Výpočetní složitost - 2016

18. října 2016 21:37

- 2/0 zk
- zkouší se z prostoru látky
- letní přednáška strukturální složitost
  (od příštího roku Struk. slož. a Výp. obž alternují v létě)

- Výpočetní složitost
  - kolik kroků je potřeba k nalezení řešení problému
    & kolik prostoru, náhodnosti, ...
  → význam mimo informatiku - algoritmický proces
    - fyzika, biologie, ekonomie, ...

tento semestr: neuniformní výpočetní modely

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

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

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

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

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

uniformní algoritmus A pro f: $x \in \{0,1\}^* \to A(x)$
$A(x) = f(x)$

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

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

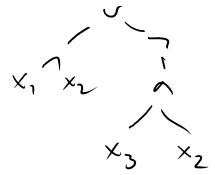
$f_n = f|_{\{0,1\}^n}$

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

Př.: 1) A_n má v sobě tabulku f_n
2) A_n má v sobě procedůru pro hašování dat velikosti n.
3) A_n používá jiný postup na vstupech malé a velké délky

• neuniformní modely:

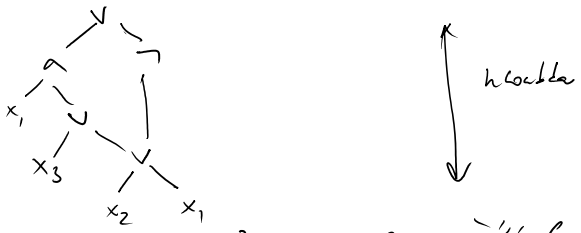
• Booleovské formule



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

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.

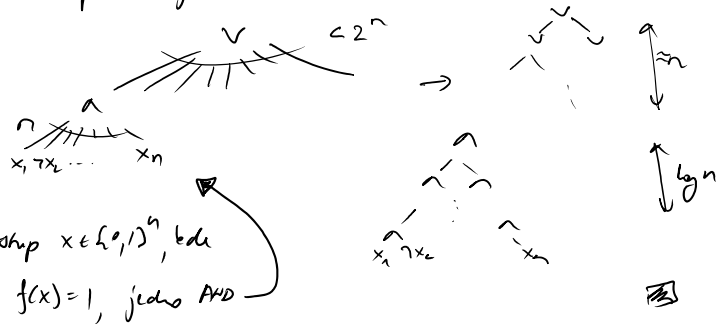
• uvidíme dále – branching programy,
aritmetické výrazy
polynomy, ...

zajímá nás velikost, hloubka, šířka, ... obvodu, fly, ...
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 $\log 2^n$, který počítá f_n.

• odhad lze zlepšit 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, který ji počítá, $\frac{2^n}{\dots}$.

AND, OR a unárními NOT, který ji používá,
je velikosti 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
$\leq s^{2s} \cdot (n+3)^s$
↑ # spojnic ↑ 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 $C_n \leq \mathcal{O}(2^n/n)$.

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 závisí 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: $\exists \varepsilon > 0 \; \forall n \; |C_n| \geq 5n$.

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 $\leq S(n)$ }

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

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

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

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

P/f ... — " — $|g(n)| \leq f(n)$.

Věta: $\forall f$ $\quad f \in P/poly \iff f$ 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 Koucky at 25. 10. 2016 23:05

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

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

$a_1 = \overbrace{0\cdots0}^{n}$, $a_2 = 00\cdots01$, ... $a_{2^n} = \overbrace{11\cdots1}^{n}$

$C_0 := \{$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.

end.

- algoritmus A $\mathcal{S}$ je v EXP, pracuje v $DTIME(2^{n^{2k}})$.
- Žádná rodina obvodů $\{C_n\}_{n\geq 0}$ velikosti $\leq n^k + k$ nepočítá stejnou fci jako A. ∎

Věta: $\forall k\ \exists L \in PSPACE$ t.ž. $L \notin SIZE(n^k + k)$. Důk: stejný $\mathcal{S}$.

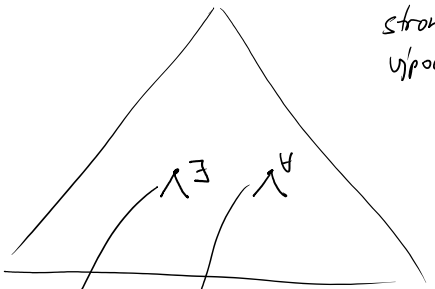
Otázka: Pro kterou nejmenší třídu lze odstranit podobné tvrzení?

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

důkaz níže

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

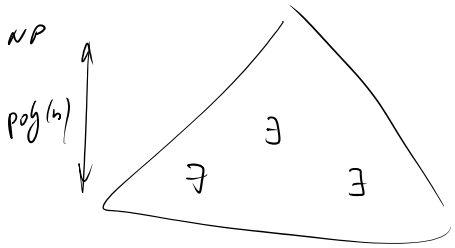
alternující Turingovy stroje: zobecnění nedeterminismu

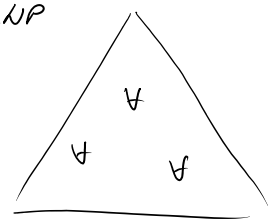


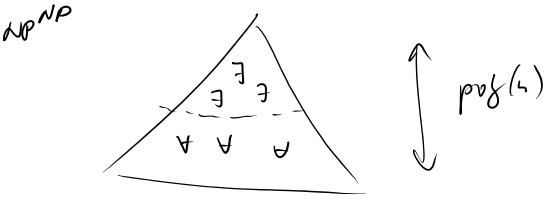
strom možných výpočtů na daném vstupu x (řešením)

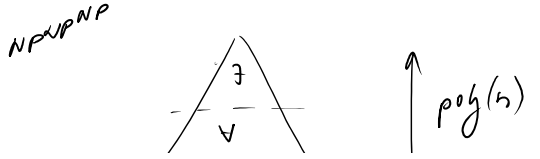
konf. je přijímající $\Leftrightarrow$ alespoň z jedna z následujících konf. je přijímající

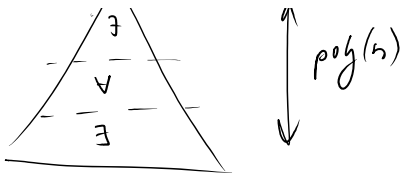
konfigurace je přijímající $\Leftrightarrow$ obě následující konfigurace jsou přijímající





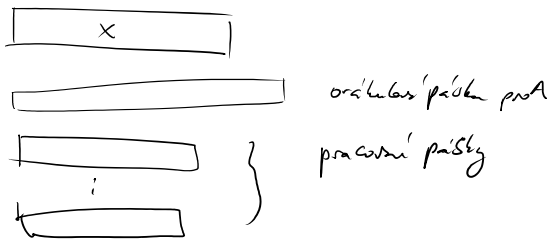






Orákula:

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



dotazy $u \in A$ vyřízeny v jedné fázi

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

P^A ... deterministický —//— .

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

$\Sigma_3\text{-SAT}$ fle $\quad \exists x_1 x_2 \dots x_{n^k} \forall y_1 \dots y_{n^k} \exists z_1 \dots z_{n^k} \varphi(\vec{x}, \vec{y}, \vec{z})$

$\Sigma_k\text{-SAT}$ ... k alternujících kvantifikátorových bloků

$\Sigma_3\text{-SAT}$ je úplný pro $NP^{NP^{NP}}$, tj.

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

a každý problém $L \in NP^{NP^{NP}}$ je 1-převeditelný na $\Sigma_3\text{-SAT}$ 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):

$S = \{a 01^n 01^m$; a je počátek úseků prvočíselných

tabulky fcí $f: \{0,1\}^n \to \{0,1\}$, kterých je
počítaná obvody velikosti $\le m\}$

$S \in NP$ ... pro vstup $x = a\,01^n 01^m$ uhodnu
obvod velikosti $\le m$ a ověřím,
že a odpovídá první tabulka
(částečně).

idea: alg pro L
na vstupu x $\quad |x| = n$
uhodnu $a \in \{0,1\}^{n^{2k}+k}$ + ř.
a není počítatelný obvody velikosti $n^k + k$,
tj. $a\,01^n 01^{n^k+k} \notin S$.
pokud x je j-tý řetězec, $j < |a|$, přijmi pokud $a_j = 1$
jinak odmítni.

problém: a není jednoznačné, alg může
přijímat všechno

řešení:
$S' = \{a\,01^n 01^m,\ a \in \{0,1\}^*$, buď $a\,01^n 01^m \in S$ nebo
Lebo $\exists$ lex. menší $a' < a$, $a \in \{0,1\}^{|a|}$
t.ř. $a'\,01^n 01^m \notin S\}$

$S' \in NP^{NP}$
$\to$ použij předchozí algoritmus s S'. ∎

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

Dk:
Dvě možnosti

1) $NP \not\subseteq PSIZE = \bigcup_{k \ge 0} SIZE(n^k + k)$
(pak závěr plyne automaticky)

2) $NP \subseteq PSIZE \Rightarrow NP^{NP^{NP}} \subseteq NP^{NP}$
$\Sigma_3 \subseteq \Sigma_2$

Dk:

φ bool. fce
$\varphi(x_1, x_2, \dots x_n)$ je splnitelná
$\Leftrightarrow$
$\varphi(x_1, \dots x_{n-1}, 0)$ nebo $\varphi(x_1, \dots, x_{n-1}, 1)$
je splnitelná

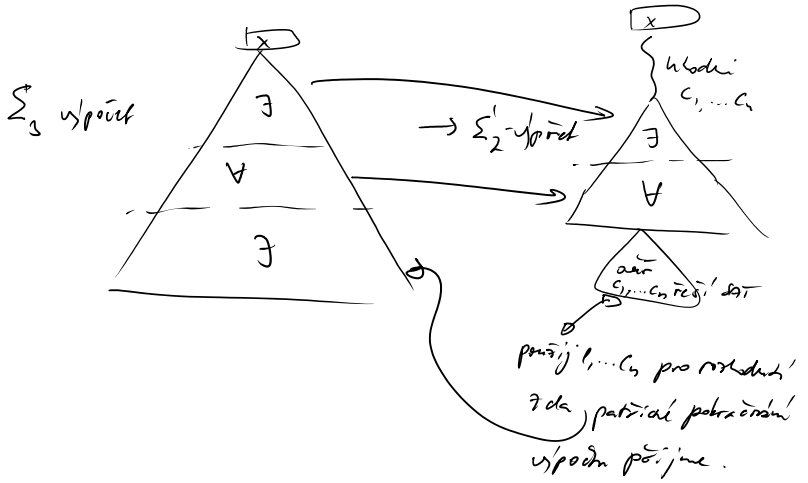
⇒ mám-li obvody $C_1, C_2, \dots C_n$, které hádají řešení SAT, musím ještě ověřit, že jsou konzistentní:

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

$$C_{|\varphi|}(\varphi) = 1 \iff \left(C_{|\varphi(x_1,\dots,0)|}(\varphi(x_1,\dots,0)) = 1\right) \vee \left(C_{|\varphi(x_1,\dots,1)|}(\varphi(x_1,\dots,1)) = 1\right)$$

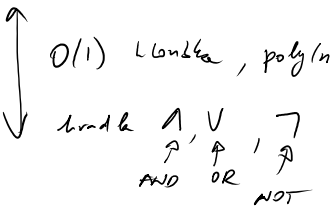
- pokud φ nemá žádné proměnné, vyhodnotí ji a tautologii, zda to odpovídá $C_{|\varphi|}(\varphi)$.

toto je coNP test.



$L \in NP^{NP^{NP}}$ t.ž. $L \notin SIZE(n^k+k)$ ⇒ $L \in NP^{NP}$ t.ž. —"— ☠

AC^0 - obvody



- 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 různý výstup)
- neumí - spočítat paritu vstupu tj. $\sum x_i \bmod 2$

⇒ nemůžeme vyvrátit dělitelnost binárně zapsaných čísla.

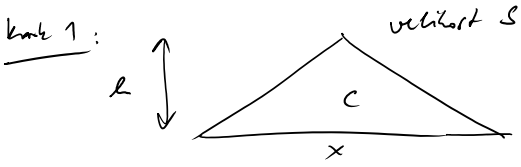
Razborov – Smolensky '87

- $\forall p, q$ prvočísla, $MOD\text{-}p \notin AC^0[q]$
  $p \neq q$
  ↑ polynomiálně velké AC^0-obvody nemůže o hradlech pro $MOD\text{-}q$ $[\sum x_i \neq 0 \bmod q]$

pozdějším: Furst-Saxe-Sipser '83, Ajtai '83, Håstad '86
$MOD\text{-}2 \notin AC^0$

Dk: dělitelnost ve dvou krocích (malé stupně)
1) aproximace obvodu polynomem nad $GF[q]$
2) $MOD\text{-}p$ nelze aproximovat polynomem malého stupně nad $GF[q]$.

krok 1:



Bůno: OR, $MOD\text{-}q$, NOT
na hladině i
indukčním odspodu přiřadím hradlu 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)

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

c)

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

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

pro první zvolený vstup $x \notin W_{i-1}$, kde $\bigvee_i g_i = 1$
tj. $\sum_{i=1}^{\ell} p_{g_i}(x_i) > 0$
↑ součet nad $\mathbb{Z}$.

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

↳ součet nad $\mathbb{Z}$.

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

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

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

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

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

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

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

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

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

$\rightarrow \forall AC^0[q]$ obvod velikosti $\le 2^{n^{1-1/4h}}$ a hloubky h $\exists$ pol. 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: Spárem: 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 $y_1, \dots y_n \in \{-1,1\}^n \setminus W'$

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

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

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

$$f(y_1, \dots y_n) = p_2' \cdot \ell_1 + \ell_2$$

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

- Přibližný počítání je v AC^0

[Ajtai - Ben-Or '83]

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

Tvrz: $AMAJ(x) \in AC^0$.
... "funkce konzistentní s $AMAJ(x)$ je v AC^0."

Dk: obrod sestrojíme náhodně

vezmeme pevné $x \in \{0,1\}^n$ a počítejme pst., že náš obrod 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 náhodně zvolené $i \in \{1, \dots, n\}$	$\leq \frac{1}{4}$	$\geq \frac{3}{4}$
$C_1(x) = \wedge$ (10 log n nezávislých kopií $C_0(x)$)	$\leq \frac{1}{n^{20}}$	$\geq \left(\frac{3}{4}\right)^{10 \log n} \geq \frac{1}{n^{10}}$
$C_2(x) = \vee$ (n^{15} nezávislých	$\leq \frac{1}{n^5}$	$\geq 1 - \left(1 - \frac{1}{n^{10}}\right)^{n^{15}} \geq 1 - e^{-n^5}$

kopií $C_0(x)$)

$C_2(x) = \vee$ (n^{15} nezávislých kopií $C_1(x)$) $\quad \le \frac{1}{n^5}$ $\quad \Big| \quad \ge 1-\left(1-\frac{1}{n^{10}}\right)^{n^{15}} \ge 1-e^{-n^5}$

$C_3(x) = \wedge$ (n^2 nezávislých kopií $C_2(x)$) $\quad << 2^{-n^2}$ $\quad \Big| \quad >> 1-e^{-n^4}$

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

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

- nejvýše 2^n různých x

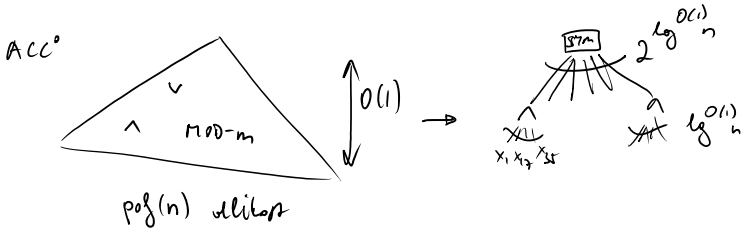
$$\Pr_{C_3}\left[C_3(x) \text{ dává špatný výsledek na nějakém } x \right] \le 2^{-n}.$$

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

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

Michal Koucky at 29. 11. 2016 22:19

Razborov hlavní obvodová [Beigel-Tarui "$\circ_n$ ACC"]



ISYM ... počítá "symetrickou" fci, tj. funkci, která zajímá pouze jedničky na vstupu, ale ne na jejich rozmístění.

Pozn.: užitečné z hlediska zkoumání omezení ACC° obvodů

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

- Razborov-Smolensky šlo použít na redukci hlavní 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

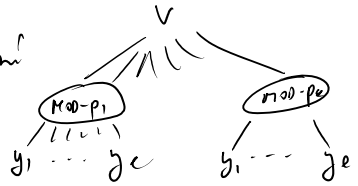
pro jednoduchost budeme předpokládat, že $a_1 = a_2 = \dots = a_k = 1$.

(např.: $m = 6$, o tom Raz.-Smol.

nic netřídí)

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

je ekvivalentní

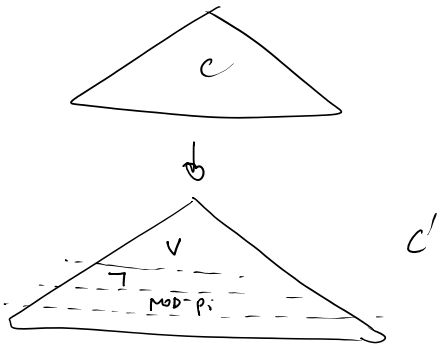


$m \mid \Sigma x_i$ iff $\forall j = 1,\dots,k \quad p_j \mid \Sigma x_i$.

Dk: "⇒" triviální

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

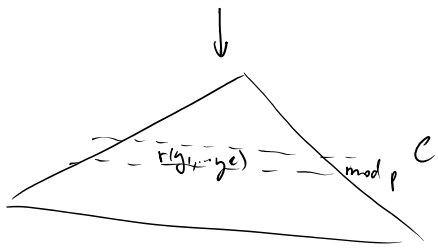
ACC⁰ obvod C zredukujeme do požadovaného tvaru v několika krocích



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

a) ¬ nebo ∨
b) MOD-p_i kde p_i je sekající v rámci vrstvy.

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



každé hradlo nahradíme polynomem, každá vrstva asociovaná s nějakým prvočíslem p, pojmy na další vrstvě dávají 0/1 při počítání mod p.

(s velkou pravděpodobností, konstrukce je randomizovaná)

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

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

a_{ij} zvolíme náhodně

pro pevné $y_1, \dots y_\ell$

$$\Pr_{a_{ij}}\left[r_{OR}(y_1, \dots y_\ell) = OR(y_1, \dots y_\ell) \right] \geq 1 - \frac{1}{2^{4\lg n}} = 1 - \frac{1}{n^8}.$$

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 $x \in \{0,1\}^n$

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

(odpovídající hodnotě příslušného hradla; pro n dostatečně velké)

C''':

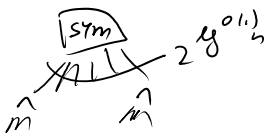
n nezávislých náhodných zvolíme obvody C''.

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.

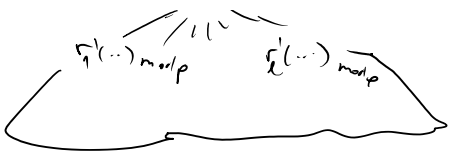
C''' zkonvertujeme do podoby



postupným kolapsem horních dvou vrstev.

$C''' =$

$$h : \mathbb{Z} \to \{0,1\} \qquad h(x) = \begin{cases} 0 & x \leq \frac{n}{2} \\ 1 & x > \frac{n}{2} \end{cases}$$



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

chceme zkolabovat $r(y_1, \dots y_k)$ s polynomy $r_1'(\dots)$

poutá substituce $r(r_1'(\dots), r_2'(\dots), \dots r_k'(\dots))$

nesbírá, neboť $r_1'(\dots), \dots r_k'(\dots)$ dávají 0/1 pouze při mod p. Vystrčit mod p ven by takto nepomohlo.

→ použijeme speciální polynomy, které nám to umožní

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

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

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

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

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

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

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

Tvrzení: P_{2^i} splňuje (*).

Dk: $i = 1$

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

$x \equiv 0 \bmod p \Rightarrow x = cp+1 \Rightarrow 3\underbrace{(2cp+1)}_{x^2 \bmod p^2} - 2\underbrace{(2cp+1)(cp+1)}_{x^3 \bmod p^2} =$

$= 6cp + 3 - 4cp - 2cp - 2 = 1 \pmod{p^2}$ ✓

$i > 1$

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

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

$x \equiv 0 \bmod p \Rightarrow y = P_{2^{i-1}}(x) = cp^{2^{i-1}} + 1$

$\Rightarrow P(n) = c' n^{2^i} + 1 \bmod p^{2^i}$ ✓

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

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

Tvrzení: 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 $P_{\frac{k}{2}}(x) \in N \le 5^{\left(\frac{k}{2}\right)^2 - 1}$

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: $r(y_1, \dots y_c)$ je polynom normy N.

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

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

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

Dk: $r(a_1 \mod m, a_2 \mod m, \dots a_c \mod m) =$

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

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

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

---

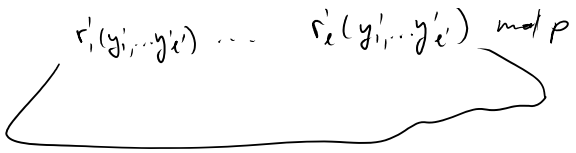
opakovaně kolokujeme horní dvě vrstvy v C^{III}

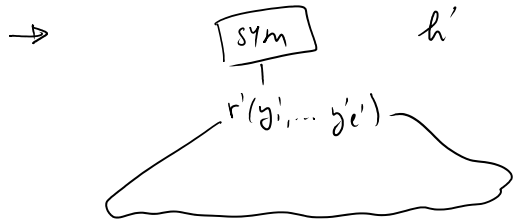
$h : \mathbb{Z} \to \mathbb{Z}$

sym

$r(y_1, \dots y_c)$

C^{III}





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

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

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

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

konstanta je korektní.

pokud stupeň $r, r'_1,\dots r'_c = \log^{O(1)} n$,

& norma $r \leq 2^{\log^{O(1)} n}$ & norma $r'_1,\dots r'_c \leq \log^{O(1)} n$

pak $k \leq \log^{O(1)} n$, stupeň $r' \leq \log^{O(1)} n$,

norma $r' \leq 2^{\log^{O(1)} n}$. ∎

→ výsledná h a r mi dá potřebný tvar

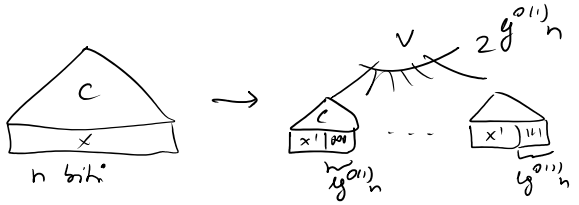
Michal Koucky at 29. 11. 2016 23:21

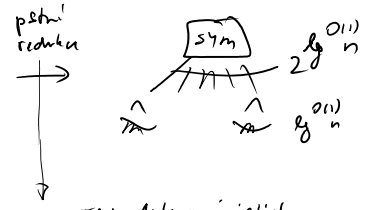
Testování splnitelnosti ACC⁰ obvodu.

ACC⁰ obvod C s n bity vstupu $x_1,\dots x_n$.

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

- Triviální alg. $\text{poly}(n)\cdot 2^n$ čas
- Lze lépe: alg. $2^{n - \log^{O(1)} n}$ čas.





lze urdělat deterministicky
(výběr coeficientů a_{ij})

C v čase $2^{\log^{O(1)} n}$ přeberu vrcholy, tj. spočítám

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

↳ stupeň $\le \log^{O(1)} n$
norma $\le 2^{\log^{O(1)} n}$.

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

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

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

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

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

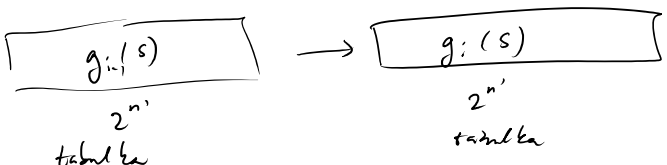
$\forall x$; $\quad r(x) = \sum_{T \subseteq S_x} g(T)$.

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

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

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

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 s časem $2^{n'} \cdot n'$

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 $DTIME(2^{n - \lg^{O(1)} n})$

(lze zlepšit na $DTIME(2^{n - n^\varepsilon})$, také s žádnou hlubokou obvodu)

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

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

(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 Co.L-úplných vstupů

φ_x kódujeme tabulkou výpočtu M pro x.

- tedy dán φ_x velikosti $O(2^n \cdot 2^n) = O(2^{2n})$ my potřebujeme $O(2^n poly(n))$ což lze získat pomocí vylepšení technikou.

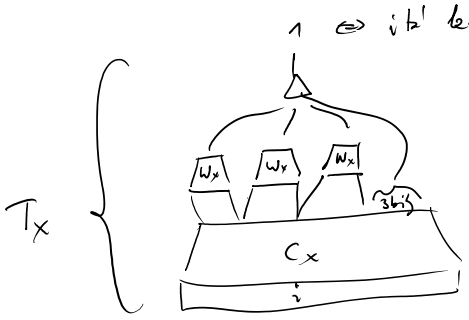
- existuje poly-time alg., který pro x sestrojí polynomiálně velký obvod C_x kódující φ_x.

C_x na vstupu $i \in \{0,1\}^{n + O(\lg n)}$ vypíše indexy

tří proměnných obsažených v ité disjunkci φ_x spolu se třemi bity určujícími, které z těchto proměnných jsou v ité disjunkci negované.

- v E^{NP} lze najít pro dané $x \in L$ 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 kódující 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 ověř, že jejich kombinace dělají, co by měly. ▨

Výpočty s nápovědou

- definice viz dříve

Věta: $P/poly = PSIZE$

↳ for počítatelné 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

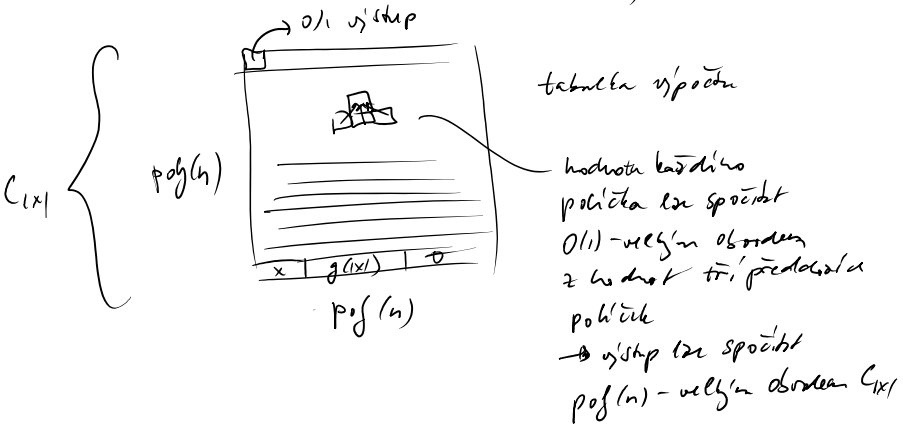
běží v čase $poly(|C_{|x|}|) = poly(|x|)$

$\rightarrow L \in P/poly$ ∎

"$\subseteq$" $L \in P/poly$ tj. máme poly-time alg. a radící fci $g(n)$ t.ž.

na vstupu x, $g(|x|)$ algoritmus vydá, zda $x \in L$.

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



$\rightarrow C_1, C_2, \ldots \quad L \in PSIZE$ ∎

$NP/poly$ ... "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 počitatelně nedeterministickými obvody polynomiální velikosti.

Dk: obdobný předchozí větě.

EXP/poly: $L \subseteq EXP/poly$, 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

$$EXP/poly = P/poly$$

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

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

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

Věta: $coNEXP \subseteq NEXP/poly$

Důsledek: $coNEXP/poly = NEXP/poly$

Dk:
pro $L \in coNEXP$ zkusme nedeterministický alg. A 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.

$\forall x \quad x \notin L \Leftrightarrow B$ přijímá x.

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

alg. A: na vstupu x, $g(|x|)$

pobž $n = |x|$.

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. $A \Leftrightarrow x$ 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)

$\overline{L'} \in NP$ 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 $\left(n + 2^{n^k}\right)^{k'} \simeq 2^{k' n^k}$

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

- Obdobná implikace $NEXP = coNEXP \Rightarrow coNP = NP$ není známá a zřejmě by šlo pravděpodobně

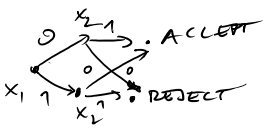
Branching pgms:

(BP Ordered Binary Decision Diagrams, Switching & rectifier)

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.

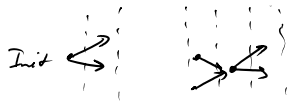
vyhodnocení b.p.: začíná se v uzlu Init, přečte
proměnnou příslušející aktuálnímu
uzlu a následuje hranu konzistentní
s hodnotou přečtené proměnné.
dorazí buď do ACCEPT nebo REJECT
→ výsledek výpočtu.

Př:



počítá $x_1 \oplus x_2$

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



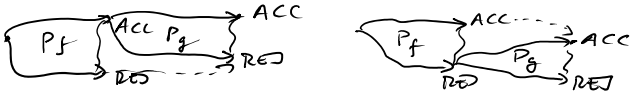
hrany pouze mezi sousedními vrstvami

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

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

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

Kombinování b.p.: $f \wedge g \wedge \ldots$ $f \vee g \vee \ldots$



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

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

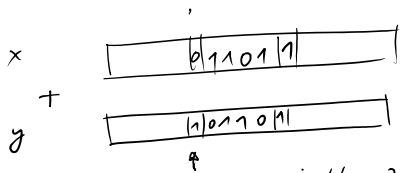
log-space výpočet s polynomiální radící fcí g

→ $x, g(|x|)$ 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

$$c_{ij} = (x_j \wedge y_j) \wedge \bigwedge_{\ell=i+1}^{j-1} (x_\ell \vee y_\ell)$$

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

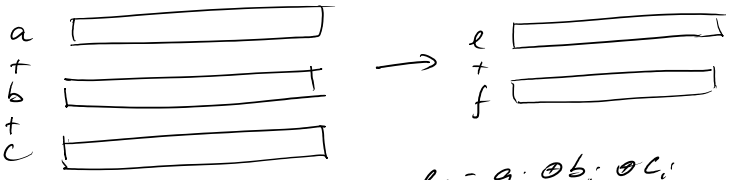
$$z_i = x_i \oplus y_i \oplus c_i \quad \leftarrow x+y=z \quad i = 1, \dots, n$$

$$z_0 = c_0$$

→ AC^0 obvod → NC^1 obvod ▨

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

$\sum^t x_i \in NC^1$



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

- Součet tří čísel zredukuji na součet dvou čísel

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

→ aplikuj $x+y$ ▨

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

4) $MAJ \notin NC^1$

⋮

[Barrington, Ben-Or&Cleve]

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

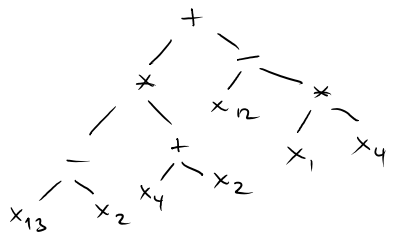
grupa

V okruhu lze odečítat

Př: $\mathbb{C}$, $\mathbb{R}$, $\mathbb{Z}$, $\mathbb{Z}_n$, $GF[p]$ (tělesa, okruh; 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:

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

r_j, r_ℓ musí být buď x_j, x_ℓ

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 oborem 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

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

$r_i \leftarrow r_i \pm x_k$

(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 hodnotu 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

dohromady požadovaný efekt

pro "$-$" obdobně

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

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

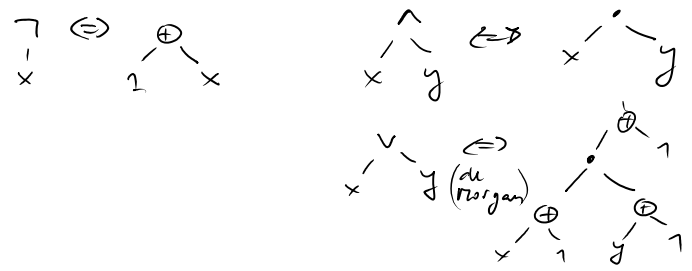
} pgmy existují dle indukčního předpokladu

požadovaný efekt

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

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

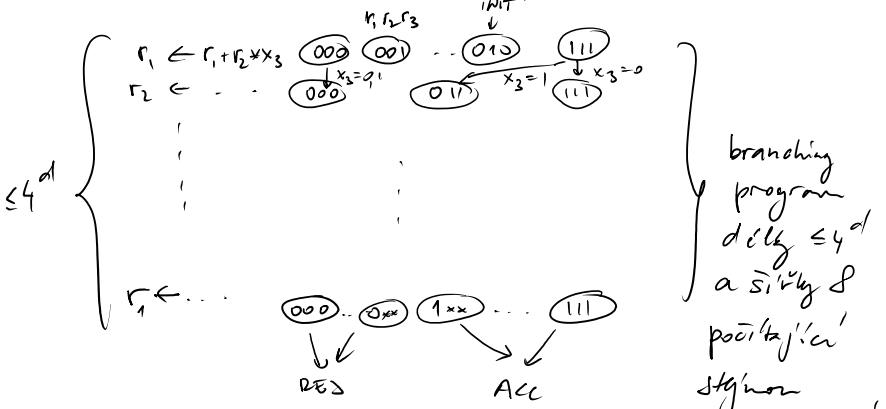
sčítání násobení



⇒ 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 registry 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



branching program délky $\leq 4^d$ a šířky 8 počítající stejnou fci jako původní fce.

Věta (Barrington '87): Pokud je fce počitatelná Booleovskou formulí hloubky d, lze ji počítat též branching pgmem 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]?

graf G, $s, t \in V(G)$

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

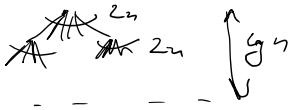
vše vyžaduje prostor $\Omega(n)$

Lze lépe? Savitchův alg.

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

ad.

→ hloubka rekurze $\lg n$



strom rekurze - arita $2n$

→ čas $n^{\lg n}$

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

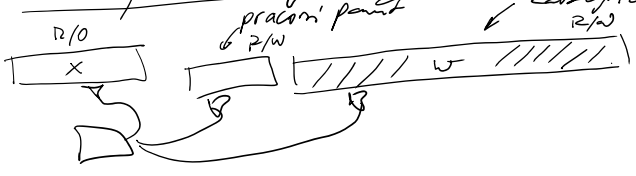
→ Důsledek: $DSPACE(\log n) \subseteq NSPACE(\lg n) \subseteq DSPACE(\lg^2 n)$

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

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

Otázka: Lze počítat grafovou dosažitelnost v čase polynomiálním a prostoru $O(\sqrt{n})$?

Katalytická výpočty



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

Možnost: kompresní - problém - w nemusí být komprimovatelné

(např. náhodné σ)

Technika Ben-Or & Cleve dovoluje vyvážit

matice $A_1 \times A_2 \dots A_n$

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

s použitím $O(\log n)$ bitů prostoru a $\tilde{O}(n^2)$ operací na klasických vstupech. Tam simulujeme tři registry r_1, r_2, r_3.

$$r_i \leftarrow r_i + r_j * x_k \quad \text{(inverze)}$$
$$r_i \leftarrow r_i - r_j * x_k$$

$\Rightarrow$ pokud lze zjistit $P \to P^{-1}$

...

Michal v 2022-05-17 10:46

Decision trees & other measures of complexity

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

$x \in \{0,1\}^n$ $\quad s_x(f) = \left| \{ i ; f(x) \neq f(x \oplus e_i) \} \right|$

... sensitivity of f at x

$$s(f) = \max_x s_x(f)$$

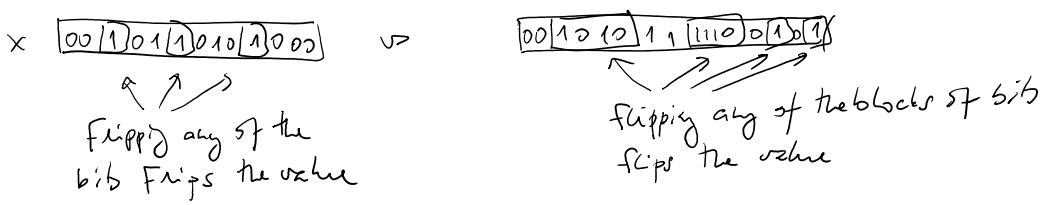
Ex: $s(\text{PARITY}) = n$

$x \in \{0,1\}^n$ $\quad bs_x(f) = \max \{ b ; \exists B_1, \dots B_b \subseteq \{1, \dots n\}, B_i\text{'s disjoint}$ s.t. $\forall i \; f(x) \neq f(x^{B_i}) \}$

$$x^B = x \oplus \bigoplus_{i \in B} e_i$$

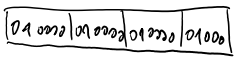
... block sensitivity of f at x

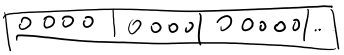
$$bs(f) = \max_x bs_x(f)$$



Ex: $f(\bar{x}_1 \dots \bar{x}_{\sqrt{n}}) = 1$ iff in some $\bar{x}_i$ there are two consecutive 1s & rest of $\bar{x}_i$ is 0.

$\bar{x}_i \in \{0,1\}^{\sqrt{n}}$

$s(f) = 2\sqrt{n}$ ↑ on input 

$bs(f) = n/2$ ↑ on 

Claim: $s(f) \le bs(f)$ trivo

Thm (Huang 2019): ("sensitivity conjecture") $bs(f) \le (s(f))^4$.

- $C : S \to \{0,1\}$ $S \subseteq \{1,\dots,n\}$

C is a certificate for f if $\forall x, x'$ consistent with C $f(x) = f(x')$.

Claim: $bs(f) \le C(f)$ trivo $bs_x(f) \le C_x(f)$

Claim: $C(f) \le bs(f) \cdot s(f)$

Pf: For given $x \in \{0,1\}^n$ consider $B_1, \dots B_b$ for $b = s_x(f)$ where $\sum_i |B_i|$ is minimal

$\cup B_i$ set to values in x is a certificate for x

o/w $\exists B_{b+1}$ $f(x) \neq f(x^{B_{b+1}})$ which ↑ disjoint from B_i's.

would give $s_x(f) \ge b+1$.

Also $|B_i| \le s(f)$ o/w x^{B_i} is sensitive to more var's than $s(f)$ violating def. of $s(f)$. ▨

Claim: $C(f), bs(f) \le D(f)$.

Pf: dec. tree gives certificates; on x, dec. tree must query each sensitive block. ▨

Claim: $D(f) \le C(f) \cdot bs(f) \le (bs(f))^3$

Pf: to build the dec. tree, use the following alg:

1) repeat $bs(f)$ - times:
   Pick a 1-certificate C consistent w/ queries thus far.
   Query all its var's. (If no 1-certificate $\Rightarrow$ output 0).
   If the input consistent with C output 1 o/w continue with next iteration.

2) output $f(y)$ for any y consistent with all the queries asked.

Claim: In 2) $\forall y, y'$ $f(y) = f(y')$

Pf: By contradiction: assume $\exists y, y'$ $f(y)=0$ $f(y')=1$

Let C_{b+1} be 1-certificate for y'.

For $i = 1, \dots b+1$ let B_i = var's on which y & C_i disagree

$f(y^{B_i}) = 1$ so y is sensitive in B_i's. It suffices to show that B_i's are disjoint to reach a contradiction with $bs(f) = b$.

$$\forall k \in B_i \Rightarrow x_k = y_k \neq C_i(k)$$
$$i \leq b$$

$\forall j > i$; C_j is consistent with $x_k \neq y_k$ so $k \notin B_j$ ☒

Thm: (Nisan - Szegedy): $D(f) \leq deg(f)^2 \cdot bs(f) \leq 2\, deg(f)^4$.

- $bs(f) \leq 2 deg(f)^2$
- $bs(f) \leq 6 \widetilde{deg}(f)^2$

$\Rightarrow$

$\widetilde{deg}$ approx. degree: $|\tilde{p}(x) - f(x)| \leq 1/3$ $\forall x \in \{0,1\}^n$.