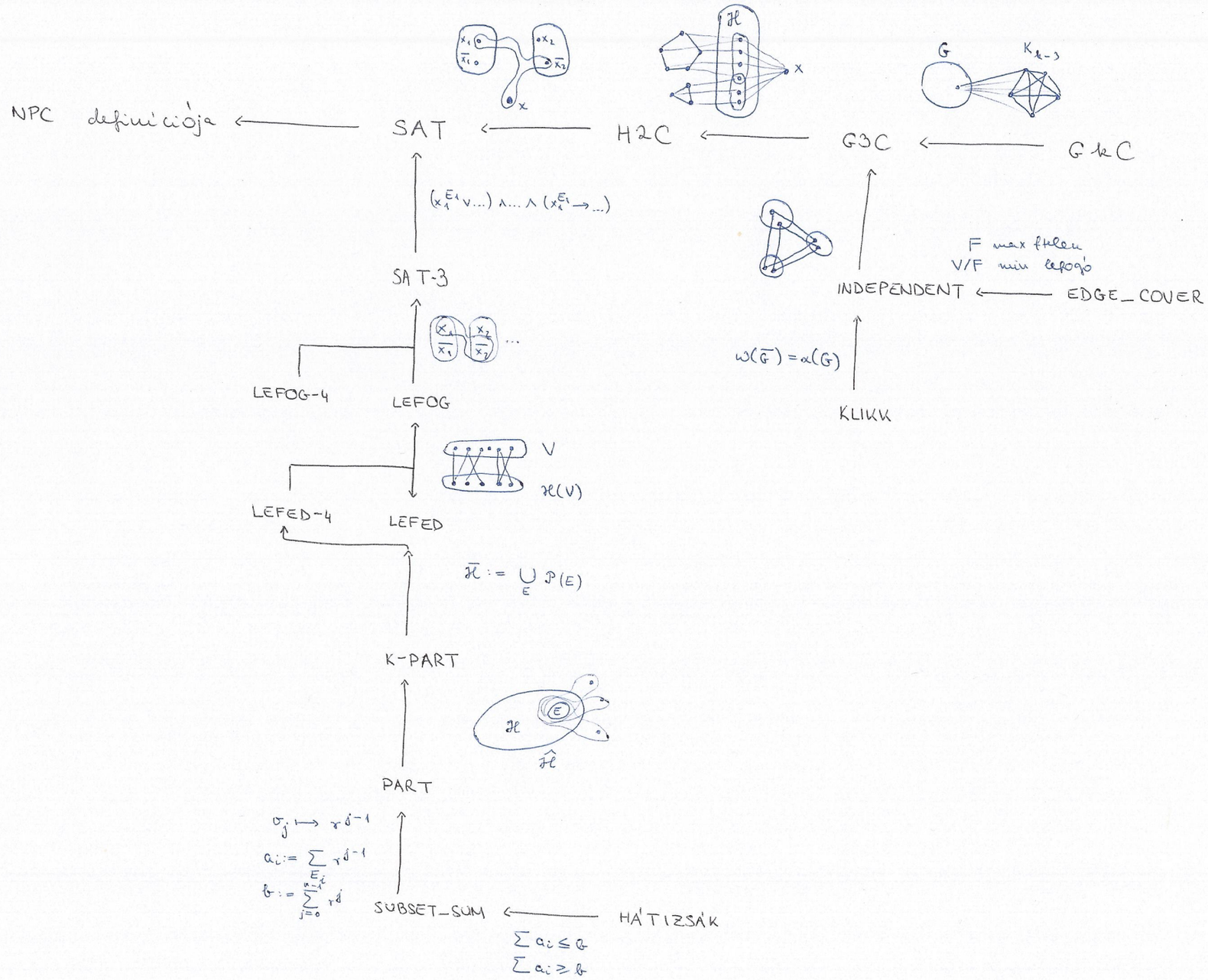

NPC definíciója
SAT
H2C
G3C
GkC
$(x_1^{E_1} \vee \dots) \wedge \dots \wedge (x_k^{E_1} \to \dots)$
SAT-3
LEFOG-4
LEFOG
LEFED-4
LEFED
V
$\mathcal{H}(V)$
$\bar{\mathcal{H}} := \bigcup_E \mathcal{P}(E)$
K-PART
PART
$\hat{\mathcal{H}}$
$\sigma_j \longmapsto r^{j-1}$
$a_i := \sum_{E_i} r^{j-1}$
$b := \sum_{j=0}^{n-1} r^j$
SUBSET_SUM
HÁTIZSÁK
$\sum a_i \le b$
$\sum a_i \ge b$
F max fokú
V/F min lefogó
INDEPENDENT
EDGE_COVER
$\omega(\bar{G}) = \alpha(G)$
KLIKK

NP: $\forall w \in L\ \exists y$ tanú: det. T-gép (w,y)-t elfogadja P időben, $|y| \le |w|^k$

coNP: L komplementere NP-beli

BPP: T random. Turing-gép, Pól. időben lefut, $x \in L$: $P(\text{elfogadás}) \ge \frac{2}{3}$, $x \notin L$: $P(\text{elutasítás}) \ge \frac{2}{3}$

PP: poly. időben lefut, $P(\text{elf.}) > \frac{1}{2}$, $P(\text{elut.}) \ge \frac{1}{2}$

RP: poly. időben lefut, $P(\text{elf.}) > \frac{1}{2}$, $P(\text{elut.}) = 1$

coRP: L komplementere RP-beli

ZPP: várhatóan poly. időben lefut, $P(\text{elf.}) = 1$, $P(\text{elut.}) = 1$

$$
\begin{array}{ccccccccc}
 & & \text{coRP} & \subseteq & \text{coNP} & & & & \\
 & & \cup\!\!\!\!= & & \cap & & & & \\
P & \subseteq & \text{ZPP} & & \text{BPP} & \subseteq & \text{PP} & \subseteq & \text{PSPACE} \\
 & & \cap & & \cup\!\!\!\!= & & \cup\!\!\!\!= & & \\
 & & \text{RP} & \subseteq & \text{NP} & & & &
\end{array}
$$

BPP $\iff$ csak 1 állapotban van véletlen
$\iff$ van egy véletlen szalag
$\iff$ p valószínű pénzérmét dobálunk (p rép)
$\iff$ $\frac{2}{3}$ helyett $\frac{1}{2} + n^{-c} < f(n) < 1 - 2^{-n^d}$, (n az input hossza)

ZPP $\iff$ $\frac{1}{2}$ valószínűséggel helyes, $\frac{1}{2}$ valószínűséggel nem tudom.
min. max.

- SAT
- 3-SAT minden klóz legfeljebb 3 elemű
- SAT-3 minden változó max 3 literálban fordul elő
- H2C $\mathcal{H}$ k színnel színezhető, ha $\forall$ hipergráfhoz létezik
- G3C, GkC $(k \ge 3)$
- INDEPENDENT $= \{(G,k) \mid \alpha(G) \ge k\}$ a max free csúcshalmaz mérete
- EDGE_COVER $= \{(G,k) \mid G$ élei lefedhetők k csúccsal$\}$
- KLIKK $= \{(G,k) \mid$ van k elemű klikk$\}$ w kérdésben
- LEFOG $= \{(\mathcal{H},k) \mid \mathcal{H}$ hipergráf lefogható k elemű csúcshalmazzal$\}$
- LEFED $= \{(\mathcal{H},k) \mid \mathcal{H}$ hipergráf csúcsai fedhetők k hiperéllel$\}$
- LEFOG olyan $\mathcal{H}$-ra, amely minden csúcs legfeljebb 4 élben van benne.
- LEFED, ha $\forall$ él elemszáma max. 4
- K-PART $= \{(\mathcal{H},\ell) \mid$ van ℓ éle $\mathcal{H}$-nak, amik particionálják a csúcshalmazt$\}$
- PART $= \{\mathcal{H} \mid$ kiválasztható $\mathcal{H}$-nak élei, amik particionálnak$\}$
- SUBSET_SUM $= \{(a_1,\dots,a_n,b) \mid \exists I \subseteq n: \sum_I a_i = b\}$
- HÁTIZSÁK $= \{(s_1,\dots,s_m,r_1,\dots,r_m,b,k) \mid \exists I \subseteq m: \sum_I s_i \le b,\ \sum_I r_i \ge k\}$
- HAMILTON
- van csúcsok fölött tartalmazó kör
- term. számok egy halmaza felosztható-e két egyenlő összegű részre.

SAT-2

2-SAT

INDEPENDENT$_k$

Véges automata $M = (Q, \Sigma, \delta, q_0, F)$

Q: állapotok, δ: átmeneti fv., F: végállapotok

$L(M)$: M által felismertek

L reguláris $\iff \exists M: L \in L(M)$. $\cap, \cup$, komplementerre zárt

Pumpálós lemma: $\forall L$ reg. $\exists n$ $\forall z \in L$, $|z| \geq n$

$\implies \exists u, v, w: z = uvw$, $|uv| \leq n$, $\forall i: uv^iw \in L$

Boole-hálózat: $\neg$ $\wedge$ $\vee$

mélység: leghosszabb út $\to$ log n-ben polinomiális

méret: kapuk száma $\to$ $O(n)$-ben polinomiális

bemenetek megadása

RAM-gép $X[i] \in \mathbb{Z}$ $i \in \mathbb{Z}$

$X[0]$ az input hossza

$X[1], X[2], \ldots, X[X[0]]$ input

$X[i] := 0, 1, -1$

$X[i] := X[i] \pm X[j]$

$X[i] := X[j]$

$X[i] := X[X[j]]$

$X[X[i]] := X[j]$

If $X[i] \leq 0$ GOTO (címke)

L rekurzív, ha $\exists T$: $\forall w$ inputon megáll, és $w \in L$-re 1-et ad, $w \notin L$-re 0-t.

f rekurzív ha $\exists T$. $\forall w$ inputon megáll, és $f(w)$-t ad.

L rek. felsorolható, ha $L = \emptyset$ vagy $\exists f$ rekurzív: $L = \mathrm{im}(f)$

L co-rek. fels., ha $\overline{L}$ rek. fels.

L rek. fels. $\iff \exists T$: pontosan az L-beliekre áll meg

L rekurzív $\iff$ L rek. fels. és co-rek. fels.

L rek., $L' \subseteq L$-re L' rekurzív $\iff$ L' rek. fels. és $L \setminus L'$ rek. fels.

Eldönthetetlen: T megáll-e az üres inputon.

n hosszú bemenetre futásidő $\leq \mathrm{time}_T(n)$,

tárhely $\leq \mathrm{space}_T(n)$

$f: \mathbb{N} \to \mathbb{N}$-re $\mathrm{DTIME}(f(n)) = \{L : T$ felismeri L-et és $\forall n: \mathrm{time}_T(n) \leq f(n)\}$

$$P = \bigcup_{\substack{i \in \mathbb{N} \\ c > 0}} \mathrm{DTIME}(c \cdot n^i)$$

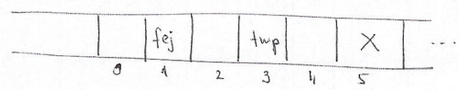
Tétel. $\forall T$ Turing-géphez $\exists R$ RAM-gép, hogy minden inputra T leáll $\Leftrightarrow$ R leáll és az output ugyanaz.

Ha T t lépést tesz meg, akkor R $O(t)$ lépést tesz meg és max. $O(\log t)$ bites számokat használ.

B: Tfh. $T = (1, \Sigma, \Gamma, \alpha, \beta, \gamma)$, ahol $\Sigma = \{0, 1, 2\}$, ahol 0 a $*$ jelnek felel meg (ez a számítást könnyíti meg).

Tfh. továbbá, hogy $\Gamma = \{1, \dots, r\}$, $1 = \text{START}$, $r = \text{STOP}$

Páros mezőben lesznek az adatok, $X[1]$ a fej helyét jelöli, $X[3]$ temporális mező.



A program P_i és Q_{ij} blokkokból áll, $i \in \Gamma$, $j \in \Sigma = \{0, 1, 2\}$

P_i: az i állapotban való olvasás spec. P_r legyen üres (egyetlen üres sor)

Q_{ij}: az i állapotban j-t olvasunk

$P_i :=$
```
X[3] := X[ X[1] ]          // olvasás az X[1] helyről
if X[3] ≤ 0 then goto Q_i0 // 0-t olvasunk
X[3] := X[3] - 1
if X[3] ≤ 0 then goto Q_i1 // 1-et olvasunk
X[3] := X[3] - 1
if X[3] ≤ 0 then goto Q_i2 // 2-t olvasunk
```

$Q_{ij} :=$
```
X[3] := 0                  // ez fogja tartalmazni β(i,j)-t
X[3] := X[3] + 1 }
      ⋮          } β(i,j) darab sor
X[3] := X[3] + 1 }
X[X[1]] := X[3]            // β(i,j) beírása
X[1] := X[1] + γ(i,j)
X[1] := X[1] + γ(i,j)      // azért kell kétszer, mert a ps alatt páros dolgozunk
X[3] := 0
if X[3] ≤ 0 then goto P_α(i,j)   // biztos utasítás
```

Tehát a program:
```
X[1] := 0
P_1
⋮
P_r
Q_0,0
⋮
Q_r-1,2
```

Ez nyilván minimális, a blokkok hossza konstanssal becsülhető $\Rightarrow$ $O(t)$ lépést tesz R.

Ha T nem hagyja el a $-N, N$ tartományt, akkor R-ben legfeljebb $O(\log N)$ méretű számok vannak, $t \geq N$. $\square$

Tétel. Minden R RAM-gép-programhoz $\exists$ T Turing-gép, hogy R megáll $\Leftrightarrow$ T megáll, az output ugyanaz megálláskor, és ha R futásideje t, akkor T lépésszáma $O(t^2)$.

B: Minden sor egy egy szalagon Turing-gépet.

az előző sor outputszalagja lesz a következő sor inputszalagja.

A RAM-gép műveletei kötül a címzés a legbonyolultabb (összeadni írásban kell, minden más könnyű).

Négyszalagos Turing-gépet csinálunk, az egyik szalagon a memória, a másik hámon a rész-T-gépek elférnek (pusztán lehet sokkal nagyvonalúbban is)

Sohsa nem törlünk csak a szalagon, mindig a végére logoljuk a változtatást.

	*	*	a		*		i		*	*	b		*	j		*	*		...

Kiolvasáskor a végéről kell visszafelé menni. $\longrightarrow$ $X[i] := a$, $X[j] := b$.

A címzés így $O(t)$ lépés, minden más se bonyolultabb $\Rightarrow t \cdot O(t) = O(t^2)$ elég. □

Tétel. Az L nyelv rfs $\Leftrightarrow$ $\exists$ T Turing-gép, hogy T Σ_0^* elemei közül csak L-ne áll le.

B: Ha L rfs: $L = \emptyset$-ra triv: csak egy ∞ ciklus kell.

$L \neq \emptyset$-ra $\exists f$ rek. fv., $\operatorname{Im} f = L$, T' kiszámolja f-et

$\hat{T}$ sorban állítja elő Σ_0^* elemeit, T' eszközt hozzáfér,

Ha y az input, akkor elindítjuk $\hat{T}$-ot és a kimenetén T'-t, és nézzük, hogy T' outputja y-e: ha igen, leállunk, ha nem, megyünk tovább.

Ha $\exists$ T: $L = \emptyset$-ra triv:

$\hat{T}$ által előállított j-edik szó legyen w_j.

$\tilde{T}$: a $w_{j-\lfloor\sqrt{j}\rfloor^2}$ szót futtatja T-t j lépésen át, ha valahol leáll, akkor kiírjuk, különben írunk *-ot.

$j - \lfloor\sqrt{j}\rfloor^2$ minden $\mathbb{N}$-beli számot ∞ sokszor előállít

$\Rightarrow$ ha egy w szót a T leáll, akkor $\tilde{T}$ azt is elég időt nézi

$\Rightarrow$ $\tilde{T}$ előállítja L-et. outputként. □

Tétel. Legyen T 2-szalagos univ. T-gép. $L_T := \{w \in \Sigma_0^* \mid T$ leáll, ha mindkét szalagján w az input$\}$

$\Rightarrow L_T$ rek. fels., de nem rekurzív.

B: 1) T' a w inputtal fusson úgy, hogy T-t futtatja (w, w) inputtal

$\Rightarrow$ T' pontosan L_T-n áll le $\Rightarrow L_T$ rfs.

2) L_T rekurzív. $\Rightarrow \Sigma_0^* \setminus L_T$ rekurzív. $\Rightarrow \Sigma_0^* \setminus L_T$ rfs. $\Rightarrow \exists$ T': T' $\Sigma_0^* \setminus L_T$-n áll meg

$\Rightarrow \exists$ T'' 1-szalagos, ami $\Sigma_0^* \setminus L_T$-n áll meg (egyszalagos utánzás)

Legyen T'' programja p. a T gépen.

Ekkor $p \in L_T \Leftrightarrow$ T megáll (p, p) inputtal $\overset{\text{univerzalitás}}{\Leftrightarrow}$ T'' megáll a p inputtal $\Leftrightarrow$

$\underset{\text{defi.}}{\Leftrightarrow}$ T' megáll a p inputtal $\underset{\text{defi}}{\Leftrightarrow}$ $p \in \Sigma_0^* \setminus L_T$ ↯ □

Köv. $\Sigma_0^* \setminus L_T$ nem rfs. (valójából csak ennyi az indirekt feltevésből elég). □

Ha $\forall$ Turing-gép $\forall$ inputjára tudnánk, hogy leáll-e $\Rightarrow L_T$ rek lenne.

Schwartz-lemma. f n-változós, d fokú, $f \not\equiv 0$. Ha $\xi_1, \dots, \xi_n \in \{1, \dots, N\}$ független egyenletes eo., akkor $\mathbb{P}(f(\xi_1, \dots, \xi_n) = 0) \leq \frac{n \cdot d}{N}$.

B: $n = 1$-re összesen legfeljebb d gyök van $\Rightarrow$ az eltalálásának valósége $\leq \frac{d}{N}$.

Teljes indukció n-re.

$$f(x_1, \dots, x_n) = x_1^0 \cdot f_0(x_2, \dots, x_n) + x_1^1 f_1(x_2, \dots, x_n) + \dots + x_1^l f_l(x_2, \dots, x_n)$$

$l \leq d$, $f_l \not\equiv 0$, ftht. $l \geq 1$.

$\xi = (\xi_1, \dots, \xi_n)$, $\xi^* = (\xi_2, \dots, \xi_n)$

$$\mathbb{P}(f(\xi) = 0) = \underbrace{\mathbb{P}(f(\xi) = 0 \mid f_l(\xi^*) = 0)}_{\leq 1} \cdot \underbrace{\mathbb{P}(f_l(\xi^*) = 0)}_{\leq \frac{(n-1) \cdot d}{N}} + \underbrace{\mathbb{P}(f(\xi) = 0 \mid f_l(\xi^*) \neq 0)}_{\leq \frac{d}{N}} \cdot \underbrace{\mathbb{P}(f_l(\xi^*) \neq 0)}_{\leq 1} \leq$$

indukció

az $n=1$ eset miatt. ha ξ^* nőpitett, akkor az x_1-ben d fokú

$$\leq \frac{(n-1)d}{N} + \frac{d}{N} = \frac{n \cdot d}{N}$$

$N := 2n \cdot d \longrightarrow \mathbb{P}(f(\xi) = 0) \leq \frac{1}{2}$ $\square$

Teljes párosítás létezése

$\{v_1, \dots, v_n\}$ és $\{w_1, \dots, w_n\}$ szinonimapár, ha $\{w_j, v_i\} \in E$, akkor $a_{ij} := x_{ij}$, különben 0, $A := (a_{ij})_{ij}$. $\det A = \sum_{\pi} (\pm a_{1\pi(1)} \cdots a_{n\pi(n)})$ polinom.

Egy tag nem $0 \iff \{v_1 w_{\pi(1)}, \dots, v_n w_{\pi(n)}\}$ t.párosítás

Ha nincs t.p. $\Rightarrow$ $\det A = 0$. Ha van $\Rightarrow$ van nemnulla tag, ami nem eshet ki, mert bármely két tagban van különböző változó. $\square$

BitCoin

Nem lerövidíthető, nincs hozzá központi hatóság (mint pl. nemzeti bank)

$f: A \twoheadrightarrow B$ nagyfokú könnyen számolható, f^{-1}-et nehéz. Bányászat: B-beli sok 0-t tartalmazó elem ősét keresni.

Peer-to-peer rendszer csomópontjaiban tárolt adatbázis (mindenkinél megvan (pár 100 MiB)) arról, hogy egy BTC-t ki bányászott és mikor kinek adta tovább. Nem embereket rögzít, hanem címeket. (Egy embernek több is lehet.) Csak véges sok BTC létezhet, de osztható.

Ki, hogy valaki tényleg akar-e fizetni egy következőt, a nyilvános-privátkulcs igazolással ellenőrzi. A tranzakciókat blokkokban tárolják. Ezek írása nem könnyű (és az írásáért is fizetnek): egy ösztönzésre van itt is, de könnyebb, mint a BTC-bányászat. Emiatt nem ennek többszörösét eladni, amelyik tranzakció előtt rögzítve van, az fog megváltozni. A legrosszabb blockchain az évvégés.

Tétel. $\forall f(n) \geq n$ rek. függvény $\exists L$ nyelv, hogy L rekurzív és $L \notin DTIME(f(n))$

B: Legyen T 2-szalagos univ. T-gép, $L_T := \{w \mid (w,w)$ inputon T leáll $\leq f(|w|)^3$ lépésben$\}$

L_T rekurzív: a $\hat{T}$ gép kiszámítja χ_{L_T}-t, ahol $\hat{T}$ w inputra $f(|w|^3)^3$-öt kiszámítja (ezt lehet, mert f rek.), és futtatja T-t (w,w) inputon ennyi lépésig. Ha leáll, output 1, kül. 0.

↯ $L \in DTIME(f(n))$.

$\Rightarrow \exists T_1$ k-szalagos Turing-gép, ami az L-beli tagságot $f(n)$ időben eldönti

k szalagosról egyszalagos áttérés (T_2) $O(f(n)^2) = c \cdot f(n)^2$ időben való elvégzésével.

Módosítás (T_3): ha $w \notin L_T$, akkor álljon le T_3 $c \cdot f(n)^2$ lépésben (mint T_2), de ha $w \in L_T$, akkor fusson a végtelenségig.

$P :=$ T_3 programja T-n.

↯ Ha $p \in L_T \iff T$ leáll (p,p)-n $f(|p|)^3$ időben $\iff T_3$ ~~nem~~ leáll le p-n. $f(|p|^3)^3$ időben.

def. szimuláció

De $p \in L_T$-re T_3 nem áll le. ↯

↯ Ha $p \notin L_T \iff T_3$ leáll p-n $f(|p|)^2$ időben $\iff T$ leáll (p,p)-n $f(|p|)^3$ időben

$\Downarrow$

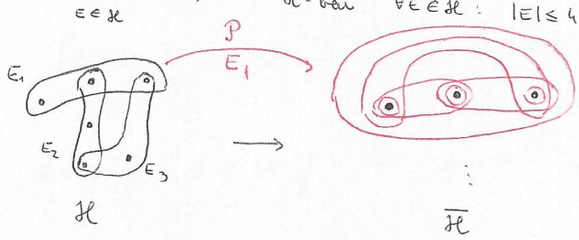
$p \in L_T$ ↯. ↯ □

Tétel. $K\text{-PART} = \{(\mathcal{H}, \ell) \mid$ van ℓ particionáló ugyanele $\mathcal{H}$-nak$\} \in NPC$.

B: A tanú az ℓ db él. Poli. ellenőrzés az incidenciamátrixban: azt kell ellenőrizni, hogy ezen élek oszlopaiban $\forall$ sorban pontosan 1 db 1-es van.

$LEFED4 \propto K\text{-PART}$: $(\mathcal{H}, \ell) \in LEFED4 \iff (\bar{\mathcal{H}}, \ell) \in K\text{-PART}$, ahol

$\bar{\mathcal{H}} := \bigcup_{E \in \mathcal{H}} \mathcal{P}(E)$ $\mathcal{H}$-ban $\forall E \in \mathcal{H}$: $|E| \leq 4 \Rightarrow |\mathcal{P}(E)| \leq 2^4 = 16 \Rightarrow \bar{\mathcal{H}}$ mérete poli. $\mathcal{H}$-ban, az előállítása is.



Ha $(\mathcal{H}, \ell) \in LEFED\text{-}4 \to E_{i_1}, \dots, E_{i_{\ell'}}$ fedner, $\ell' \leq \ell$

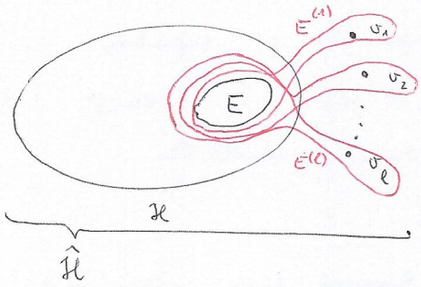
$\to$ alkalmazzuk a metszeteket elhagyva $E'_{i_1} \subset E_{i_1}, \dots, E'_{i_{\ell'}} \subset E_{i_{\ell'}}$, $(E'_{i_j})_j$ particionál $\bar{\mathcal{H}}$-ben. (Esetleg kell üreseket hozzávenni!)

Ha $(\bar{\mathcal{H}}, \ell) \in K\text{-PART} \to E'_{i_1}, \dots, E'_{i_\ell}$ mint az előbb, $\exists E_{i_1}, \dots, E_{i_\ell}$, hogy $E'_{i_1} \subset E_{i_1}, \dots, E'_{i_\ell} \subset E_{i_\ell} \to (E_{i_j})$ lefed $\mathcal{H}$-ben. (Esetleges üresek elhagyhatók.) □

Tétel. PART = $\{ \mathcal{H} \mid$ van partíciója $\} \in$ NPC

B: Tanú a partíció, ellenőrzés incidenciamátrixszal mint K-PART-nál.

K-PART $\propto$ PART: $(\mathcal{H}, \ell) \in$ K-PART $\iff \hat{\mathcal{H}} \in$ PART



Ha $(\mathcal{H}, \ell) \in$ K-PART $\to E_1, \dots, E_\ell$ partíció $\mathcal{H}$-ban $\to E_1^{(1)}, \dots, E_\ell^{(\ell)}$ partíció $\hat{\mathcal{H}}$-ban.

Ha $\hat{\mathcal{H}} \in$ PART $\to$ a v_i-k miatt a particionáló élek száma legfeljebb ℓ. Ezekből a bennük levő v_i-ket eldobva $\mathcal{H}$-beli legfeljebb ℓ méretű partíció $\to$ néhány $\emptyset$ hozzávételével ℓ elemű. $\square$

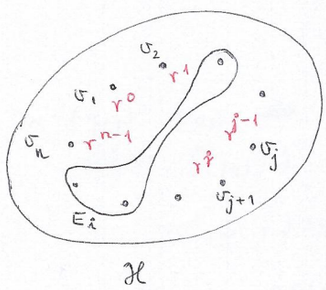
Tétel. SUBSET_SUM $= \{ (a_1, \dots, a_n, b) \mid \forall a_i, b \in \mathbb{N}, \ \exists I \subseteq \{1, 2, \dots, n\} : \sum_{i \in I} a_i = b \}$

B: Az I tanú, az ellenőrzés $\sum_I a_i$ kiszámítása és összehasonlítása b-vel.

PART $\propto$ SUBSET_SUM: $\mathcal{H} \in$ PART $\iff (a_1, \dots, a_n, b) \in$ SUBSET_SUM.

$m := |\mathcal{H}|, \quad n := |V(\mathcal{H})|$

$r := m + 1$



$\longrightarrow$

$\forall E_i \in \mathcal{H}: \quad a_i := \sum_{v_j \in E_i} r^{j-1}$

$b := \sum_{j=0}^{n-1} r^j$

Ha $\mathcal{H} \in$ PART $\to E_{i_1}, \dots, E_{i_\ell}$ particionál $\to a_{i_1} + \dots + a_{i_\ell} = \sum_{j=0}^{n-1} r^j = b$.

Ha $(a_1, \dots, a_n, b) \in$ SUBSET_SUM $\to b = \sum_{i \in I} a_i = \sum_{j=0}^{n-1} b_j r^j$

az r-es számrendszerbeli felírás egyértelműsége miatt $\forall b_j = 1$ $\square$

Tétel. HÁTIZSÁK $= \{ (s_1, \dots, s_m, r_1, \dots, r_m, b, k) \mid \exists I \subseteq \{1, \dots, m\} : \sum_{i \in I} s_i \le b, \ \sum_{i \in I} r_i \ge k \}$

B: Tanú I.

SUBSET_SUM $\propto$ HÁTIZSÁK: $a_i = s_i = r_i, \quad b = k \to \sum_I a_i \le b, \ \sum_I a_i \ge b$ $\square$

Tétel. $\forall k \in \mathbb{N}\ \forall \Sigma\ \exists\ k+1$ szalagos Σ feletti univerzális Turing-gép.

B: Legyen $S = (k, \Sigma, \Gamma_S, \alpha_S, \beta_S, \gamma_S)$ tetszőleges.

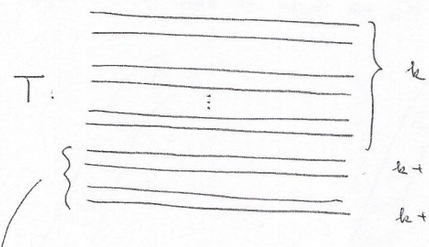
Γ_S elemei kódolhatók 0-1-sorozatokkal

S egy lehetséges kódja $h\ g\ \alpha_S(h,g)\ \beta_S(h,g)\ \gamma_S(h,g)$, ha $h \in \Sigma^k$, $g \in \Gamma_S$

$\uparrow$ a fejek által olvasott betűk

$P_S :=$ ezen kódok tetsz. sorrendű konkatenációja $\forall g\ \forall h$-ra.

$\longrightarrow$ véges szó



$k+1 \longrightarrow$ erre írjuk S aktuális állapotát (g_S) (ezek nyilvántartására nem használhatjuk Γ-t, mert az nem univerzális), kezdetben START

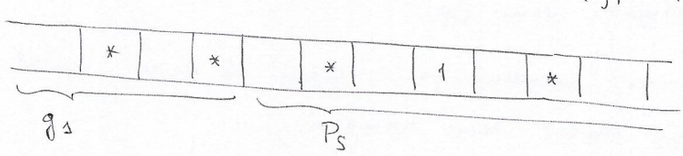
$k+2 \longrightarrow$ erre írjuk P_S-t.

A szimuláció lépései: T megnézi, hogy a $k+2$-edik szalagon hol van az $\underbrace{\text{első } k \text{ szalagról olvasott jelekre}}_{h}$ és a $k+1$-edik szalagon lévő g állapotnak megfelelő feljegyzés, és leolvasva utána, hogy $\alpha_S, \beta_S, \gamma_S$ mi volt. $\longrightarrow$ elvégzi a műveleteket.

Ez $k+2$ szalagos szimuláció.

A $k+1$-edik szalagon csak $\lceil \log |\Gamma_S| \rceil$ mezőt használunk.

$k+1$-edik és $k+2$-edik szalag összolvasztható: pozitív helyeken van P_S, negatív helyekre írjuk g_S-t. Ehhez 2 fej kell, de megoldható 1-gyel is: duplázás, $*$, ha nincs ott a fej, 1, ha ott van (közt. 1-esből 2 lesz)



$\square$

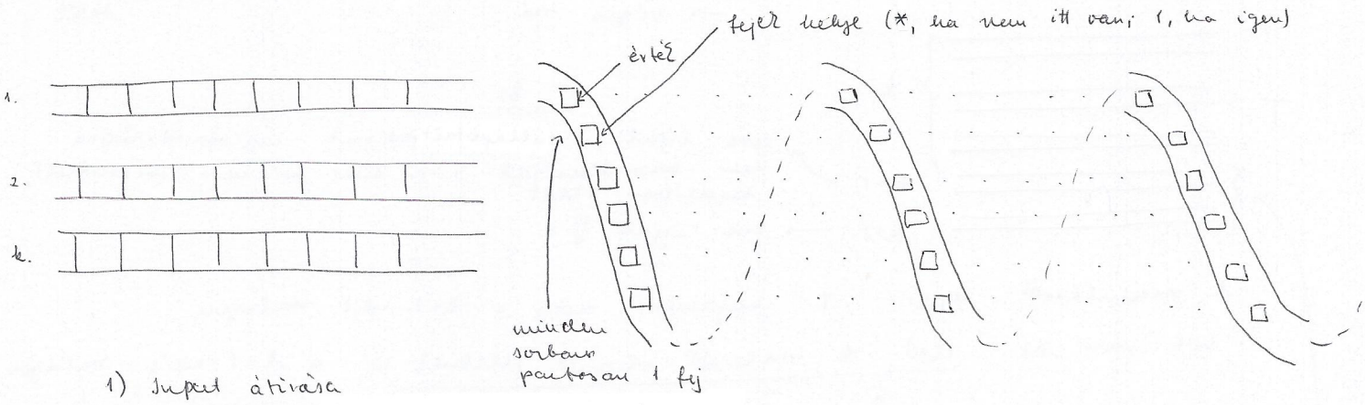
Tétel. $\forall S = (k, \Sigma, \Gamma_S, \alpha_S, \beta_S, \gamma_S)\ \exists T = (1, \Sigma, \Gamma, \alpha, \beta, \gamma)$:

① $\forall w \in \Sigma_0^*$: S megáll w-n $\Leftrightarrow$ T megáll w-n

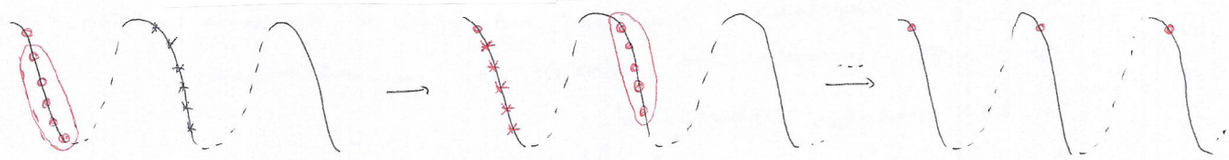
② megálláskor S és T utolsó szalagján ugyanaz van

③ ha S t lépésben áll le, akkor T $O(t^2)$ lépésben áll le.

B: S szalagjainak tartalmát kell 1 szalagra átírni



1) Input átírása



Majd fejek sorában az 1-eseket beírja a keresőpontokat.

2) Γ-t úgy csináljuk, hogy Γ_S-et számon tudjuk tartani

Tudunk mod k számolni (k fix), így azt is tudja T, hogy a fejek S melyik szalagján vannak épp.

Egy végigolvasás $\longrightarrow$ megállapítja, hogy S fejei miket olvasnának

$\Rightarrow$ kiszámítja, hogy miket kéne csinálni.

α a megfelelő módon lép

β felírásokat még egyszer végig kell menni.

γ átlépéseket még egyszer végig kell menni.

3) Output írása: az 1) lépés visszafelé.

Ha T összesen M mezőt lép, akkor $M = O(t)$.

1) és 3): $O(M^2) = O(t^2)$

2): $O(M)$ egy lépés $\Rightarrow$ $O(Mt) = O(t^2)$ $\Big\}$ $O(t^2)$ lépés elég

$\square$

OTP

A és B előre megállapodnak egy titkos r kulcsban. Az x üzenet kódolja $x \oplus r$, ebből a visszafejtés: $(x \oplus r) \oplus r = x \oplus (r \oplus r) = x \oplus 0 = x$.

r csak egyszer használható, $\{0,1\}^n$-beli véletlen, egyenletes, koordinátánként ftlen eloszlású.

Titkos kulcscsere

A: p prím, $a \in \mathbb{N}$, $1 < s < p-1$ $\longrightarrow$ küld: $a, p, a^s \bmod p$

B: $1 < t < p-1$ $\longrightarrow$ küld: $a^t \bmod p$

A titkos kulcs $r = a^{ts} \bmod p = (a^t)^s \bmod p = (a^s)^t \bmod p$.

Hátránya: ebből s és t csak d. loggal fejthető vissza, és az nem gyors.

Nyilvános kulcsú titkosítás f_i nyilvános, f_i titkos kulcs, $M(\cdot,\cdot)$ kódolás

1) $M(M(x, g_i), f_i) = x$ (el tudja olvasni a neki írt üzenetet)

2) $M(M(x, f_i), g_i) = x$ (tudja igazolni magát)

3) x ne legyen kiszámolható $M(x, g_i)$-ből f_i nélkül

RSA

p_1 és p_2 véletlen prímek (véletlenszám, PRIME P)

$m := p_1 p_2$ $\quad \varphi(m) = (p_1 - 1)(p_2 - 1)$

g véletlen, $(\varphi(m), g) = 1$.

$f :=$ g mult. inverze, azaz $f \cdot g \equiv 1 \ (\varphi(m))$ (kiterj. eu. alg.)

m nyilvános, g a nyilvános kulcs, f a privát kulcs.

$M(x, g) := x^g \bmod m$

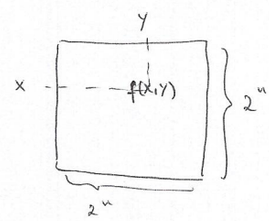
1) $M(M(x,g), f) = x^{fg} \bmod m = x^{\varphi(m)\cdot l + 1} \bmod m = x \bmod m$

2) $M(M(x,g), f) = x \bmod m$

$f: \{0,1\}^n \times \{0,1\}^n \longrightarrow \{0,1\}$ Kiszámítási: az egyik tudja és ezt tudja a másik.

$\kappa(f) := \min_{A} \max_{x,y} [A$ által (x,y)-ra kommunikált bitek száma$] \leq n$

Mehlhorn – Schmidt: $\kappa(f) \geq \log_2(r(M_f))$



B: Minden információ partícionálja M_f-et. 1 bit legfeljebb felezi:

$r(M_f^1) \geq \frac{1}{2} r(M_f)$ $\quad$ A oszlopokat, B sorokat

$\Rightarrow r(M_f^{(k)}) \geq \frac{1}{2^k} r(M_f)$

Ha f ki van számítva, a sorok vagy oszlopok homogének $\Rightarrow r(M_f^{(k)}) = 1 \Rightarrow$ kell legalább $\lceil \log_2 r(M_f) \rceil$ lépés. □

Köv. $M_{ID} = I_{2^n} \Rightarrow n \geq \kappa(ID) \geq \log_2 2^n = n. \Rightarrow \kappa(ID) = n.$ □

$\kappa_0(f) := \min_{ET} \max_{x,y}$ [meggyőző biz. hossza $f(x,y) = 0$-ra] $\quad \kappa_1(f) := \min_{ET} \max_{x,y}$ [megy. biz. hossza $f(x,y) = 1$-re].

$\kappa_0(ID) \leq \log n + 1$. $\quad \kappa_1(f) \leq \kappa(f)$, $\quad \kappa_0(f) \leq \kappa(f)$

Tétel. t min, hogy M_f 1-eseit fedhetők 2^t db csupa 1 résmátrixszal. $\Rightarrow \kappa_1(f) = t$. Ugyanígy 0-val.

B: Előre megbeszélnek egy résmátrixfelbontást $\Rightarrow \kappa_1(f) \leq \log_2 2^t = t$.

Ha $H_\alpha = \{(x,y) \mid A$ és B elfog. α-t$\} \Rightarrow H_\alpha$ résmátrix, csupa 1-es.

Minden α-ra véve a H_α résmátrixok fedik M_f-et.

Véletlen alg. ID-re A: $1 < p < n^2$, elküldi X mod p-t és P-t
($\to 2\log n + 2\log n$ bit)

B: ha $X \equiv Y \ (p) \to 1$, különben 0. (1 bit)

Ezt ismételjük 100-szor, ha van 0, akkor a végeredmény 0.

$$\mathbb{P}(X \equiv Y \ (p) \mid X \neq Y) = \mathbb{P}(p \mid X-Y \mid X-Y \neq 0) \leq \frac{n}{n^2/2\log n} < \frac{1}{2}$$ elég nagy n-re.

$0 \leq X, Y \leq 2^n - 1 \Rightarrow$ p legfeljebb n prímszám közül kerül ki

DISJ = $\mathrm{DISJ}(X,Y) = \begin{cases} 1 & X \cdot Y = 0 \\ 0 & X \cdot Y \neq 0 \end{cases}$ $\quad r(\mathrm{DISJ}) = n$ MS miatt.

Fa legrövid. táv. kivő élek $\log n + \log n = 2\log n$ bit

Gráf. $\mathcal{X}$ fiúk, $\mathcal{Y}$ lányok

A: van-e $\deg(v) \geq \frac{n}{2} \to v$ vagy 0
B: van-e $\deg(w) \leq \frac{n}{2} \to w$ vagy 0
} $\log n$ bit mindkét esetben

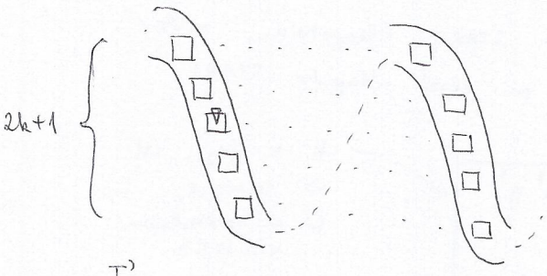
0,0 $\to \mathcal{X} \cap \mathcal{Y} = \emptyset$

v szomszédait kiolvasja, w szomszédait is. $\to \underbrace{\log n \cdot \log n}_{1 \text{ élszám kiírása}}$ bit. +1 az 1 élszám

Lineáris gyorsítási tétel. Ha $f(n) \geq n$, és $L \in DTIME(f(n))$, akkor $\forall 0 < c < 1$-re $\exists T'$ Turing-gép, ami L-et ismeri fel és $time_{T'}(n) \leq c \cdot f(n) + O(n)$.

B: Feltesszük, hogy a T egyszalagos* T-gép L-et $f(n)$ lépésben ismeri fel.

Legyen $k := \lceil \frac{1}{c} \rceil$, akkor $\frac{1}{k} < \frac{1}{1/c} = c$. Akkor k-szoros gyorsítást végzünk.



T' 1 lépése szimulálja T k lépését

T' ismeri az aktuálisban lépest mindkét irányba a következő k mezőt is

$\Rightarrow$ ki tudja számítani a köv. k lépést T-nek

Input átírása $O(n)$ lépés, outputot nem kell átírni, mert csak 0 vagy 1.

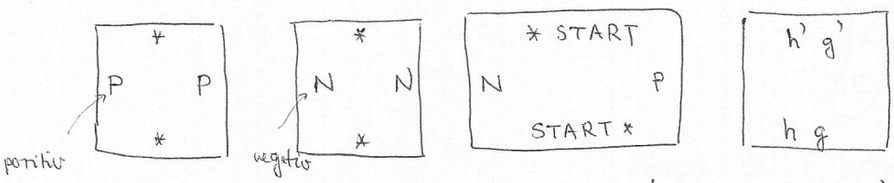
* 1 szalag esetén mindegyikkel megoldjuk ugyanezt.

Tétel. L_{KIRAK} nem rekurzív.

B: $T = (k, \Sigma, \Gamma, \alpha, \beta, \gamma)$ tetszőleges.

Ebből konstruálunk olyan K_T kirakat, amivel pontosan akkor lehet kirakni, ha T nem megáll az üres input.

Ha $k > 1$ vagy T nem csak a legelején van START állapotban, akkor könnyű áttérni olyan T-re, amire $k = 1$ és csak az első állapota START.

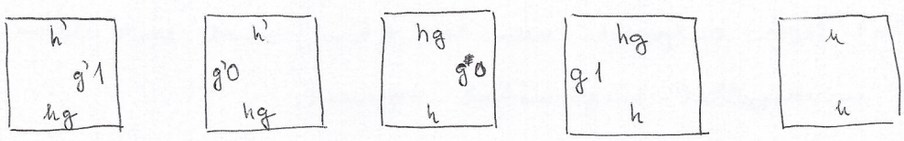


itt a fej, itt is marad, és a tartalom változik.

kezdődominó

$\alpha(h,g) = g'$
$\beta(h,g) = h'$

$\left.\begin{array}{l}\forall h \in \Sigma \\ \forall g \in \Gamma\end{array}\right\}$ véges



$\alpha(g,h) = g'$
$\beta(h,g) = h'$
$\gamma(h,g) = 1$

$\alpha(g,h) = g'$
$\beta(h,g) = h'$
$\gamma(h,g) = -1$

most lép el innen

most lépett ide

üres itt a fej

A (p,q) négyzet átírása a p-edik mező q-adik lépés utáni tartalma alapján.

Felülre: amilyen jel most a tartalma, alulra: ami az előbb volt. } h

Ha a q-adik lépés után a p-edik mezőn is állt a fej; és az állapot g: felülre a g, alulra amilyen állapotból jött.

Ha most lépett balról ide: bal oldalra g1
jobbról ide: jobb oldalra g0

+ középpontos tükrözések az alsó félsíkra.

Ha T nem áll le az üres input → K_T-vel a parkettázás megtehető → $K_T \in L_{KIRAK}$

Ha T leáll ⇒ $K_T \notin L_{KIRAK}$.

Mivel a megállási probléma eldönthetetlen, L_{KIRAK} nem lehet rekurzív.

{Ki. ha rekurzív lenne, akkor $K_T \overset{?}{\in} L_{KIRAK}$ eldönthető, ami épp a megállási probléma.} □

Köv. L_{KIRAK} nem rfs. □