan lyukszalagos **nemdeterminisztikus** Turing-gép, amely **logaritmikus tárral** kiszámítja adott G gráfra és annak x csúcsára **az x-ből elérhető csúcsok számát.**

Megj. Egy **nemdeterminisztikus lyukszalagos gép akkor számít ki egy $F(x)$ függvényt**, ha minden x bemenetre minden számítási sorozat végén h vagy „nem” állapotban áll meg (tehát nem léteznek végtelen számítási sorozatok), és minden x bemenetre létezik olyan számítási sorozat, melynek végén h állapotban áll meg. Ebben az esetben az utolsó szalag tartalma mindig $F(x)$. Ha még a munkaszavak hosszának összege mindig legfeljebb $f(|x|)$, akkor a gép $f(n)$ tárral számítja ki $F(x)$-et.

Biz. Jelölje n a csúcsok számát, és $k = 0, \ldots, n-1$ esetén legyen $S(k)$ az x-ből legfeljebb k hosszú úton elérhető csúcsok halmaza. Nekünk $|S(n-1)|$-et kell meghatároznunk.

Világos, hogy $S(0) = \{x\}$, $|S(0)| = 1$.

Az $|S(k)|$ meghatározásához $|S(k-1)|$-et használjuk fel,

továbbá egy olyan eljárást, amely választ ad az $u \overset{?}{\in} S(k)$ kérdésre. $u = 1, 2, \ldots, n$-re meghívjuk ezt az eljárást:

$m := 0$; válasz:=hamis
Minden $v = 1, \ldots, n$ csúcsra
  ha $v \in S(k-1)$ akkor m:=m+1
  ha továbbá (v, u) él vagy $v = u$, akkor válasz:=igen;
ha végül $m < |S(k-1)|$ akkor „nem”
különben az eredmény válasz értéke.

Itt a $v \in S(k-1)$ kérdésre egy „pontatlan” nemdeterminisztikus algoritmus ad választ, mely megsejt egy legfeljebb k hosszú csúcssorozatot és ellenőrzi, hogy az x-ből v-be vezető út-e.

Tárigény: arányos az egy csúcs tárolásához szükséges területtel: $\mathcal{O}(\log n)$. (Az út egyes pontjait egyenként sejtjük meg.)

Köv. Ha $f(n) \geq \log n$ megengedett bonyolultsági függvény, akkor

$$\mathrm{NSPACE}(f(n)) = \mathrm{coNSPACE}(f(n))$$

Biz. Tegyük fel, hogy L-et eldönti egy $f(n)$ tárkorlátos nemdeterminisztikus Turing-gép, melyről feltehető, hogy mindig megáll. Adott n hosszú bemeneten a konfigurációk száma $\leq c^{f(n)}$ Így nemdet. Turing-géppel $\mathcal{O}(f(n))$ tárral meghatározhatjuk a kezdő konfigurációból elérhető konfigurációk számát, majd ennek alapján – az előző algoritmusban adott módszerhez hasonlóan – eldönthetjük, hogy ezek között van-e („igen”, . . .) alakú.

Megj.

$$\begin{aligned} \mathrm{TIME}(f(n) &= \mathrm{coTIME}(f(n) \\ \mathrm{SPACE}(f(n)) &= \mathrm{coSPACE}(f(n)) \end{aligned}$$

Visszavezetések

Legyenek A és B problémák. B **legalább olyan nehéz, mint** A, ha létezik olyan hatékony módszer, azaz f rekurzív függvény, amely az A tetszőleges x bemenetéhez hozzárendeli a B egy $f(x)$ bemenetét (példányát) úgy, hogy x az A-nak akkor és csak akkor igen példánya, ha $f(x)$ a B igen példánya. Az f függvényre további megszorítást teszünk.

Def. Azt mondjuk, hogy az L_1 nyelv (logaritmikus tárral) **visszavezethető** L_2-re, ha létezik olyan $\mathcal{O}(\log n)$ tárral kiszámítható R szófüggvény, hogy minden x bemenetre

$$x \in L_1 \iff R(x) \in L_2.$$

Áll. Ha R egy M (lyukszalagos) Turing-géppel kiszámított visszavezetés, akkor M minden x bemeneten **polinom időben megáll**.

Biz. Adott n hosszú x bemeneten a **konfigurációk száma** $\mathcal{O}(nc^{\log n}) = \mathcal{O}(n^k)$. Ezek egyike sem fordulhat elő kétszer egy számítási sorozatban.

Tehát minden logaritmikus tárral való visszavezetés polinom idejű visszavezetés is, továbbá $|R(x)|$ az $|x|$ polinom függvényével korlátozható.

HAMILTON-ÚT

Adott: G irányított gráf.

Kérdés: Létezik-e olyan út, mely minden csúcsot pontosan egyszer látogat meg?

SAT

Adott: konjunktív normál alakú Boole-formula.

Kérdés: kielégíthető-e?

Áll. HAMILTON-ÚT visszavezethető a SAT-ra.

Biz. G csúcsai: $1, \ldots, n$. A formula felírásához az x_{ij}, $1 \leq i, j \leq n$, változókat használjuk. (Az x_{ij} igaz volta annak felel meg, hogy a j csúcs i-edik egy Hamilton-úton.)

A formulát 5 részformula konjunkciójaként állítjuk elő:

(1) $$\forall j \exists i \; x_{ij} \quad - \quad \bigwedge_j (x_{1j} \vee \ldots \vee x_{nj})$$

(2) $$\forall j \forall i_1 < i_2 \; (\neg x_{i_1 j} \vee \neg x_{i_2 j}) \quad - \quad \bigwedge_j \bigwedge_{i_1 < i_2} (\neg x_{i_1 j} \vee \neg x_{i_2 j})$$

A kettő együtt: $\forall j \exists ! i \; x_{ij}$

(3) $$\forall i \exists j \; x_{ij} \quad - \quad \bigwedge_i (x_{i1} \vee \ldots \vee x_{in})$$

(4) $$\forall i \forall j_1 < j_2 \; (\neg x_{i j_1} \vee \neg x_{i j_2}) \quad - \quad \bigwedge_i \bigwedge_{j_1 < j_2} (\neg x_{i j_1} \vee \neg x_{i j_2})$$

Együtt: $\forall i \exists ! j \; x_{ij}$ Eddig csak n-től függ.

$\forall (i,j) \notin G$ az i csúcs után nem következhet j:

(5)
$$\bigwedge_{(i,j)\notin G} \bigwedge_{k=1}^{n-1} (\neg x_{ki} \vee \neg x_{k+1j})$$

Adott G gráfra a formula logaritmikus tárral elkészíthető, és G-ben akkor és csak akkor van Hamilton-út, ha a formula kielégíthető.

***Def.* Hálózat:**
körmentes irányított gráf,
a csúcsok címkézettek: $\wedge, \vee, \neg,$ *igaz*, *hamis*, x_i
$\wedge, \vee$ címke esetén a csúcs be foka: 2
$\neg$ címke esetén a csúcs be foka: 1
hamis, *igaz*, x_i címke esetén a csúcs be foka: 0.

Általában megköveteljük még:
minden csúcs elérhető a 0 befokú csúcsokból,
pontosan egy csúcs kifoka 0.

Amennyiben a változók közül az $x_1, \ldots, x_n$ címkék fordulnak elő a hálózatban (legfeljebb), akkor a hálózat kiszámít egy $\{\textit{igaz}, \textit{hamis}\}^n \to \{\textit{igaz}, \textit{hamis}\}$ Boole-függvényt.

HÁLÓZAT-KIÉRTÉKELÉS

Adott: változómentes hálózat

Kérdés: igaz-e az értéke?

HÁLÓZAT-KIELÉGÍTHETŐSÉG

Adott: hálózat

Kérdés: a változóknak lehet-e úgy logikai értéket adni, hogy a hálózat érték igaz legyen?

Áll. Az ELÉRHETŐSÉG visszavezethető a HÁLÓZAT-KIÉRTÉKELÉS-re.

Biz. Ld. könyv (182. old / 8.2 Példa).

Áll. A HÁLÓZAT-KIELÉGÍTHETŐSÉG visszavezethető a SAT-ra.

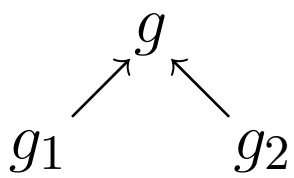
Minden g csúcsnak feleljen meg a g változó.

g **címkéje** x $\mapsto$ $x \Leftrightarrow g$, azaz $(\neg x \vee g) \wedge (\neg g \vee x)$

g **címkéje** *igaz* $\mapsto$ g

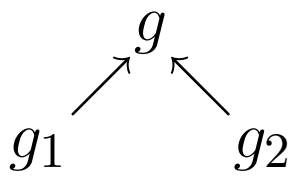
g **címkéje** *hamis* $\mapsto$ $\neg g$

g **címkéje** $\vee$:



$\mapsto$ $g \Leftrightarrow g_1 \vee g_2$, azaz $(\neg g_1 \vee g) \wedge (\neg g_2 \vee g) \wedge (\neg g \vee g_1 \vee g_2)$

g **címkéje** $\wedge$:



$\mapsto$ $g \Leftrightarrow g_1 \wedge g_2$, azaz $\neg g \Leftrightarrow \neg g_1 \vee \neg g_2$ azaz $(g_1 \vee \neg g) \wedge (g_2 \vee \neg g) \wedge (g \vee \neg g_1 \vee \neg g_2)$

g **címkéje** $\neg$:

$\mapsto$ $g \Leftrightarrow \neg g_1$, azaz $(\neg g \vee \neg g_1) \wedge (g_1 \vee g)$

g **kimenő kapu** $\mapsto$ g

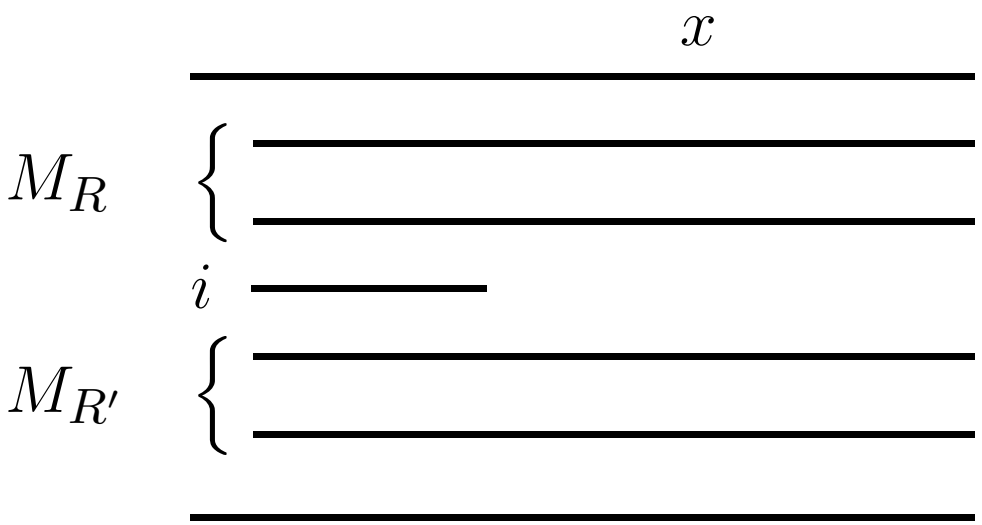
- A keresett formula: a fenti formulák konjunkciója.
- Adott hálózathoz a formula elkészíthető logaritmikus tárral. Továbbá a hálózat akkor és csak akkor kielégíthető, ha a formula az.
- Ha ugyanis a formula kielégíthető, akkor tekintsünk egy h kielégítő értékelését. Ez minden egyes g kapuhoz és x_i változóhoz meghatároz egy logikai értéket. Rendeljük minden kapuhoz ezt a logikai értéket. Ekkor a kimeneti kapu értéke a hálózat értéke amikor minden x_i vátozó értékét $h(x_i)$-nek választjuk.
- Ha a hálózat kielégíthető, akkor válasszuk meg minden x_i változó értékét úgy, hogy a hálózat értéke igaz legyen. Számítsuk ki a hálózat minden kapujának értékét: Az így kapott értékelés kielégíti a formulát.

- Az egyik irány: Ha H_g-t kielégíti a H_g változóinak egy h kiértékelése, akkor h-nak az a folytatása, mely a H_g minden g' csúcsához (mint változóhoz) a $H_{g'}$ értékét rendeli a h értékelés mellett, kielégíti F_g-t.
- A másik irány: Ha F_g-ét kielégíti egy h értékelés (mely értéket ad a H_g változóinak és minden csúcsának), akkor h megszorítása H_g változóira kielégíti a H_g hálózatot.

- Világos, hogy a visszavezetés reflexív.
- ***Tétel.*** A visszavezetés tranzitív: Ha R az L_1-nek L_2-re, R' az L_2-nek L_3-ra való visszavezetése, akkor $R \circ R'$ összetett függvény L_1-nek L_3-ra való visszavezetése.

  Biz. Legyen M_R és $M_{R'}$ az R-et illetve R'-t logaritmikus tárral kiszámító lyukszalagos Turing-gépek. Elkészítünk egy olyan lyukszalagos Turing-gépet, mely logaritmikus tárral kiszámítja az $R \circ R'$ függvényt. (Ez nem lehet a hagyományos szuperpozíció, mert $R(x)$ sokkal hosszabb lehet, mint $\log |x|$.)

Trükk: Nem tároljuk M_R kimenő szalagját, csak a mutató pozícióját (kezdetben ez 1).



$M_{R'}$ **-nek szüksége van a következő karakterre:** folytatjuk M_R szimulálását mindaddig, amig ezt ki nem írja

$M_{R'}$**-nek szüksége van az előző karakterre:** elölről kezdjük M_R szimulálását.

Def. Azt mondjuk, hogy egy $\mathcal{C}$ bonyolultsági osztály zárt a visszavezetésre, ha valahányszor L visszavezethető L'-re és $L' \in \mathcal{C}$, mindig teljesül, hogy $L \in \mathcal{C}$.

Áll. Az **L**, **NL**, **P**, **NP**, **PSPACE**, **EXP** osztályok zártak a visszavezetésre.

Biz. **L** és **NL** esetén az előző tételhez (tranzitivitás) hasonlóan. A többi osztály esetén hagyományos szuperpozíció is megfelel.

- $\mathbf{L}$ bármely két nemtriviális L, L' elemére L visszavezethető L'-re.

  Áll. Egy $\mathcal{C}$ bonyolultsági osztály akkor és csakis akkor zárt a visszavezetésre, ha $\mathrm{co}\mathcal{C}$ zárt.

Teljesség

Def. Legyen $\mathcal{C}$ egy bonyolultsági osztály. Egy L nyelvet $\mathcal{C}$-**nehéz**nek nevezünk, ha minden $L' \in \mathcal{C}$ nyelv visszavezethető L-re. Ha még $L \in \mathcal{C}$ is teljesül, akkor L $\mathcal{C}$**-teljes**.

Spec: **NP-teljes**, **P-teljes**, stb. nyelvek, ill. problémák.

Áll. Tfh. $\mathcal{C}$ zárt a visszavezetésre és az L nyelv $\mathcal{C}$-teljes. Akkor $\mathcal{C}$ az összes olyan L' nyelvből áll, amely visszavezethető L-re. Tehát L reprezentálja az egész osztályt!

Áll. Tfh. L nyelv $\mathcal{C}$-teljes, $L' \in \mathcal{C}$ és L visszavezethető L'-re. Ekkor L' is $\mathcal{C}$-teljes.

Áll. L nyelv $\mathcal{C}$-teljes $\iff$ $\bar{L}$ $\mathrm{co}\mathcal{C}$-teljes.

Egy **P**-teljes probléma

Tétel. A HÁLÓZAT-KIÉRTÉKELÉS probléma **P**-teljes.

Biz. Csak azt mutatjuk meg, hogy a probléma **P**-nehéz. E célból legyen $L \in P$, be kell látnunk, hogy L visszavezethető a HÁLÓZAT-KIÉRTÉKELÉS-re.

Mivel $L \in P$, létezik olyan polinom időkorlátos egyszavas M Turing-gép, mely eldönti L-et. Tfh. $p(n) - 2$ az M polinom időkorlátja, ahol $p(n) > 2$.

Ha megengedünk üres átmeneteket is, úgy feltehető, hogy M minden n hosszú x bemeneten pontosan $p(n) - 2$ lépésben áll meg „igen" vagy „nem" állapotban.

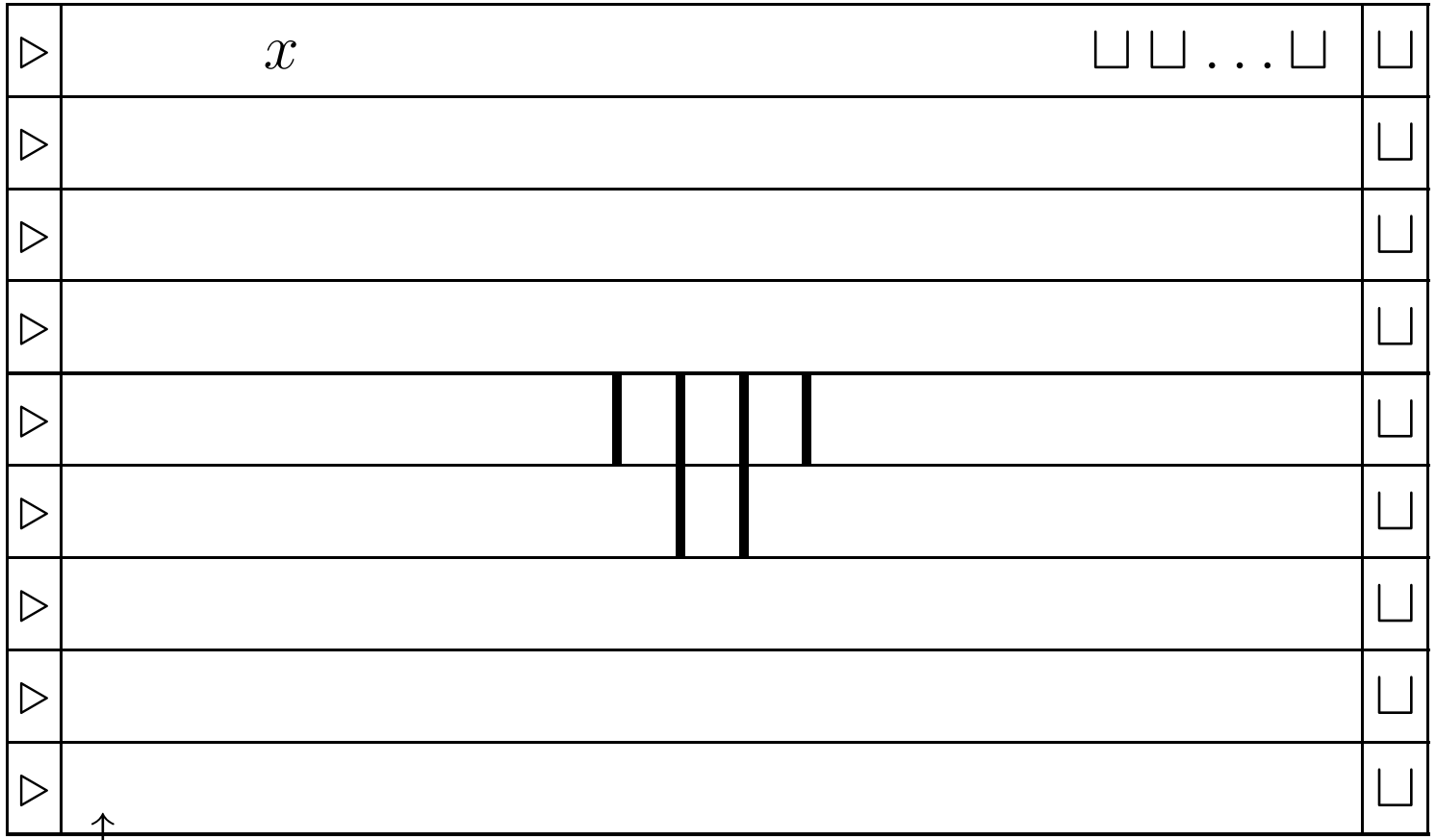
Azt is feltehetjük, hogy megálláskor a mutató a $\triangleright$ szimbólumot követő karakteren áll, és ez a karakter az „igen" vagy „nem" állapot.

M működését az x n hosszú bemeneten leírhatjuk egy $(p(n)-1)\times p(n)$ méretű $T=(T_{ij})$ táblázattal,

$$T_{ij}\in\Gamma=\Sigma\cup\{\sigma_q\ :\ \sigma\in\Sigma,\ q\in K\}$$

Itt σ_q jelentése: a gép a q állapotban van, a mutató az adott karaktert jelöli ki, mely σ.

Az egyszerűség kedvéért most feltesszük, hogy induláskor a mutató a $\triangleright$ szimbólumot követő karakterre mutat, és a $\triangleright$ szimbólumra soha nem kerül a mutató.


x
$\sqcup\sqcup\ldots\sqcup$
„igen” v. „nem”

A szélső elemek adottak. Minden belső elem a felette lévő 3 függvénye valamely M-től függő

$$f : \Gamma^3 \to \Gamma$$

függvény szerint.

Amennyiben G elemeit bitsorozatokkal kódoljuk, f helyettesíthető

$$f_l : \{\mathit{igaz}, \mathit{hamis}\}^{3m} \to \{\mathit{igaz}, \mathit{hamis}\}$$

($l = 1, ..., m, \quad m = \lceil \log |\Gamma| \rceil$) függvényekkel.

Az x bemenethez tartozó $R(x)$ hálózatot úgy kapjuk, hogy a táblázat minden belső pozícióját helyettesítjük egy egyszerű hálózattal (melyek véges sok, x-től függetlenül megadható hálózattípusból kerülnek ki).

szélső elem: a megfelelő szimbólum bináris kódjához tartozó csak kezdőcsúcsokat tartalmazó m csúcsú hálózat.

belső elem: $3m$ bemenettel és m kimenettel rendelkező, az $f_1, \ldots, f_m$ függvényeket kiszámító hálózat.

A belső elemek helyére kerülő hálózatok bemeneteit rendre azonosítjuk a felette lévő 3 hálózat kimeneteivel.

az egész hálózat kimenete: az utolsó sor 2. hálózat 1. kimenete.

$1 \rightarrow$ „igen”

$0 \rightarrow$ „nem”

Világos hogy,

- $R(x)$ elkészíthető logaritmikus tárkorlátos Turing-géppel,
- $x \in L \iff R(x)$ értéke *igaz*.

Cook tétele

Tétel. A SAT probléma **NP**-teljes.

Az nyilvánvaló, hogy $SAT \in \mathbf{NP}$. Mivel a HÁLÓZAT-KIELÉGÍTHETŐSÉG visszavezethető SAT-ra, így elegendő belátni, hogy a HÁLÓZAT-KIELÉGÍTHETŐSÉG**NP**-nehéz. (Persze akkor **NP**-teljes is.)

Tétel'. A HÁLÓZAT-KIELÉGÍTHETŐSÉG**NP**-teljes.

Biz. Legyen L egy M nemdeterminisztikus egyszavas Turing-géppel polinom időben eldöntött nyelv. A HÁLÓZAT-KIÉRTÉKELÉS**P**-teljességének bizonyítását követve, adott x bemenethez elkészítünk egy T táblázatot, mely a nemdeterminisztikus választásoktól is függ. Feltehető, hogy minden konfigurációban 2 választás lehetséges, így c egy $t = p(|x|)$ hosszú bitsorozat, ahol p az M időkorlátját megadó polinom függvény.

A belső $T_{i,j}$ elemek értéke most a $T_{i-1,j-1},\ T_{i-1,j},\ T_{i-1,j+1}$ értékeken kívül a c_i-től is függ, ami 0 vagy 1.

Így az f_l függvények:

$$\{\mathit{igaz}, \mathit{hamis}\}^{3m+1} \longrightarrow \{\mathit{igaz}, \mathit{hamis}\}.$$

Az $R(x)$ hálózatot a korábbiakhoz hasonlóan készítjük el azzal a különbséggel, hogy a hálózat rendelkezik még $c_1, \ldots, c_t$ bemeneti kapukkal, amelyek $x_1, \ldots, x_t$ változókkal címkézettek.

Az NP osztály egy jellemzése

Def. Legyen $R \subseteq \Sigma^* \times \Sigma^*$. Azt mondjuk, hogy R **polinom időben eldönthető**, ha az

$$L_R = \{x; y \ : \ R(x, y)\}$$

nyelv az.

Azt mondjuk, hogy R **polinomiálisan kiegyensúlyozott**, ha van olyan $p(n)$ polinom, hogy

$$R(x, y) \Rightarrow |y| \leq p(|x|)$$

Áll. $L \in \mathbf{NP} \Leftrightarrow \exists R$ polinom időben eldönthető, polinomiálisan kiegyensúlyozott reláció, hogy

$$L = \{x \ : \ \exists y \ R(x, y)\}$$

Biz. $\Rightarrow$ Tfh. $L \in \mathbf{NP}$. Létezik L-et eldöntő, $p(n)$ polinom időkorlátos nemdet. Turing-gép.

Legyen $R = \{(x, y) \ : \ y$ az x-hez tartozó elfogadó számítási sorozatot kódol. $\}$

$\Leftarrow$ Legyen $L = \{x \ : \ \exists y \ R(x, y)\}$, ahol R polinom időben eldönthető és polinomiálisan kiegyensúlyozott: $R(x, y) \Rightarrow |y| \leq p(|x|)$.

Az N nemdet. Turing-gép adott x esetén nemdeterminisztikusan generáljon egy tetszőleges y szót, amelyre $|y| \leq p(|x|)$, majd az L_R-et polinom időben eldöntő Turing-gépet szimulálva ellenőrizze, hogy $R(x, y)$ teljesül-e.

A kielégíthetőség változatai

Def. Legyen $k \geq 1$. A *k*SAT a SAT azon speciális esete, amelyben minden tag pontosan k (nem feltétlenül különböző) literálból áll.

Áll. 3SAT**NP**-teljes, és így $k \geq 3$ estén *k*SAT**NP**-teljes.

Biz.

$$
\begin{array}{rcl}
l & \mapsto & l \vee l \vee l \\
l_1 \vee l_2 & \mapsto & l_1 \vee l_2 \vee l_2 \\
l_1 \vee l_2 \vee l_3 & \mapsto & l_1 \vee l_2 \vee l_3 \\
l_1 \vee l_2 \vee l_3 \vee l_4 & \mapsto & (l_1 \vee l_2 \vee x) \wedge (\neg x \vee l_3 \vee l_4) \\
\vdots & & \vdots \\
l_1 \vee l_2 \vee \ldots \vee l_n & \mapsto & (l_1 \vee l_2 \vee x) \wedge (\neg x \vee l_3 \vee y) \wedge \\
 & & (\neg y \vee l_4 \vee z) \wedge \ldots \wedge (\neg u \vee l_{n-1} \vee l_n)
\end{array}
$$

Áll. Legyen $\varphi = c_1 \wedge c_2 \wedge \ldots \wedge c_k$ konjunktív normálformájú formula. Minden c_i taghoz legyen c_i' az előzőekben adott kifejezés, ahol minden kifejezésben új változókat használunk. Ekkor φ akkor és csak akkor kielégíthető, ha $\varphi' = c_1' \wedge c_2' \wedge \ldots \wedge c_k'$ az.

Mivel SAT visszavezethető 3SAT-ra és 3SAT $\in$ **NP**, ezért SAT**NP**-teljességéből következik, hogy 3SAT is **NP**-teljes.

2SAT:
Legyen φ olyan k.n.f., hogy φ minden tagja 2 literálból áll.

$G(\varphi)$:
csúcsok: A φ-ben előforduló változók és negáltjaik.
élek: (α, β) él $\Leftrightarrow \neg\alpha \vee \beta$ vagy $\beta \vee \neg\alpha$ a φ tagja
(azaz ha $\alpha \Rightarrow \beta$ vagy $\neg\beta \Rightarrow \neg\alpha$ a φ tagja.)

Áll. Akkor és csak akkor vezet α-ból β-ba irányított út, ha létezik $\neg\beta$-ból $\neg\alpha$-ba vezető irányított út.

Tétel. A φ formula akkor és csak akkor kielégíthető, ha nincs olyan x változó, amelyre létezik $G(\varphi)$-ben irányított út x-ből $\neg x$-be és vissza.

Biz. Tfh. létezik olyan x változó, hogy x-ből elérhető $\neg x$, és $\neg x$-ből elérhető x.

Belátjuk, hogy a változók tetszőleges T kiértékelésére φ hamis.

Tfh. $T(x)$ igaz (a $T(x)$ hamis eset hasonlóan kezelhető). Mivel $T(\neg x)$ hamis, az x-ből $\neg x$-be vezető úton létezik olyan (α, β) él, hogy $T(\alpha)$ igaz és $T(\beta)$ hamis. Ekkor $G(\varphi)$ konstrukciója szerint a megfelelő tag ($\neg\alpha \vee \beta$ vagy $\beta \vee \neg\alpha$) hamis, így φ hamis.

Tfh. egyetlen x változóra sem létezik x-ből $\neg x$-be és $\neg x$-ből x-be vezető út. Elkészítjük a változók egy olyan T kiértékelését, melyre φ igaz.

Válasszunk egy olyan x változót, amelynek igazságértékét még nem rögzítettük. Ha x-ből nem érhető el $\neg x$, akkor legyen $\alpha := x$, különben (ekkor $\neg x$-ből nem érhető el x) $\alpha := \neg x$.

α-hoz és minden α-ból elérhető csúcshoz rendeljük az igaz értéket, és tagadásaikhoz rendeljük a hamis értéket.
(Ezek pontosan azok, melyekből elérhető $\neg\alpha$.)

T jól definiált, **nem lehet**:
$\alpha \rightsquigarrow \beta$ és $\alpha \rightsquigarrow \neg\beta$, mert ekkor $\neg\beta \rightsquigarrow \neg\alpha$ és $\beta \rightsquigarrow \neg\alpha$ miatt $\alpha \rightsquigarrow \neg\alpha$ lenne.

$\alpha \rightsquigarrow \beta$ esetén **nem lehet**, hogy β korábban hamis értéket kapjon, mert akkor $\neg\beta \rightsquigarrow \neg\alpha$ alapján α már korábban hamis értéket kapott volna.

Valahányszor (α, β) él és $T(\alpha)$ igaz, $T(\beta)$ is igaz. Ezért φ **a** T **kiértékelés mellett igaz.**

Köv. 2SAT $\in$ **NL**, így 2SAT $\in$ **P**.

Biz. Mivel **NL**= **coNL**, elegendő azt belátni, hogy 2SAT-KOMPLEMENTER $\in$ **NL**.

Ehhez azt kell eldönteni, adott φ esetén, hogy létezik-e $G(\varphi)$-ben valamely x változóra x-ből $\neg x$-be és vissza vezető út.

Ez ugyanúgy eldönthető nemdeterminisztikus tárral, mint az ELÉRHETŐSÉG.

NP-teljes gráfelméleti problémák

FÜGGETLEN-CSÚCSHALMAZ
Adott: $G = (V, E)$ irányítatlan gráf, K szám.
Kérdés: Létezik-e K elemű független csúcshalmaz.

Tétel. A FÜGGETLEN-CSÚCSHALMAZ probléma **NP**-teljes.

Biz. Világos, hogy a probléma **NP**-ben van. Megmutatjuk, hogy 3SAT visszavezethető a FÜGGETLEN-CSÚCSHALMAZ-ra.

Legyen φ a 3SAT egy példánya, $\varphi = c_1 \wedge \ldots \wedge c_m$. A G gráf álljon m darab háromszögből, melyek a c_i tagoknak felelnek meg, s melyek csúcsai a c_i-kben lévő literáloknak felelnek meg. Ezen kívül még kössünk össze éllel minden ellentétes literált. Végül legyen $K = m$.

φ kielégíthető $\Leftrightarrow$ G-ben létezik K elemű független csúcshalmaz.

KLIKK

Adott: G=(V,E) irányítatlan gráf, K szám.

Kérdés: Létezik-e K elemű klikk (teljes részgráf)?

Tétel. KLIKK**NP**-teljes.

Biz. Világos, hogy KLIKK $\in$ **NP**. Legyen $(G = (V, E), K)$ a FÜGGETLEN-CSÚCSHALMAZ probléma egy példánya. Feltehető, hogy $K \leq |V|$. Legyen $G^c = (V, E^c)$ a G komplementer gráfja. Világos, hogy G-ben akkor és csak akkor létezik K elemű független csúcshalmaz, ha G^c -ben létezik K elemű klikk.

CSÚCSLEFEDÉS

Adott: $G = (V, E)$ gráf, K szám.

Kérdés: Létezik-e olyan K elemű csúcshalmaz, hogy minden él illeszkedik a halmaz egy csúcsához.

Tétel. A CSÚCSLEFEDÉS**NP**-teljes.

Biz. Legyen $(G = (V, E), K)$ a FÜGGETLEN-CSÚCSHALMAZ probléma egy példánya, ahol $K \leq |V|$. G-nek akkor és csak akkor létezik K elemű független csúcshalmaza, ha élei lefedhetőek $|V| - K$ elemű csúcshalmazzal.

HAMILTON-ÚT

Adott: $G = (V, E)$ irányítatlan gráf.

Kérdés: Létezik-e olyan út, melyben minden csúcs pontosan egyszer fordul elő?

Tétel. A HAMILTON-ÚT**NP**-teljes.

Tétel. A TSP (E) probléma **NP**-teljes

Biz. A HAMILTON-ÚT problémát visszavezetjük a TSP (E) problémára. Legyen G n csúcsú gráf az $\{1, \ldots, n\}$ halmazon. Ha $i \neq j$ és $[i, j]$ él, akkor d_{ij} legyen 1, különben legyen $d_{ij} = 2$. G-ben akkor és csak akkor létezik Hamilton-út, ha létezik $\leq n + 1$ összköltségű körút.

3-SZÍNEZÉS
Adott: G gráf.
Kérdés: Kiszínezhetőek-e a gráf csúcsai 3 színnel úgy, hogy a szomszédos csúcsok különböző színűek legyenek.

Tétel. A 3-SZÍNEZÉS **NP**-teljes.

Megj.
A 2-SZÍNEZÉS **P**-ben van.
A 4-SZÍNEZÉS **triviális** síkbarajzolható gráfokra.

NP-teljes problémák halmazokra és számokra

HÁRMASÍTÁS

Adott: Három azonos méretű halmaz, F (fiúk), L (lányok), H (házak), és egy $R \subseteq F \times L \times H$ reláció.

Kérdés: Megadható-e az R-beli hármasok egy olyan részhalmaza, melyben minden fiú, lány és ház pontosan egyszer szerepel?

Világos, hogy HÁRMASÍTÁS $\in$ **NP**.

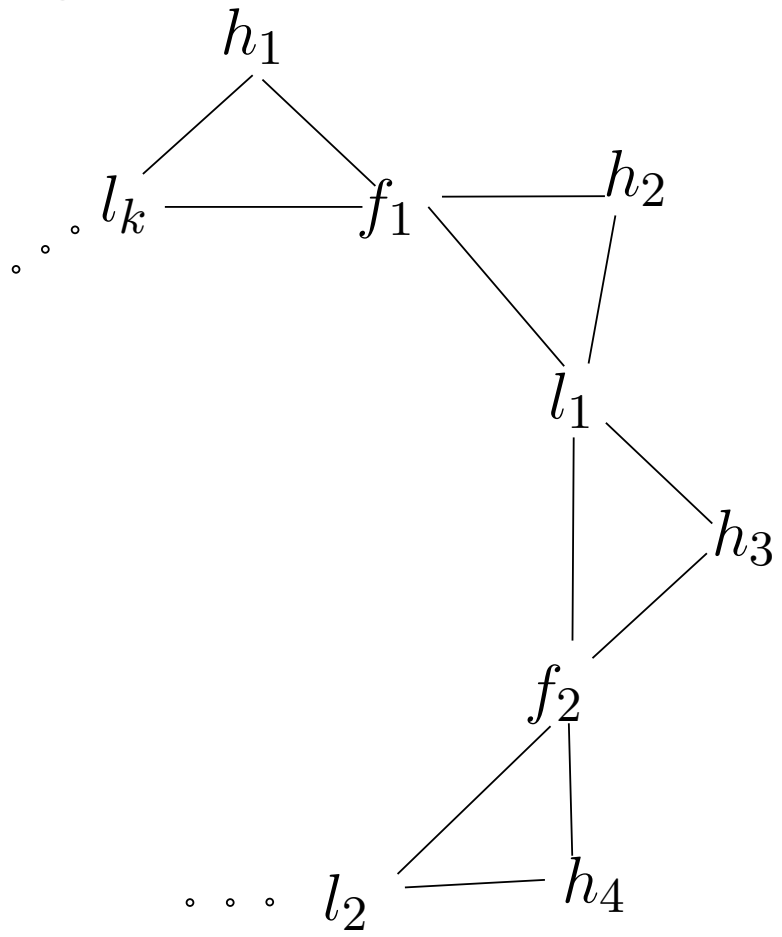
Tétel. A HÁRMASÍTÁS**NP**-teljes.

Megj. PÁROSÍTÁS $\in$ **P**.

Biz. A 3SAT-ot vezetjük vissza HÁRMASÍTÁS-ra.

$$\varphi = c_1 \wedge \ldots \wedge c_m$$

Adott x változóra jelölje k az x és $\neg x$ előfordulásainak maximumát. Az x változóhoz felvesszük $2k$ hármast; az ábrán látható háromszögeket:



Az x minden egyes előfordulásához hozzárendelünk egy páratlan indexű h_i -t , a $\neg x$ minden előfordulásához pedig egy páros indexű h_i-t.

Mivel az ábrán szereplő fiúk és lányok más hármasokban nem szerepelnek, minden hármasításban minden második háromszög szerepel:

$(f_1, l_1, h_2), (f_2, l_2, h_4), \ldots$ x igaz

$(f_1, l_k, h_1), (f_2, l_1, h_3), \ldots$ x hamis

Minden c_j taghoz felveszünk még 3 hármast, melyek (f, l, h) alakúak, ahol f, l csak a c_j -től függnek, h pedig azon 3 ház valamelyike, melyek a tagban szereplő literáloknak felelnek meg.

Az eddig elkészített R relációra érvényes:

Akkor és csak akkor adható meg az R egy olyan részhalmaza, melyben minden fiú és lány pontosan egyszer, és minden ház legfeljebb egyszer szerepel, ha φ kielégíthető.

A konstrukció befejezése:

Eddig $H \geq 3m$ házat és $\frac{H}{2} + m \leq \frac{H}{2} + \frac{H}{3}$ fiút és ugyanannyi lányt használtunk.

Jelölje K a $H - (\frac{H}{2} + m)$ különbséget. Felveszünk még K lányt és fiút, melyek mindegyik házba belerakhatók.

HALMAZLEFEDÉS

Adott: Egy U halmaz részhalmazainak $C = (S_1, \ldots, S_n)$ rendszere és egy K szám.

Kérdés: Kiválasztható-e K darab az S_i-k közül úgy, hogy ezek egyesítése U?

HALMAZPAKOLÁS

Adott: Egy U halmaz részhalmazainak $C = (S_1, \ldots, S_k)$ rendszere, és egy K szám.

Kérdés: Kiválasztható-e az S_i-k közül K páronként diszjunkt halmaz?

PONTOS LEFEDLÉS HÁRMASOKKAL

Adott: Egy $3m$ elemű U halmaz és 3-elemű részhalmazainak $(S_1, \ldots, S_n)$ rendszere.

Kérdés: Kiválasztható-e m darab S_i úgy, hogy ezek egyesítése U? (A kiválasztott S_i-k nyilván diszjunktak.)

Tétel. A HALMAZLEFEDÉS, HALMAZPAKOLÁS, PONTOS LEFEDLÉS HÁRMASOKKAL problémák **NP**-teljesek.

Biz. A HÁRMASÍTÁS, a PONTOS LEFEDLÉS HÁRMASOKKAL spec. esete.

A PONTOS LEFEDLÉS HÁRMASOKKAL, a HALMAZLEFEDÉS, valamint a HALMAZPAKOLÁS spec. esete is.

EGÉSZ ÉRTÉKŰ PROGRAMOZÁS

Adott: Egy n-változós, egész együtthatós lineáris egyenlőtlenség-rendszer.

Kérdés: Létezik-e egész értékű megoldás?

A HALMAZLEFEDÉS felfogható az EGÉSZ ÉRTÉKŰ PROGRAMOZÁS spec. eseteként:

$$Ax \geq 1, \quad \sum_{i=1}^{n} x_i \leq B, \quad 0 \leq x_i \leq 1$$

ahol A oszlopai a halmazrendszer elemeinek felelnek meg.
Azt is meg lehet mutatni, hogy az EGÉSZ ÉRTÉKŰ PROGRAMOZÁS**NP**-ben van.

Tétel. Az EGÉSZ ÉRTÉKŰ PROGRAMOZÁS**NP**-teljes.

Az EGÉSZ ÉRTÉKŰ PROGRAMOZÁS egy másik spec. esete: HÁTIZSÁK

Adott: n elem mindegyikének w_i súlya és v_i értéke, valamint W és K.

Kérdés: Kiválasztható-e ismétlés nélkül néhány elem úgy, hogy összértékük $\geq K$ és összsúlyuk $\leq W$?

Tétel. A HÁTIZSÁK**NP**-teljes.

A **coNP** osztály

Def. **coNP** $= \{L : L^c \in$ **NP**$\}$

Példák.

ÉRVÉNYESSÉG

Adott: φ Boole-formula

Kérdés: Érvényes-e, azaz azonosan igaz-e φ?

HAMILTON-ÚT KOMPLEMENTER

Adott: G gráf

Kérdés: Igaz-e, hogy G nem tartalmaz Hamilton-utat?

Áll. Az ÉRVÉNYESSÉG és HAMILTON-ÚT KOMPLEMENTER problémák **coNP**-ben vannak.

Egy probléma akkor van

- **NP**-ben, ha minden „igen" példányához létezik tömör, polinom időben ellenőrizhető bizonyíték, és csak az „igen" példányokhoz létezik ilyen bizonyíték.
- **coNP**-ben van, ha minden „nem" példányhoz létezik tömör, polinom időben ellenőrizhető cáfolat, és csak a „nem" példányokhoz létezik ilyen cáfolat

Áll. ÉRVÉNYESSÉG és HAMILTON-ÚT KOMPLEMENTER problémák **coNP**-teljesek.

Áll. **P**$\subseteq$ **NP**$\cap$ **coNP**

Áll. Tfh $\mathcal{C}$ zárt a visszavezetésre és L $\mathcal{C}$-teljes. Ekkor $\mathcal{C} = \mathrm{co}C \Leftrightarrow L \in \mathrm{co}C$.

Köv. **NP**=co**NP**$\Leftrightarrow$ SAT $\in$ co**NP**$\Leftrightarrow$ ÉRVÉNYESSÉG $\in$ **NP**

PRÍMEK

Adott: p szám (binárisan)

Kérdés: Prím szám-e p?

Áll. PRÍMEK $\in$ **coNP**.

Tétel. Egy $p > 1$ szám akkor és csakis akkor prímszám, ha létezik olyan $1 \leq r < p$, amelyre:

(a) $r^{p-1} = 1 \bmod p$

(b) A $p-1$ minden q prímszám osztójára $r^{(p-1)/q} \neq 1 \bmod p$.

Példa. $p = 5 \longrightarrow r = 3$

$3^4 = 1 \bmod 5$

$3^2 = 4 \bmod 5$

Tétel. (Pratt) PRÍMEK $\in$ **NP** $\cap$ **coNP**.

Biz. Legyen adott p. Feltehető, hogy $p > 2$. A p prímszám voltának bizonyítéka:

$$(r, q_1, \ldots, q_k),$$

ahol $1 \leq r < p$, $r^{p-1} = 1 \bmod p$, továbbá $q_1, \ldots, q_k$ a $p-1$ összes prímosztói, és így $k \leq \lceil \log p \rceil$ és $r^{(p-1)/q_i} \neq 1 \bmod p$, $i = 1, \ldots, k$.

Jelölés. $l = \lceil \log p \rceil$

Áll. $r^{(p-1)} \bmod p$ meghatározható valamely m-re $\mathcal{O}(l^m)$ időben.

Biz. Képezzük az $r \bmod p$, $r^2 \bmod p$, ..., $r^{2^l} \bmod p$ számokat, majd ezek közül összeszorozzuk azon r^{2^i}-ket, melyekre $p-1$ bináris reprezentációjában az i-edik helyiértékű bit 1.

Áll. l-ben polinom időben ellenőrizhető, hogy eljuthatunk-e $p-1$-ből az 1-be q_i-kel való osztások segítségével, ahol minden egyes q_i-t legalább egyszer felhasználjuk. Továbbá l-ben polinom időben ellenőrizhető az is, hogy minden i-re érvényes-e az $r^{(p-1)/q_i} \neq 1 \bmod p$ egyenlőtlenség.

De $(r, q_1, \ldots, q_k)$ **nem teljes bizonyíték**, mert valahányszor $q_i \neq 2$, annak is bizonyítékát kell adni, hogy q_i prím.
A teljes bizonyíték:

$$C(p) = (r, q_1, C(q_1), \ldots, q_k, C(q_k)),$$

ahol $C(q_i)$ a q_i prímszám voltának bizonyítéka
(ez üres, ha $q_i = 2$).

Áll. : a teljes bizonyíték hossza $\mathcal{O}(l^2)$

Áll. : Adott p-re a $C(p)$ bizonyíték helyessége l-ben polinom időben ellenőrizhető.

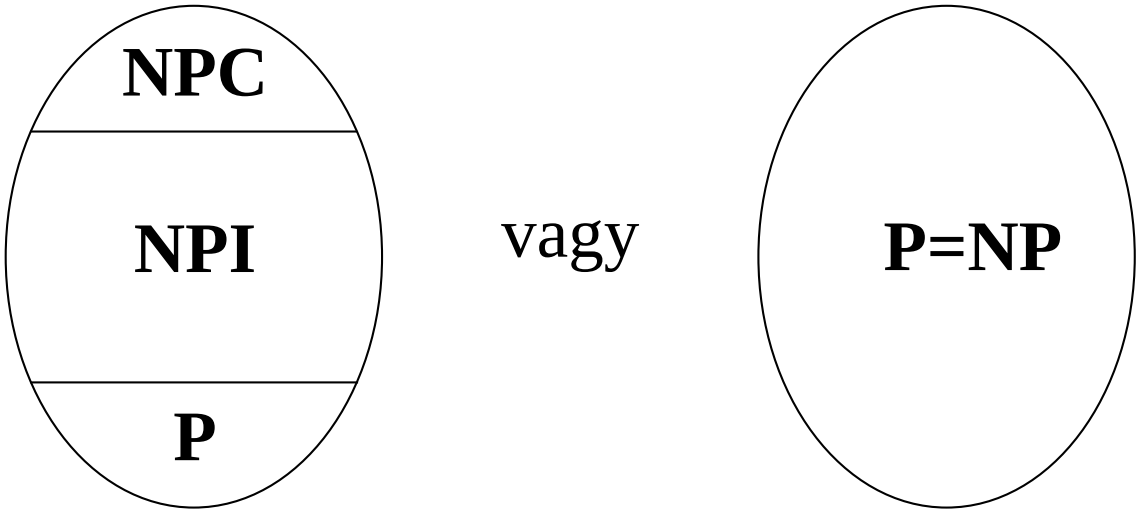
Biz. : A fentiek alapján, ld. könyv.

Egy új (2002. augusztusi) eredmény:
Tétel. (M. Agrawal, N. Kayal és N. Saxena) PRÍMEK $\in$ **P**

Néhány további eredmény az **NP** és a **P** osztályokról

Tétel. Ha **P** $\neq$ **NP**, akkor létezik olyan $L \in$ **NP**$-$**P**, mely nem **NP**-teljes.

Tehát



NPC: **NP**-teljes nyelvek (**NP**-complete)

NPI: Azon (**NP**$-$**P**)-beli nyelvek, amelyek nem **NP**-teljesek (**NP**-intermediate).

Def. Az $M^?$ **orákulumos**, determinisztikus vagy nemdeterminisztikus **Turing-gép** olyan többszavas Turing-gép, mely rendelkezik egy speciális kérdés szóval (szalaggal), valamint $q_?, q_{igen}, q_{nem}$ állapotokkal.

Legyen $A \subseteq \Sigma^*$, ahol Σ az M ábécéje. Ekkor az M^A Turing gép, melyben az **az orákulumot az A nyelvvel realizáljuk**, úgy működik, mint egy közönséges Turing-gép, kivéve, ha a $q_?$ állapotban van. Ekkor a q_{igen} vagy a q_{nem} állapotba megy át annak megfelelően, hogy a kérdés szó az A nyelvben van-e.

Időbonyolultság: mint a közönséges esetben,
minden kérdés egyetlen lépésnek számít.

Def. Legyen $\mathcal{C}$ bonyolultsági osztály, A tetszőleges nyelv.

$\mathcal{C}^A$: Azon nyelvek, amelyek eldönthetőek ugyanolyan típusú orákulumos Turing-géppel az orákulum A-val való realizálása mellett, mint a $\mathcal{C}$ osztályba eső nyelveket eldöntő Turing-gépek.

Legyen $\mathcal{C}'$ is bonyolultsági osztály.

$$\mathcal{C}^{\mathcal{C}'} = \bigcup_{A \in \mathcal{C}'} \mathcal{C}^A$$

Példa.

$\mathbf{P}^{\mathbf{P}} = \mathbf{P}$

$\mathbf{NP} \cup \mathbf{coNP} \subseteq \mathbf{P}^{\mathbf{NP}} = \mathbf{P}^{\mathrm{SAT}}$

Tétel. Van olyan A orákulum, melyre $\mathbf{P}^A = \mathbf{NP}^A$.

Lemma. Ha $A \in$ **PSPACE**, akkor $\mathbf{P}^A \subseteq$ **PSPACE**, és $\mathbf{NP}^A \subseteq$ **NPSPACE** $=$ **PSPACE**.

Lemma. Ha A **PSPACE**-teljes, akkor **PSPACE** $\subseteq \mathbf{P}^A$.

A tétel bizonyítása:
Legyen A **PSPACE**-teljes , később látni fogjuk, hogy ilyen A létezik. Ekkor:

$$\mathbf{PSPACE} \subseteq \mathbf{P}^A \subseteq \mathbf{NP}^A \subseteq \mathbf{PSPACE}.$$

Tétel. Van olyan B, melyre $\mathbf{P}^B \neq \mathbf{NP}^B$.

Tehát a **P**$\neq$**NP** sejtés csak olyan gondolatmenettel bizonyítható, mely nem relativizálható tetszőleges orákulumra.

Logaritmikus tár

Tétel. Az ELÉRHETŐSÉG probléma **NL**-teljes.

Biz. Azt már tudjuk, hogy a probléma **NL**-ben van. Legyen most az L egy **NL**-beli nyelv. Ekkor L eldönthető egy N, $\log n$ tárkorlátos (pontos) nemdeterminisztikus lyukszalagos Turing-géppel.

Egy n hosszú x bemeneten a konfigurációk száma: $\mathcal{O}(n^k)$ valamely k-ra. Feltehető, hogy egy elfogadó konfiguráció van.

Így a konfigurációs gráfot tekintve az ELÉRHETŐSÉG egy példányához jutunk, mely akkor és csak akkor "igen" példány, ha $x \in L$.

Mivel a konfigurációs gráf elkészíthető logaritmikus tárral, készen vagyunk.

Köv. AZ ELÉRHETETLENSÉG probléma **NL**-teljes.

Biz. **NL**=**coNL**

Tétel. A 2SAT probléma **NL**-teljes.

Biz. Azt már tudjuk, hogy 2SAT $\in$ **NL**. Tekintsük az ELÉRHETETLENSÉG egy példányát. Az előző tétel bizonyítása miatt feltehető, hogy az adott gráf körmentes.

Minden csúcshoz rendeljünk egy x változót. A 2SAT azon φ példányát készítjük el, mely az összes $\neg x \vee y$ tagokból áll, ahol (x, y) él, továbbá az s és $\neg t$ tagokból, ahol s a kezdőcsúcs és t a cél.

Így: φ kielégíthető $\iff$ s-ből nem érhető el t.

Tfh. T a φ kielégítő kiértékelése. Ekkor $T(x) = 1$ valahányszor x elérhető s-ből, és $T(t) = 0$. Így t nem érhető el.

Tfh. t nem érhető el s-ből. Legyen T azon kiértékelés, melyre $T(x) = 1$ akkor és csak akkor, ha x elérhető s-ből. Ekkor T kielégíti φ-t.

Alternáló Turing gép

Def. Az alternáló Turing gép olyan $N = (K, \Sigma, \Delta, s)$ nemdeterminisztikus Turing-gép, melyben K a $K_{\text{ÉS}}$ és K_{VAGY} diszjunkt halmazok egyesítése, és amely minden számítási sorozata véges (feltehető, hogy a gép pontos).

Tekintsük az N számítási fáját az x bemeneten. Azt mondjuk, hogy egy C konfiguráció **végül elfogadó**, ha

- C-ben a gép „igen” állapotban van, vagy
- $K_{\text{ÉS}}$-állapotban van, és **minden** közvetlen leszármazottja végül elfogadó, vagy
- K_{VAGY}-állapotban van, és **valamely** közvetlen leszármazottja végül elfogadó.

Az N gép akkor **fogadja el** az x bemenetet, ha a kezdő konfiguráció végül elfogadó, különben elutasítja azt.
Az $f(n)$ idő vagy tárkorlátos Turing-gép fogalmát értelemszerűen definiáljuk.

Def. Legyen $f(n)$ megengedett bonyolultsági függvény. (Idő esetén $f(n) > n$.)

$\mathrm{ATIME}(f(n))$: az összes $f(n)$ időkorlátos alternáló Turing-géppel eldönthető nyelvek.

$\mathrm{ASPACE}(f(n))$: az összes $f(n)$ tárkorlátos alternáló Turing-géppel eldönthető nyelvek.

AL$= \mathrm{ASPACE}(\log n)$

AP$= \mathrm{ATIME}(n^k)$

Világos, hogy $\mathrm{NTIME}(f(n)) \subseteq \mathrm{ATIME}(f(n))$ és $\mathrm{NSPACE}(f(n)) \subseteq \mathrm{ASPACE}(f(n))$.

Tétel. **AL**$=$**P**.

Biz. Láttuk, hogy a HÁLÓZAT-KIÉRTÉKELÉS probléma **P**-teljes. Valójában már a MONOTON-HÁLÓZAT-KIÉRTÉKELÉS is **P**-teljes, mert minden változómentes hálózat monotonná tehető. Mivel mindkét osztály zárt a visszavezetésre, **AL**$=$ **P** teljesül, ha a MONOTON-HÁLÓZAT-KIÉRTÉKELÉS**AL**-teljes.

Tétel. MONOTON-HÁLÓZAT-KIÉRTÉKELÉS $\in$ **AL**.

Biz. Adott monoton hálózat kiértékelését kezdje az alternáló Turing-gép a kimeneti kapunál. Ha ez igaz vagy hamis, a kiértékelésnek vége. Ha ÉS-kapu, akkor a gép ÉS-állapotba, különben VAGY-állapotba lép, és nemdeterminisztikusan a két ős valamelyikénél folytatja a kiértékelést. Ha végül igaz kapuhoz ér, „igen” állapotba, különben „nem” állapotba megy át. E közben mindig csak az aktuális kapu sorszámát kell tárolni.

Tétel. MONOTON-HÁLÓZAT-KIÉRTÉKELÉS**AL**-nehéz.

Biz. Megmutatjuk, hogy tetszőleges $L \in$ **AL** nyelv visszavezethető a MONOTON-HÁLÓZAT-KIÉRTÉKELÉS-re.

Legyen $N = (K, \Sigma, \Delta, s)$ olyan pontos, alternáló Turing-gép, mely $\log n$ tárral eldönti az L nyelvet. Feltehető, hogy minden nem elfogadó vagy elutasító konfigurációnak 2 leszármazottja van. A konfigurációs gráf adott x bemenet esetén körmentes. Minden C csúcsot helyettesítünk egy C' kapuval. Ez
ÉS kapu ha az állapot $K_{\text{ÉS}}$-ben van,
VAGY kapu, ha az állapot K_{VAGY}-ban van,
igaz kapu, ha az állapot „igen”
hamis kapu, ha az állapot „nem”.

Amennyiben C közvetlen leszármazottai C_1 és C_2, akkor a hálózatban C' ősei C'_1 és C'_2. A kimeneti csúcs a kezdőkonfiguráció.

Világos, hogy a hálózat értéke akkor és csakis akkor igaz, ha $x \in L$. Minden konfiguráció mérete $\mathcal{O}(\log n)$. Így könnyen látható, hogy a hálózat elkészíthető logaritmikus tárral.

Tétel. Ha $f(n) \geq \log n$ megengedett, akkor $\mathrm{ASPACE}(f(n)) = \mathrm{TIME}(k^{f(n)})$.

A poliniomiális tár

Def. QSAT :

Adott: $\exists x_1 \forall x_2 \ldots Q_m x_m \varphi$ kvantifikált Boole-formula Prenex normálalakban úgy, hogy a φ kvantormentes formula, melyben legfeljebb az $x_1, \ldots, x_m$ változók fordulnak elő.

Kérdés: Igaz-e az adott formula?

Megj.

- A kvantorok szigorú alternálása nem lényeges.
- Az sem lényeges, hogy az első kvantor $\exists$.

Áll. A QSAT probléma **PSPACE**-ben van.

Biz. Legyen adott a $\exists x_1 \forall x_2 \ldots Q_m x_m \varphi$ formula. Ebből $\mathcal{O}(m)$ tárral elkészítünk egy olyan hálózatot, mely gráfja egy m mélységű, kiegyensúlyozott bináris fa.

- Páratlan szinten: **VAGY** kapu
- Páros szinten: **ÉS** kapu
- Levél: **igaz** vagy **hamis** attól függően, hogy a „megfelelő” kiértékelés mellett φ igaz vagy hamis.

Egy rekurzív algoritmussal a mélységgel arányos, tehát $\mathcal{O}(m)$ tárral kiértékeljük a hálózatot.

A visszavezetés tranzitivitásának bizonyításában megismert módszer szerint a két algoritmust összetehetjük – $\mathcal{O}(m)$ **tárkolátos Turing-gépet kapunk.**

Megj. A probléma akkor is **PSPACE**-ben van, ha

- a kvantorok nem feltétlenül váltakoznak.
- φ-ben az $x_1, \ldots, x_m$ változókon kívül egyéb változók is előfordulnak, és azt kérdezzük, kielégíthető-e az egész formula.
- A formula nem feltétlenül Prenex alakú.
- Azt kérdezzük, hogy a formula hamis-e.

Tétel. A QSAT probléma **PSPACE**-teljes.

Biz. Mivel **PSPACE**$=$**coPSPACE**, a QSAT probléma akkor és csakis akkor **PSPACE**-teljes, ha QSAT-KOMPLEMENTER**PSPACE**-teljes.

Legyen adott az $L \in$ **PSPACE** nyelv, melyet eldönt az $\mathcal{O}(n^{k'})$ tárkorlátos M Turing-gép. Adott x, n hosszú bemeneten a konfigurációk száma $\mathcal{O}(2^{n^k})$ valamely k-ra. Az egyszerűség kedvéért feltesszük, hogy a konfigurációk száma legfeljebb 2^{n^k}.

Minden konfigurációt kódoljunk egy n^k hosszú bitsorozattal. Legyen

$$\textbf{ÚT}(a, b, i)$$

akkor és csakis akkor ha a-ból elérhető a b legfeljebb 2^i hosszú úton. Megmutatjuk, hogyan lehet felírni olyan

$$\psi_i(A, B)$$

formulát az $A = \{a_1, \ldots, a_{n^k}\}, B = \{b_1, \ldots, b_{n^k}\}$ szabad változókkal, hogy ψ_i akkor és csak akkor igaz a változók valamely kiértékelése mellett, ha a változók értékei olyan a, b konfigurációkat kódolnak, hogy ÚT(a, b, i) teljesül.

Így $x \in L \iff \psi_{n^k}(A, B)$ igaz, amikor A helyébe a kezdő konfigurációt, B helyébe az elfogadó konfigurációt kódoló kiértékelést helyettesítjük. A formulák logaritmikus tárral elkészíthetők lesznek.

$\psi_0(A, B)$: azt fejezi ki, hogy minden i-re $a_i = b_i$ vagy a B konfiguráció egy lépésben következik A-ból. Megadható $\mathcal{O}(n^k)$ hosszú formulával.

$$\psi_{i+1}(A, B) : \exists Z[\psi_i(A, Z) \wedge \psi_i(Z, B)].$$

A formulák exponenciálisan hosszúak lesznek!

$\psi_{\mathbf{i+1}}(\mathbf{A}, \mathbf{B})$:
$\exists Z \forall X \forall Y[((X = A \wedge Y = Z) \vee (X = Z \wedge Y = B)) \Rightarrow \psi_i(X, Y)]$.

ψ_i kvantorai előre hozhatóak. A mag diszjunktív normálformája:
ψ_i magjának normálformája $+$ $\mathcal{O}(n^k)$ hosszú tag.

Így ψ_{i+1} mérete: $\psi_i(A, B)$ mérete $+\mathcal{O}(n^k)$.

Most egy összefüggést igazolunk az alternáló polinom idő és a polinomiális tár között. Már definiáltuk az **AP** = $\mathrm{ATIME}(n^k)$ osztályt.

Tétel. A QSAT probléma **AP**-teljes.

Biz. A QSAT $\in$ **AP** bizonyításához: egy alternáló Turing-gép egymás után megsejti az $x_1, x_2, \ldots$ változók értékét, az egzisztenciálisan kvantifikált változókét K_{VAGY} állapotban, az univerzálisan kvantifikált változókét $K_{\text{ÉS}}$ állapotban. Végül ellenőrzi, hogy a megtalált kiértékelés kielégíti-e a formula magját.
Időigény: polinomiális.

Teljesség: A Cook-tétel bizonyításának változata. A nemdet. választásokhoz tartozó változókat lekötjük $\exists$ vagy $\forall$ kvantorral aszerint, hogy az állapot a K_{VAGY} vagy $K_{\text{ÉS}}$ halmazba esik-e. Feltehető, hogy csak alternálnak – minden második azonosan kvantifikált.

Köv. **AP = PSPACE**

Biz. Mindkét osztály zárt a visszavezetésre és QSAT mindkettőben teljes.

A QSAT felfogható **kétszemélyes játékként**:

- Játékosok: $\exists$, $\forall$
- Lépések felváltva: x_1 értéke, x_2 értéke, . . .
- $\exists$ célja: kielégíteni a formula magját
- $\forall$ célja: hamissá tenni

FÖLDRAJZI JÁTÉK**:**

Adott: $G = (V, E)$ irányított gráf, 1 kezdőcsúcs.

Kérdés: Az I. játékos nyer-e az alábbi játékban?

- Az I. játékos kezd az 1 csúcs megnevezésével.
- Minden lépésben minden játékosnak egy olyan csúcsot kell választani, amely még nem szerepelt, és amelybe vezet él az előzőleg megnevezett csúcsból.
- Egy játékos veszít, ha végül nem tud ilyen csúcsot megnevezni.

Áll. A FÖLDRAJZI JÁTÉK probléma **PSPACE**-ben van.

Biz.

- A lépéssorozatok hossza a bemenet méretével polinomiális.
- Létezik olyan polinom tárigényű algoritmus, mely adott állásra megkonstruálja az állás rákövetkezőit, vagy ha ilyen nincs, eldönti, melyik játékos nyert.

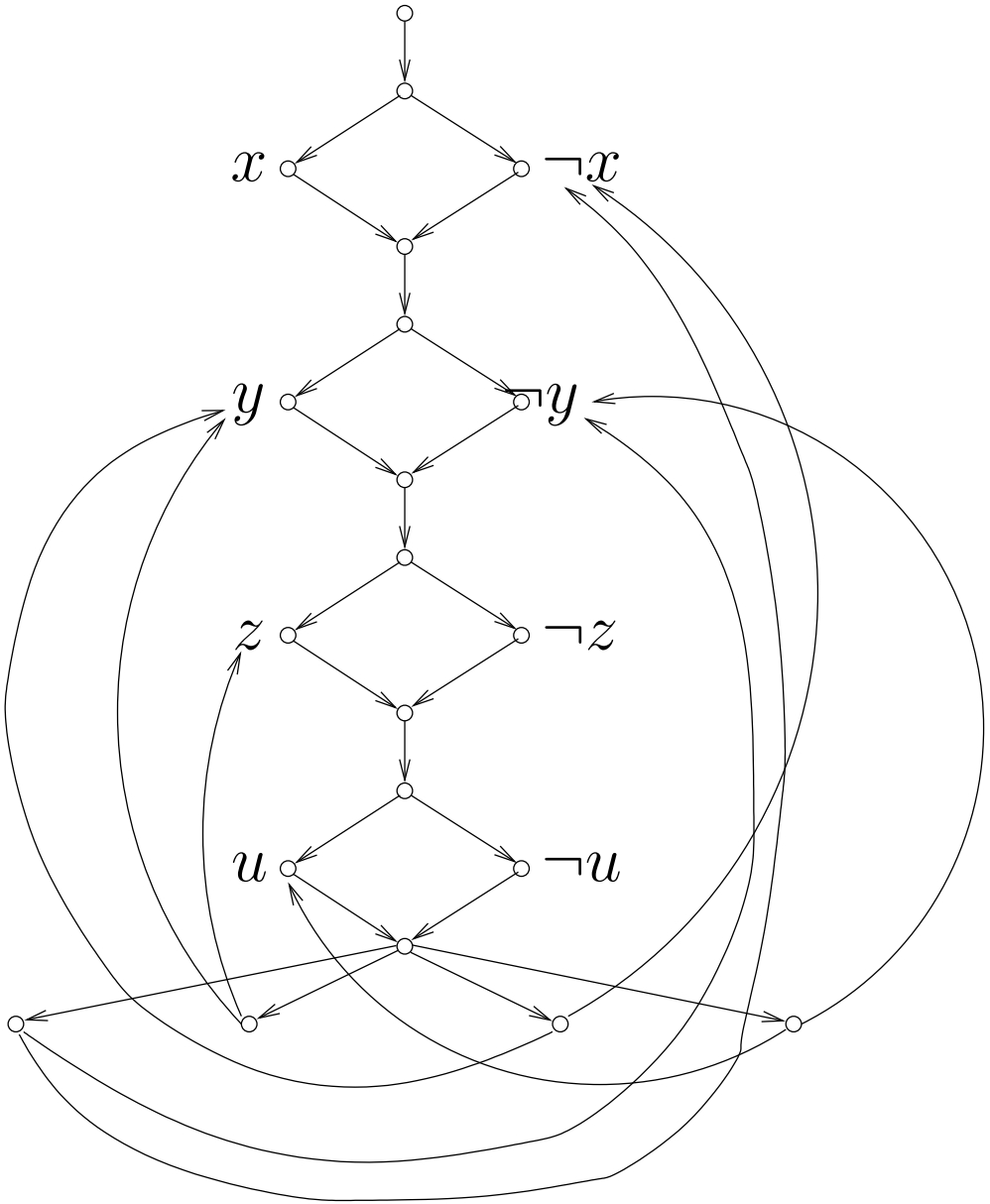
Minden ilyen kétszemélyes játék **PSPACE**-ben van:

- Polinom tárral elkészítjük az adott játék fáját.
- Mélységgel arányos tárral eldöntjük, kinek van nyerő stratégiája.

A két algoritmus összetehető: polinom tárkorlát.

Áll. A FÖLDRAJZI JÁTÉK probléma **PSPACE**-teljes.

$$\exists x \forall y \exists z \forall u (\neg x \vee \neg y) \wedge (y \vee z) \wedge (y \vee \neg x) \wedge (\neg y \vee u)$$



- Az első játékos dönti el az egzisztenciálisan, a második az univerzálisan kvantifikált változó értékét, mely a címkével ellentétes.
- A második játékos választ egy tagot.
- Az első játékos akkor és csak akkor nyer, ha a tag valamely literálja igaz (visszavezető él választható)

Mivel ez minden tagra igaz, az első játékos nyer $\Longleftrightarrow$ a formula igaz.

HELYBEN ELFOGADÁS**:**

Adott: M det. Turing-gép és x bemenő szó.

Kérdés: Elfogadja-e M az x-et úgy, hogy szava valamikor is hosszabb lenne, mint $|x| + 1$?

Tétel. HELYBEN ELFOGADÁS**PSPACE**-teljes.

REGULÁRIS KIFEJEZÉSEK EKVIVALENCIÁJA :

Adott: Két reguláris kifejezés

Kérdés: Ugyanazt a nyelvet jelölik-e?

Tétel. A REGULÁRIS KIFEJEZÉSEK EKVIVALENCIÁJA**PSPACE**-teljes.

VÉGES AUTOMATÁK EKVIVALENCIÁJA:

Adott: M_1 és M_2 véges nemdeterminisztikus automaták

Kérdés: M_1 és M_2 ekvivalensek-e?

Tétel. A VÉGES AUTOMATÁK EKVIVALENCIÁJA **PSPACE**-teljes.

A probléma determinisztikus automatákra is **PSPACE**-teljes.

Túl a **PSPACE** osztályon

Def. **EXP** = $\mathrm{TIME}(2^{n^k})$

NEXP = $\mathrm{NTIME}(2^{n^k})$

Világos, hogy **EXP** $\subseteq$ **NEXP**, és tudjuk, hogy **PSPACE** $\subseteq$ **EXP**.

Nem ismert, hogy az **EXP** és **NEXP** osztályok megegyeznek-e.

Tétel. Ha **P** = **NP**, akkor **EXP** = **NEXP**.

Gráfok tömör reprezentációja:

- $G = (V, E)$ $\qquad V = \{0, 1, \ldots, 2^m - 1\}$
- Megadható egy $f(\overrightarrow{x}, \overrightarrow{y})$ $2m$ változós Boole-függvénnyel.
- Az f megadható egy $2m$ bemeneti kapuval rendelkező hálózattal.

Hálózatok hasonló módszerrel megadható tömören.

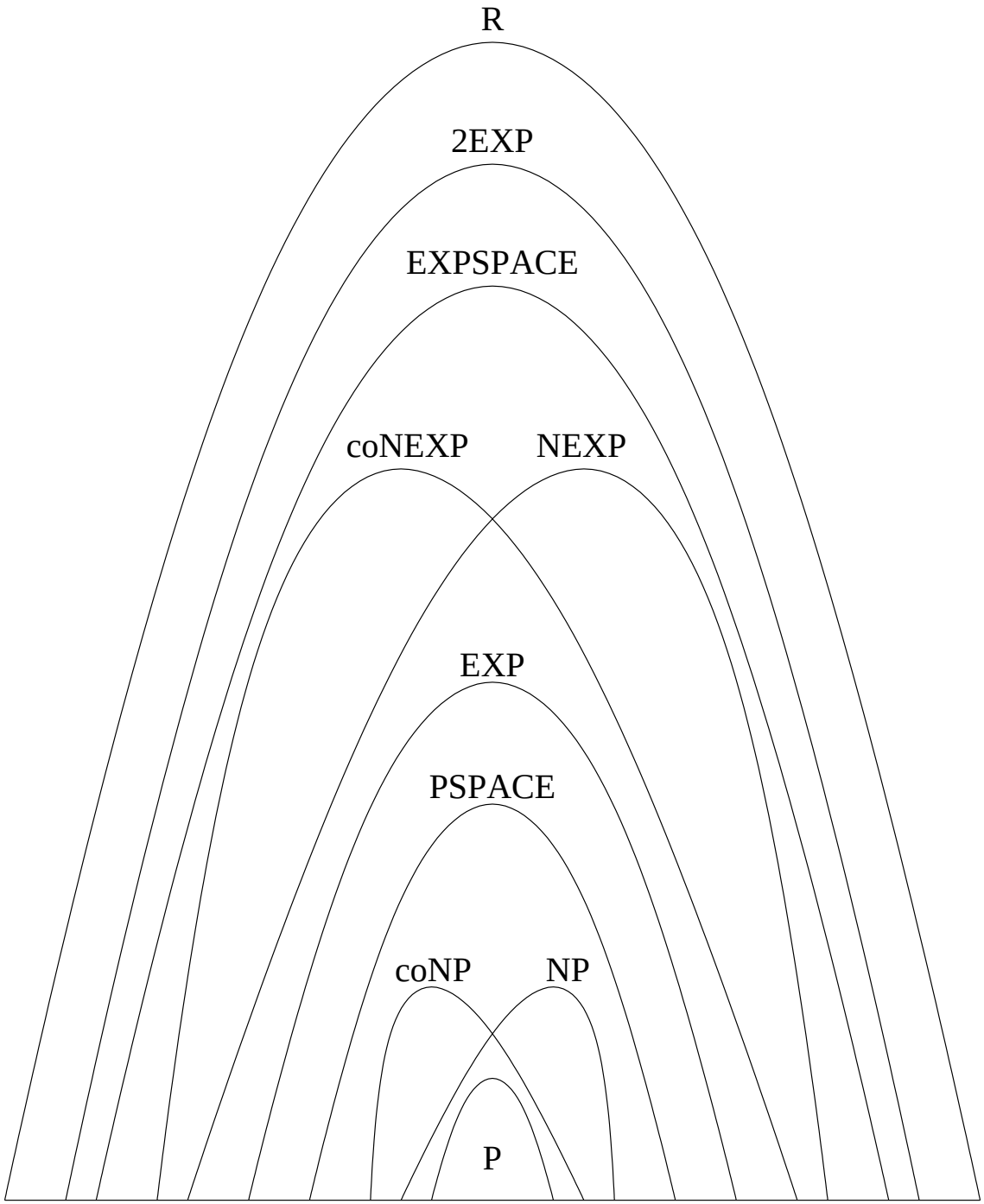
Tétel. A TÖMÖR HAMILTON-ÚT és TÖMÖR HÁLÓZAT-KIELÉGÍTHETŐSÉG problémák **NEXP**-teljesek.

Tétel. A TÖMÖR HÁLÓZAT-KIÉRTÉKELÉS probléma **EXP** teljes.

Def. **EXPSPACE** = $\mathrm{SPACE}(2^{n^k})$

Világos, hogy **NEXP** $\subseteq$ **EXPSPACE**.

Az alábbi probléma **EXPSPACE**-teljes: Adott két reguláris kifejezés, melyekben négyzetreemelés is szerepelhet, ekvivalensek-e a kifejezések?


R
2EXP
EXPSPACE
coNEXP
NEXP
EXP
PSPACE
coNP
NP
P

Randomizált számítások

PÁROSÍTÁS :

Adott: $G = (U, V, E)$ páros gráf, ahol

$$U = \{ u_1, \ldots, u_n \}, \quad V = \{ v_1, \ldots, v_n \}, \quad E \subseteq U \times V$$

Kérdés:

Létezik-e teljes párosítás, azaz olyan $M \subseteq E$ halmaz, hogy

$$\forall\, u_i \; \exists!\, v_j \quad (u_i, v_j) \in M$$
$$\forall\, v_j \; \exists!\, u_i \quad (u_i, v_j) \in M$$

Szimbolikus determináns

Legyen A^G az az $n \times n$-es mátrix, melynek (i,j)-dik eleme

- az x_{ij} változó, ha $(u_i, v_j) \in E$,
- 0 különben.

Áll. A $\det(A^G)$ **szimbolikus determináns** akkor és csakis akkor nem nulla, ha létezik teljes párosítás.

Biz.

- $\det(A^G) = \sum_{\Pi} \sigma(\pi) \prod_{i=1}^{n} A^G_{i,\pi(i)}$.
- $\forall \pi : \prod_{i=1}^{n} A^G_{i,\pi(i)} \neq 0 \Leftrightarrow \pi$ teljes párosítás.
- $\pi_1 \neq \pi_2, \neq 0 \Rightarrow \prod_{i=1}^{n} A^G_{i,\pi_1(i)}$ tartalmaz olyan változót, mely nem fordul elő $\prod_{i=1}^{n} A^G_{i,\pi_2(i)}$-ben.

Lemma. Legyen $\pi(x_1, \ldots, x_m)$ adott m-változós, nem azonosan nulla, minden változójában legfeljebb d-ed fokú polinom, és legyen $M > 0$. Ekkor azon $(x_1, \ldots, x_m) \in \{\, 0, 1, \ldots, M-1 \,\}^m$ rendezett m-esek száma, melyekre $\pi(x_1, \ldots, x_m) = 0$, legfeljebb mdM^{m-1}.

Biz. m szerinti teljes indukcióval.

m = 1: Legfeljebb d zérushely.

m > 1: Legyen $\pi(x_1, \ldots, x_m) =$
$p_0(x_1, \ldots, x_{m-1})x_m^r + p_1(x_1, \ldots, x_{m-1})x_m^{r-1} + \ldots + p_r(x_1, \ldots, x_{m-1})$.

Tfh $(x_1, \ldots, x_m) \in \{\, 0, 1, \ldots, M-1 \,\}^m$, melyre $\pi(x_1, \ldots, x_m) = 0$.

(a) $p_0(x_1, \ldots, x_{m-1}) = 0 \rightarrow (m-1)dM^{m-2} \cdot M$

(b) $p_0(x_1, \ldots, x_{m-1}) \neq 0 \rightarrow dM^{m-1}$

Összesen: mdM^{m-1}

Köv. Az előző feltételek esetén, ha $M = 2m$ és $d = 1$, akkor azon $(x_1, \ldots, x_m) \in \{\, 0, 1, \ldots, M-1 \,\}^m$ rendezett m-esek száma, melyekre $\pi(x_1, \ldots, x_m) = 0$, **legfeljebb** $\frac{M^m}{2}$, azaz az összes m-esek **fele**.

Randomizált algoritmus PÁROSÍTÁS-ra

(1) Válasszunk véletlenszerűen $i_1, \ldots, i_m$ egész számokat a $\{\, 0, 1, \ldots, 2m-1 \,\}$ halmazból, ahol m az élek száma.

(2) Számítsuk ki Gauss eliminációval a $\det(A^G(i_1, \ldots, i_m))$ értéket.

(3) Ha ez nem 0, akkor a válasz **„igen”**: létezik párosítás.

(4) Különben a válasz **„nem”**: G-ben (valószínűleg) nem létezik teljes párosítás.

Megj.

- „Igen” válasz sohasem téves.
- „Nem” válasz esetén a tévedés valószínűsége $\leq 1/2$
- Így k-szori megismétlés esetén „nem” válasz esetén a tévedés valószínűsége $\leq 1/2^k$.
- Polinom időigény.

A „Fermat próba”

Tétel. Tfh p prímszám és $0 < a < p$ egész. Ekkor

$$a^{p-1} \equiv 1 \bmod p$$

Hipotézis. Ha $n > 1$ nem prímszám, akkor a nemnulla a maradékok legalább felére igaz, hogy

$$a^{n-1} \not\equiv 1 \bmod n$$

Adódna egy randomizált polinomidejű algoritmus az

$$\text{ÖSSZETETT-SZÁM} = \text{co-PRÍMEK}$$

problémára.

Adott $n > 2$ esetén:

(1) Válasszunk ki egy $a \neq 0$ véletlen maradékot.

(2) Ha $a^{n-1} \not\equiv 1 \bmod n$ akkor n **összetett szám**.

(3) Ellenkező esetben n **valószínűleg prím**.

Téves (negatív) válasz valószínűsége $\leq 1/2$, k-szori ismétlés után pedig legfeljebb $1/2^k$.

Da a HIPOTÉZIS **nem igaz …**

Def. Legyen p páratlan prímszám, az a pedig p nemnulla maradéka. Ekkor az

$$(a \mid p) = a^{(p-1)/2} \bmod p$$

kifejezést az a és p **Legendre-szimbolumának** nevezzük.

Áll. $(a \mid p) \in \{ p-1, 1 \}$ $\qquad$ $((a \mid p) \in \{ -1, 1 \})$

Def. Legyenek M, N relatív prímek, N páratlan. Legyen $N = q_1 \dots q_k$, ahol minden q_i prím. Ekkor

$$(M \mid N) = \prod_{i=1}^{k} (M \mid q_i).$$

Tétel. Ha N páratlan összetett szám, akkor az N-nel relatív prím $M < N$ számok legalább felére teljesül, hogy

$$(M \mid N) \neq M^{(N-1)/2} \bmod N.$$

Az algoritmus

Adott $N \geq 1$ páratlan egész

- Generáljunk 1 és $N-1$ között egy véletlen egész számot, legyen ez M.
- Számoljuk ki (M, N) értékét.
- Ha $(M, N) > 1$, akkor a válasz **„igen”**: N összetett szám.
- Ellenkező esetben számítsuk ki $(M \mid N)$ és $M^{(N-1)/2}$ értékét, majd hasonlítsuk össze a két eredményt.
  - Ha nem egyeznek meg, a válasz ismét **„igen”**: N összetett szám.
  - Ellenkező esetben a válasz **„nem”**: N valószínűleg prím.

Def. Legyen L nyelv. L-hez tartozó polinom idejű, **Monte-Carlo típusú Turing-gép** olyan pontos, nemdeterminisztikus, $p(n)$ polinom idejű Turing-gép, melynek minden nem végső konfigurációjában pontosan két nemdeterminisztikus választása van, és amelyekre teljesül:

- $x \in L$ esetén a számítási sorozatok **legalább fele** „igen” válasszal végződik.
- $x \notin L$ esetén **minden** számítási sorozat „nem” válasszal végződik.

Def. **RP**: polinom idejű, MC-típusú Turing-géppel eldönthető nyelvek.

Áll. **P** $\subseteq$ **RP** $\subseteq$ **NP**.

Köv. **P** $\subseteq$ **coRP** $\subseteq$ **coNP**.

Def. **ZPP** $=$ **RP** $\cap$ **coRP**

Polinom idejű, Las-Vegas algoritmussal (Turing-géppel) eldönthető nyelvek, ill. problémák.
Ha egy ilyen algoritmust k-szor függetlenül végrehajtunk, annak valószínűsége, hogy nem kapunk **végleges választ** legfeljebb $1/2^k$.