

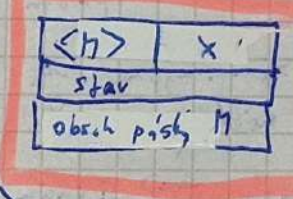
B1 $L_u := \{ \langle n, x \rangle \mid x \in L(M) \}$
 univ. TH a nerozh. L_u

předp. bin. vstupní ab.,
 jednopásmový,
 poč. stav q_0 ,
 jediný příj. stav q_1

TH $(Q, \Sigma, \delta, q_0, \{q_1\})$
 $\delta(q_i, x_j) = (q_k, x_\ell, \{L, R, N\})$
 $\#(i)_B | (j)_B | (k)_B | (l)_B | \langle n, N \rangle \#$

000 = 0
 001 = 1
 010 = 1
 011 = #
 100 = L
 101 = R
 110 = N

TH U:



Věta: L_u je nerozhodnutelný
 Dk: Gödelovo číslo H : index $\langle n \rangle$



$T_{i,j} = \begin{cases} 1 & \text{if } x_j \in L(M_i) \\ 0 & \text{otherwise} \end{cases}$

všechny TH

jazyk DIAS: doplněk diagonály $(\{ \langle M \rangle \langle n \rangle \notin L(M) \})$
 $\rightarrow$ liší se od každého řádku
 $\Rightarrow$ není čísl. rozh.

ukážeme: DIAS $\rightarrow L_u$
 vstup: $\langle n \rangle$, nechť máme TH U rozhodující L_u

$\rightarrow$ pustíme $U(\langle n, \langle n \rangle \rangle) \Rightarrow$ určíme rozhodnutí DIAS

B2
 RAM a ekv. s TH

neomezený # registrů $r_0, r_1, \dots$ na přír. čísla
 (na začátku všude nuly)

instr.: ADD(r_i, r_j, r_k), SUB(r_i, r_j, r_k), LOAD(r_i, val),
 JNZ($r_i, line$), PRINT(r_i), READ(r_i),
 if ($r_i > 0$): goto line

COPY($[r_i]^{src}, r_j^{dst}$), COPY($r_i^{src}, [r_j]^{dst}$)
 $r_j \leftarrow [r_i]$ $r_{[r_i]} \leftarrow [r_i]$

program = posl. instr.
 $I_0, I_1, \dots$

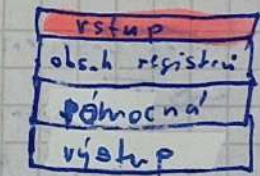
TH $\rightarrow$ RAM:

předp. pásky omezená jen doprava,
 poč. stav q_0

Snaž se stav, na začátku 0 (q_0)
 p... pozice hlavy
 T[]... obsah pásky
 J zakódujeme do programu
 if ($s = \dots \wedge T[p] = \dots$)
 else if ($c = \dots$)

RAM $\rightarrow$ TS:

4 pásky:



$\rightarrow$ #číslo reg. | obsah reg. #...

číslo instr. ve stavu, program v δ

B3

vlastnosti
(číst.)
rozhl. jaz.

DF.: L^* ... jazyk ^{konkretizace} slov z L

$$L_1 \cdot L_2 \dots \left\{ w_1 w_2 \mid \begin{matrix} w_1 \in L_1 \\ w_2 \in L_2 \end{matrix} \right\}$$

uz. vlastnosti:	rozhl.	číst. rozhl.
$\cap$	✓	✓
$\cup$	✓	✓
$\cdot$	✓	✓
$*$	✓	✓
deplét	✓	✗

Věta (Postova): L rozhl. $\Leftrightarrow L \in \overline{L}$ číst. rozhl.

DL.: $\Rightarrow L \in \overline{L}$
 $\overline{L}$ - znegujeme ✓

$\Leftarrow$ paralelně spustíme TM pro L a TM pro $\overline{L}$
 $\Rightarrow$ jeden z nich přijme ✓

Důst. vine: L a $\overline{L}$ číst. rozhl., ale ne rozhl.
 $\Rightarrow \overline{L}$ není číst. rozhl.
(jinak by byly L i $\overline{L}$ číst. rozhl. a z Postovy věty by L byl rozhl.)

Věta ($\exists$ -kvantifikace): L je číst. rozhl.

$\Leftrightarrow \exists R$ rozhodující:

$$L = \{ x \mid \exists y: \langle x, y \rangle \in R \}$$

DL.: $\Rightarrow$ x a y budou # odděleny, po kterých ten TM pro L přijme

$\Leftarrow$ z a y zkonstruujeme všichni v shortlex uspoř.

enumerátory: DF.: Enum. pro jazyk L je TM

s vyhrazenou výst. písmen, na kterém vypisuje $\forall$ slova z L

Věta: L je číst. rozhl. $\Leftrightarrow \exists$ enum. pro L

DL.: $\Rightarrow$ vine: $\exists$ rozhl. jazyk R z věty o $\exists$ -kvant.

$\rightarrow$ generujeme $\langle x, y \rangle$ v shortlex pořadí
a necháme s : od k říct, zda L přijme
 $\Rightarrow$ když jo, tak x vypíšeme na výst.
písmen

$\Leftarrow$ simulujeme ten enumerátor a sledujeme, zda někdy vypíše to vstupní slovo

Věta: L rozhl. $\Leftrightarrow \exists$ enum. pro L vypisující v shortlex uspoř.

DL.: $\Rightarrow$ generujeme x v shortlex, na nich simulujeme TM pro L , když přijme, tak to vypíšeme

$\Leftarrow$ zase simulujeme ten enum., a když vypíše slovo $>$ vstup, tak zastavíme

A1

Riceova věta

DF.: $A \leq_m B \equiv \exists f$ totální výčisl.:
 $x \in A \iff f(x) \in B$

Věta (Riceova): Bud' C třída část. rozh. jazyků, definujme $L_C := \{ \langle n \rangle \mid L(n) \in C \}$.
 Pak L_C je rozh. $\iff$ je C triviální (nebo všechny jazyky)

Def.: $C = \emptyset$: rozh. $\checkmark$

$C, \forall$ část. rozh. jaz.: $L_C = \Sigma^*$ $\checkmark$ rozh.

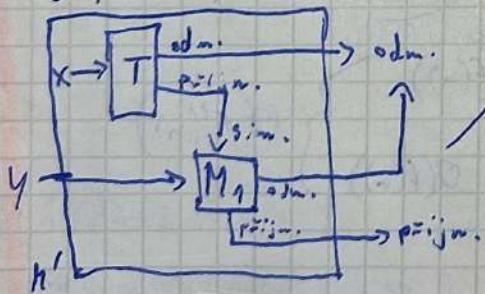
C netriviální: Předp., že $\emptyset \notin C$,
 jinak místo s C pracujeme s $\bar{C}$.

Vezměme lib. $L_n \in C, \emptyset \neq L_n \neq \Sigma^*$
 Vytvoříme TH n s následující strukturou:
 rozhodující L_n

$n_1, \dots, n_k$ přijímající L_n

vstup: $\langle T, x \rangle$

1) vykonáme kód TH h' :



2) postavíme TH rozhodující, zda jazyk $\in L_C$ na vstup $\langle n' \rangle$

$L(n') = \begin{cases} L_n & \iff x \in L(T) \\ \emptyset & \iff x \notin L(T) \end{cases}$
 tedy umíme rozhodnout L_n , což ale víme, že nejde
 $(L_n \leq_m L_C)$

B4

zákl. hrady slot. + dikuz
 $NTIME(f(n)) \subseteq SPACE(f(n))$

DF.: NTH pracuje v čase, $f(n) \equiv \forall$ vstup
 vydá výsledek po nejvýše $f(|vstup|)$ krocích

NTH pracuje v prostoru $f(n) \equiv \forall$ vstup zastaví a
 použije nejvýše $f(|vstup|)$ buněk pásky (pracovní)

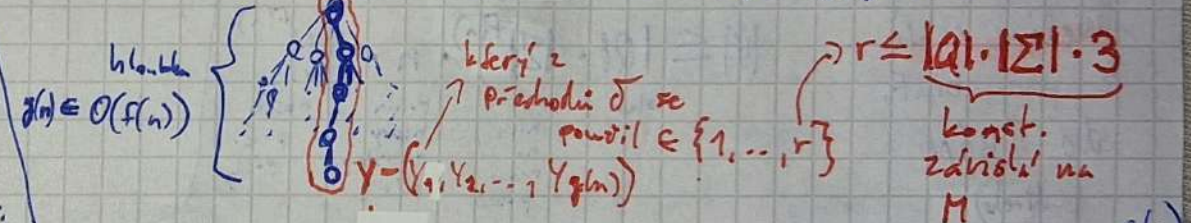
$NTIME(f(n)) :=$ třída jazyků přijímaných NTH pracujících
 čase $O(f(n))$

každý výpočet

$NSPACE(f(n)) :=$ třída jazyků přijímaných NTH
 pracujících v prost. $O(f(n))$

Věta: $NTIME(f(n)) \subseteq SPACE(f(n))$

Def.: strom výpočtu NTH n nad vstupem $x, |x| = n$:



$\rightarrow$ můžeme generovat všechny $y \in \{1, \dots, r\}^{g(n)}$,
 podle toho simulovat ten výpočet
 a když přijme, tak přijmeme

ALE vezmeme $g(n) \rightarrow$ zkusíme místo něj
 $k = 1, 2, \dots$

přijmeme všechny simulace odmítnou $\rightarrow$ odmítneme

$\rightarrow k \leq g(n) \iff$

n'

$\Rightarrow$ zřejmě $L(n') = L(n)$

Věta (o prakt. a čísn): $\forall f(n) \geq \lg_2(n)$

Def.: Def. graf konfiguracji:

$V \Rightarrow$ Konfiguration

$$E: (v_1, v_2) \in E \equiv "v_1 \xrightarrow{\sigma} v_2"$$

6) velikost tega grafu:

$$|V| \leq |Q| \cdot f(n) \cdot n \cdot |\Sigma|^{f(n)}$$

zhu	p. lita	p. lita	obsah
	hlay	hlay	prac.
	n.	na vstup.	přítel
	prac.		
	přítel	přítel	

$$\begin{aligned} &= 2^{1_{\gamma_2}|Q| + 1_{\gamma_2}f(n) + 1_{\gamma_2}n + f(n)1_{\gamma_2}|Z|} \\ &\leq 2^{F(n) \cdot (1_{\gamma_2}|Q| + 1 + 1 + 1_{\gamma_2}|Z|)} \end{aligned}$$

C_n - konst. závislá na
jazyku

$$= 2^{c_n} \cdot f(n)$$

$$|E| \leq |V|^2 \leq 2^{2cn} F(n)$$

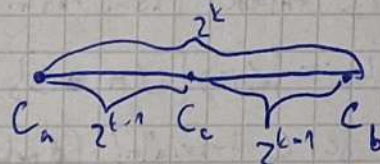
(for graph 9: stock constructed on demand z & a. akt. kont.)

$$C_L \geq k \cdot \epsilon_M$$

Volta (Saviceva): $\forall f(n) \geq \lg^2(n)$: $\text{NSPACE}(f(n)) \subseteq \text{SPACE}(f^2(n))$

Dk.: zase jako v **(B5)** budeme pracovat s grafem

conf. $G_{n,k}$, ale masine si h/odat prastor:
(ne DFS/BFS!)



costa $C_a \rightarrow C_b$ de' lly $2^k \exists \Leftrightarrow$

$\exists$ cest $C_a \rightarrow C_c$ díky 2^{k-1} a

$\exists$ rest $C_c \rightarrow C_b$ de l'hy 2^{k-1}

chceme: $C_0^x \xrightarrow{zk} C_F$

$\uparrow$ $\uparrow$

Přetv.
konf. jedno jediné
 přímění!

≤ $C_m F(n)$
neznáme!

→ klasické všechny možnosti, hledat to je

nezrúme!

2. časina
 ↓
 4. časina
 8. časina
 16. časina
 32. časina
 64. časina
 128. časina
 256. časina
 512. časina
 1024. časina
 2048. časina
 4096. časina
 8192. časina
 16384. časina
 32768. časina
 65536. časina
 131072. časina
 262144. časina
 524288. časina
 1048576. časina
 2097152. časina
 4194304. časina
 8388608. časina
 16777216. časina
 33554432. časina
 67108864. časina
 134217728. časina
 268435456. časina
 536870912. časina
 1073741824. časina
 2147483648. časina
 4294967296. časina
 8589934592. časina
 17179869184. časina
 34359738368. časina
 68719476736. časina
 137438953472. časina
 274877906944. časina
 549755813888. časina
 1099511627776. časina
 2199023255552. časina
 4398046511104. časina
 8796093022208. časina
 17592186044416. časina
 35184372088832. časina
 70368744177664. časina
 140737488355328. časina
 281474976710656. časina
 562949953421312. časina
 1125899906842624. časina
 2251799813685248. časina
 4503599627370496. časina
 9007199254740992. časina
 18014398509481984. časina
 36028797018963968. časina
 72057594037927936. časina
 144115188075855872. časina
 288230376151711744. časina
 576460752303423488. časina
 1152921504606846976. časina
 2305843009213693952. časina
 4611686018427387904. časina
 9223372036854775808. časina
 18446744073709551616. časina
 36893488147419103232. časina
 73786976294838206464. časina
 147573952589676412928. časina
 295147905179352825856. časina
 590295810358705651712. časina
 1180591620717411303424. časina
 2361183241434822606848. časina
 4722366482869645213696. časina
 9444732965739290427392. časina
 18889465931478580854784. časina
 37778931862957161709568. časina
 75557863725914323419136. časina
 151115727451828646838272. časina
 302231454903657293676544. časina
 604462909807314587353088. časina
 1208925819614629174706176. časina
 2417851639229258349412352. časina
 4835703278458516698824704. časina
 9671406556917033397649408. časina
 19342813113834066795298816. časina
 38685626227668133590597632. časina
 77371252455336267181195264. časina
 154742504910672534362390528. časina
 309485009821345068724781056. časina
 618970019642690137449562112. časina
 1237940039285380274899124224. časina
 2475880078570760549798248448. časina
 4951760157141521099596496896. časina
 9903520314283042199192993792. časina
 19807040628566084398385987584. časina
 39614081257132168796771975168. časina
 79228162514264337593543950336. časina
 158456325028528675187087900672. časina
 316912650057057350374175801344. časina
 633825300114114700748351602688. časina
 1267650600228229401496703205376. časina
 2535301200456458802993406410752. časina
 5070602400912917605986812821504. časina
 10141204801825835211973625643008. časina
 20282409603651670423947251286016. časina
 40564819207303340847894502572032. časina
 81129638414606681695789005144064. časina
 162259276829213363391578010288128. časina
 324518553658426726783156020576256. časina
 649037107316853453566312041152512. časina
 1298074214633706907132624082305024. časina
 2596148429267413814265248164610048. časina
 5192296858534827628530496329220096. časina
 10384593717069655257060992658440192. časina
 20769187434139310514121985316880384. časina
 41538374868278621028243970633760768. časina
 83076749736557242056487941267521536. časina
 166153499473114484112975882535043072. časina
 332306998946228968225951765070086144. časina
 664613997892457936451903530140172288. časina
 1329227995784915872903807060280344576. časina
 2658455991569831745807614120560689152. časina
 5316911983139663491615228241121378304. časina
 10633823966279326983230456482242756608. časina
 21267647932558653966460912964485513216. časina
 42535295865117307932921825928971026432. časina
 85070591730234615865843651857942052864. časina
 170141183460469231731687303715884105728. časina
 340282366920938463463374607431768211456. časina
 680564733841876926926749214863536422912. časina
 1361129467683753853853498429727072845824. časina
 2722258935367507707706996859454145691648. časina
 5444517870735015415413993718908291383296. časina
 10889035741470030830827987437816582766592. časina
 21778071482940061661655974875633165533184. časina
 43556142965880123323311949751266331066368. časina
 87112285931760246646623899502532662132736. časina
 174224571863520493293247799005065324265472. časina
 348449143727040986586495598010130648530944. časina
 696898287454081973172991

✓
Zkoušime
1, 2, ...
a zastavíme
se, když už
zvýšková!

prostoru
vedat' shraj
vyssi silu
(neni dos.
ziden' kof.
mutivajci
prostor
→ $f(x) + 1$)
aktualni

A3

DF: Funkce $f(n) \geq \lg_2(n)$ je prost. konstr.
 $\equiv$ "ke spočítání té hodnoty nepotřebuje větší prostor, než je ta samotná hodnota"
 (zobrazuje n na bin. repr. $f(n)$ v prostoru $O(f(n))$)

det.
prost.
hierarchie

Věta (det. prost. hier.): $\forall f(n)$ prost. konstr. funkci:
 $\exists L$ rozhodnutelný v prost. $O(f(n))$, ale ne v prost. $o(f(n))$

Dk: Definujme jazyk $L(D) = \{ \langle n \rangle \mid \text{simulace } M(\langle n \rangle) \text{ odmítne v prost. } O(f(n)) \}$

$\rightarrow$ $\odot$ rozh. v prost. $O(f(n))$
 $\odot \forall L'$ rozh. v prost. $o(f(n))$: $L' \neq L(D)$: $D(\langle n \rangle 10^{10})$
 přijme
 $\Rightarrow M$ to odmítne
 $\Rightarrow$ liší se jejich jazyky!!
 poznámka: TM

Musíme ale ještě dorešit asymptotiku:
 to, že $g(n) < c f(n)$ platí až od nějakého n_0 !
 $\Rightarrow$ D nebude jazyk $\langle n \rangle$, ale $\langle n \rangle 10^k$,
 aby ty vstupy byly dost dlouhé na to, aby se projevila asymptotika

A promyslet tu simulaci, aby kontrolovala ten prostor: $f(n)$ je prost. konstr. $\Rightarrow$ musíme si vyznačit $f(n)$ buňek na páse pro tu simulaci: (a šloť na to prostor $f(n)$)

A zařídit, aby se simulace nezacyklila
 - to by se stalo, když se nějaká konfigurace opakuje
 když je $2^{f(n)}$ možností
 po $2^{f(n)}$ krocích simulace zůstane v dané konfiguraci

A4

DF: Funkce $f(n) \in \Omega(\lg_2(n))$ je čas. konstr. $\equiv$ "ke spočítání $f(n)$ nepotřebuje víc času, než je ta hodnota $f(n)$ "
 (zabr. n na bin. repr. $f(n)$ vyčísl. v čase $O(f(n))$)

det.
čas.
hier.

Věta (det. čas. hier.): $\forall$ čas. konstr. $f(n) \exists L$ rozh. v čase $O(f(n))$, ale ne v čase $o(f(n)/\lg_2 f(n))$

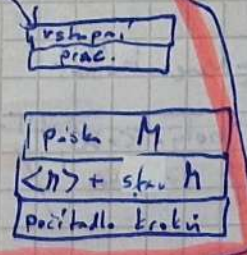
Dk: Podobně jako u A3, ale zájma nás čas a musíme lip rozmyslet tu simulaci: (přdp. $\forall TM$ jednopásmové)

• počítání si bráky - počítadlo inicializujeme na $f(n)/\lg_2 f(n) \rightarrow$ manipulace v čase $O(\lg_2 f(n))$
 čas. konstr. $\checkmark$!!

• univ. TM umí simulaci v kvadratickém čase, my ale potřebujeme lepší: "ty stopy si budeme držet u hlavy."

počítadlo posouváme v každém kroku simulace
 $\Rightarrow$ zarovnáme s hlavou
 $\langle n \rangle$ a stav posouváme v každém kroku M
 $\Rightarrow$ zarovnáme s akt. blokem

D má:
 2 pásky:
 3 stopy:
 prac. pásky



$O(f(n)/\lg_2 f(n))$
 kroky,
 každý
 trvá!

$O(\lg_2 f(n))$
 $O(f(n))$

Pokud M pracuje v čase $g(n)$,
 ke jeho simulaci provede v čase $O(g(n))$
 $O(Kn^2)$
 $\rightarrow$ konst. pro to M

B6 DF: Verifikátor jazyka L je alg V t.t.

$$L = \{x \mid \exists y: V(x, y) \text{ přijme}\}$$

$L \leftarrow \text{certifikát}$

Polynomiální verifikátor je verifikátor, který pracuje v poly. čase vzhledem k $|x|$

duň def.
tridy NP
a jejich
ekvivy.

DF: $L \in NP \equiv L$ má polynomiální verifikátor

Věta: $L \in NP \Leftrightarrow L \in \text{UNTIME}(n^k)$

DL: $(\Leftarrow)$: Necht' L je rozhodn. v čase $g(n)$ NTA M
 $\Rightarrow \forall$ větu toho výpočtu má délku $\leq g(n)$

$\Rightarrow$ použijeme ji jako certifikát
 (ocistujeme si možné přechody δ
 $\rightarrow 0, 1, \dots, r$
 $L \leftarrow$ "101.101.3 $\rightarrow$ konst.)

$y \in \{0, 1, \dots, r\}^{g(n)}$
 $\leftarrow$ polynomiální certifikát

verifikátor projde y přechody a zkontroluje to
 $\rightarrow$ polynomiální

$(\Rightarrow)$: Máme poly. verifikátor $V \Rightarrow$ ten simulujeme s tím, že když chce číst nějaký znak certifikátu, tak je nedeterministicky zvolíme

B7

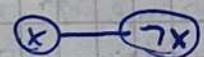
3-SAT
 $\downarrow$
 vrch.
 pokr.
 (VP)

proměnné $x_1, \dots, x_n$

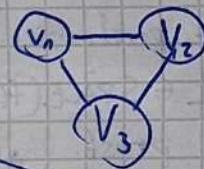
klauzule $C_1, \dots, C_m$, každá obs. právě 3 literály

Sestrojíme graf G :

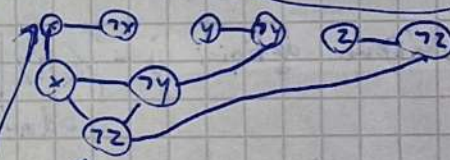
1) $\forall$ proměnnou x :



2) $\forall$ klauzuli $C = \{v_1, v_2, v_3\}$:



3) propojíme odpovídající literály s teni: průsečíky:



4) $k := n + 2m$

5) najdeme vrch. pokrytí velikosti k v G

Tvrzení: $\varphi := \{C_1, \dots, C_m\}$ splnitelná $\Leftrightarrow \exists$ vrch. pokrytí vel. k v G

DL: $(\Leftarrow)$ z těch proměnných se vždy musí vybrat právě jedna, z trojčlenného dva vrcholy
 $\Rightarrow \forall$ klauzule je splnitelná tím literálem, který není vybrán v tom Δ

$(\Rightarrow)$ podle toho spln. ohodnotíme všechny sestrojit ("pozpátkn")

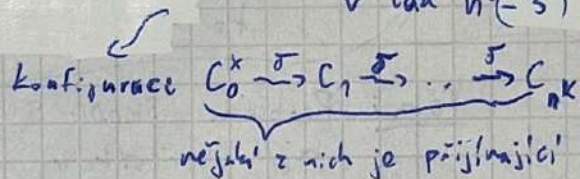
A5 Věta (Cook-Levinová): Pokud $SAT \in P$, pak $P=NP$

Cook-Levin Def. Plyne z věty níže

Věta: SAT je NP-úplný

Def.: $SAT \in NP$ ✓

SAT je NP-těžký: nějaký jazyk přijímá NTH n v čase $n^k(-3)$



def. Tablo T pro $n(x)$ je matice $n^k \times n^k$

• řádky: $C_0, \dots, C_n$ konfigurace

- značí a kon. #
- stav před políčkem s hlávkou
- ty konf. po sobě následují

• přijímající $\equiv \exists$: T_i obsahuje přijímající konf.

V_3 trojice formule φ (není v CNF, ale mohla by):

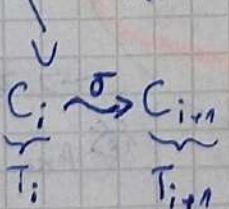
$$\varphi = \varphi_{\text{cell}} \wedge \varphi_{\text{start}} \wedge \varphi_{\text{move}} \wedge \varphi_{\text{accept}} \Rightarrow SAT$$

$\forall$ buňka $\in S := Q \cup \Sigma \cup \{\#\}$

$$x_{i,j,s} = \begin{cases} 1 & \Leftrightarrow T_{i,j} = s \\ 0 & \text{jinak} \end{cases}$$

první jednota $\forall i, j$
• aspoň 1
• max. 1

na 1. řádku je C_0



okno: 2×3 - efekt σ

↳ podle σ zakládáme, že musí být přípustné

B11

co-NP: $= \{A \mid \bar{A} \in NP\}$

Př.: TAUT: $= \{\langle \varphi \rangle \mid \varphi \text{ je tautologie}\} \in \text{co-NP}$:

φ není tautologie $\Leftrightarrow \langle \varphi \rangle$ má model $\Leftrightarrow$
"SAT(φ)" přijme
NP

Def.: A je co-NP-úplný $\equiv A$ je co-NP-těžký
 $\wedge A \in \text{co-NP}$

• A je co-NP-těžký $\equiv \forall B \in \text{co-NP}: B \leq_m^P A$

Věta: A je co-NP-úplný $\Leftrightarrow \bar{A}$ je NP-úplný

Def.: vime: $A \leq_m^P B \Leftrightarrow \bar{A} \leq_m^P \bar{B}$

$\Rightarrow$ Př.: SAT je co-NP-úplný

Větička: $A \in \text{co-NP} \Leftrightarrow \exists$ poly. verifikátor V :

$$A = \{x \mid \forall y: V(x, y) \text{ přijme}\}$$

Def.: $\exists$ " $\exists V \in NP$ "
verifikátor $\in NP$ pro $\bar{A}$

Vime: $P \in NP \cap \text{co-NP}$

asi ostrá inkluze, ale nevíme
→ možná $P = NP = \text{co-NP}?$

B10 DF: $f: \Sigma^* \rightarrow \mathbb{N} \in \#P \equiv \exists \text{ poly. verifikator}$

#P,
#P-úplnosť,
ťažkosť
počítaní
cyklů
v grafu

$\forall a \text{ poly. } p(n): \forall x \in \Sigma^*:$

$$f(x) = |\{y \in \{0,1\}^{p(x)} \mid V(x,y) \text{ přijme}\}|$$

DF: $f \leq_p g \equiv f$ lze vyčíslit v
poly. čase algoritmem, který má přístup
k orákulu funkce g .

→ poly. # datoví

DF: $g \in \#P$ je #P-úpln $\equiv \forall f \in \#P: f \leq_p g$

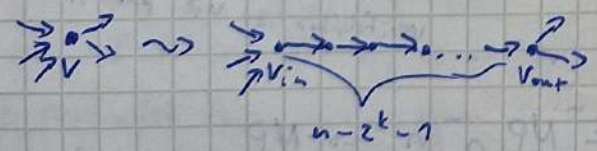
Př: Počítání cyklů v grafu

- nalezení cyklů: DFS - jednoduché
- spočítání cyklů: těžké - ukážeme

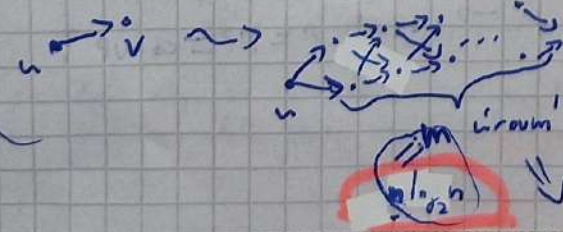
Fakt: HK (Ham. kružnice) je NP-úplný problém

ukázkové HK → #cyklů:

1) graf G upravíme, aby měl 2^k vrcholů:



2) každou hranu nahradíme takto:



3) G obsahuje HK

$$\Leftrightarrow \# \text{cyklů v } G' \geq n^{n^2}$$

• $v \in \text{Greni HK}$: $\forall$ cyklů $\checkmark$ délky max. $n-1$
a je jich n^{n-1}
 $\Rightarrow v \in G'$ je nejvýš $(2^{n(n-1)}) \cdot n^{n-1}$

$$\begin{aligned} \text{• HK v } G: \# \text{cyklů v } G' &\geq (2^n)^n = n^{n^2} \\ &\parallel n^{n^2-1} < n^{n^2} \end{aligned}$$

díky tomu
se z cyklů délky l v G
stane 2^{ml} cyklů v G'

$$\frac{1}{2} \log_2 n \Rightarrow 2^m \text{ or. cest}$$

B13 → podmíněné dolní odhady

Příklad
zjemněle
redukce

→ platnost nějaké hypotézy

Hyp. (OV): A - množina n ddim. vektorů
 B - ...

$$\Rightarrow \exists x_A \in A, x_B \in B: x_A x_B = 0?$$

jen o!

Hyp. říká, že pro každé $\delta, c > 0$
to nejde vyřešit v čase $O(n^{2-\delta} d^c)$

Rozn: Problém SHODA S REGEXEM
nejde vyřešit v čase $O((nm)^{2-\epsilon})$ pro
každé $\epsilon > 0$, pokud platí hyp. o OV.

Def: Ukázkové OV → REGEX:

- Z A vytráčíme REGEX takto:

$$F(v) := \bigcap_{i=1 \dots d} g(v_i) \quad \forall v \in A$$

$$g(x) := \begin{cases} 0 & \text{pokud } x \in A \\ \{0,1\} & \text{pokud } x = 0 \end{cases}$$

$$R(A) := \bigcap_{v \in A} F(v)$$

- Z B vytráčíme text: dává za sebe
všechny z B oddělené # → $T(B)$

→ Zapišme se, zda $R(A)$ se vyskytuje v $T(B)$

• Ano $\Leftrightarrow$ jsou tam nějaké hyp. OV

→ kdyby šlo rychleji rozhodnout
REGEX, pak lze i rychleji rozhodnout
OV

(38)

DF.: Parametrizovaný problém je jazyk

$$L \subseteq \Sigma^+ \times \mathbb{N}$$

kon. abeceda

parametr

$$\langle I, k \rangle$$

instance
velikosti
 $|I| + k$

FPT,
kernel
+
kernelizace
vrch.
pokr. (VP)

DF.: Parametrizovaný problém $L \in \text{FPT}$

$\equiv$ lze jej rozhodnout algoritmem
 $A \in \text{TIME}(f(k) \cdot |I|^c)$ pro nějakou
alg. vyčísl. funkci f a konst. c

DF.: $O^*(f(k)) := O(f(k) \cdot n^c)$ pro lib. konst. c

DF.: A je kernel pro problém $L \equiv$

- $A \in P$!
- Pro instance $\langle I, k \rangle, \langle I', k' \rangle = A(I, k)$:

$$\langle I, k \rangle \in L \Leftrightarrow \langle I', k' \rangle \in L$$
- $\exists$ vyčísl. funkce $g(k)$ t.j. $\forall$ instance $\langle I, k \rangle$,

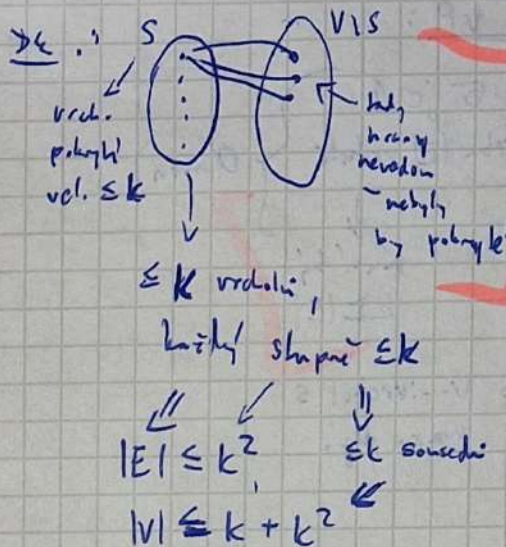
$$\langle I', k' \rangle = A(I, k) : |I'| + k' \leq g(k)$$

velikost
kernelu

Pr.: kernelizace VP:

- odstraníme izolované vrcholy (1)
- odstraníme vrcholy stupně $\geq k+1$ a snížíme k (2)

Lemma: když už nejde aplikovat žádný z pravidel (1), (2),
tak $|E| \leq k^2$ a $|V| \leq k + k^2$



Pravidlo (3):

když nejde aplikovat ani (1), ani (2)
a $|E| > k^2$ v $|V| > k + k^2$,
tak tam není to VP vel. $\leq k$

↓
Search VP(G, k)

- čas $O(2^k(m+n)) = O^*(2^k)$
- 0) if $E = \emptyset$ return $\emptyset$
- 0.5) if $k = 0$ return NONE
- 1) vyber hranu $\{u, v\} \in E$
- 2) odeber $u \rightarrow$ rek. $(G - u, k-1)$
- 3) odeber $v \rightarrow$ rek. $(G - v, k-1)$

- ↑
hloubka rek. $k = 2$ rek. volání
- ↑
čas na hloubku $O(m+n)$

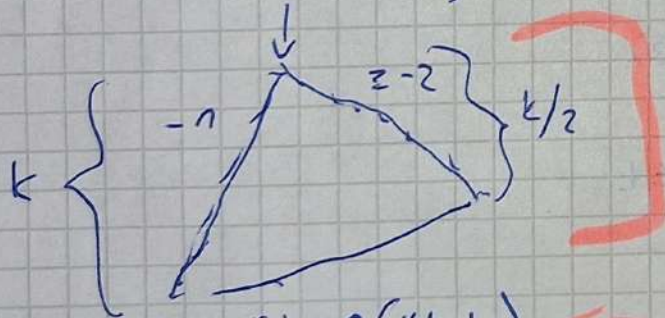
B9

Probl. s omezením hloubkou pro VP:

FPT (viz B8)
+ param. alj. na VP
pomocí probl. dárání s omezením hloubkou

- základní případ: • max. deg. v G je 1
→ je potřeba |E| vrcholů
• $k \leq 1$
 - řešení: • VES
NEBO
• $N(v) \subseteq S$ pro v - vrchol s max. deg.
- $O(m+n)$

čas: $O(\gamma(k)(m+n))$



$T(k) \in O(\# \text{listy})$

$$T(i) = \begin{cases} T(i-1) + T(i-2) & \text{pro } i \geq 2 \\ 1 & \text{jinak} \end{cases}$$

čas: $O(T(k)(m+n))$

chceme: $T(k) \leq \lambda^k \cdot c$

$$\rightarrow T(k) \leq \lambda \cdot \lambda^{k-1} + \lambda \cdot \lambda^{k-2} \leq \lambda \cdot \lambda^k$$

$$\lambda + 1 \leq \lambda^2 \rightarrow \lambda = 1.6181$$

Jde ještě vylepsit základní případ: pokud max. deg. ≤ 2 ,
pak $\forall$ komp. G je cesta nebo cyklus

$$T(i) = \begin{cases} T(i-1) + T(i-3) & \text{pro } i \geq 3 \\ 1 & \text{jinak} \end{cases}$$

$$\lambda \leq 1.4656$$

$$O^*(1.6181^k)$$

B12

Pseudo-RNG,
jednosměrná
funkce
+
kryptografie

DF: $G: \{0,1\}^n \rightarrow \{0,1\}^{\ell(n)}$ je pseudo-RNG

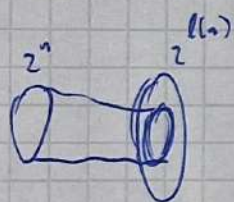
$\equiv$ • G je vyčísl. det. poly. alg.

• $\ell(n) > n \quad \forall n$
↳ stretch

• $\forall$ pravd. poly. alg. $A: \left| \frac{P[A(y)=1]}{y \in \{0,1\}^{\ell(n)}} - \frac{P[A(G(y))=1]}{y \in \{0,1\}^n} \right| \leq \epsilon(n)$

↳ "na říct, jestli rekurec na vstup byl vygenerovaný tím PRNG, nebo ne"

zanedbatelná ($\equiv \forall k \in \mathbb{N} \exists n_k: \forall n > n_k: \epsilon(n) < \frac{1}{n^k}$)



Věta: Pokud $P=NP$, tak $\nexists$ žádný PRNG

Důk: certifikát: $y \in \{0,1\}^n$

pro
 $x \in \{0,1\}^{\ell(n)}$

jazyk všech $x \in \{0,1\}^{\ell(n)}$
 $\in NP = P \rightarrow$ v poly. čase umíme určit, zda x tam patří, či ne

DF: $f: \{0,1\}^* \rightarrow \{0,1\}^*$ je jednosměrná funkce (OWF)

$\equiv$ • je vyčísl. v poly. čase

• je těžko invertovatelná:

$\forall A$ prac. v poly. proud. čase

$\exists \epsilon(n)$ zanedbatelná:

$\forall n \in \mathbb{N}: P[A(f(x)) \in f^{-1}(f(x))] \leq \epsilon(n)$
 $x \in \{0,1\}^n$

Věta: PRNG $\exists \iff \exists$ OWF

Kandidáti na OWF: např. faktorizace, disk. log.

$f_{g,p}(x) := g^x \bmod p$

$$\rightarrow \left| \frac{P[A(y)=1]}{2^n} - \frac{P[A(G(y))=1]}{1} \right| = 1 - 2^{n-\ell(n)} \geq 1 - \frac{1}{2} = \frac{1}{2}$$

$\frac{2^n}{2^{\ell(n)}} = 2^{n-\ell(n)}$ $\downarrow$ ($\ell(n) > n$) $\downarrow$ není zanedbatelná

Pr.: Bit-commitment

PRNG G s $\ell(n) = 3n$

Bob

Alice

r - náh. 3n-bit.

y - náh. n-bit.

bit b

$c := \begin{cases} G(y) & \text{pro } b=1 \\ G(y) \oplus r & \text{pro } b=0 \end{cases}$

$b = \begin{cases} 1 & \text{pro } c=G(y) \\ 0 & \text{pro } c=G(y) \oplus r \end{cases}$

odhalení

y