Výpočetní složitost - 2016

18. října 2016 21:37

- 2/0 zk
- zkouší se z probrané látky
- letní přednáška Strukturální složitost
  (od příštího roku Strukt. slož. a Výp. slož. alternují v létě)

- Výpočetní složitost
  - kolik kroků je potřeba k nalezení řešení problému

  & kolik prostoru, náhodnosti, ...

  → významná mimo informatiku - algoritmické procesy
    - fyzika, biologie, ekonomie, ...

<u>tento semestr:</u> neuniformní výpočetní modely

výpočetní problém : $f : \{0,1\}^* \to U$ (vstup → výstup)

$U = \{0,1\}$ ... booleovská funkce, rozhodovací problém

$U = \mathbb{Z}, \mathbb{Q}$ ... obecná číselná funkce, optimalizace (nejkratší cesta, ...)

$U = \{0,1\}^*$

částečné funkce (promise problems, gap problems, ...)

<u>uniformní algoritmus A pro f</u> : $x \in \{0,1\}^* \longrightarrow A(x) = f(x)$

<u>neuniformní algoritmus $\mathcal{A}$ pro f</u> : $f = \{f_n\}_{n \geq 0}$

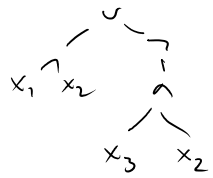
$f_n : \{0,1\}^n \to \{0,1\}$

$f_n = f|_{\{0,1\}^n}$

$\mathcal{A} = \{A_n\}_{n \geq 0}$ A_n počítá f_n.

<u>Př.:</u>
1) A_n má v sobě tabulku f_n
2) A_n má v sobě prvočíslo pro hašování dat velikosti n.
3) A_n použije jiný postup na vstupech sudé a liché délky

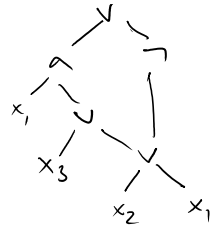
- neuniformní modely:
  - Booleovské formule

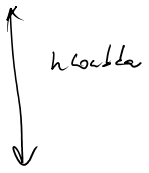


$f(x_1, x_2, x_3) = \ldots$

posloupnost formulí $\{\phi_n\}_{n \geq 0}$, ϕ_n počítá f_n.

  - Booleovské obvody





posloupnost obvodů $\{C_n\}_{n \geq 0}$; C_n počítá f_n.

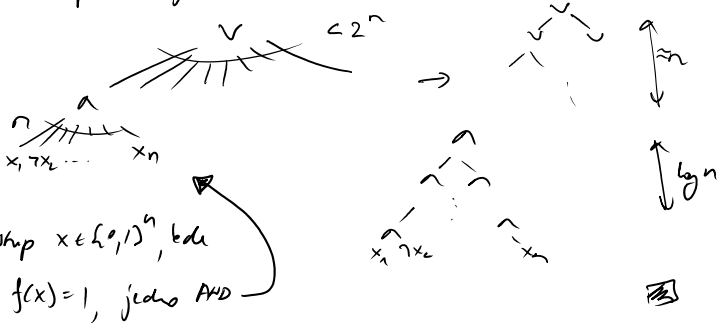
- uvidíme další – branching programy,
  aritmetické výrazy
  polynomy, ...

zajímají nás velikost, hloubka, šířka ... obvodu, f_n, ...
jak roste s velikostí vstupu.

Věta: $\forall f_n : \{0,1\}^n \to \{0,1\}$ existuje obvod s binárními AND, OR a unárním NOT velikosti $\log n \, 2^n$, který počítá f_n.

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

Dk: obvod implementující DNF nebo CNF formuli pro f_n



pro každý vstup $x \in \{0,1\}^n$, kde $f(x) = 1$, jedno AND

Věta: $\exists f_n : \{0,1\}^n \to \{0,1\}$ t.ž. nejmenší obvod s bin.

AND, OR a unárními NOT, kterých je použití,
je velikost alespoň $\frac{2^n}{10n}$.

Dk: počítací argument:
- 2^{2^n} různých fcí $f_n: \{0,1\}^n \to \{0,1\}$
- počet obvodů velikosti s
$\leq s^{2s} \cdot (n+3)^s$
↑ # spojic ↑ počet možných hradel

pro $s = \frac{2^n}{10n}$ je $\# < 2^{2n \cdot \frac{2^n}{10n}} \cdot (n+3)^{\frac{2^n}{10n}} << 2^{2^n}$

⇒ většina funkcí potřebuje exponenciálně velké obvody.

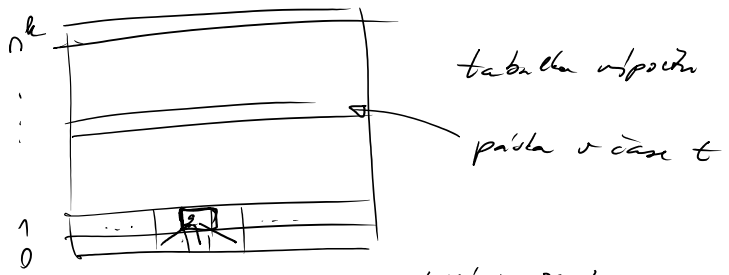
- $\forall f: \{0,1\}^* \to \{0,1\}$ existuje posloupnost obvodů $\{C_n\}_{n \geq 0}$ velikost $C_n \leq O(2^n/n)$.

Otázka: $f \in NP \Rightarrow$ f má posloupnost obvodů polynomiální velikosti (tj. C_n má velikost $n^{O(1)}$)

Domněnka: SAT nemá polynomiálně velké obvody

Věta: $f \in P \Rightarrow f$ má polynomiálně velké obvody

Dk:



každé políčko záleží pouze na třech předchozích

⇒ obvod velikosti $O(n^{2k})$

Důsledek:
- NP nemá polynomiálně velké obvody ⇒ $P \neq NP$.

Domněnka (Kolmogorov) Funkce v P mají lineárně velké obvody

„nejlepší možný odhad": $f \in P$ t.ž. $|C_n| \geq 5n$.

Def: $s(n): \mathbb{N} \to \mathbb{N}$ $SIZE(s(n)) = \{ f: \{0,1\}^* \to \{0,1\}$, $\exists \{C_n\}_{n \geq 0}$, C_n počítá f_n }

& C_n má velikost $\leq S(n)$ }

$P \subseteq \bigcup_{k \geq 0} SIZE(n^k + k)$

Spor k algoritmům: rádcí funkce $g: \mathbb{N} \to \{0,1\}^*$

- algoritmus A + rádcí funkce g
- při výpočtu na vstupu velikosti n dostaneme zadarmo $g(n)$, tj. A rozšíříme $(x, g(|x|))$ jako svůj vstup.
- měříme délku $|g(n)|$

$P/poly$ ... funkce pro které existuje algoritmus pracující v poly-čase s rádcí fcí g, kde $|g(n)| \leq n^k$ pro nějaké k nezávislého n.

P/f ... — " —
$|g(n)| \leq f(n)$.

Věta: $\forall f$ $f \in P/poly \iff f$ má obvody polynomiální velikosti.

Dk: "$\Leftarrow$" $g(n)$ = popis obvodu C_n
A vyhodnotí C_n na daném vstupu

"$\Rightarrow$" $g(n)$ zadrátujeme do obvodu simulujícího výpočet A. ∎

Michal Koucký at 25. 10. 2016 23:05

Věta: $\forall k \; \exists L \in EXP \; \forall \bar{c} \cdot L \notin SIZE(n^k + k)$

Dk: Algoritmus A pro L:
na vstupu x délky n

$a_1 = \overbrace{0\cdots0}^{n}$, $a_2 = 00\cdots01$, ... $a_{2^n} = \overbrace{111\cdots1}^{n}$

$C_0 := \{$obvody velikosti $\leq n^k + k\}$

$i := 0$

opakuj dokud $C_i \neq \emptyset$:

- pokud většina obvodů v C_i dává na a_i výstup 0, definuj $t_i := 1$, jinak $t_i := 0$.
- $C_{i+1} = \{ C \in C_i : C(a_i) = t_i \}$
- $i := i + 1$;

pokud je $x = a_j$, kde $j < i$, vystup t_j
jinak výstup 0.

ud.

- algoritmus A ⁹ je v EXP, počítající v DTIME($2^{n^{2k}}$).
- žádná rodina obvodů $\{C_n\}_{n \geq 0}$ velikosti $\leq n^k + k$ nespočítá stejnou fci jako A. ∎

Věta: $\forall k \; \exists L \in PSPACE$ t.ž. $L \notin SIZE(n^k + k)$. Dk: stejný ⁹.

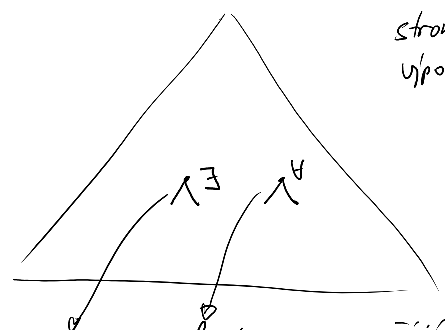
Otázka: Pro kterou nejmenší třídu lze dokázat podobné tvrzení?

Věta: (Karp-Lipton): $\forall k, \exists L \in NP^{NP^{NP}}$ t.ž. $L \notin SIZE(n^k + k)$

důkaz níže

$NP^{NP^{NP}}$?

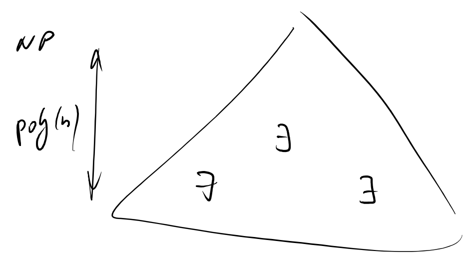
alternující Turingovy stroje: zobecnění nondeterminismu

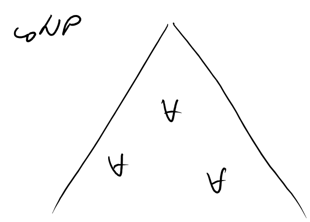


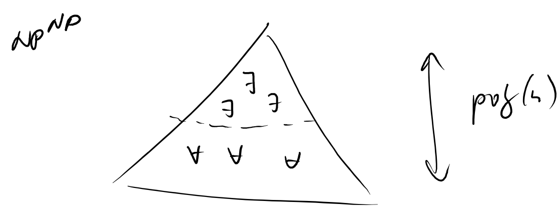
strom možných výpočtů na daném vstupu x (pesnim)

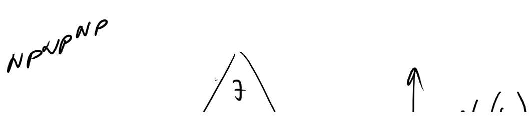
konf. je přijímající ⟺ alespoň z jedna z následujících konf. je přijímající

konfigurace je přijímající ⟺ obě následující konfigurace jsou přijímající

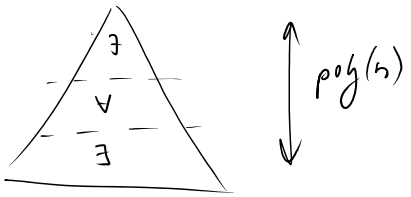






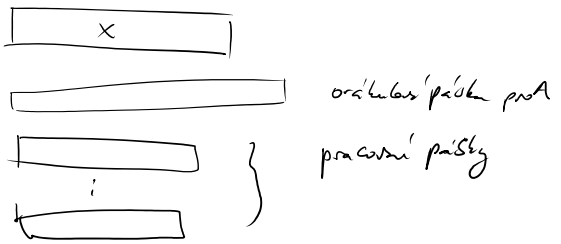


$NP^{NP^{NP}}$



Orákula:

$A \subseteq \{0,1\}^*$



dotazy $u \in A$ vyřízeny v jednom kroce

NP^A ... nedeterministický výpočet v pol. čase s orákulem A

P^A ... deterministický —"— .

$\Sigma_k\text{-SAT} = \{\varphi;\ \varphi$ je pravdivá Σ_k-SAT fle$\}$

Σ_3-SAT fle $\quad \exists x_1 x_2 \dots x_{n^k}\ \forall y_1 \dots y_{n^k}\ \exists z_1 \dots z_{n^k} \varphi(\overline{x}, \overline{y}, \overline{z})$

Σ_k-SAT ... k alternujících kvantifikátorových bloků

Σ_3-SAT je úplný pro $NP^{NP^{NP}}$, tj.

$\Sigma_3\text{-SAT} \in NP^{NP^{NP}}$

a každý problém $L \in NP^{NP^{NP}}$ je polynomiálně převeditelný na Σ_3-SAT v pol. čase.

$NP^{NP^{\cdots^{NP}}}\Big\}k = \Sigma_k \qquad \Sigma_1 = NP$

$L \subseteq \{0,1\}^* \qquad coL = \{0,1\}^* \setminus L$

$coNP^{NP^{NP^{\cdots^{NP}}}} = \Pi_k \qquad \Pi_1 = coNP$

$PH = \bigcup_k \Sigma_k$ ... polynomiální hierarchie.

$PH \subseteq PSPACE$

Dk: (Karp-Lipton):

$S = \{ a01^n01^m$; a je počáteční úsek prarodiční

tabulky fce $f: \{0,1\}^n \to \{0,1\}$, která je

počítaná obvod velikosti $m \}$

$S \in NP$ . . . pro vstup $x = a01^n01^m$ uhodnu obvod velikosti $\leq m$ a ověřím, že a odpovídá jeho tabulce (částečně).

idea: alg pro L

na vstupu x $|x| = n$

uhodni $\underline{a} \in \{0,1\}^{n^{2k}+k}$ t.ž.

$\underline{a}$ není počítaná obvodem velikosti n^k+k,

tj. $a01^n01^{n^k+k} \notin S$.

pokud x je j-tý řetězec, $j < |a|$, přijmi pokud $a_j = 1$

jinak odmítni.

problém: $\underline{a}$ není jednoznačné, alg ↗ může přijímat všechno

řešení:

$S' = \{ a01^n01^m,\ a \in \{0,1\}^*$, buď $a01^n01^m \in S$ nebo $\exists$ lex. menší $a' < a$, $a \in \{0,1\}^{|a|}$ t.ž. $a01^n01^m \notin S \}$

$S' \in NP^{NP}$

$\to$ použij předchozí algoritmus s S'. ⊠

Věta: $\forall k \geq 0\ \exists L \in NP^{NP}$ t.ž. $L \notin SIZE(n^k+k)$

Dk: Dvě možnosti

1) $NP \nsubseteq PSIZE = \bigcup_{k>0} SIZE(n^k+k)$

(pak závěr plyne automaticky)

2) $NP \subseteq PSIZE \Rightarrow NP^{NP^{NP}} \subseteq NP^{NP}$

$\Sigma_3 \subseteq \Sigma_2$

Dk: φ bool. fce

$\varphi(x_1, x_2, \ldots x_n)$ je splnitelná

$\Leftrightarrow$

$\varphi(x_1, \ldots x_{n-1}, 0)$ nebo $\varphi(x_1, \ldots, x_{n-1}, 1)$

je splnitelná

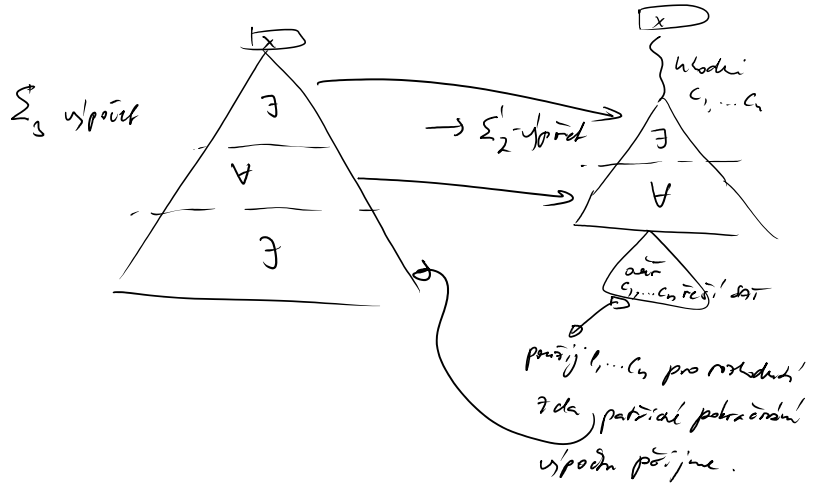
⇒ máme-li obvody $C_1, C_2, \dots C_n$, které hádají řešení SAT, můžeme si ověřit, že jsou konzistentní:

$\forall$ bool fce φ s velikostí zápisu $\le n$ ověř, zda

$$C_{|\varphi|}(\varphi) = 1 \quad \text{iff} \left(C_{|\varphi(x_1 \dots x_{n-1}, 0)|}(\varphi(x_1 \dots x_{n-1}, 0)) = 1 \right.$$
$$\left. \vee\ C_{|\varphi(x_1 \dots x_{n-1}, 1)|}(\varphi(x_1 \dots x_{n-1}, 1)) = 1 \right)$$

- pokud φ nemá žádné proměnné, vyhodnoť ji a zkontroluj, zda to odpovídá $C_{|\varphi|}(\varphi)$.

toto je coNP test.



$L \in NP^{NP^{NP}}$ t.ž. $L \notin SIZE(n^k + 6)$

$\Rightarrow L \in NP^{NP}$ t.ž. —"—

AC^0 - obvody

$O(1)$ hloubka, $poly(n)$ velikost

hradla $\wedge, \vee, \neg$

$\wedge$ AND, $\vee$ OR, $\neg$ NOT

- umí - sečíst dvě čísla v binárním zápisu
  - spočítat počet jedniček, pokud je jejich počet menší než $\log^{O(1)} n$.
  - rozhodnout zda $\sum x_i \le \log n$.
  - rozhodnout zda $\sum x_i \ge \frac{3}{4} n$ nebo $\sum x_i \le \frac{1}{4} n$

(pro ostatní vstupy dá nějaký výstup)

- neumí – spočítat paritu vstupů tj. $\sum x_i \bmod 2$

  ⇒ neumí vynásobit dvě binárně zapsaná čísla.

<u>Razborov – Smolensky '87</u>

- $\forall p, q$ prvočísla, $p \neq q$, MOD-$p \notin AC^0[q]$

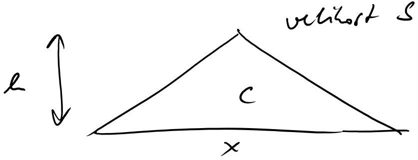
  ↑ polynomiálně velké AC^0-obvody navíc s hradly pro MOD-q $[\sum x_i \not\equiv 0 \bmod q]$

<u>předtím</u>: Furst – Saxe – Sipser '83, Ajtai '83, Håstad '86
MOD-2 $\notin AC^0$

<u>Dk</u>: důkaz ve dvou krocích

1) aproximace obvodu polynomem malého stupně nad $GF[q]$
2) MOD-p nelze approximovat polynomem malého stupně nad $GF[q]$.

<u>krok 1</u>:



<u>BÚNO</u>: OR, MOD-q, NOT

induktivně odspodu přiřadím hradlům na hladině i obvodu C polynom p_g nad $GF[q]$, který bude souhlasit s hodnotou g na všech vstupech x, až na "malou" množinu W_i.

- $W_0 = \emptyset$

$h > 0$:

a)

$p_g = 1 - p_{g'}$

b)

$$p_g = \left(\sum_{i=1}^{l} p_{g_i}\right)^{q-1}$$

c)

$$p_g = 1 - \prod_{j=1}^{k} \left(1 - \left(\sum_{i=1}^{l} a_{ji}\, p_{g_i}\right)^{q-1}\right)$$

pro náhodně zvolená $a_{ji} \in \{0,1\}$

pro pevně zvolený vstup $x \notin W_{i-1}$, kde $\forall g_i = 1$

$\exists$: $\sum_{i=1}^{t} p_{g_i}(x_i) > 0$

t … součet nad $\mathbb{Z}$.

$$Pr_{a_{i,j}}\left[p_g(x) = 0\right] \le \left(\frac{1}{2}\right)^k$$

nebo $Pr_{a_i}\left[\sum_i a_i p_g(x) \equiv 0 \mod 2\right] \le \frac{1}{2}$

$\le S_i$ hradel na i-té hladině

$Pr_{a_{ij}}$[některý z polynomů na i-té hladině počítaný špatně na x] $\le S_i \left(\frac{1}{2}\right)^k$

$\Rightarrow$ $\exists$ výběr koeficientů a_{ij} pro hradla na i-té hladině, $\forall \bar{x}$

$$|W_i| \le |W_{i-1}| + 2^n \cdot S_i \left(\frac{1}{2}\right)^k$$

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

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

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

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

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

2) polynom stupně $O(n^{1/3})$ nad $GF[q]$ nemůže počítat MOD-p na $2 - o(2^n)$ vstupech.

Dk: Sporem: pro $p = 2$

p_2 … polynom pro MOD-2 na $\{0,1\}^n \setminus W$

$$p_2'(y_1, \dots y_n) = 1 - 2\, p_2\left(\frac{1-y_1}{2}, \frac{1-y_2}{2}, \dots \frac{1-y_n}{2}\right)$$

$p_2' : \{-1,1\}^n \to \{-1,1\}$

$p_2'(y_1, \dots y_n) = \prod_{i=1}^{n} y_i$ pro $y_1, \dots y_n \in \{-1,1\}^n \setminus W'$

$W' =$ " $\frac{1-2w}{2}$ "

Nech' $f : \{-1,1\}^n \setminus W' \to GF[q]$

protože $x_i^2 = 1$, f lze reprezentovat multilineárním polynomem nad $GF[q]$

$$f(y_1, \dots y_n) = p_2' \cdot l_1 + l_2$$

kde l_1 a l_2 jsou multilineární polynomy stupně $\leq n/2$

$$\prod_{i \in I} y_i = \prod_{i=1}^{n} y_i \cdot \prod_{i \notin I} y_i \qquad \text{pro } y_1, \dots y_n \in \{-1,1\}^n \setminus W'$$

$\hookrightarrow p_2'(y_1, \dots y_n)$

$\Rightarrow$ f lze reprezentovat polynomem stupně

$$\leq \frac{n}{2} + O(n^{1/3}).$$

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

$$\#\text{fcí } \{-1,1\}^n \setminus W \to GF[q] \geq q^{2^n - |W'|} \geq 3^{2^n - o(2^n)}$$

spor ▨

- pro $p \neq 2$ se použije podobný trik

- Přibližné počítání je v AC^0

[Ajtai - Ben-Or '83]

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

Tvrzení: $AMAJ(x) \in AC^0$.

... "funkce kontrolující s $AMAJ(x)$ je v AC^0."

Dk: obvod sestrojíme náhodně

Vezmeme pevné $x \in \{0,1\}^n$ a počítejme pst, že náš obvod počítá $MAJ(x)$ na x správně:

	$\sum x_i \leq \frac{1}{4}n$ $Pr[C(x)=1]$	$\sum x_i \geq \frac{3}{4}n$ $Pr[C(x)=1]$
$C_0(x) = x_i$, pro náhodné zvolené $i \in \{1,\dots,n\}$	$\leq \frac{1}{4}$	$\geq \frac{3}{4}$
$C_1(x) = \bigwedge$ ($10 \log n$ náhodných)	$\leq \frac{1}{n^{20}}$	$\geq \left(\frac{3}{4}\right)^{10 \log n} \geq \frac{1}{n^{10}}$

$i \in \{1,\dots,n\}$

$C_1(x) = \wedge$ (10 log n nezávislých kopií $C_0(x)$) $\le \frac{1}{n^{20}}$

$C_2(x) = \vee$ (n^{15} nezávislých kopií $C_1(x)$) $\le \frac{1}{n^{5}}$

$C_3(x) = \wedge$ (n^{2} nezávislých kopií $C_2(x)$) $\ll 2^{-n^2}$

$\ge \left(\frac{3}{4}\right)^{10\log n} \ge \frac{1}{n^{10}}$

$\ge 1-\left(1-\frac{1}{n^{10}}\right)^{n^{15}} \ge 1-e^{-n^5}$

$\gg 1-e^{-n^4}$

$\Rightarrow$ pro pevně zvolené x t.ž. $\sum x_i \le \frac{1}{4}n$ nebo $\sum x_i \ge \frac{3}{4}n$

$\Pr_{C_3}[C_3(x)$ dává špatný výsledek$] \le 2^{-n^2}$

• nejvýše 2^n různých x

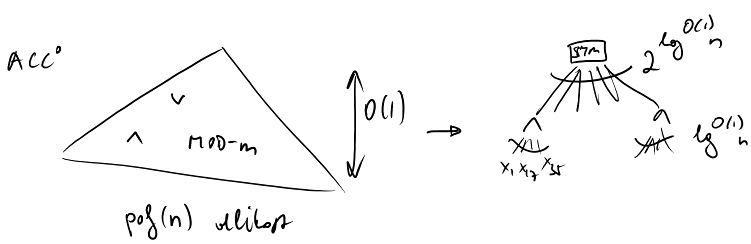
$\Pr_{C_3}[C_3(x)$ dává špatný výsledek na nějakém x$] \le 2^{-n}$.

$\rightarrow$ náhodně zvolený obvod C_3 počítá AMAJ s pstí $\ge 1-2^{-n}$.

$\Rightarrow \exists C_3$, který počítá AMAJ. QED

Michal Koucky at 29. 11. 2016 22:19

Redukce hloubky obvodu [Beigel-Tarui "On ACC"]



[SYM] ... počítá "symetrickou" fci, tj. funkci, která závisí pouze na počtu jedniček na vstupu, ale ne na jejich rozmístění.

Proč? : užitečné z hlediska zkoumání omezení ACC^0 obvodů

$$ACC^0 = \bigcup_{m \ge 1} AC^0[m]$$

• Razborov-Smolensky lze použít na redukci hloubky pro $AC^0[q]$, kde q je mocnina prvočísla. Tady nás zajímají složená m.

$m = p_1^{a_1} \cdot p_2^{a_2} \cdots p_k^{a_k}$ $\quad p_1,\dots,p_k$ jsou různá prvočísla

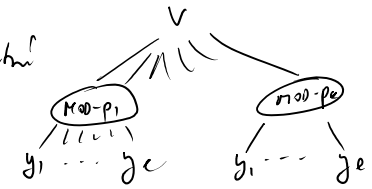
pro jednoduchost budeme předpokládat,

že $a_1 = a_2 = \dots = a_k = 1$.

(Např.: $m = 6$, o tom Raz.-Smol. nic nevíme)

- $MOD\text{-}m(x) = \begin{cases} 0 & m \mid \Sigma x_i \\ 1 & m \nmid \Sigma x_i \end{cases}$

je ekvivalentní

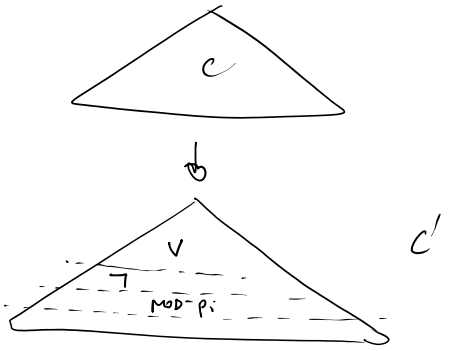


$m \mid \Sigma x_i$ iff $\forall j = 1, \dots, k \quad p_j \mid \Sigma x_i$.

Dk: "$\Rightarrow$" triviální

"$\Leftarrow$" $p_1, p_2, \dots p_k \mid \Sigma x_i \Rightarrow n.s.n.(p_1, \dots p_k) \mid \Sigma x_i$ ☒
m

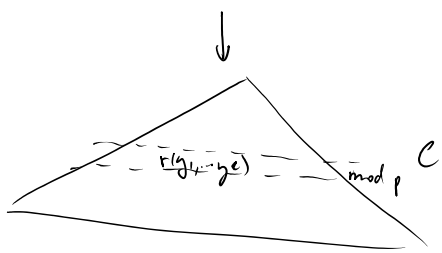
ACC^0 obvod C zredukujeme do přiřazovacího tvaru v několika krocích



MOD-m rozepíšeme jako MOD-p_i a hradla rozvrstvíme tak, že v každé vrstvě jsou pouze hradla jednoho typu:

a) ¬ nebo V
b) MOD-p_i kde p_i je stejné v rámci vrstvy.

→ C' má stále konstantní hloubku ($O(k)$) a polynomiální velikost.



V každé hradlo nahradíme polynomem, každá vrstva

asociovaná s nějakým prvočíslem p, pojmy
na další vrstvě dávají 0/1 při počítání $\bmod p$.
(s velkou pravděpodobností, konstrukce je randomizovaná)

1) $\neg(y_1) \Rightarrow r_\neg(y_1) = 1 - y_1 \mod 2$

2) $OR(y_1, \dots y_\ell) \Rightarrow r_{OR}(y_1, \dots y_\ell) = 1 - \prod_{i=1}^{4n^2} \left(1 - \sum_{j=1}^{\ell} a_{ij} y_j\right) \mod 2$

a_{ij} volby náhodně

pro pevné $y_1, \dots y_\ell$

$$\Pr_{a_{ij}}\left[r_{OR}(y_1, \dots y_\ell) = OR(y_1, \dots y_\ell) \right] \geq 1 - \frac{1}{2^{4n^2}} = 1 - \frac{1}{n^{4n}}.$$

3) $MOD\text{-}p(y_1, \dots y_\ell) \Rightarrow r_{MOD\text{-}p}(y_1, \dots y_\ell) = \left(\sum_{i=1}^{\ell} y_i\right)^{p-1} \mod p.$

Podle malé Fermatovy věty $\rightarrow$ 0/1 pro $\forall y_i$.

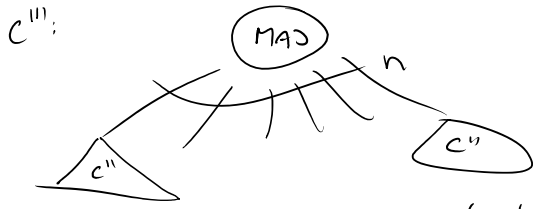
$$\text{tedy } \forall y_i \quad \sum y_i \neq 0 \Rightarrow \left(\sum y_i\right)^{p-1} = 1.$$

C'' je pravděpodobnostní: pro daný vstup $x \in \{0,1\}^n$

$$\Pr_{a_{ij}}\Big[\text{v obvodu } C'' \text{ některý pojmenovaný na vstupech z předchozí vrstvy nedává správnou hodnotu} \Big] \leq \frac{1}{n^{4n}} \cdot \text{velikost } C''' \leq \frac{1}{64}.$$

odpovídající hodnotě příslušného hradla

pro n dostatečně velké



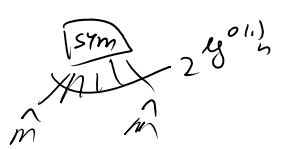
n nezávislých náhodně zvolených obvodů C''.

pro daný vstup $x \in \{0,1\}^n$, $\Pr_{a_{ij}}\left[C'''(x) \neq C(x) \right] \leq \binom{n}{n/2} \cdot \left(\frac{1}{64}\right)^{n/2}$

$$\leq 2^n \cdot \frac{1}{8^n} = \frac{1}{4^n}.$$

$\Rightarrow$ lze zafixovat a_{ij} v C'' tak, že C''' dává správný výstup na všech vstupech x.

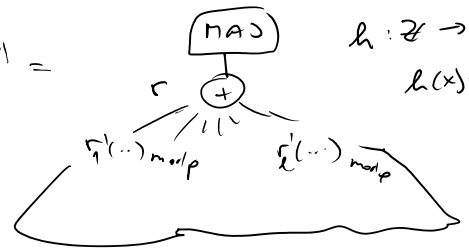
C''' zkonvertujeme do podoby



postupným kolapsem horních dvou vrstev.

postupným kolapsem horních dvou vrstev.

$C^{(n)}$ =



$h : \mathbb{Z} \to \{0,1\}$

$h(x) = \begin{cases} 0 & x \le \frac{n}{2} \\ 1 & x > \frac{n}{2} \end{cases}$

$r(y_1, \dots y_n) = \sum_{i=1}^{n} y_i$

chceme zkolabovat $r(y_1, \dots y_t)$ s pojmy $r_1'(\dots)$ ...

poctivá substituce $r(r_1'(\dots), r_2'(\dots), \dots r_t'(\dots))$

nestačí, protože $r_1'(\dots), \dots r_t'(\dots)$ dávají 0/1 pouze při mod p. Vystrčit mod p ven by takhle nepomohlo.

→ použijeme speciální polynomy, které nám to umožní

pro polynom $r(y_1, \dots y_t)$, norma $r()$ je součet koeficientů v absolutní hodnotě nad $\mathbb{R}$.
$r(\dots)$

použijeme polynomy $P_k(x) : \mathbb{Z} \to \mathbb{Z}$ +-.

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

(*) $x = 0 \bmod p \Rightarrow P_k(x) = 0 \bmod p^k$
$x = 1 \bmod p \Rightarrow P_k(x) = 1 \bmod p^k$

$P_2(x) = 3x^2 - 2x^3$

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

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

Tvrzení: P_{2^i} splňuje (*).

Dk: $i = 1$

$x = 0 \bmod p \Rightarrow x = cp \Rightarrow p^2 \mid 3x^2 - 2x^3$

$x = 0 \bmod p \to x = cp+1 \Rightarrow 3(2cp+1) - 2(2cp+1)(cp+1) =$

$3(2cp+1) \equiv x^2 \bmod p^2$ $\quad 2(2cp+1)(cp+1) \equiv x^3 \bmod p^2$

$\pmod{p^2}$

$= 6cp + 3 - 4cp - 2cp - 2 = 1$ ✓

$i > 1$

$x = 0 \mod p \Rightarrow y = P_{2^{i-1}}(x) = c \cdot p^{2^{i-1}}$

$\Rightarrow p^{2^i} \mid P_2(y)$

$x = 0 \mod p \Rightarrow y = P_{2^{i-1}}(x) = c p^{2^{i-1}} + 1$

$\Rightarrow P_2(y) = c' p^{2^i} + 1 \mod p^{2^i}$ ✓

▨

Tvrz.: pro $k = 2^i$ stupeň $P_k(x) \le k^2 - 1$

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

Dk.: indukcí:

stupeň $P_k(x) \le 3\left[\left(\frac{k}{2}\right)^2 - 1\right] = \frac{3}{4}k^2 - 3 \le k^2 - 1$ ✓

norma $P_k(x)$:

norma $P_{\frac{k}{2}}(x) = N \le 5^{\left(\frac{k}{2}\right)^2 - 1}$

norma $P_k(x) \le 3 \cdot N^2 + 2N^3 \le 5N^3 \le$

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

▨

def.: $x \overline{\mod} m = \begin{cases} x \mod m & \text{pokud } x \mod m < \frac{m}{2} \\ (x \mod m) - m & \text{pokud } x \mod m \ge \frac{m}{2} \end{cases}$

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

Lema: $r(y_1, \dots y_\ell)$ je polynom normy N.

Nechť $m^k \ge 2N+1$. Pak $\forall a_1, \dots a_\ell$ t.ž. $(a_i \mod m) \in \{0,1\}$

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

$= r(P_k(a_1), P_k(a_2), \dots, P_k(a^k)) \overline{\mod} m^k$.

Dk.: $r(a_1 \mod m, a_2 \mod m, \dots a_\ell \mod m) =$

$= r(P_k(a_1) \mod m^k, P_k(a_2) \mod m^k, \dots P_k(a_\ell) \mod m^k)$

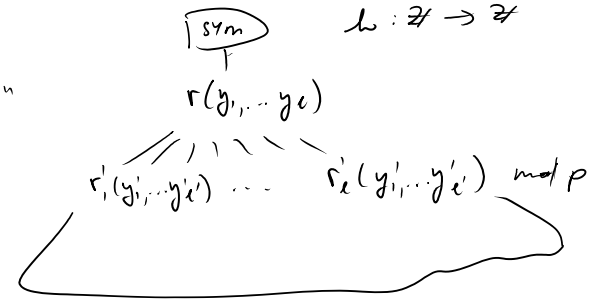
$= r(P_k(a_1) \mod m^k, P_k(a_2) \mod m^k, \dots P_k(a_\ell) \mod m^k) \overline{\mod} m^k$

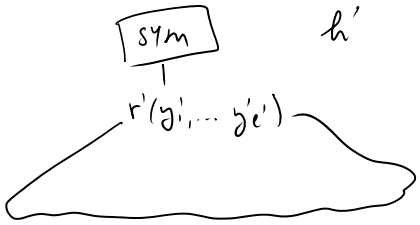
$= r(P_k(a_1), \dots P_k(a_\ell)) \overline{\mod} m^k$ ▨

---

opakovaně kolabujeme horní dvě vrstvy v C'''

opakovaně kolabujeme horní dvě vrstvy s C'''





nechť $p^k \geq 2 \text{norma}(r) + 1$

$$r'(y'_1, \dots y'_{\ell'}) = r\left(P_k(r'_1(y'_1, \dots y'_{\ell'})), \dots, P_k(r'_\ell(y'_1, \dots y'_{\ell'}))\right)$$

$$h'(z) = h(z \bmod p^k).$$

jelikož $r'(y'_1, \dots y'_{\ell'}) \bmod p^k = r(r_1(y'_1, \dots y'_{\ell'}) \bmod p, \dots r'_\ell(y'_1, \dots y'_{\ell'}) \bmod p)$

konstrukce je korektní.

pokud stupeň $r, r'_1, \dots r'_\ell = \log^{O(1)} n$,
& norma $r \leq 2^{\log^{O(1)} n}$ & norma $r'_1, \dots r'_\ell \leq \log^{O(1)} n$
pak $k \leq \log^{O(1)} n$, stupeň $r' \leq \log^{O(1)} n$,
norma $r' \leq 2^{\log^{O(1)} n}$. ∎

→ všechna h a r mi dají potřebný tvar

Michal Koucký at 29. 11. 2016 23:21

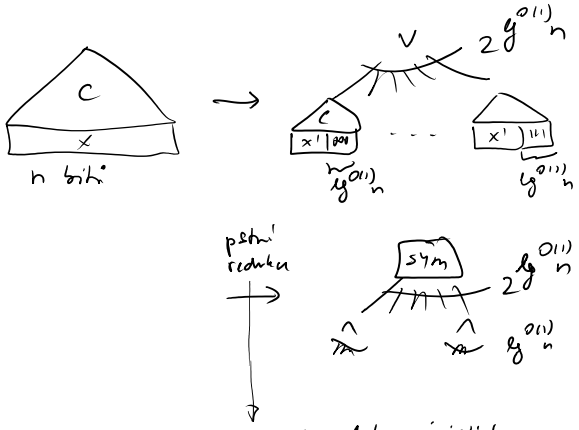
Testování splnitelnosti ACC^0 obvodu.

ACC^0 obvod C s n bitý vstupy $x_1, \dots x_n$.

$\exists a_1, \dots a_n \in \{0,1\}$ t.ž. $C(a_1, \dots a_n) = 1$?

- Triviální alg. $\text{poly}(n) \cdot 2^n$ čas

• Ještě lépe: alg. $2^{n - \lg^{(0)} n}$ čas.



lze udělat deterministicky
(výběr coefficientů a_{ij})

C v čase $2^{\lg^{(0)} n}$ přeformulujeme, tj. spočítáme

h a $r(x_1, \dots x_{n'})$

↳ stupeň $\le \lg^{(0)} n$

norma $\le 2^{\lg^{(0)} n}$.

$n' = n - \lg^{(0)} n$

Chci ověřit, zda $\exists x_1, \dots x_{n'} \in \{0,1\}$ t.ž.

$h(r(x_1 \dots x_{n'})) = 1$?

• triviální způsob – zkus všechny možnosti

→ čas $2^{n'} \cdot \text{norma}(r) \ge 2^n$!

→ tudy cesta nevede

• lépe: $x \in \{0,1\}^{n'}$ $\quad S_x = \{ i ; x_i = 1 \}$

$g(S)$ = koef. monočlenu $\prod_{i \in S} x_i$ v $r(x_1, \dots x_{n'})$

$\forall S \subseteq \{1, \dots n'\}$, $|S| \le \lg^{(0)} n'$, $g(S)$ lze spočítat v čase $2^{\lg^{(0)} n}$.

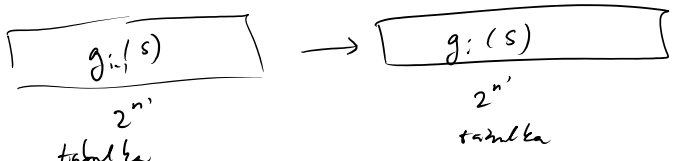
$\forall x$; $\quad r(x) = \sum_{T \subseteq S_x} g(T)$.

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

$T \cap \{i+1, \dots n'\} = S \cap \{i+1, \dots n'\}$

$g_0(S) = g(S)$ $\quad$ & $\quad g_{n'}(S_x) = r(x)$.

Spočítám $g_i(S)$ pro $\forall S \subseteq \{1, \dots n'\}$.



$$g_i(S) = \begin{cases} g_{i-1}(S) & i \notin S \\ g_{i-1}(S) + g_{i-1}(S \setminus \{i\}) & i \in S \end{cases}$$

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

pokud známe $g_0(S) = g(S)$.

→ z tabulky $g_{n'}(S_x) = r(x)$ s pomocí tabulky pro k určíme splnitelnost $C(x)$.

→ splnitelnost ACC^0 obvodu lze rozhodnout v $DTIME\left(2^{n-\frac{n}{g^{O(1)}n}}\right)$

(lze zlepšit na $DTIME(2^{n-n^\varepsilon})$, kde ε závisí na hloubce obvodu)

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

ukážeme slabší výsledek $ENP \not\subseteq ACC^0$

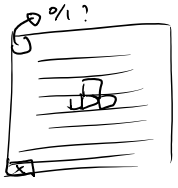
(silnější výsledek používá [IKW]: $NEXP \subseteq P/poly \Rightarrow NEXP$ má svědky v $P/poly$)

Dk:
- uvažujme $L \in NTIME(2^n)$ přijímaného strojem M.
- ukážeme, že pokud $NEXP \subseteq ACC^0$, pak $L \in NTIME(o(2^n))$. To je ve sporu s nedeterministickou časovou hierarchií:
$$NTIME(o(t(n))) \subsetneq NTIME(t(n+1))$$

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

$x \in L$ iff φ_x je splnitelná

~ ekvivalent Cook-Levinovy věty



φ_x kódujeme tabulkou výpočtu M pro x.

- tedy dal φ_x velikosti $O(2^n \cdot 2^n) = O(2^{2n})$

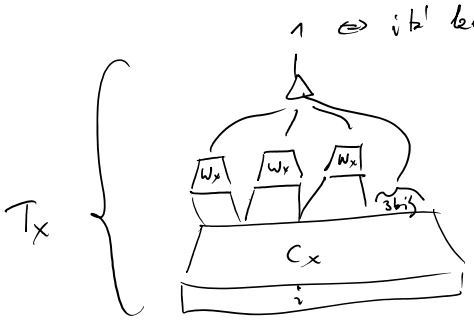
my potřebujeme $O(2^n \cdot poly(n))$
což lze získat pomocí vylepšení
techniky.

- existuje pol-time alg., který pro x sestrojí polynomiální
velký obvod C_x řešící φ_x.

C_x na vstupu $i \in \{0,1\}^{n+O(\lg n)}$ vypíše index
i-té proměnné obsažené v i-té disjunkci φ_x
spolu se třemi bity určujícími, které z těchto
proměnných jsem v i-té disjunkci negovaná

- v E^{NP} lze najít pro danou $x \in L$ splňující
ohodnocení φ_x. (Binárním vyhledáváním na proměnné φ_x)
- pokud je $E^{NP} \subseteq ACC^0$ pak pro dané $x \in L$
existuje ACC^0 obvod W_x
řešící splnitelné ohodnocení φ_x. Obvod
W_x bere jako vstup index proměnné a spočítá
její hodnotu ve splňujícím ohodnocení pro φ_x.

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



pomocí algoritmu pro ACC^0-SAT lze otestovat
v čase $o(2^n)$, zda T_x dává 1 pro všechna i.

→ algoritmus pro L: uhodni C'_x (ACC^0 obvod
ekvivalentní C_x), ověř, že $C'_x \equiv C_x$, uhodni W_x
a otestuj, že T_x dává vždy 1.
→ $NTIME(o(2^n))$.

uhodni posloupnost obvodů
ekvivalentních jednotlivým hradlům

(podobvodům) v C_x a pomocí ACC^0-SAT ověří že jejich kombinace dělají, co by měly. ■

Výpočty s nápovědou

- definice viz dříve

Věta: $P/poly = PSIZE$

↳ fce počitatelné polynomiálně velkými obvody.

Dk: „$\supseteq$" $L \in PSIZE \Rightarrow \exists$ posloupnost obvodů $\{C_n\}_{n \geq 1}$ počítající $L \cap \{0,1\}^n$.

$$|C_n| = poly(n)$$

definuj radící fci $g(n)$ = "binárně zakódovaný popis obvodu C_n"

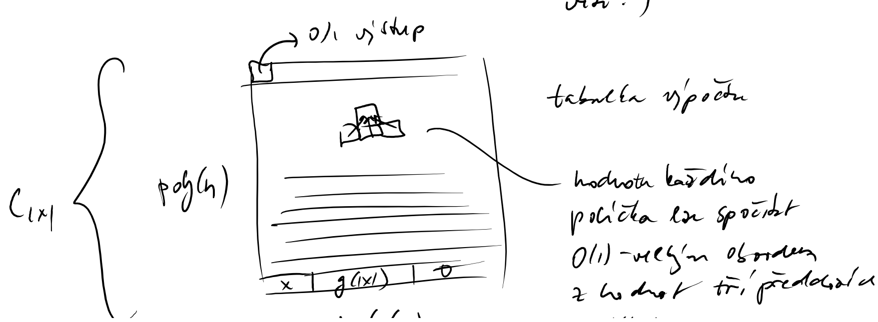
$L \in P/poly$:
- radící fce
- alg na vstupu x dostane $g(|x|)$, tedy popis $C_{|x|}$, který vyhodnotí na vstupu x a vydá to jako výstupní hodnotu

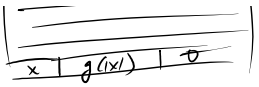
běží v čase $poly(|C_{|x|}|) = poly(|x|)$

$\Rightarrow L \in P/poly$ ■

„$\subseteq$" $L \in P/poly$ tj. máme poly-time alg. a radící fci $g(n)$ t.ž. na vstupu x, $g(|x|)$ algoritmus vydá, zda $x \in L$.

Algoritmus se dá simulovat polynomiálně velkým obvodem, ve kterém je vstup x a nápověda $g(|x|)$ je zadrátovaná (jako v Cook-Levinově větě.)





poly(n)

O(1) - velký obvod
z hodnot tří předchozích
políček
→ výstup lze spočítat
poly(n) - velký obvod $C_{|x|}$

→ $C_1, C_2, \ldots$ $L \in$ P/size ∎

NP/poly ... "neuniformní NP"

$L \in$ NP/poly, pokud existuje radící fce
$g: \mathbb{N} \to \{0,1\}^*$ a nedeterministický
algoritmus A pracující v poly-čase, t.ž.
$\forall x$ $\quad x \in L \iff A$ přijímá $x, g(|x|)$.

• nedeterministické obvody:

obvod C se vstupem $x \in \{0,1\}^n$ a $y \in \{0,1\}^{p(n)}$

říkáme, že C přijímá x, pokud $\exists y$ t.ž.

$C(x,y) = 1$.

Věta: $L \in$ NP/poly $\iff$ L je počitatelné nedeterministickými obvody polynomiální velikosti

Dk: obdobný předchozí větě.

EXP/poly: $L \subset$ EXP/poly, pokud existuje radící fce $g: \mathbb{N} \to \{0,1\}^*$, kde $|g(n)| = poly(n)$, a algoritmus A pracující v čase 2^{n^k}, pro nějakou konstantu k, t.ž.

$\forall x$ $\quad x \in L \iff A$ přijme $x, g(|x|)$.

Lema: EXP $\subseteq$ P/poly $\Rightarrow$ EXP/poly $\subseteq$ P/poly

Dk: cvičení ∎

EXP/poly $\overset{\Uparrow}{=}$ P/poly

$\Rightarrow$ nevíme, zda EXP/poly $\neq$ P/poly, neboť
nevíme, zda EXP $\subseteq$ P/poly

NEXP/poly: def. obdobně EXP/poly & NP/poly

$L \in$ coNEXP/poly, pokud $\overline{L} \in$ NEXP/poly.

Věta: $coNEXP \subseteq NEXP/poly$

Důsledek: $coNEXP/poly = NEXP/poly$

Dk.:

pro $L \in coNEXP$

A chceme nedeterministický alg. pracující v čase 2^{n^k} a radící fci $g: \mathbb{N} \to \{0,1\}^*$, t.ž. $\forall x \quad x \in L \iff A$ přijímá $x, g(|x|)$

$L \in coNEXP$, tj. existuje nedet. alg. B pracující v čase $2^{n^{O(1)}}$ t.ž.
$\forall x \quad x \notin L \iff B$ přijímá x.

radící fce $g(n) = |L \cap \{0,1\}^n|$ binárně zakódovaná, tj. $\in n+1$ bitů.

alg. A: na vstupu x, $g(|x|)$
pověz $n = |x|$.
uhodni $2^n - g(n)$ různých řetězců délky n a přijímající výpočty algoritmu B na těchto řetězcích.
Pokud uhodnu správně, přijmi x jestliže není jedním z uhodnutých řetězců přijímajících alg. B.
Jinak odmítni.

x je přijato alg. $A \iff x$ není přijato alg. B

A běží v čase $2^n \cdot 2^{n^{O(1)}} = 2^{n^{O(1)}}$ ∎

Věta: $NP = coNP \Rightarrow NEXP = coNEXP$

Dk: "padding argument"

$L \in NEXP$ ... L je přijímán nedet. algoritmem A v čase 2^{n^k}.

$L' = \{ x \# 0^{2^{n^k}} ; x \in L \}$

zjevně $L' \in NP \Rightarrow L' \in coNP$ (předpoklad)

$L' \in NP$ a je přijímán nedet. alg A' v čase $n^{k'}$.

nedet. alg A'' pro $\overline{L}$: na vstupu x, vygeneruj $x \# 0^{2^{n^k}}$ a spusť alg. A'.

A'' běží v čase $\left(n + 2^{n^k}\right)^{k'} \approx 2^{k' n^k}$

$\Rightarrow$ A'' je NEXP alg. pro $\overline{L}$. ☒

- Obdobná implikace NEXP = coNEXP $\Rightarrow$ coNP = NP není známá a lemmata typu pravda

---

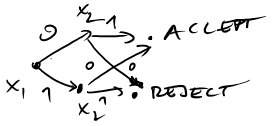
Branching pgms:

(BP, Ordered Binary Decision Diagrams, Switching & rectifier networks)

Branching program: orientovaný acyklický graf
jeden zdrojový uzel "Init",
dva cílové uzly "ACCEPT", "REJECT"
každý vnitřní uzel označen proměnnou,
z každého vnitřního uzlu vychází dvě
hrany označené 0 a 1.

Výpočet b.p.: začíná ve vrcholu Init, přečte
proměnnou příslušející aktuálnímu
vrcholu a následuje hranu konzistentní
s hodnotou přiřazenou proměnné.
dorazí-li do ACCEPT nebo REJECT
$\rightarrow$ výsledek výpočtu.

Př:



počítá $x_1 \oplus x_2$

zajímají nás:
- velikost b.p. = počet vrcholů b.p.
- délka b.p. = nejdelší cesta v b.p.
- šířka b.p. = pro vrstevný b.p., maximální velikost vrstvy.

hrany pouze mezi sousedními vrstvami

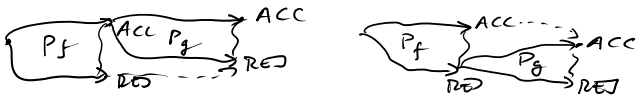
Př: $\bigoplus$ $x_1 \oplus x_2 \oplus \cdots x_n$ lze spočítat b.p. velikosti $2n + O(1)$, šířky 2

a délky n.

2) $\sum_i x_i \bmod p$ → b.p. velikost $pn + O(p)$
šířka p
délka n

3) NAJ → b.p. velikost, šířka, délka ? viz dále

Kombinační b.p.: $f \wedge g \wedge \dots$ $f \vee g \vee \dots$



→ AC^0 ... b.p. velikosti $poly(n)$ a šířky $O(1)$.
ACC^0 ... + dělky

Věta: $f \in L/poly \iff f$ je počitatelný branching programy polynomiální velikosti:
($\{f_n\}_{n\geq 1}$ je počitatelný pgmy $\{P_n\}_{n\geq 1}$)

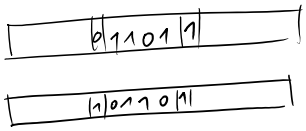
log-space výpočet s polynomiální radící fa g ↑

→ $x, g(|x|)$ je vstup pro log-space výpočet

Dk: obdobný
$P/poly \equiv PSIZE$ ☒

NC^1: obvody hloubky $O(\lg n)$ sestávající z binárních $\vee, \wedge, \neg$.

Př: 1) $x + y \in NC^1$ $\quad x, y \in \{0,1\}^n$



$i < j$ $\quad c_{ij}$ ... j-tá pozice generuje carry, které probublá až na i-tou pozici

$$c_{ij} = (x_j \wedge y_j) \wedge \bigwedge_{l=i+1}^{j-1} (x_l \vee y_l)$$

$$c_i = \bigvee_{j=i+1}^{n} c_{ij}$$

$$z_i = x_i \oplus y_i \oplus c_i \qquad \leftarrow x + y = z$$

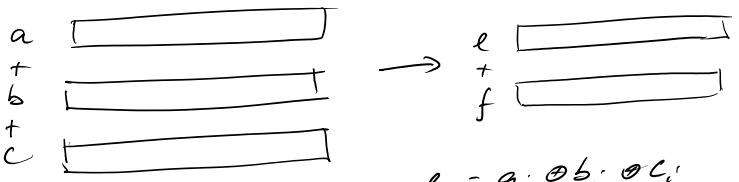
$i = 1, \dots, n$

$z_0 = c_0$

→ AC^0 obvody → NC^1 obvody ☒

2) $x_1, x_2, \dots x_n \in \{0,1\}^n$ -- n-bitová čísla

$\sum_i x_i \in NC^1$



$$\ell_i = a_i \oplus b_i \oplus c_i$$
$$f_{i-1} = [a_i + b_i + c_i \geq 2]$$

- Součet tříd čísel zredukuje na součet dvou čísel

  rekurzivně součet n čísel $\to$ $2/3\,n \to \left(\frac{2}{3}\right)^i n \to \dots \to 2$ čísla

  $\to$ aplikuj $x+y$ ☐

3) $X \cdot y \in NC^1$

4) $MAJ \in NC^1$

⋮

[Barrington, Ben-Or&Cleve]

Okruh $(R, +, \cdot)$ — nemusí být multiplikativní grupa

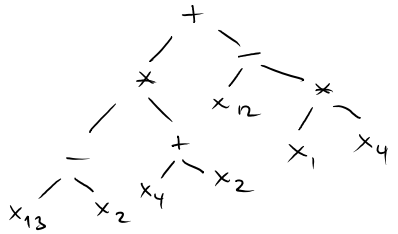
$+$: grupa

V okruhu lze odečítat

Př: $\mathbb{C}$, $\mathbb{R}$, $\mathbb{Z}$, $\mathbb{Z}_n$, $GF[p]$ — tělesa okruh; $\mathbb{Z}_n$ počítání „modulo"

těleso = okruh, kde lze i dělit, tj. s multiplikativní grupou

Formule nad $R(+,\cdot)$



registrový model: registry $r_1, \dots r_k$ ← pracovní registry (R/W)
$x_1, \dots x_n$ ← vstupní registry (R/O)

obsahují hodnoty z R

program: posloupnost instrukcí typu:

$$r_i \leftarrow r_j \overset{*}{\pm} r_\ell$$

r_j, r_ℓ můžou být i x_j, x_ℓ

Př: $r_1 \leftarrow x_{13} - x_2$

Př:

$r_1 \leftarrow x_{13} - x_2$
$r_2 \leftarrow x_4 + x_2$
$r_3 \leftarrow x_1 * x_4$
$r_4 \leftarrow x_{12} - x_3$
$r_5 \leftarrow r_1 * r_2$
$r_5 \leftarrow r_5 - r_4$

} spočte hodnotu výrazu výše

Kolik registrů potřebujeme na vyhodnocení výrazu?

1) ≤ velikost formule ✓
2) ≤ hloubka formule + 1 ✓
3) ≤ 3 (4) ✓

Věta: (Ben-Or, Cleve)

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

Dk:

používáme instrukce

$r_i \leftarrow r_i \pm r_j * x_k \qquad i \neq j$
$r_i \leftarrow r_i \pm x_k$

(potřeba jeden registr navíc pro simulaci těchto instrukcí předchozími instrukcemi)

Cíl: pro formuli $f(x_1 \dots x_n)$ zkonstruovat pgm ekvivalentní $r_1 \leftarrow r_1 + r_2 * f(x_1 \dots x_n)$
a $r_1 \leftarrow r_1 - r_2 * f(x_1 \dots x_n)$
který zachová původní hodnoty r_2 a r_3

pokud na počátku nastavíme $r_1 = 0$ a $r_2 = 1$,
pak na konci máme $r_1 = f(x_1 \dots x_n)$.

cíl – konstrukce indukcí:

1) $f(x_1, x_2, \dots x_n) = x_i$

$r_1 \leftarrow r_1 + r_2 * x_i$ ... přímo instrukce

„–" se udělá obdobně

2) $f(x_1 \dots x_n) = g(x_1 \dots x_n) + h(x_1, \dots x_n)$

$r_1 \leftarrow r_1 + r_2 * g(x_1 \dots x_n)$
$r_1 \leftarrow r_1 + r_2 * h(x_1 \dots x_n)$

} pgmy existují dle indukčního předpokladu

dokonaný požadovaný efekt

pro "-" obdobně

3) $f(x_1 \dots x_n) = g(x_1 \dots x_n) * h(x_1 \dots x_n)$

$r_3 \leftarrow r_3 + r_2 * g(x_1 \dots x_n)$
$r_1 \leftarrow r_1 + r_3 * h(x_1 \dots x_n)$
$r_3 \leftarrow r_3 - r_2 * g(x_1 \dots x_n)$
$r_1 \leftarrow r_1 - r_3 * h(x_1 \dots x_n)$

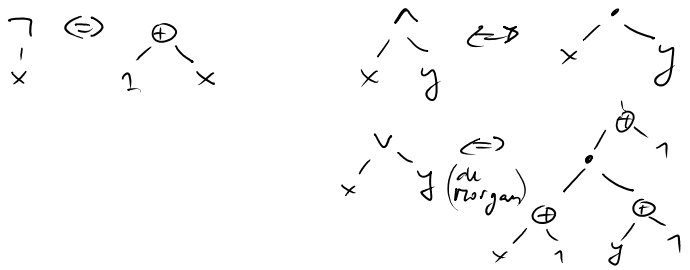
} pgmy existují dle indukčního předpokladu

požadovaný efekt

délka pgm je $\leq 4^d$, použité 3 registry ☑

Př: obecně $GF[2] = (\{0,1\}, \oplus, \cdot)$ (těleso)

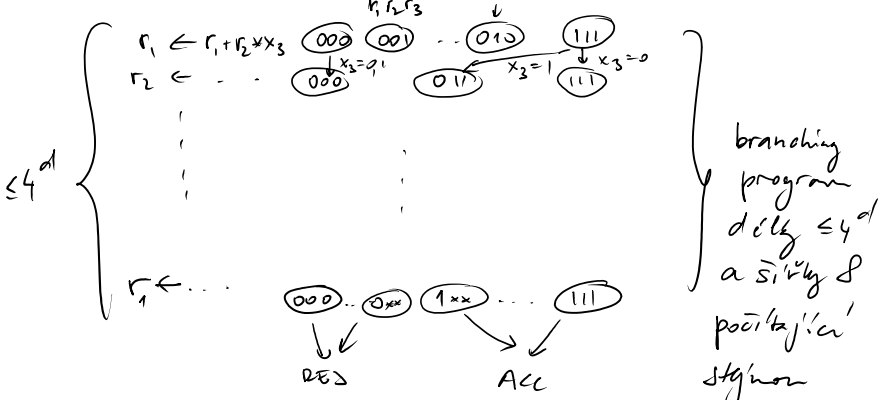
parita, násobení



⇒ Booleovská formule sestávající z $\wedge, \vee, \neg$ se dá přepsat jako aritmetická fce nad $GF[2]$, hloubka se zvětší nejvýše 3-krát.

→ pro danou fci nad $GF[2]$ hloubky d můžeme sestrojit registrový pgm se třemi regis. r_1, r_2, r_3, každý z nich si pamatuje jeden bit.

⇒ konfigurace (obsah) registrů v daném kroce se dá popsat 3 bity ⇒ 8 možností



REG ACC stejnou fci jako původní fci.

Věta (Barrington '87): Pokud je fce počitatelná Booleovskou formulí hloubky d, lze ji počítat též branching programem délky $\leq 4^d$ a šířky 5.

Předchozí konstrukce dává o něco slabší výsledek délka $\leq 4^{3d}$ a šířka 8.

- graph Reachability

Vstup: G, s, t (graf, $\in V(G)$) Cíl: [Vede z s cesta do t]?

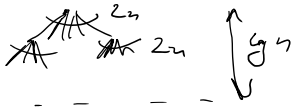
Algoritmy:
- Dijkstra
- BFS, DFS
- ...

vše vyžaduje prostor $\Omega(n)$

Lze líp? Savitchův alg.

```
Reach(s,t,k):
  if k≤1 return [(s,i)∈E(G)];
  for u∈V(G) do
    if Reach(s,u,k/2) & Reach(u,t,k/2)
    then return true;
  return false;
end.
```

→ hloubka rekurze $\lg n$



strom rekurze - arita $2n$

→ čas $n^{\lg n}$

prostor - $\lg n \cdot O(\lg n) = O(\lg^2 n)$

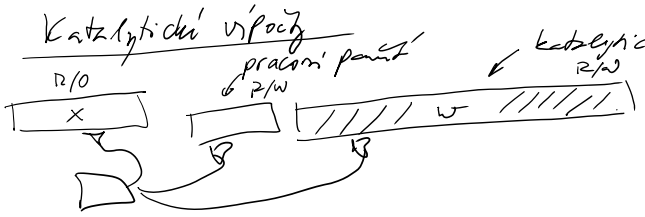
→ Důsledek: $DSPACE(\lg n) \subseteq NSPACE(\lg n) \subseteq DSPACE(\lg^2 n)$

Obecně: $DSPACE(s) \subseteq NSPACE(s) \subseteq DSPACE(s^2)$

- Nejlepší polynomiální alg. používá prostor $O\left(\frac{n}{2^{\sqrt{\lg n}}}\right)$

Otázka: Lze počítat graph reachability v čase
pologaritmickém a prostoru $O(\sqrt{n})$?

Katalytická výpočty



lze libovolně zapisovat, ale na konci výpočtu musí obsahovat původní obsah.

Možnost: kompresní – problém – v nemusí být komprimovatelné (např. náhodné v)

Technika Ben-Or & Cleve dovoluje vynásobit

matice $A_1 \times A_2 \dots A_n$

$A_i \in \mathbb{Z}^{n \times n}$ přes $GF[2]^{n \times n}$

s použitím $O(\log n)$ bitů prostoru a $\hat{O}(n^2)$ prostoru na katalytické pásce. Tam simulují tři registry r_1, r_2, r_3.

$$\frac{r_i \leftarrow r_i + r_j * x_k}{r_i \leftarrow r_i - r_j * x_k}$$ ↺ inverze

$\Rightarrow$ pomocí lze zinvertovat $P \to P^{-1}$

...