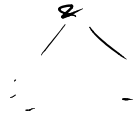
- Dw
- stroje

- uniformní ... algority, TS, RAM, ...
- neuniformní ... bool fce, bool. obvody, ...

Př.



bool fce

formule f_i pravje na vstupech velikosti i

$\{f_i\}_{i \geq 1}$ ... posloupnost bool. fcí (rodina)

- můžeme zkoumat parametry f_i v závislosti na velikosti vstupu.

např.: velikost f_i jako funkce i

- k čemu je to dobré,
  1) realita může být neuniformní
  2) cesta pro porozumění výpočtům
- radící fce $g: \mathbb{N} \to \{0,1\}^*$
  - algoritmus A + radící fce g.
  - při výpočtu na vstupu velikosti n dostaneme zadarmo $g(n)$.

$A(x, g(|x|))$

Př.: P/poly ... problémy pro které $\exists$ alg. A

pomocí rádců v pol. čase a radící fce g, kde délka g roste polynomiálně, t.ž. (A, g) řeší daný problém

P/f … —||— , kde délka g roste jako f.

… "neuniformní P"

<u>Tvrzení</u>: Pokud g je počitatelná v pol. čase pak fce počítaná (A, g) je v P.

$\Rightarrow$ "uniformita nápovědy"

<u>Tvrzení</u>: f je v $P/poly$ $\Leftrightarrow$ f je počitatelná bool. obvody pol. velikosti.

<u>Dk</u>: "$\Leftarrow$" easy … nápovědou je popis obvodu
"$\Rightarrow$" obvod je schopen simulovat výpočet; nápověda se zadrátuje

"P vs. NP"

$$NP \not\subseteq P/poly \Rightarrow P \neq NP$$

$\rightarrow$ <u>Otázka</u>: $\exists$ fce $f \in NP$ t.ž. nelze počítat obvody pol. velikosti?

<u>Domněnka (Kolmogorov)</u>: Fce v P mají obvody lineární velikosti.

- Nejlepší dolní odhad <u>$5n$</u>.

(2)

Věta: $\forall f : \{0,1\}^n \to \{0,1\}$ existuje obvod s binárními AND, OR & NOT velikosti $O(n2^n)$ počítající f.

- lze zlepšit na $O\left(\frac{2^n}{n}\right)$

Věta: $\exists f : \{0,1\}^n \to \{0,1\}$ t.ž. nejmenší obvod s binárními AND, OR & NOT, který ji počítá je alespoň velikosti $\frac{2^n}{3n}$.

Dk: "counting" ☒

Věta: $\forall k \; \exists L \in EXP$ t.ž. $L \in SIZE(kn^k + k)$.

Dk. na vstupu x

$x_1 = 0\ldots0, \quad \ldots \quad x_{2^n} = \underbrace{1\ldots1}_{n}$

C_0 = obvody velikosti $\leq kn^k + k$.

$i = 0$

opakuj pokud $C_i \neq \emptyset$

- pokud většina obvodů v C_i dává výstup 0 na x_i, def $t_i := 1$ jinak $t_i := 0$
- $C_{i+1} = \{C \in C_i,$ t.ž. $C(x_i) = t_i\}$.
- $i \leftarrow i+1$

pokud x je j-tý prvek kde $j < i$, pak výstup t_j, jinak výstup 0.

konec.

→ **Věta:** $\forall k \; \exists L \in PSPACE$ t.ž. $L \in SIZE(n^k)$.

Otázka: Pro jakou nejmenší třídu lze ukázat podobné tvrzení?

Věta (Karp – Lipton): $\forall k$, existuje f v $NP^{NP^{NP}}$ t.ž. f nemá obvody velikosti kn^k.

Dk: $c = S(a, 111\ldots1, 111\ldots1)$ $\quad a \in \{0,1\}^*$

a je počáteční úsek prvních tabulky fce na n bitech, počitatelné obvodem velikosti n^k S

$S \in NP$

---

na vstupu x: uhodni $a \in \{0,1\}^n$, zjisti zda $(a, 1^n, 1^{knk}) \in S$. Pokud ne, rozhodni x podle tabulky počínající a, tj. $a0^{2^n - n}$.

→ problém, může být více takových a.

$S' = \{ (a, 1^n, 1^m) ;\ a \in \{0,1\}^*$, buď $a \in S$ nebo existuje lex. menší $a' < a$ t.ž. $a' \notin S \}$

$S' \in NP^{NP}$

→ použij předchozí algoritmus s S'. ■

Věta (k-l) lze zlepšit na $NP^{NP} \subseteq SIZE(n^k)$

- buď $NP \not\subseteq P/poly$ pak závěr plyne automaticky
- nebo $NP \subseteq P/poly$ ⇒ ∃ poly velký obvod pro SAT.

⇓

$S' \in NP/poly$

- Tento obvod lze uhodnout a ověřit během NP^{NP} výpočtu
- po jeho uhodnutí, "$S' \in NP/poly$"

⇒ alg. v důkazu ...

„je v NP". ∎

„NP vs coNP" ?

„NEXP vs coNEXP"

↳ nedeterministický exponenciální čas

víme: $NEXP \subseteq coNEXP/O(n)$

(tzn. $NEXP/O(n) = coNEXP/O(n)$)

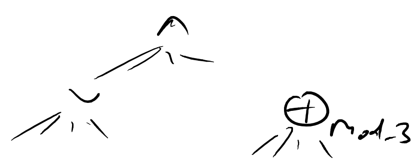
↳ důkaz: nápověda udává počet přijímaných vstupů délky n. ∎

③

Fact: $MAEXP \not\subseteq P/poly$

Fact: $\forall k \quad ZPP^{NP} \not\subseteq SIZE(n^k)$ (double check)

Razborov - Smolensky ≈ '87



$AC^0[3]$

AC^0 ... hloubka $O(1)$, velikost $n^{O(1)}$

AND, OR, NOT libovolného stupně

$AC^0[m]$ ... navíc MOD_m libovolného stupně

$$MOD_m(x_1, \dots x_k) = \begin{cases} 0 & m \mid \sum x_i \\ 1 & \text{jinak} \end{cases}$$

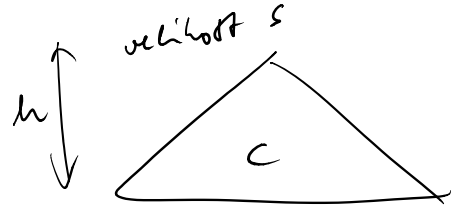
- Razborov - Smolensky: $\forall$ různá prvočísla p, q, $MOD_p \notin AC^0[q]$
- Furst - Saxe - Sipser '81, Ajtai '83, Håstad '85: $MOD_2 \notin AC^0$

$MOD_2 \notin AC^0$

→ důkaz ve dvou krocích:

1) aproximace obvodu polynomem nad $GF[2]$
2) MOD_p velké aproximační polynom malého stupně

Krok 1:



Báze: OR, MOD_2, NOT

indukcí podle hloubky obvodu přiřadíme každému hradlu g polynom p_g nad $GF[2]$, který bude souhlasit s g na všech vstupech až na malou množinu. W_h ... množina špatných vstupů pro hradla na h-té úrovni

$W_0 = \emptyset$

$h > 0$

a) ¬ g

g'

$p_g = 1 - p_{g'}$

b) MOD_2

$g_1 \dots g_\ell$

$p_g = \left(\sum_i p_{g_i}\right)^{2-1}$

c) OR

$g_1 \quad g_\ell$

$p_g = 1 - \prod_{j=1}^{k} \left(1 - \left(\sum_i a_{ji} p_{g_i}\right)^{2-1}\right)$

a_{ji} ... náhodně zvolené

pro pevně zvolený vstup $x \notin W_{h-1}$, kde $\bigvee_i g_i = 1$, $\forall j$:

$\sum_i a_{ji} (x_i) > 0$

pro pevně zvolený vstup $x \notin W_{h-1}$, kde $\exists_i\, g_i = 1$, tj. $\sum_i p_{g_i}(x_i) > 0$ ← součet nad W

$$\Pr_{a_{ij}} \left[p_g(x) = 0 \right] \le \left(\frac{1}{q}\right)^k$$

S hradel na h-té úrovni

$\rightarrow$ $\Pr$ [některý z polynomů na h-té úrovni počítá špatně na x] $\le S \cdot \left(\frac{1}{q}\right)^k$

$\exists$ výběr koeficientů a_{ij} pro hradla na h-té úrovni, t.ž.

$$|W_h| \le |W_{h-1}| + 2^n \cdot S \left(\frac{1}{2}\right)^k$$

$$\Longrightarrow |W_h| \le O\left(2^n \cdot S \left(\frac{1}{2}\right)^k\right)$$

$$\deg p_g \le ((q-1)k)^h$$

$S = 2^{n^{1/4h}}$ $\quad k = n^{1/3h}$ $\quad q \ge 2$

$$\deg p_C = O(n^{1/3})$$

$\rightarrow$ $\forall$ $AC^0[q]$ obvod velikosti $\le 2^{n^{1/4h}}$ a hloubky $\le h$ $\exists$ polynom nad $GF[q]$, který počítá stejnou funkci na všech vstupech až na množinu vstupů $o(2^n)$.

2) polynom stupně $O(n^{1/3})$ nad $GF(q)$ nemůže

počítat MOD_p na $2 - o(2^n)$ vstupu.

spor. že počítá MOD_2 vzdor tomu $W \subseteq \{-1,1\}^n$

Dk: Nechť $f : \{-1,1\}^n \setminus W \longrightarrow GF[q]$

$|W| = o(2^n)$

$x_i^2 = 1 \qquad x_i \in \{-1,1\}$

f lze reprezentovat polynomem nad $GF[q]$ multilineárním

pz počítá $x_1, x_2 \dots x_n$ na $\{-1,1\}^n \setminus W$ $\Rightarrow f = p_2 \cdot l_1 + l_2$

kde l_1, l_2 jsou multilineární stupně nejvýše $n/2$.

$\Rightarrow$ f lze reprezentovat polynomem stupně $\leq n/2 + O(n^{1/3})$

$\#$ polynomů $\leq 3^{\sum_{i=0}^{\frac{n}{2}+O(n^{1/3})} \binom{n}{i}} \leq 3^{2^{n-1}+o(2^n)}$

$\#f$ $\geq 3^{2^n - W} > 3^{2^n - o(2^n)}$ spor

---

(4) • Přibližné počítání je v AC^0

(approx. MAJ $\in AC^0$ [Ajtai-Ben-Or '83])

$$aMAJ(x) = \begin{cases} 0 & \sum x_i \leq \frac{1}{4} \\ 1 & \sum x_i \geq \frac{3}{4} \end{cases}$$

• $aMAJ(x) \in AC^0$

Dk: náhodně sestrojíme obvod

z pevné volbě

$\sum x_i \leq \frac{1}{4}$ | $\sum x_i \geq 3/4$

x jsem nevím

	$\sum x_i \le \frac{1}{4}$ $\Pr[C(x)=1]$	$\sum x_i \ge \frac{3}{4}$ $\Pr[C(x)=1]$
$C_0(x)$ = náhodně zvolený vstupní bit	$\le \frac{1}{4}$	$\ge 3/4$
$C_1(x) = \Lambda$(10 log n náhodných obvodů C_0)	$\le \frac{1}{n^{20}}$	$\ge \left(\frac{3}{4}\right)^{10\log n} \ge \frac{1}{n^{10}}$
$C_2(x) = \vee$(n^15 náhodných obvodů $C_1(x)$)	$\le \frac{1}{n^5}$	$\ge 1-(1-\frac{1}{n^{10}})^{n^{15}} \ge 1-e^{-n^5}$
$C_3(x) = \Lambda$(n^2 náhodných obvodů $C_2(x)$)	$\ll 2^{-n^2}$	$\gg 1-e^{-n^4}$

pro pevné zvolené x, kde $\sum x_i \le \frac{1}{4}$ nebo $\ge 3/4$

$$\Pr[\,C(x) \text{ dává špatný výsledek}\,] \le 2^{-n^2}$$

vezměte 2^n různých x

$$\Pr[\,C(x) \text{ dává špatný výsledek na nějakém } x\,] \le 2^{-n}$$

$\longrightarrow$ náhodně zvolený $C(x)$ počítá s MA

(s vysokou pravděpodobností)

$\longrightarrow \exists C$, který počítá s MA

∎

- počítání až do $\log(n)$ je v AC^0
  - Chceme $H = \{h : \{1,\dots n\} \to \{1,\dots,\log^3 n\}\}$

    t.ž. $\forall S \subseteq \{1,\dots,n\}$ $|S| \le \log n$

    $\exists h \in H$ $h(S)$ je prosté, tj. $|h(S)| = |S|$.
  - Takové H existuje velikosti $2\log^2 n$.

Dk: $H = \{ h_p(x) = x \bmod p \mid p \text{ je jedno z prvních } 2\log^2 n$

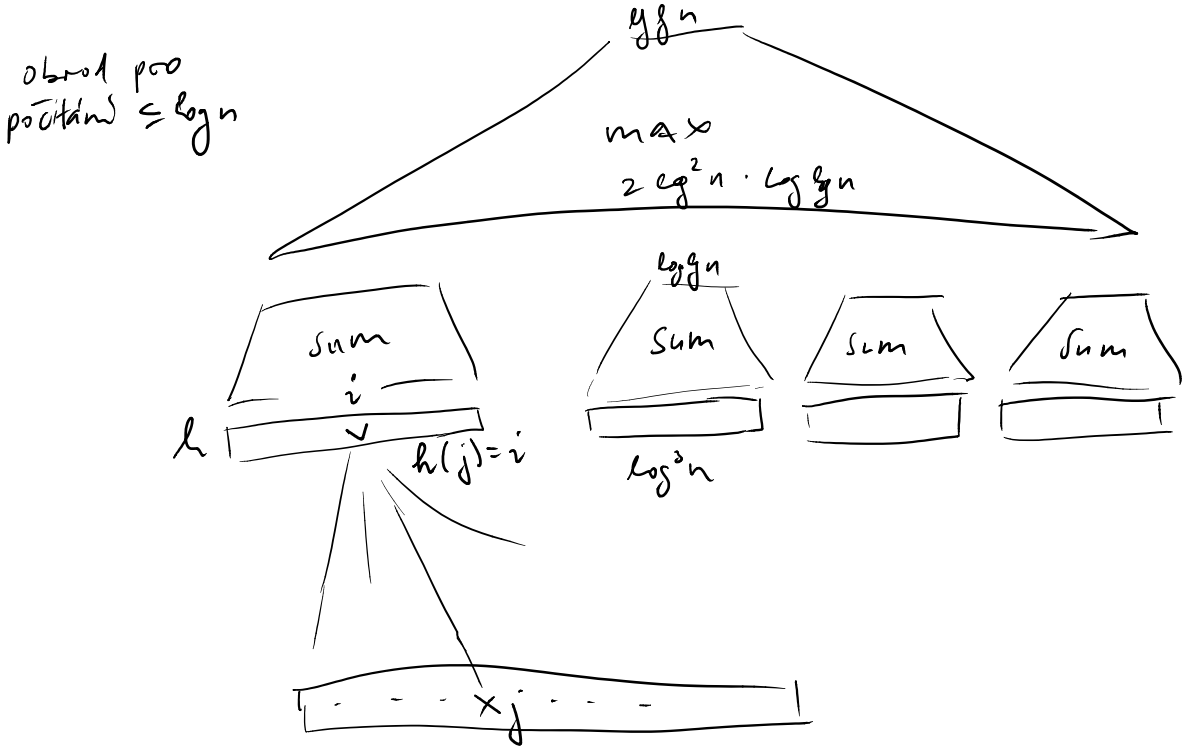
prvočísel }

dle čínské věty o zbytcích

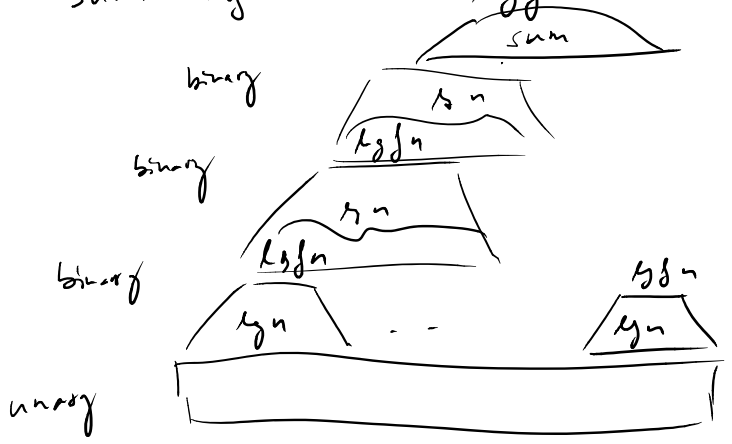
$\forall x \neq x' \in \{1,\dots,n\}$, $h(x) = h(x')$ pro nejvýše $\log n$ fcí $h \in H$.

pro danou $S \subseteq \{1,\dots,n\}$, nejvýše $|S| \cdot \log n$ špatných fcí $h \in H$

$\longrightarrow$ $\exists$ dobré $h \in H$ pro každou S velikosti $\leq \lg n$



sum $\log^3 n$ bit $\longrightarrow$ $\log \lg n$ bit



$(\lg \lg n)^2$

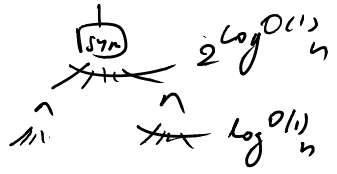
$\lg n \cdot \lg \lg n$

$\log^2 n$

$\log^3 n$ bit

Redukce hloubky obvodu

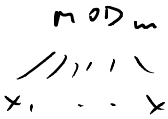
ACC 



$m = p_1^{a_1} \cdot p_2^{a_2} \cdot \ldots \cdot p_k^{a_k}$ $p_1 \ldots p_k$ prvočísla, různá

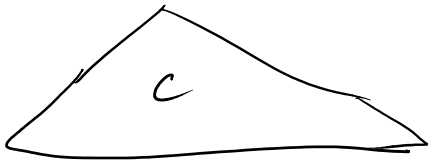
pro jednoduchost budeme uvažovat pouze případ

$a_1 = a_2 = \ldots = a_k$

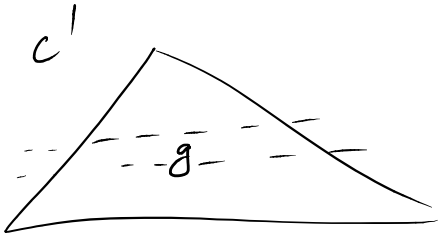
- 

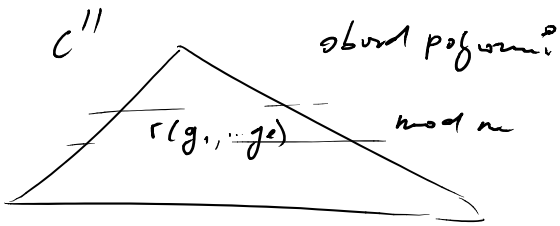
je ekvivalentní 

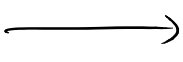
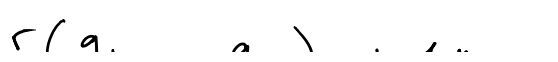
1 iff $m \mid \sum_{i=1}^{\ell} x_i$ 1 iff $m \mid \sum_{i=1}^{\ell} x_i$

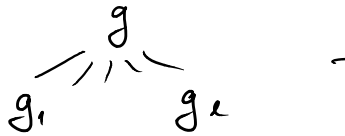
- ACC 

vstupy prvé z předchozí vrstvy, všechny hradla na dané vrstvě stejného typu.

- 





$\longrightarrow \quad r(g_1, \dots g_\ell) \mod m$

a) $g = \mathrm{MOD}_m(y_1, \dots y_\ell) \quad \Rightarrow \quad r = \sum_{j=1}^{\ell} y_j \mod m$

$r(y_1 \dots y_\ell) \bmod m \in \{0,1\}$

pro $y_1 \dots y_\ell \in \{0,1\}$

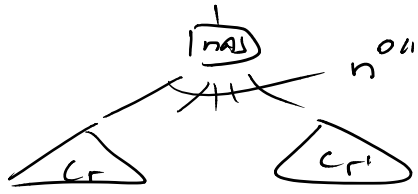
b) $g = \mathrm{OR}(y_1 \dots y_\ell)$

$$r = 1 - \prod_{i=1}^{\log^2 n} \left(1 - \sum_{j=1}^{\ell} a_{ij}\, y_j\right) \mod 2$$

a_{ij} náhodné

- každá vrstva C'' má přiřazeno modul m, a pojmy v dané vrstvě jsou za 0/1 ostrupeň také 0/1 mod m.

C'' je pěkné $\longrightarrow$ C'''



zafixujeme náhodné bity v nezávislých kopiích tak, abychom měli vždy správný výsledek

- zredukujeme hloubku C'' – postupně kolabujeme hrany dvě vrstvy pod MAJ a dostáváme jiným symetrickou funkci.

potřebujeme pojmy: $P_k(x) : \mathbb{Z} \to \mathbb{Z}$

$\forall m, \forall k, \forall x$

$x = 0 \mod m \iff P_k(x) = 0 \mod m^k$

$x = 1 \mod m \iff P_k(x) = 1 \mod m^k$

$x = 1 \mod m \iff P_k(x) = 1 \mod m^k$

- $P_2(x) = 3x^2 - 2x^3$
- $P_{2^i}(x) = P_2(P_{2^{i-1}}(x))$ pro $i > 1$

  $P_k(x) = P_{2^j}(x)$ pro $2^{j-1} < k < 2^j$

Dk:

- $x = 0 \mod m \Rightarrow x = cm \Rightarrow P_2(x) = c'm^2 \Rightarrow P_2(x) = 0 \mod m^2$

  $x = 1 \mod m \Rightarrow x = cm+1 \Rightarrow P_2(x) = 6cm + 3 - 2(2cm+1)(cm+1)$

  $= 6cm + 3 - 4cm - 2 - 2cm$

  $= 1 \mod m^2$

- indukce podle i

  $x = 0 \mod m \Rightarrow P_{2^{i-1}}(x) = 0 \mod m^{2^{i-1}}$

  $y = 0 \mod m' \Rightarrow P_2(y) = 0 \mod (m')^2$

  $P_{2^i}(x)$ m^{2^i}

  $x = 1 \mod m \Rightarrow \ldots 1$

  1 $\quad P_{2^i}(x) = 1 \mod m^{2^i}$

- $\deg P_k = k^2 - 1$ pro $k = 2^i$

$$3 \cdot \left[\left(\frac{k}{2}\right)^2 - 1\right] = \frac{3}{4}k^2 - 3 \leq k^2 - 1 \quad \checkmark$$

- norma $P_k(x) \leq 5^{k^2-1}$

$3N^2 + 2N^3 \leq 5N^3$

$$5 \cdot \left(5^{\left(\frac{k}{2}\right)^2 - 1}\right)^3 \leq 5^{\left(\frac{k^2}{4}-1\right)3+1} \leq 5^{\frac{3k^2}{4}-2}$$

součet koeficientů v absolutní hodnotě

- $x \overline{\bmod} m = \begin{cases} x \bmod m, & \text{pokud } x \bmod m < \frac{m}{2} \\ (x \bmod m) - m & \text{jinak} \end{cases}$

  tj. $x \overline{\bmod} m \in \left[-\frac{m}{2}, \frac{m}{2}\right]$

Lemma. $r(x_1, \dots x_\ell)$ polynom normy N.

$m^k \geq 2N+1$

Pak $\forall a_1, \dots a_\ell$ t.ž. $(a_i \bmod m) \in \{0,1\}$

je

$r(a_1 \bmod m, a_2 \bmod m, \dots, a_\ell \bmod m) =$

$r(P_k(a_1), P_k(a_2), \dots P_k(a_\ell)) \overline{\bmod} m^k$

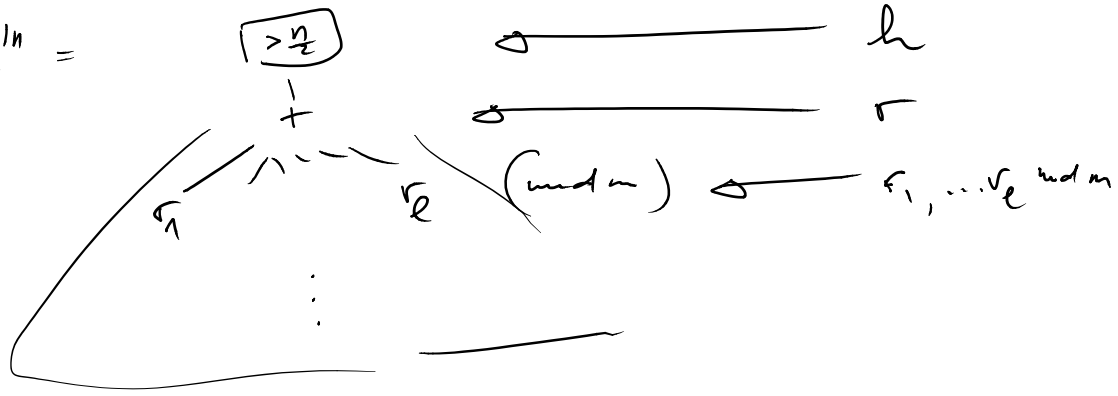
Dk: $r(a_1 \bmod m, \dots a_\ell \bmod m) =$

$= r(P_k(a_1) \bmod m^k, \dots P_k(a_\ell) \bmod m^k)$

$= r(P_k(a_1) \bmod m^k, \dots P_k(a_\ell) \bmod m^k) \overline{\bmod} m^k$

$= r(P_k(a_1), \dots, P_k(a_\ell)) \overline{\bmod} m^k$ ∎

$C^{in} =$



opakovaně zkalkulujeme horní dvě vrstvy:

zvol k t.ž. $m^k \geq 2\,\mathrm{norm}(r) + 1$

recall. $(r_i \bmod m) \in \{0,1\}$ na 0/1 odpovídá r_i

$$r(r_1(y_1,\dots y_t) \bmod m, \dots r_\ell(y_1,\dots y_t) \bmod m) =$$
$$= r(P_k(r_1(y_1,\dots y_t)), \dots, P_k(r_\ell(y_1,\dots y_t))) \bmod m^k$$

↑ lemma

polož $r'(y_1,\dots y_t) = r(P_k(r_1(y_1,\dots y_t)), \dots P_k(r_\ell(y_1,\dots y_t)))$

$h'(z) = h(z \bmod m^k)$

Pak $\forall y_1,\dots y_t \in \{0,1\}$

$$h(r(r_1(y_1,\dots y_t) \bmod m, \dots r_\ell(y_1,\dots y_t) \bmod m)) =$$
$$= h(r'(y_1,\dots y_t) \bmod m^k) = h'(r'(y_1,\dots y_t)).$$

$\deg r' \le \log^{O(1)} n$, pokud $k = \log^{O(1)} n$ a $\deg r_i = \log^{O(1)} n$

$\text{norm } r' \le 2^{\log^{O(1)} n}$

reference: Beigel-Tarui: "On ACC".

Michal at 23. 4. 2014 14:49

[Williams 2010]: $NEXP \not\subseteq ACC^0$

vhodnější slabší výsledek: $E^{NP} \not\subseteq ACC^0$

- silnější výsledek plyne s použitím [IKW]: $NEXP \subseteq P/poly$
  $\Rightarrow NEXP$ má svědky v $P/poly$

- pro důkaz bude nutné potřebovat rychlý algoritmus pro

ACC^0-SAT: vstup: C ... ACC^0 obvod s n proměnnými.

a velikosti $n^{\log n}$

výstup: $[\exists x : C(x) = 1]$?

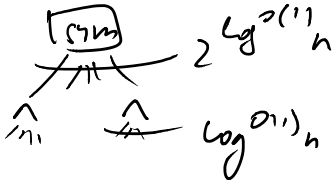
- $ACC^0\text{-}SAT \in DTIME(2^{n-n^\varepsilon})$

pro $3SAT \in DTIME(1.\ldots^n)$
$k\text{-}SAT \in DTIME((2-\varepsilon)^n)$

ETH (exponential time hypothesis)

Dva kroky

1) $C \in ACC^0 \longrightarrow C''$

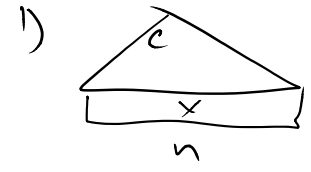


pouze $n' = n - \log^{O(1)} n$ proměnných

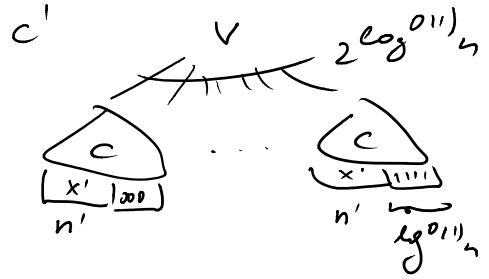
2) v čase $n' 2^{n'}$ určí splnitelnost C''

C'' splnitelné $\Leftrightarrow$ C splnitelné

1)



$\longrightarrow$



$C \in SAT \equiv C' \in SAT \equiv C'' \in SAT$

redukce hlavolamů z minula

předpokládejme $C'' \Rightarrow$ [h] $h : \mathbb{Z} \to \{0,1\}$

$r(x_1 \ldots x_n)$

p
s.t. $\log^{O(1)} n$, velikost $2^{\log^{O(1)} n}$

2) $x \in \{0,1\}^n$ $S_x = \{ i ;\ x_i = 1 \}$

$g(S)$ = koef. mnohočlenu $\prod_{i \in S} x_i$ v $r(x_1 \dots x_n)$

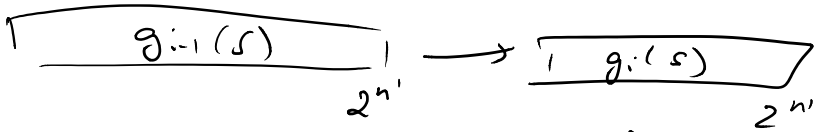
↳ spočitatelné v čase $2^{\log^{O(1)} n}$ $\forall S,\ |S| \le \log^{O(1)} n$

$\forall x ;\quad r(x) = \sum_{T \subseteq S_x} g(T)$

$\Rightarrow\quad g_i(S) = \sum_{T \subseteq S} g(S)$

$T \cap \{i+1, \dots n\} = S \cap \{i+1, \dots n\}$

- $g_0(S) = g(S)$ & $g_n(S_x) = r(x)$
- $g_i(S) = \begin{cases} g_{i-1}(S) & i \notin S \\ g_{i-1}(S) + g_{i-1}(S \setminus \{i\}) & i \in S \end{cases}$



v čase $2^{n'}$

$\Rightarrow$ lze spočítat $g_n(S)$ v čase $n' \cdot 2^{n'}$

pokud známe tabulku pro $g_0(S) = g(S)$

$\rightarrow$ z tabulky $g_n(S) = r(x)$ s pomocí tabulky pro $\underline{h}$

určím úplnost $C''(x')$

$\leftarrow$ úplnost $C(x)$

– dal $DTIME(2^{n'} \cdot poly(n'))$ alg. pro $n = n - \log^{O(1)} n$

(lze až pro $n' = n - n^{\varepsilon}$ záleží na hloubce C)

Před. $EXP^{NP} \subseteq ACC^0$ (*)

Důkaz $\exists L \in NTIME(2^n) \Rightarrow L \in NTIME(o(2^n))$

↑ spor s nedet. časovou hierarchií.

- $\forall x \; \exists \varphi_x$ ... CNF formule velikosti $2^n \cdot poly(n)$ t.ž.

  $x \in L$ iff φ_x je splnitelná

  $x \to \varphi_x$ je počítatelné polynomiálním obvodem C_x

- existuje alg. v E^{NP}, který pro obvod C_x nalezne splňující ohodnocení pro φ_x kódované obvodem C_x

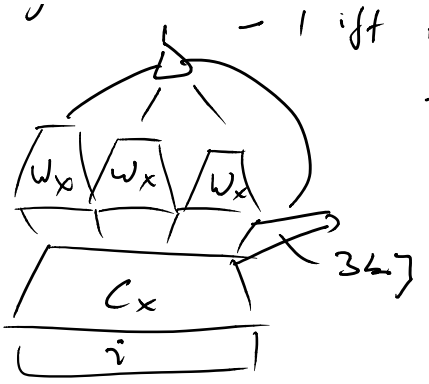
  jelikož $E^{NP} \in ACC^0$, existuje obvod W_x (z ACC^0), který kóduje splňující ohodnocení pro φ_x.

Brání: C_x na vstupu $i \in \{0,1\}^{n+O(\log n)}$ vypíše indexy tří proměnných obsažených v i-té disjunkci φ_x spolu se znaky určujícími, zda proměnná jsou negované!

Kdyby C_x byl ACC^0 obvod, pak následující obvod by byl také ACC^0:

– 1 iff i-tá klauzule φ_x je splněná pro W_x

T_x :



– 1 iff i-tá kolonka ψ_x je splněná pro W_x

pomocí algoritmu pro ACC⁰-SAT lze otestovat v čase $o(2^n)$, zda T_x dává 1 pro $\forall i$.

→ alg. pro L: uhodni C'_x (ACC⁰ obvod ekvivalentní C_x), ověř, že je $C'_x \equiv C_x$, uhodni W_x, otestuj, že T_x dá vždy 1.

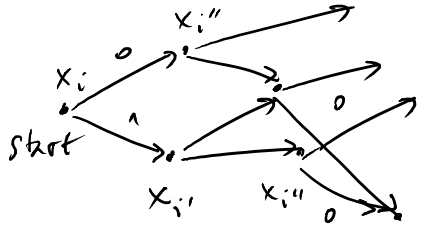
→ NTIME($o(2^n)$)

pomocí ACC⁰-SAT (⊕xc) ↑ netriviální, ale elementární

Michal at 29. 4. 2014 21:18

Branching Programs

BP, OBDD, switching & rectifier networks



. . .

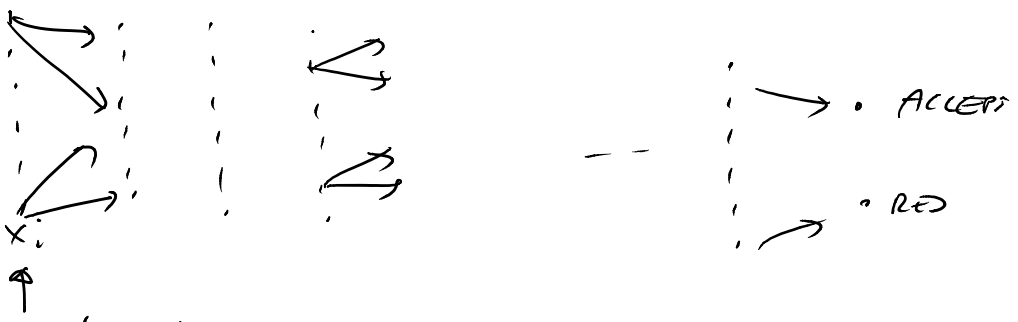
→ ACCEPT

→ REJECT

- $L \in DSPACE(\log n) \Rightarrow L$ má branching program

polynomiální velikost.

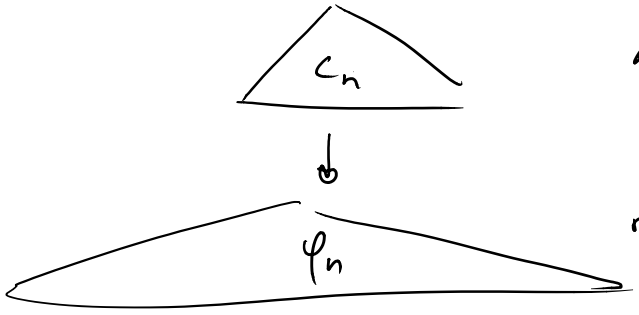
Dk:



- možná konfigurace stroje pro L
- každá konfigurace čte právě jeden bit vstupu x_i:
  → přejde do jiné konfigurace vzniklé na hodnotě x_i

- $L/poly$ = branching prog. poly velikosti.
- $L \in ACC^0 \Rightarrow$ L má BP polynomiální délky a konstantní šířky

Dk:



ACC^0 obvody

rozbal na formuli

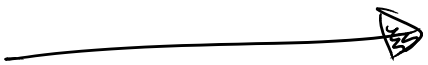
ACC^0

tj. fce dosahující AND, OR, mod-m

$poly(n)$ velikost

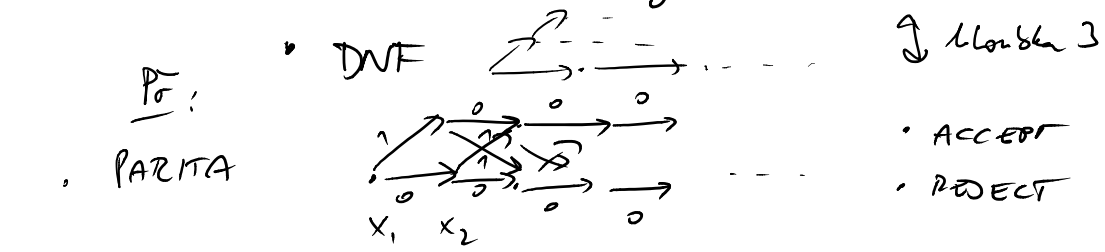
- φ_n lze vyhodnotit zleva doprava s použitím zásobníku konstantní velikosti

x_1 x_2 x_3

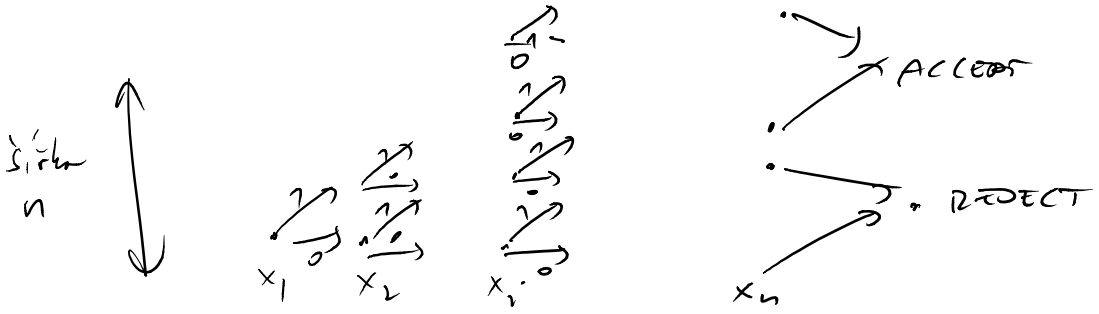


po přečtení přísušných proměnných uprav hodnoty na zásobníku

→ BP šířky 2^d, aby v daný okamžik reprezentoval možné hodnoty na zásobníku.



• MAJ Konstantní šířka?



• Barringtonova věta

Booleovskou formuli hloubky d lze simulovat branching programem šířky 5 & délky 2^d.

• Ben-Or & Cleve

Formuli s $+, \cdot$ nad okruhem R hloubky d s proměnnými $x_1, \dots, x_n$ lze simulovat registrovým programem délky 4^d se třemi registry (nad R).

↓

instrukce: $r_i \leftarrow r_i \pm r_j * x_k \quad i \neq j$

$r_i \leftarrow r_i \pm x_k$

$r_i \leftarrow r_i \pm c \quad c \in \mathbb{R}$

Dk: Pro formuli φ, ať $r_1 \leftarrow r_1 + r_2 * \varphi(x_1, \dots x_n)$

- indukce podle složitosti formule:

složitost 1: $f_k = x_i$

$r_1 \leftarrow r_1 + r_2 * x_i$ ✓

složitost i: a) $f_k = f(x_1, \dots x_n) + g(x_1, \dots x_n)$

$r_1 \leftarrow r_1 + r_2 * f(x_1, \dots x_n)$
$r_1 \leftarrow r_1 + r_2 * g(x_1, \dots x_n)$
} existuje dle ind. předpokladu

b) $f_k = f(x_1, \dots x_2) * g(x_1, \dots x_2)$

$r_3 \leftarrow r_3 + r_2 * f(x_1, \dots x_2)$
$r_1 \leftarrow r_1 + r_3 * g(x_1, \dots x_2)$
$r_3 \leftarrow r_3 - r_2 * f(x_1, \dots x_n)$
$r_1 \leftarrow r_1 - r_3 * g(x_1, \dots x_2)$

existuje dle ind. předpokladu

→ pgm $r_1 \leftarrow r_1 + r_2 * \varphi(x_1, \dots x_n)$

délky $\leq 4^d$, max 3 registry ∎

Nastav $r_1 = 0$, $r_2 = 1$, r_3 = cokoliv

→ $r_1 \leftarrow \varphi(x_1, \dots x_n)$ ✓

→ $r_1 \leftarrow \varphi(x_1, \ldots x_n)$ ✓

→ registrový pgm se 3 Booleovskými registry lze simulovat Turingův pgm nad Σ.

(slabší verze Turingovy věty)