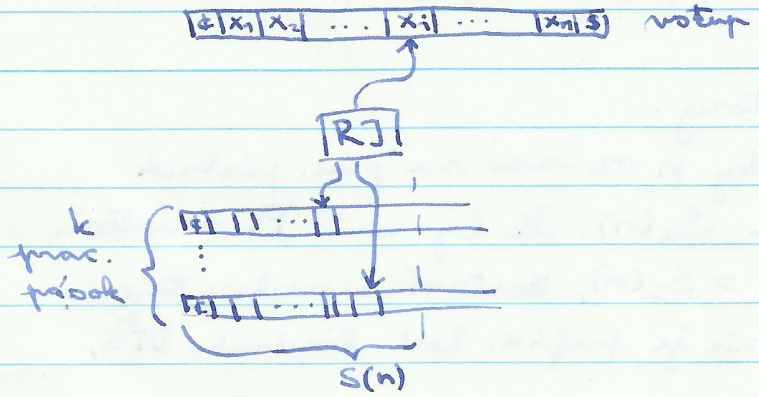
24.09.2014

VÝPOČTOVÝ MODEL:

k-páskový T-stroj



ZLOŽITOSTNÉ TRIEDY

DTIME(T(n)) - trieda jazykov akceptovateľných DTS v čase T(n)

NTIME(T(n)) ——— || ——— NTS ——— || ———

DSPACE(S(n)) ——— || ——— DTS s pamäťou S(n)

NSPACE(S(n)) ——— || ——— NTS ——— || ———

$P = \bigcup_{k>0} DTIME(n^k)$ $PSPACE = \bigcup_{k>0} DSPACE(n^k)$ $EXP = \bigcup_{k>0} DTIME(2^{n^k})$

$NP = \bigcup_{k>0} NTIME(n^k)$ $NSPACE = \bigcup_{k>0} NSPACE(n^k)$ $NEXP = \bigcup_{k>0} NTIME(2^{n^k})$

Def.: Fcia S(n) je páskovo konštrukovateľná, ak ex. DTS ktorý na každom vstupe dĺžky n nepoužije pre žiadnej prac. pásky viac než S(n) políčok a na 1. prac. páske použije práve S(n) políčok.

Ozn.: DTS M s 1 prac. páskou. Kód pre M (ozn. $\langle M \rangle$) je binárny reťazec

$$111\,kód_1\,11\,kód_2\,11 \ldots 11\,kód_n\,111$$

kde každý $kód_i$ je tvaru

$$0^k 1 0^l 1 0^m 1 0^r 1 0^s 1 0^t,$$

ktorý kóduje prechod. fcia $\delta(q_i, a_k, a_l) = (q_m, d_r, a_s, d_t)$

(Podobne kódujeme k-páskové stroje.)

Prefixový kód pre M je $\underbrace{11\ldots1}_{n}\langle M \rangle$ pre ľubov. n.

Veta: (O pamäťovej hierarchii)

Nech $\log n \leq S_1(n) = o(S_2(n))$, kde $S_2(n)$ je pás. konšt.
Potom $DSPACE(S_1(n)) \subsetneq DSPACE(S_2(n))$.

D.:

Nech $\bar{M}$ je DTS ktorý:

(i) Na vstupe dĺžky n označí na prac. páskach pamäť rozsahu $S_2(n)$. Ak by mal $\bar{M}$ v ďalšom použiť pamäť $> S_2(n)$, zastaví a neakceptuje.

(ii) Ak vstup w nie je prefix. kód žiadneho DTS, $\bar{M}$ neakceptuje.

(iii) Nech w je prefix. kód nejakého DTS M. Potom $\bar{M}$ (rešpektujúc (i)) simuluje $2^{S_2(n)}$ krokov stroja M na w a $\bar{M}$ akceptuje iff M neakc. počas $2^{S_2(n)}$ krokov.

Pre všetky M (↑ DTS s pásk. abc. $S_1(n)$): $L(M) \neq L(\bar{M})$. (Jeraat dokázaťeme.)

$L(\bar{M}) \in DSPACE(S_2(n))$, tj. $L(\bar{M}) \notin DSPACE(S_1(n))$

Nech M má k pások, t páskových symbolov, q stavov.
Potom $\forall$ vstupy dĺžky n platí:

$$(n+2)\, q\, ((S_1(n)+1)\, t^{S_1(n)})^k \leq \text{max. \# krokov } M \text{ (bez cyklu)}$$

Pre každý dostatočne dlhý vstup w ($|w|=n$) platí:

(a) M akceptuje w iff M akceptuje w počas najviac $(n+2)\, q\, ((S_1(n)+1)\, t^{S_1(n)})^k \leq 2^{S_2(n)}$ krokov
(lebo $\log n \leq S_1 < o(S_2(n))$)

Pre dostatočne dlhý pref. kód stroja M platí:

(b) $\bar{M}$ je schopný (nepresiahnuc pamäť $S_2(n)$) simulovať $2^{S_2(n)}$ výpočtov M na w.

(a) $\wedge$ (b) $\wedge$ (iii) $\Rightarrow$ $\bar{M}$ akceptuje w $\Leftrightarrow$ M neakceptuje w

Preto pre ľubovoľný M s pam. zl. $S_1(n)$: $L(M) \neq L(\bar{M})$. □

01.10.2014

Def.: Fcia $T(n)$ je <u>časovo konštruovateľná</u> ak existuje DTS, ktorý na každom vstupe dĺžky n vykoná práve $T(n)$ krokov.

Veta: (O časovej hierarchii)
Nech $T(n) \leq T_1(n) = O(T_2(n)/\log T_1(n))$, kde $T_2(n)$ je časovo konštruovateľná. Potom $DTIME(T_1(n)) \subsetneq DTIME(T_2(n))$

D.: Podobný dôkazu predošlej vety.

Veta: (Savitch)
Nech $S(n) \geq \log n$ je páskovo konštruovateľná.
Potom $NSPACE(S(n)) \subseteq DSPACE(S^2(n))$.

D.: M - ľub. NTS s pamäťovou zložitosťou $S(n)$ majúci k pások, q stavov, t páskových symbolov. Na vstupe dĺžky n sa M môže dostať do $\leq q(n+2)((S(n)+1)t^{S(n)})^k \leq c^{S(n)}$ rôznych konfigurácií (pre nejaké c).
↳ Označ.: $S(n)$-ohraničené konfigurácie

Jeda: ak $x \in L(M)$ potom $\exists$ akceptačný výpočet M na x majúci $\leq c^{S(n)}$ krokov.

Zrejme: M sa dostane z konf. C_1 do konf. C_2 počas $\leq i$ krokov iff sa M dostane z C_1 do C_3 počas $\leq \lfloor i/2 \rfloor$ krokov a z C_3 do C_2 počas $\leq \lceil i/2 \rceil$ krokov.

Algoritmus pre simulovanie M:

```
begin nech x je vstup, |x|=n, nech C0 je poč. konf. M
; foreach S(n)-ohranič. akc. konf. Cf:
;     if TEST(C0, Cf, c^S(n)) then accept
end

procedure TEST(C1, C2, i): // zistí, či M prejde z C1 do C2 počas ≤ i krokov
  if C1 = C2 alebo M prejde z C1 do C2 v 1 kroku then return true
  if i > 1 then:
     foreach S(n)-ohr. konf. C3:
        if TEST(C1, C3, ⌊i/2⌋) and TEST(C3, C2, ⌈i/2⌉) then return true
  return false
```

Uvedený algoritmus možno implementovať na DTS M', ktorý:
- jednu z pások používa ako zásobník
- pri každom volaní proc. TEST pridá do zásobníka 1 blok obsahujúci C_1, C_2, C_3, TEST$(C_1, C_3, \lfloor i/2 \rfloor)$, TEST$(C_3, C_2, \lceil i/2 \rceil)$ a adresu návratu pre volanie TEST (tj. volanie medzi if a and alebo medzi and a then alebo z hl. programu).

$\Rightarrow$ rozsah bloku je $O(S(n))$, lebo $i \leq c^{S(n)} \Rightarrow$ hĺbka rekurzie (= # blokov v zásobníku) je $\leq 1 + \log_2 c^{S(n)} = O(S(n))$ (lebo 3. parameter TEST-u je zmenšený na polovicu $\lfloor i/2 \rfloor$ pri každom volaní) $\Rightarrow M'$ má pam. zl. $O(S^2(n))$.

$\Rightarrow M'$ možno prerobiť na DTS s páskovou zložitosťou $S^2(n)$ akc. rovnaký jazyk (redukcia pásky – viac susediacich symbolov $\Rightarrow$ 1 symbol). □

Veta: (a) $\mathrm{DTIME}(T(n)) \subseteq \mathrm{NTIME}(T(n)) \subseteq \mathrm{DSPACE}(T(n))$
$\forall$ čas. zložitosť $T(n)$

(b) $\mathrm{DSPACE}(S(n)) \subseteq \mathrm{NSPACE}(S(n)) \subseteq \bigcup_{c>0} \mathrm{DTIME}(c^{\log n + S(n)})$
$\forall$ pás. zložitosť $S(n)$

Dôk.: (b) M – ľubov. NTS s pam. zl. $S(n)$ (q stavov, k pások, t pásk. symbolov)
x – vstup pre M, $|x| = n$
$G = (V, E)$ – orientovaný
$V = \{ C \mid C$ je konf. $M \} \Rightarrow |V| \leq (n+2) q (S(n)+1)^k t^{S(n)} \leq d^{\log n + S(n)}$ (vľavo vo predp. nie je $S(n) \geq \log n$ ↓)
$E = \{ (C, C') \mid M$ prejde z C do C' v 1 kroku $\}$
$x \in L(M)$ iff $\exists$ cesta v G z poč. konf. do niekt. akcept.
Jo, či v G ∃ taká cesta možno zistiť DTS pomocou vhodne implementovaného alg. (Dijkstra, BFS, ...) v čase
$\leq |V|^m \leq (d^{\log n + S(n)})^m \leq c^{\log n + S(n)}$ □

Dôsledok: $\mathrm{DSPACE}(\log n) \subseteq \mathrm{NSPACE}(\log n) \subseteq P \subseteq NP \subseteq PSPACE = NPSPACE = EXP = NEXP$
(b) (b) (Savitch) (b)

Tvrdenie: (Zjednodušená veta o medzere a zrýchľovaní)

Existuje rekurzívna fcia $s(n) \geq n$ t.ž. $DSPACE(s(n)) = DSPACE(2^{s(n)})$.

(Pozn.: nie je tu podmienka o páske, konštruovateľnosti.)

Dôk.: Nech $m \in \mathbb{N}$, M - ľubov. DTS, x - ľubov. vstup pre M

$s :=$ max. veľkosť použ. pamäte počas výpočtu M na x, $s \in \mathbb{N} \cup \{\infty\}$

$$TEST(m, M, x) = \begin{cases} 1, & \text{ak } s \in \langle m+1, 2^m \rangle \\ 0, & \text{ak } s \notin \langle m+1, 2^m \rangle \end{cases}$$

Existuje alg. počítajúci TEST $\forall m, M, x$?

Áno: $t :=$ # rôznych konf., do kt. sa môže M dostať na x používajúc pamäť $\leq 2^m$. Simulujme M na x počas $t+1$ krokov. Sú 3 možnosti:

1.) M sa zastavil do času t $\Rightarrow$ vieme určiť s aj $TEST(m, M, x)$.

2.) V čase $t+1$ má M použitú pamäť $\leq 2^m$ $\Rightarrow$ nejaká konf. sa zopakovala, M sa zacyklil $\Rightarrow$ vieme s aj TEST

3.) V čase $t+1$ má M použitú pamäť $> 2^m$ $\Rightarrow TEST(\dots) = 0$ lebo $s > 2^m$

Nech $M_1, M_2, M_3, \dots$ je ľubov. efektívne očíslovanie všetkých DTS (tj. k číslu i vieme skonštruovať M_i).

Algoritmus pre $s(n)$:

Nech $s_1 < s_2 < \dots < s_p$ sú všetky veľkosti použitej pamäte v čase zastavenia (ak nejakého) niektorého M_ℓ, $\ell \leq n$ na niektorom jeho vstupe x, $|x| = n$

(tie s-ká sú pre celú množinu strojov M_ℓ, resp. dvojice (M_ℓ, x) pre ľubov. M_ℓ, $\ell \leq n$ a ľubov. x, kde sa M_ℓ na x zastavil)

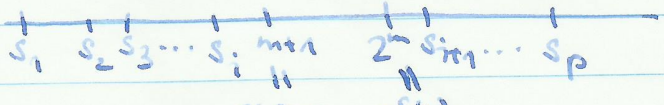
Cieľ - určiť $s(n)$: (pre $m \leftarrow n, n+1, n+2, \dots$)

tak, aby $s_i \notin \langle s(n)+1, 2^{s(n)} \rangle$ pre $i = 1, 2, \dots, p$ (*)

ak $TEST(m, M_j, x) = 0$ $\forall M_j$ $(j \leq n)$

a zároveň $\forall$ vstup x pre M_j, $|x| = n$, pričom $s(n) = m$ (pre také $s(n)$ platí (*))

↳ hľadáme medzeru:



Bude vždy nejaká? Áno, v "najhoršom" prípade s_p.

Nech $L \in DSPACE(2^{S(n)}) \Rightarrow \exists$ DTS M_k akceptujúci L s pamäťou $\leq 2^{S(n)}$. Nech x je vstup pre M_k dĺžky $n \geq k$.

M_k je det. $\Rightarrow M_k$ sa zastaví (alebo zacyklí) na x s použitou pamäťou $s_j \leq 2^{S(n)}$, $1 \leq j \leq p$.

$(*) \Rightarrow s_j \leq S(n) \Rightarrow M_k$ reprezentuje pamäť $S(n)$ na ľubovoľnom vstupe x, $|x| \geq k$.

$\Rightarrow L \in DSPACE(S(n))$. □

08.10.2014

Veta: (Gap Theorem)

Pre každú rekurzívnu fciu $g(n) \geq n$ existuje rekurzívna fcia $S(n) \geq n$ t.ž. $DSPACE(S(n)) = DSPACE(g(S(n)))$

D.: Podobný dôkazu vety vyššie.

Veta: (Speed-up Theorem)

Pre každú rekurzívnu fciu $f(n) \geq n^2$ existuje rekurzívny jazyk L t.ž. pre ľub. DTS M akceptujúci L v čase $T(n)$ existuje DTS M' akceptujúci L v čase $T'(n)$ t. ž. $f(T'(n)) \leq T(n)$ pre skoro všetky n (pre $\forall$ okrem kon. počtu).

Pozn.: ak $f(n) = 2^n \Rightarrow 2^{T'(n)} \leq T(n) \Rightarrow T'(n) \leq \log T(n) \Rightarrow T''(n) \leq \log\log T(n)$

... a tak ďalej.

D.: Bez dôkazu.

Nasleduje dôkaz Cook-Levinovej vety (iný ako v TEA).

Def.: $L \subseteq \Sigma^*$ je polynomiálne transformovateľný na $L_0 \subseteq \Sigma_0^*$ ak $\exists$ DTS s pol. časovou zložitosťou, ktorý vstup $x \in \Sigma^*$ pretransformuje na $y \in \Sigma_0^*$ t.ž.

$x \in L \Leftrightarrow y \in L_0$

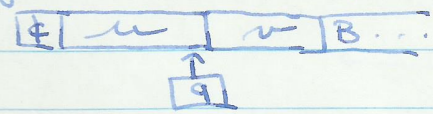
Def.: L_0 je NP-úplný, ak $L_0 \in NP$ a každý $L \in NP$ je polynomiálne transformovateľný na L_0.

Veta: (Cook-Levin)

SAT je NP-úplný.

Dôk.: SAT $\in$ NP (zrejmé). — dopĺňa nekonečnou

Nech $L \in NP$, $L \subseteq \Sigma^*$. Budeme uvažovať TS s 1 páskou. Dá sa dokázať: pre L $\exists$ polynóm $p(n)$ a NTS $M = (\Sigma', Q, \delta, q_0, q_s)$ akceptujúci L v čase $p(n)$. Konfigurácia M: $\not\in u q v$



Nech $x \in \Sigma^*$, $|x| = n$, D je ľubov. reťazec (nad $\Sigma' \cup Q$), $D = D_0 D_1 \dots D_{p(n)}$, $|D_i| = p(n)+2$ $\forall i$. D reprezentuje nejaký akceptačný výpočet dĺžky $p(n)$ stroja M na x iff D_0 je poč. konf., M môže v 1 kroku prejsť z D_i do D_{i+1} $\forall i$ a $D_{p(n)}$ je nejaká akcept. konfigurácia. (*)

Náš cieľ: pre dané x (s ohľadom na M a p) skonštruovať (v pol. čase vzhľadom k $|x|$) bool. výraz $E_x = F \wedge G \wedge H \wedge I$, ktorý bude splniteľný (t.j. $E_x \in SAT$) iff $\exists D$ spĺňajúce (*) (tj. iff $x \in L$).

Konštrukcia E_x: E_x obsahuje bool. premenné $y_{j,s}$, kde $1 \le j \le (p(n)+1)(p(n)+2)$, $s \in \Sigma' \cup Q$.

Interpretácia: $y_{j,s} = 1$ iff j-tý symbol D je s.

(Jedna pre $y_{i,s}$ pre rôzne s, pre nové i bude práve 1 $y_{i,s} = 1$.)

Zostrojme F, G, H, I tak, aby:

(a) F bol splniteľný iff pre každé j $(1 \le j \le (p(n)+1)(p(n)+2))$ práve 1 premenná $y_{j,s}$ $(s \in \Sigma' \cup Q)$ má hodnotu 1.

(b) výraz G resp. H resp. I kontrolovali, či D, určené hodnotami $y_{j,s}$ spĺňajúcimi F, má vlastnosť (*) (G - prvú, H - druhú, I - tretiu časť).

Konštrukcia F, G, H, I:

$$F = F_1 \wedge F_2 \wedge \dots \wedge F_h, \quad h = (p(n)+1)(p(n)+2)$$

$$F_j = \left(\bigvee_{s \in \Sigma' \cup Q} y_{j,s}\right) \wedge \neg\left(\bigvee_{s,s' \in \Sigma' \cup Q, s \neq s'} (y_{j,s} \wedge y_{j,s'})\right) \text{ pre } j = 1, \dots, h$$

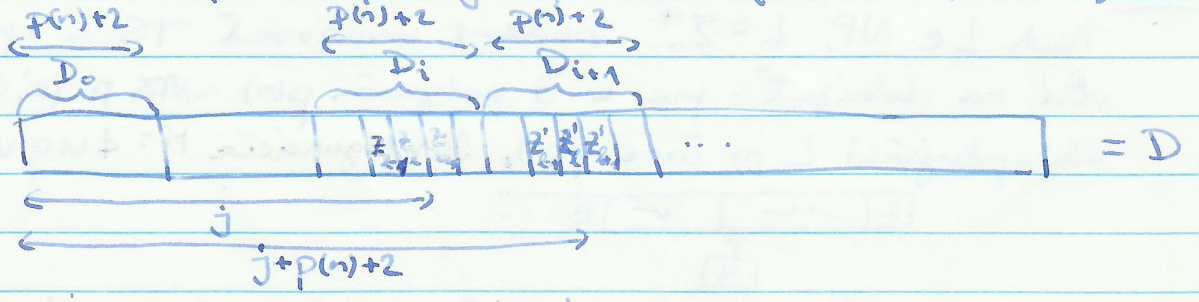
Nech $x = x_1 \dots x_n$, $x_i \in \Sigma$ $\forall i$; $D_0 = \not\in q_0 x_1 \dots x_n \overbrace{B \dots B}^{p(n)-n}$

$$G = y_{1,\not\in} \wedge y_{2,q_0} \wedge y_{3,x_1} \wedge \dots \wedge y_{n+2,x_n} \wedge y_{n+3,B} \wedge \dots \wedge y_{p(n)+2,B}$$

$I = y_{t,q_f} \vee y_{t+1,q_f} \vee \dots \vee y_{t+p(n)+1,q_f}$, $t = p(n)\cdot(p(n)+2)+1$

(čiže I zabezpečí, že niekde v poslednom bloku je q_f

... to, že $D_{p(n)}$ je správna konf. zabezpečí H)



H bude kontrolovať trojice znakov

$$H_i^* = \bigwedge_{i(p(n)+2)+1 \le j \le (i+1)(p(n)+2)} \left(\bigwedge_{r,s,t \in \Sigma'} (y_{j-1,r} \wedge y_{j,s} \wedge y_{j+1,t} \Rightarrow y_{j+p(n)+2,s}) \right)$$

↳ toto je situácia, že v 3ici symbolov nie je stav

(to nám jednoznačne určí, stredný symbol

v zodp. trojici v D_{i+1})

$$H_i^{**} = \bigwedge_{\text{-''- } \le j \le \text{-''-}} \left(\bigwedge_{\substack{r,s \in \Sigma' \\ q \in Q}} y_{j-1,r} \wedge y_{j,q} \wedge y_{j+1,s} \Rightarrow \bigvee_{tuv \in K_{rqs}} (y_{j+p(n)+1,t} \wedge y_{j+p(n)+2,u} \wedge y_{j+p(n)+3,v}) \right)$$

$i = 0 \dots p(n)-1$

kde:

$K_{rqs} = \{ tuv \mid t,u,v \in \Sigma' \cup Q,\ tuv$ môže vzniknúť z rqs podľa $\delta \}$

↳ toto je situácia, že stredný symbol z trojice je stav

(to určí veľa možných 3íc symbolov na zodp.

miestach v D_{i+1} - podľa K_{rqs})

$$H = H_0^* \wedge H_1^* \wedge \dots \wedge H_{p(n)-1}^* \wedge H_0^{**} \wedge H_1^{**} \wedge \dots \wedge H_{p(n)-1}^{**}$$

(pozn.: j ide od 1 pre $i=0$, takže $j-1=0$, čo

by bol problém, lebo $y_{0,s}$ nie je zadef.

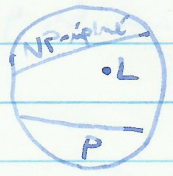
... ošetrenie nejak špeciálne, ale to dá určiť

len technické detaily)

Rôznych premenných je $|D| \cdot |\Sigma' \cup Q|$ = polynóm od $|x|$

... dá sa veriť, že táto transf. je polynomiálna $\square$

Veta: Ak $P \neq NP$, potom $\exists L \in NP$ t. ž. $L \notin P$ a L nie je NP-úplný.



Def.: Jazyk $L \subseteq \Sigma^*$ je riedky, ak $\exists$ polynóm t. ž. $|L \cap \Sigma^n| \leq p(n)$ $\forall n$.

Veta: (Mahaney)
Žiaden riedky jazyk nie je NP-úplný, ak $P \neq NP$.

TU KUS CHÝBA
(VIĎ ANIČKINE POZNÁMKY)

22.10.2014

Veta: Nech A je ľub. NP-optimalizačný problém, ktorého rozhodovací problém L možno akceptovať deterministicky v čase $T(n)$. (L nemusí byť NP-úplný.)
Potom hodnotový problém pre A možno riešiť deterministicky v čase $r(n)T(s(n))$ pre vhodné polynómy r, s.
D.: je uvedený v časti (1) dôkazu predošlej vety.

Veta: (a) Nech A je ľub. NP-optimalizačný problém a nech L je ľub. NP-úplný problém, kt. možno akceptovať deterministicky v čase $T(n)$. Potom konštrukčný problém pre A možno riešiť det. v čase $r(n)T(s(n))$ pre vhodné polynómy r, s.
(b) Dôsledok: Ak $P = NP$, potom konštrukčný problém každého NP-optimalizačného problému možno riešiť deterministicky v polynomiálnom čase.
D.: (a): analogicky predošlému
(b): (a) $\Rightarrow$ $T(n)$ je polynóm, lebo $L \in NP = P \Rightarrow$ $r(n)T(s(n))$ je polynóm $\square$

Pr.: Pre daný graf G nájdi (ak existuje) nejakú hamiltonovskú kružnicu. Je to rozh., hodn. alebo konštr. problém?
→ Konštrukčný, napr. $R = \{(x,y) \mid$ graf x má ham. kr. $y\}$
$m(x,y) = 1 \ \forall x,y$, cieľ je min./max.

APROXIMOVATEĽNOSŤ NP-OPTIMALIZAČNÝCH PROBLÉMOV

Def: Nech A je NP-opt. problém s cieľom min (max), reláciou R a hodnotiacou fciou m. Nech

$m^*(x) = \min_{y \in \Sigma^*} \{ m(x,y) \mid (x,y) \in R \}$ pre cieľ min

$m^*(x) = \max_{y \in \Sigma^*}$ ——— || ——— pre cieľ max

Problém A je α-aproximovateľný, $\alpha > 1$, ak ex. det. pol. algoritmus M, ktorý pretransformuje vstup x na výstup y, tj. $M(x) = y$, pričom $(x,y) \in R$ a $\left.\begin{array}{l} m^*(x) \geq m(x, M(x)) \text{ pre cieľ min} \\ m^*(x) \leq \alpha m(x, M(x)) \text{ pre cieľ max} \end{array}\right\}$ pre skoro $\forall x$

Def.: Problém A je dobre aproximovateľný, ak je α-aproximovateľný $\forall \alpha > 1$. $(\alpha \to 1)$

Def.: Problém A je neaproximovateľný, ak nie je α-aproximovateľný pre žiadne $\alpha > 1$. $(\alpha \to \infty)$

Pr.: NP-opt. problém 0-1-knapsack
cieľ: max
$R = \{(x,y) \mid x = d(v_1)\#\ldots\#d(v_n)\#d(w_1)\#\ldots\#d(w_n)\#d(W),$
$y = d(S),\ S \subseteq \{1,\ldots,n\},\ \sum_{i \in S} w_i \leq W\}$
kde v_i - cena, w_i - váha, W - nosnosť, d - reprezentácia čísla/množ.
$m(x,y) = \sum_{i \in S} v_i$
→ Rozhodovací problém 0-1-knapsack
$L = \{x\#d(k) \mid \exists S \subseteq \{1,\ldots,n\} : \sum_{i \in S} w_i \leq W \ \sum_{i \in S} v_i \geq k\}$

Veta: L je NP-úplný.

Konštrukčný problém 0-1-knapsack

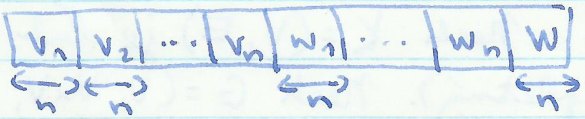
Pre vstup $v_1,\dots,v_n,w_1,\dots,w_n,W$ (formálne pre vstup x) nájdi výstup $S\subseteq\{1,\dots,n\}$ (formálne $y=d(S)$) t.ž.
$(x,y)\in R \wedge m(x,y)=\sum_{i\in S} v_i = \max_{S'\subseteq\{1,\dots,n\}}\{\sum_{i\in S'} v_i \mid \sum_{i\in S} w_i \le W\}$.

Algoritmus pre konštrukčný problém 0-1-knapsack
(alg. nájde opt. výber S t.ž. $\sum_{i\in S} v_i = \max_{S'\subseteq\{1,\dots,n\}}\{\sum_{i\in S'} v_i \mid \sum_{i\in S'} w_i \le W\}$)

(1) Vyplň tabuľku $W(0..n, 0..nV)$, kde $V=\max_i v_i$ pomocou vzťahu $W(i+1,v)=\min\{W(i,v), W(i,v-v_{i+1})+w_{i+1}\}$, začni s $W(i,v)=\infty\ \forall i,v\ (i,v\neq 0)$, $W(0,0)=0$.
Pozn.: platí $W(i,v)=\min_{S'\subseteq\{1,\dots,n\}}\{\sum_{j\in S'} w_j \mid \sum_{j\in S'} v_j = v\}$

(2) $u \leftarrow \max\{v \mid W(n,v)\le W\}$
$S\leftarrow\emptyset$
for $i\leftarrow n-1$ to 0 do
  if $W(i,u)>W(i,u-v_{i+1})+w_{i+1}$ then $S\leftarrow S\cup\{i+1\}$, $u\leftarrow u-v_{i+1}$

Časová zložitosť: $O(n^2V)$ – nie nutne polynomiálna, napr. ak sú ceny veľké
$n=\Theta(\sqrt{|x|})$
$|d(v_i)|=|d(w_i)|=|d(W)|=\Theta(\sqrt{|x|})$
$V=2^{\Theta(\sqrt{|x|})} \Rightarrow$ časová zložit. $O(|x|\cdot 2^{\Theta(\sqrt{|x|})})$ – nie pol. ke $|x|$



Aproximačný algoritmus pre konštrukčný 0-1-knapsack
(pre ľubl. α, $2\ge\alpha>1$, alg. nájde S' tž. $\sum_{i\in S} v_i \le \alpha \sum_{i\in S'} v_i$ a $\sum_{i\in S'} w_i \le W$)

(1) $d\leftarrow\left\lfloor\frac{(\alpha-1)V}{n+1}\right\rfloor$ (pre $d=0$ (tj. $(\alpha-1)V<n+1$) použi presný alg., lebo $O(n^2V)=O(\frac{n^3}{\alpha-1})$)

(2) Nájdi opt. výber S' presného alg. pre konštr. problém 0-1-k. pre vstup $v'_1,\dots,v'_n,w_1,\dots,w_n,W$ kde $v'_i=\left\lfloor\frac{v_i}{d}\right\rfloor$ $\forall i$

Časová zložitosť: $O(n^2V')$, kde $V'=\max\{v'_1,\dots,v'_n\}=v'_\ell=\left\lfloor\frac{v_\ell}{d}\right\rfloor=\left\lfloor\frac{V}{d}\right\rfloor\le\frac{V}{d}$
$\Rightarrow$ čas $O(n^2\frac{V}{d})=O(\frac{n^3}{\alpha-1})$-polynomiálny

Nepresnosť aprox. algoritmu:

Platí: $\sum_{i\in S'} v_i \geq d \sum_{i\in S'} v'_i \geq d \sum_{i\in S} v'_i \geq \sum_{i\in S} (v_i - d) \geq \sum_{i\in S} v_i - nd \quad (+)$

$v_i \geq d v'_i$; S' je opt. pre v'_i ; $1 + v'_i \geq v_i/d$; $|S| \leq n$

Platí: $\frac{nd}{\alpha-1} \leq V - \frac{d}{\alpha-1} \leq V - d = V_e - d \leq d\left\lfloor \frac{V_e}{d} \right\rfloor \leq d \sum_{i\in S'} v'_i \leq \sum_{i\in S'} v_i \quad (**)$

$d \leq \frac{(\alpha-1)V}{n+1}$; $1 < \alpha \leq 2$; $V'_e \leq \sum_{i\in S'} v'_i$ (predpoklad $w_i \leq W \ \forall i$)

$$\Rightarrow \underbrace{\sum_{i\in S} v_i}_{\text{cena opt. výberu}} \overset{(*)}{\leq} nd + \sum_{i\in S'} v_i \overset{(**)}{\leq} (\alpha-1) \sum_{i\in S'} v_i + \sum_{i\in S'} v_i = \underbrace{\alpha \sum_{i\in S'} v_i}_{\text{cena nájdeného výberu}}$$

↳ Číže α-aproximovateľnosť je zaručená

Veta: 0-1-knapsack je dobre aproximovateľný NP-opt. problém, ale jeho rozhodovací problém je NP-úplný (t.j. nepatrí do P ak P ≠ NP).

29.10.2014

Veta: Ak by pre nejaké $\alpha > 1$ existoval determ. pol. algoritmus A, ktorý by pre každý graf G s celočísl. hranami našiel ham. kružnicu H t.ž. $c(H) \leq \alpha c(H^*)$, kde $c(X)$ je cena kružnice X a H^* je najlacnejšia ham. kr. pre G, potom by P = NP.

D.: SPOROM

Nech by $\exists \alpha$, A. Nech $K = (V, E)$ je ľubov. graf (nie nutne kompletný). Nech $G = (V, V \times V, c)$ je kompletný graf t.ž. $c(u,v) = \begin{cases} 1, & \text{ak } (u,v) \in E \\ \alpha|V|+1, & \text{ak } (u,v) \notin E \end{cases}$

Nech H je ham. kr. nájdená algoritmom A pre G, nech H^* je najlacnejšia ham. kr. pre G.

Potom platí:

(1) Ak K má nejakú ham. kr. H', potom $c(H) \leq \alpha c(H^*)$ (lebo A je aprox. alg. s koeficientom α),
a $\alpha c(H^*) \leq \alpha c(H')$ (lebo H' je nejaká kruž. v G)
a $\alpha c(H') \leq \alpha |V|$ (lebo $|V| = |H'|$, $c(e) = 1 \ \forall e \in E$)
⇒ teda $c(H) \leq \alpha |V|$

(2) Ak K nemá žiadnu kružnicu (a teda aspoň 1 hrana v H nepatrí do E), potom $c(H) \geq \alpha|V|+1$.

↳ týmto spôsobom vieme rozlíšiť, či K má ham. kr., a to v polynomiálnom čase, tj. vieme riešiť NP-úplný problém v pol. čase $\Rightarrow P = NP$ □

Veta: Ak $P \neq NP$, potom ex. NP-opt. problémy A, B, C t. ž.:

(i) A je dobre aproximovateľný, ale jeho rozhod. problém nepatrí do P

(ii) B je α-aproximovateľný, ale je dobre aproximovateľný

(iii) C je neaproximovateľný

D.: A, B, C sú rovnaké až na hodnot. fcie:

cieľ = max

$R = \{(x,y) \mid x$ je graf, y je ~~pod~~postupnosť všetkých vrcholov$\}$

$$m_A(x,y) = \begin{cases} |x|, & \text{ak } y \text{ je ham. kr. pre } x \\ |x|-1, & \text{inak} \end{cases}$$

$$m_B(x,y) = \begin{cases} 2, & \text{ak } y \text{ je ham. kr. pre } x \\ 1, & \text{inak} \end{cases}$$

$$m_C(x,y) = \begin{cases} |x|, & \text{ak } y \text{ je ham. kr. pre } x \\ 1, & \text{inak} \end{cases}$$

Nech Π je DTS, ktorý v pol. čase pretransformuje graf X na postupnosť vrcholov grafu X (y_x) (v nejakom usporiad.), tj. $\Pi(x) = y_x$.

(i) A je dobre aprox., lebo

$$m_A^*(x) = \max\{m_A(x,y) \mid (x,y) \in R\} \leq |x| \leq \alpha(|x|+1) \leq \alpha\, m_A(x, \Pi(x))$$

(pre každé x a skoro $\forall x$)

Nech HAM je problém, či má graf ham. kr. Z definície $m_A(x,y) \Rightarrow x$ má ham. kr. iff $\exists y: R(x,y) \wedge m_A(x,y) \geq |x|$

tj. iff riešenie rozh. problému pre A je "áno" pre graf x a $k = |x|$ (rozh. problém: vstup x, k, otázka či $\exists y: R(x,y) \wedge m_A(x,y) \geq k$)

$\Rightarrow$ ak by sme vedeli det. v pol. čase riešiť rozhod. problém pre A, potom aj HAM $\in P \Rightarrow$ spor s $P \neq NP$. ✓

(ii) B je 2-aprox., lebo $m_B^*(x) = \max\{m(x,y) \mid (x,y) \in R\} \leq 2$
$\leq 2\, m_B(x, M(x))$, keďže $m_B(x,y) \in \{1,2\}$
Predpokladajme, že by B bol dobre aprox $\Rightarrow$ B by bol aj $\frac{3}{2}$-aprox., tj. $2m_B^*(x) \leq 3\, m_B(x, M(x))$, kde M je DPA z def. α-aproximovateľnosti. Pomocou alg. M a alg. pre výpočet m_B by sme vedeli det. a v pol. čase vypočítať aj $m_B^*(x)$ takto:
- Ak $m_B(x, M(x)) = 1$, potom $m_B^*(x) = 1$ (lebo $m_B^*(x) \in \{1,2\}$)
  lebo $2m_B^*(x) \leq 3\, m_B(x, M(x)) = 3$
- Ak $m_B(x, M(x)) = 2$, potom $m_B^*(x) = 2$,
  lebo $m_B(x, M(x)) \leq m_B^*(x) \leq 2$
  ↑ lebo m_B^* je max pre m_B

Z def. $m_B(x,y) \Rightarrow m_B^*(x) = \max\{m_B(x,y) \mid (x,y) \in R\}$
$= 2$ iff $\exists y : (x,y) \in R \wedge m_B(x,y) = 2$
t.j. iff $\exists$ ham. kr. pre x.
Preto ak by sme det. a v pol. čase vedeli vyp. $m_B^*(x)$, potom by HAM $\in P$ - spor s predp. $P \neq NP$. ✓

(iii) V skriptách. □

Veta: Ak $P \neq NP$, potom:
(1) Maximálna klika grafu a najdlhšia cesta v grafe sú neaproximovateľné NP-opt. problémy.
(2) Problém kontajnerov je $\frac{3}{2}$-aproximovateľný, ale nie $(\frac{3}{2}-\varepsilon)$-aproximovateľný NP-opt.-problém.
(3) Problém obchodného cestujúceho s trojuholníkovou nerovnosťou je $\frac{3}{2}$-aproximovateľný.
(Bez 3un je to neaprox.)
(4) 0-1-knapsack je dobre aproximovateľný

Poznámka: Problém kontajnerov
Vstup: $d_1, \ldots, d_n, m, C \in \mathbb{N}$
Otázka: Možno $d_1, \ldots, d_n$ rozdeliť do m disj. množ. tak že $\max_{i=1}^{m} \sum_{\text{množ}_i} \leq C$?

PRAVDEPODOBNOSTNÉ ALGORITMY

Ozn.:

(1) rand - vráti náhodné reálne $y \in \langle 0, 1)$

(2) grand(a,b) $\langle a, b)$

(3) irand(i,j) celé $y \in \langle i, j \rangle$

(4) brand - vráti 0 alebo 1

↳ stačí 1, zvyšné sa dajú dodefinovať pomocou nelo

rand pomocou brand:

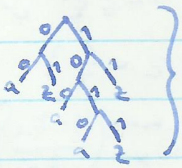
vykonaj brand r-krát s výsledkami $b_1, \dots, b_r$

return $y = 0, b_1 \dots b_r$

Def.: Pravdepodobnostný alg. A akceptuje jazyk $L \subseteq \Sigma^*$ s chybou ε $(0 < \varepsilon < 1/2)$, ak $\forall x \in \Sigma^*$ platí:

$p(A \text{ akceptuje } x \notin L) \leq \varepsilon \wedge p(A \text{ zamietne } x \in L) \leq \varepsilon$.

Pr.: Nech A používa brand



} strom pravd. výpočtu A na x

Do konkrétneho listu s hĺbkou l sa A na x dostane s pravdep. 2^{-l}

↳ pre tento konkrétny strom A akceptuje x s pravd.

$2^{-2} + 2^{-3} + 2^{-4} = 7/16$, zamietne s pr. $2^{-3} + 2^{-4} + 2^{-2} = 9/16$

Veta: (O vylepšovaní)

Nech A je pravdep. alg. akceptujúci L v čase $T(n)$ s chybou ε.

Potom $\forall \varepsilon'$, $0 < \varepsilon' < \varepsilon$ $\exists$ pravd. alg. A' akceptujúci L v čase $O(T(n))$ s chybou ε'.

D.: Algoritmus A':

A' akceptuje (zamietne) x, ak spomedzi m náhodne vybraných výpočtov A na x (A' ich vyberie a simuluje) je # akceptačných (zamietajúcich) > # zamiet. (akcept.)

↳ m je nepárne č. t. ž. $\frac{1}{2}\left(4(1-\varepsilon)\varepsilon\right)^{m/2} \leq \varepsilon'$

$0 < 4(1-\varepsilon)\varepsilon = 4\left(\frac{1}{2}+\alpha\right)\left(\frac{1}{2}-\alpha\right) = 1 - 4\alpha^2$, $\alpha \in (0, \frac{1}{2})$ < 1, preto také m existuje

Pravdepodobnosť chyby A' na x je $\leq \frac{1}{2}(4(1-\varepsilon)\varepsilon)^{m/2}$, lebo:

1.) ak $x \in L$

Nech ε_x je p. toho, že náhodne vybr. výpočet A na x zamietne

$\Rightarrow 0 \leq \varepsilon_x \leq \varepsilon$ a $1-\varepsilon_x$ = p. toho, že náh. vybr. výp. akceptuje

Pr.: p - pravd. toho, že post. S náhod. vybr. výpočtov A na x je trojica a,z,z,a,z

$$p = (1-\varepsilon_x)\varepsilon_x\varepsilon_x(1-\varepsilon_x)\varepsilon_x = \varepsilon_x^3(1-\varepsilon_x)^2$$

$\binom{5}{2}$ = # postupností s dvomi "a" a troma "z"

$\binom{5}{2}\varepsilon_x^3(1-\varepsilon_x)^2$ = pravd., že z 5 vybr. výp. sú práve 2 akc. a 3 zam.

$\sum_{j=0}^{\lfloor m/2 \rfloor} \binom{m}{j}(1-\varepsilon_x)^j \varepsilon_x^{m-j}$ = pravd. toho, že A' zamietne x (m náh. výpoč. A na x, $\frac{m}{2}$ zamietne)

(obsahuje viac zam. než akc., lebo m je nepárne)

= pravd. toho, že A' zamietne $x \in L$

= pravd. chyby A' na $x \in L$

05.11.2014 Ak $x \in L$:

Prípad 1a: $\varepsilon_x = 0 \Rightarrow$ celá suma $= 0 \leq \frac{1}{2}(4(1-\varepsilon)\varepsilon)^{m/2} \leq \varepsilon'$ ✓

Prípad 1b: $\varepsilon_x > 0 \Rightarrow 1 < \frac{1-\varepsilon_x}{\varepsilon_x}$ (lebo $0 < \varepsilon_x \leq \varepsilon < \frac{1}{2}$)

$$\Rightarrow \sum_{j=0}^{\lfloor m/2 \rfloor} \binom{m}{j}(1-\varepsilon_x)^j \varepsilon_x^{m-j} \leq \sum_{j=0}^{\lfloor m/2 \rfloor} \binom{m}{j}(1-\varepsilon_x)^j \varepsilon_x^{m-j}\left(\frac{1-\varepsilon_x}{\varepsilon_x}\right)^{m/2-j}$$

(lebo $\frac{1-\varepsilon_x}{\varepsilon_x} > 1$)

$$= \sum_{j=0}^{\lfloor m/2 \rfloor} \binom{m}{j}(1-\varepsilon_x)^{m/2}\varepsilon_x^{m/2} = (1-\varepsilon_x)^{m/2}\varepsilon_x^{m/2}\sum_{j=0}^{\lfloor m/2 \rfloor}\binom{m}{j}$$

$$= 2^{m-1}(1-\varepsilon_x)^{m/2}\varepsilon_x^{m/2} = \frac{1}{2}\left(2^2(1-\varepsilon_x)\varepsilon_x\right)^{m/2} \leq \frac{1}{2}(4(1-\varepsilon)\varepsilon)^{m/2}$$

(lebo $\varepsilon_x \leq \varepsilon \Rightarrow \varepsilon_x - \varepsilon_x^2 \leq \varepsilon - \varepsilon^2$)

(geometricky: ε_x a $1-\varepsilon_x$ vs. ε a $1-\varepsilon$ — oba obdĺžniky majú obvod 2, ktorý má väčšiu plochu? Ten "štvorcovejší", tj. $\varepsilon(1-\varepsilon) \geq \varepsilon_x(1-\varepsilon_x)$)

$\rightsquigarrow$ no a $\frac{1}{2}(4(1-\varepsilon)\varepsilon)^{m/2} \leq \varepsilon'$ (lebo m sme zvolili tak, aby to platilo). ✓

Prípad 2: ak $x \notin L$ - dôkaz je podobný. □

Dôsledok: Ak pre každé n vykoná A' na každom vstupe x dĺžky n práve $2n+1$ náhodne vybraných výpočtov A na x, potom A' akceptuje $L \cap \Sigma^n$ v čase $O(nT(n))$ s chybou $\frac{1}{2}(4(1-\varepsilon)\varepsilon)^{n+\frac{1}{2}}$.

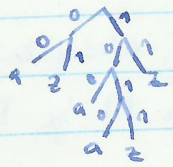
Def.: $BPP = \{L \mid \exists$ PPT alg. akceptujúci L s chybou ε $(0 \le \varepsilon < 1/2)\}$
(bounded probabilistic polynomial)

Veta: Nech $L \in BPP$, $L \subseteq \{0,1\}^*$. Potom pre L $\exists$ polynóm $p(n)$ a PPT alg. B t.ž. $\forall n$ $\exists y_n \in \{0,1\}^{p(n)}$ t.ž. $\forall x \in \{0,1\}^n$ platí: výpočet B na x s hodnotami brand určenými reťazcom y_n je správny (tj. B poznajúc y_n nerobí chyby na žiadnom $x \in \{0,1\}^n$).

D.: $L \in BPP \Rightarrow \exists$ PPT alg. akceptujúci L s chybou $\varepsilon < 1/2$.
Veta o vylepšovaní $\Rightarrow \exists$ PPT alg. A akcept. L v čase $S(n)$ s chybou $\frac{1}{7} =: \varepsilon'$
Dôsledok $\Rightarrow \exists$ PPT alg. A', ktorý v čase $O(nS(n))$ akceptuje $L \cap \{0,1\}^n$ s chybou $\frac{1}{2}(4(1-\frac{1}{7})\frac{1}{7})^{n+1/2} = \frac{1}{2}\left(\frac{24}{49}\right)^{n+1/2} < \left(\frac{1}{2}\right)^{n+3/2}$ $\forall n$ (*)

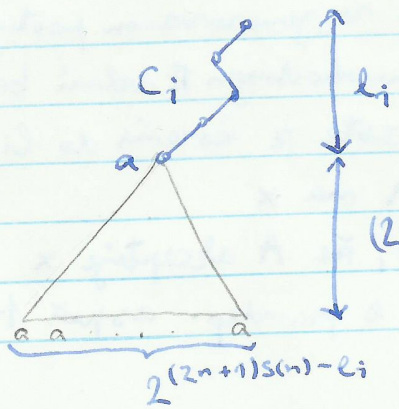
Fakt: Nech $c_1, \dots, c_r$ sú všetky nesprávne výpočty A' na x s počtom $l_1, \dots, l_r$ volaní procedúry brand. Potom $\sum_{i=1}^{r} 2^{-l_i}$ = pravdepodobnosť chyby A' na x (lebo A' vykoná c_i s pravdepodobnosťou 2^{-l_i}).

P1.:



c_1, c_2, c_3 – akceptoval $x \notin L$
$l_1 = 2, \quad l_2 = 3, \quad l_3 = 4$

Upravme A' tak, aby po ľubovoľnej výsledku (akc./zam.) vykonal na x dĺžky n ešte toľko volaní brand, aby ich celkový # bol $(2n+1)S(n)$. Nech A'' označuje upravený A'.



$\Rightarrow$ z každého c_i môže A'' vytvoriť $2^{(2n+1)S(n)-l_i}$ nesprávnych výpočtov

$\forall x \in \{0,1\}^n$: $Y_x^n := \{ y \in \{0,1\}^{(2n+1)S(n)} \mid$ výpočet A'' na x určený y_n je nespr. $\}$

$|Y_x^n|$ = # nesprávnych výpočtov A'' na x = $\sum_{i=1}^{r} 2^{(2n+1)S(n)-l_i}$

$= 2^{(2n+1)S(n)} \sum_{i=1}^{r} 2^{-l_i} = 2^{(2n+1)S(n)} \cdot$ (pravd. chyby A' na x)

$\uparrow$ fakt

$\leq 2^{(2n+1)S(n)} \left(\frac{1}{2}\right)^{n+3/2}$ (**)

$\uparrow$ (*)

$Z^n := \{0,1\}^{(2n+1)S(n)} - \underbrace{\bigcup_{x \in \{0,1\}^n} Y_x^n}$

ten sú "nesprávne" y_n (vedúce k nesprávnym výpočtom A'')

$|Z^n| \geq 2^{(2n+1)S(n)} - \sum_{x \in \{0,1\}^n} |Y_x^n| \overset{(**)}{\geq} 2^{(2n+1)S(n)} \left(1 - 2^n \left(\frac{1}{2}\right)^{n+3/2}\right)$

$= 2^{(2n-1)S(n)} \underbrace{\left(1 - \left(\frac{1}{2}\right)^{3/2}\right)}_{\geq 1/2} \geq 1 \Rightarrow Z^n \neq \emptyset$

$\Rightarrow \forall n \; \exists y_n \in Z^n \Rightarrow$ výpočet A'' na x určený reťazcom y_n je správny $\forall x$: $|x| = n$.

$\Rightarrow$ veta platí pre $B = A''$ a $p(n) = (2n+1)S(n)$. □

Veta: $P \subseteq BPP \subseteq PSPACE$

Dôk.: $P \subseteq BPP$ – z definície

Nech $L \in BPP$, nech A je pravd. alg. akceptujúci L v čase $p(n)$ (pol.) s chybou ε. Nech M je DTS s pamäťou $p(n)$, ktorý na vstupe x:

- na jednej z prac. pások postupne generuje všetky bin. postupnosti dĺžky $\leq p(|x|)$
- po vygenerovaní každej post. M simuluje A na x s hodnotami brand určenými vygenerovanou postupnosťou, pričom ignoruje postupnosti s nevhodným # volaní brand, ktoré nezodpovedajú žiadnej ceste z koreňa do listu v strome pravd. výpočtu A na x
- M priebežne počíta pravdep. toho, že A akceptuje x
- M akceptuje x, ak A akceptuje x s pravdep. aspoň $1-\varepsilon$

$\Rightarrow L \in PSPACE$

Alg. pre výpočet $\int_a^b f(x)\,dx$

kde: $\forall x \in \langle a,b\rangle: 0 \le f(x) \le c$

```
k ← 0
for i ← 1 to n do
    x ← grand(a,b)
    y ← grand(0,c)
    if y ≤ f(x) then k ← k+1
integrál ← kc(b-a)/n
```

Pre $0 < \varepsilon, \delta < 1$ platí:

$|integrál - \int_a^b f(x)\,dx| < \varepsilon$ s pravd. $\ge 1-\delta$ pre $n \ge \lceil 1/(4\varepsilon^2\delta)\rceil$

Př.: $AB \stackrel{?}{=} C$

Pravdepodobnostne v čase n^2:

z náhodný lin. vektor dĺžky n

$(zA)B \stackrel{?}{=} zC$

↳ ∀ násobenia aj porovnanie v čase n^2

TU KUS CHÝBA (INTERVIEW)

19.11.2014

Pokračovanie dôkazu:

b) Každý $L \in$ PSPACE je pol. transformovateľný na TQBF, lebo:

Nech $L \in$ PSPACE, $L \subseteq \Sigma^*$ ⇒ pre L ∃DTS M akceptujúci L v pol. pamäti $S(n)$ (M má len vstupnú pásku - doprava nekonečnú, prepisovaciu).

Čas výpočtu M na vstupe w ($|w|=n$) je ≤ # rôznych konf. v pamäti $S(n)$, čo je $\le 2^{rS(n)}$ pre vhodné $r \ge 1$.

Cieľ: pre $l = 1, 2, 4, \ldots, 2^{rS(n)}$ zostrojiť bool. formulu $\phi^l(\bar{x},\bar{y})$ t.ž. $\phi^l(\bar{C}_1,\bar{C}_2)$ je úplne kvantifikovaná a je pravdivá iff M prejde z konf. C_1 do konf. C_2 počas najviac l krokov.

$\bar{C}_1$ (resp. $\bar{C}_2$) je bool. kód konfigurácie C_1 (C_2). C_1, C_2 sú kódované podobne, ako reťazce D_i (tj. pomocou $\gamma_{i,s}$ - pozri dôkaz Cook-Levinovej vety)

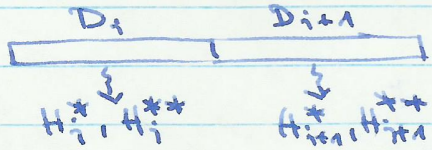
$\bar{x} = (x_1, \ldots, x_m)$ - vektor bool. premenných pre kódovanie C_1

$\bar{y} = (y_1, \ldots, y_m)$ - ——— " ——— C_2

Konštrukcia $\phi^1(\bar{x}, \bar{y})$: Použijeme formuly H_i^*, H_i^{**}, $(i=0)$, F_j z dôkazu Cook-Levinovej vety (treba v nich adekvátne nahradiť premenné y_j s premennými $x_1, \ldots, x_m, y_1, \ldots, y_m$)

$$\phi^1(\bar{x}, \bar{y}) = \Big[\underbrace{(H_i^* \wedge H_i^{**})}_{\text{kontroluje či } M \text{ prejde v kroku z } C_1 \text{ do } C_2} \vee \underbrace{(\bar{x} \Leftrightarrow \bar{y})}_{\text{kontroluje, či } C_1 = C_2}\Big] \wedge \underbrace{\bigwedge_{i(S(n)+2)+1 \le j \le (i+2)(S(n)+2)} F_j}_{\text{kontroluje správnosť kódovania}}$$

(i-čka sú rovné napr. 0 (je to jedno), lebo v dôkaze C.-L.:



teraz: konfigurácie nie "za sebou", ale "pod sebou" - to isté miesto v rôznom čase)

Konštrukcia $\phi^{2^l}(\bar{x}, \bar{y})$ pomocou $\phi^l(\bar{u}, \bar{v})$:

Platí: M prejde z C_1 do C_2 počas $\le 2^l$ krokov iff M prejde z C_1 do C_3 počas $\le l$ krokov a z C_3 do C_2 počas $\le l$ krokov.

Nepoužiteľné riešenie: $\phi^{2^l}(\bar{x}, \bar{y}) = \exists \bar{z}(\phi^l(\bar{x}, \bar{z}) \wedge \phi^l(\bar{z}, \bar{y}))$

↳ toto ale rastie exponenciálne, výsledná formula je exponenciálne dlhá (my ale potrebujeme pol. priestor, lebo ju chceme zostrojiť v pol. čase)

Použiteľné riešenie:

$$\phi^{2^l}(\bar{x}, \bar{y}) = \exists \bar{z} \forall \bar{u} \forall \bar{v} (((\bar{u} \Leftrightarrow \bar{x} \wedge \bar{v} \Leftrightarrow \bar{z}) \vee (\bar{u} \Leftrightarrow \bar{z} \wedge \bar{v} \Leftrightarrow \bar{y})) \Rightarrow \phi^l(\bar{u}, \bar{v})), \quad l = 1, 2, \ldots, 2^{r \cdot S(n)}$$

↳ vlastne to robí to isté, čo tá "nepoužiteľná" formula, lebo kvantifikátory "$\forall$" vynútia $\phi^l(\bar{x}, \bar{z})$ aj $\phi^l(\bar{z}, \bar{y})$

Dĺžka tej formuly: $O(S(n))$

Približne to $r \cdot S(n)$-krát ku dĺžke $\phi^1(\bar{x}, \bar{y}) = O(S(n))$

$\Rightarrow$ dĺžka celej formuly je $O(S^2(n))$

Splnený cieľ $\Rightarrow L$ je pol. tr. na TQBF, lebo vstup $w \in \Sigma^*$ možno det. a v pol.

čase transformovať na formulu $\phi^{2^{r \cdot S(n)}}(\bar{c}_0, \bar{c}_a)$, kde $\bar{c}_0$ je bool. kód počiatočnej konfigurácie a $\bar{c}_a$ je bool. kód akceptačnej konf.

Zrejme $w \in L$ iff $\phi^{2^{r \cdot S(n)}}(\bar{c}_0, \bar{c}_a)$ je pravdivá iff $\langle \phi^{2^{r \cdot S(n)}}(\bar{c}_0, \bar{c}_a) \rangle \in TQBF$

$\Rightarrow$ TQBF je PSPACE-úplný. □

P1.: HRA BF

Daná je bool. formula $\varphi(x_1, \ldots, x_n)$ bez kvantifikátorov.

Hru hrajú striedavo 2 hráči, H_1, H_2; začína H_1.

Hráči postupne a striedavo určujú hodnoty $b_1, \ldots, b_n$ premenným $x_1, \ldots, x_n$.

Hru vyhrá $H_1 \iff \varphi(b_1, \ldots, b_n) = 1$

P1.1: $\varphi(x_1, x_2, x_3) = (x_1 \vee x_2) \wedge (x_2 \vee x_3) \wedge (\bar{x}_2 \vee \bar{x}_3)$

Priebeh hry: H_1: $x_1 = 1$ $(= b_1)$

H_2: $x_2 = 0$ $(= b_2)$

H_1: $x_3 = 1$ $(= b_3)$

$\Rightarrow \varphi(1, 0, 1) = 1 \Rightarrow H_1$ vyhral

Def: H_1 má víťaznú stratégiu pre $\varphi(x_1, \ldots, x_n)$, ak H_1 môže vyhrať pre každú postupnosť krokov hráča H_2, t.j. ak formula $\exists x_1 \forall x_2 \exists x_3 \forall x_4 \ldots \varphi(x_1, \ldots, x_n)$ je pravdivá.

P1.2: H_1 má víťaznú stratégiu pre φ z P1.1:

H_1: $x_1 = 1$

H_2: $x_2 = b$

H_1: $x_3 = \bar{b}_2$

Def: Jazyk BF $= \{ \langle \overbrace{\varphi(x_1, \ldots, x_n)}^{\text{kód formuly } \varphi} \rangle \mid \varphi(x_1, \ldots, x_n)$ je bool. f. pre ktorú má H_1 víť. str. $\}$

Veta: Jazyk BF je PSPACE-úplný. (podobne ako TQBF)

Dôkaz: Stačí dokázať, že $BF \in PSPACE$, a že PSPACE-úplný TQBF je pol. transf. na BF.

Nech ϕ je úplne kvantifikovaná b. f. Ak ϕ je tvaru $\exists x_1 \forall x_2 \exists x_3 \forall x_4 \dots \varphi(x_1, \dots, x_n)$, potom zrejme možno $\langle\phi\rangle$ transformovať v pol. čase na $\langle\varphi\rangle$ a zrejme platí $\langle\phi\rangle \in TQBF \iff H_1$ má víť. str. pre $\varphi \iff \langle\varphi\rangle \in BF$.

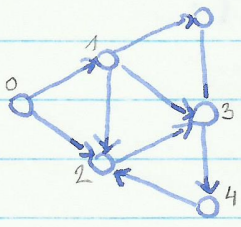
Ak ϕ je iného tvaru, napr. $\phi = \forall x_1 \exists x_2 \exists x_3 \forall x_4 \forall x_5 \varphi(x_1, \dots, x_5)$, potom $\langle\phi\rangle$ možno det. a v pol. čase transformovať na

$$\langle\varphi'(y_1, x_1, x_2, y_2, x_3, x_4, y_3, x_5)\rangle =$$
$$= \langle\varphi(x_1, \dots, x_5) \wedge \underbrace{(y_1 \vee \bar{y}_1) \wedge (y_2 \vee \bar{y}_2) \wedge (y_3 \vee \bar{y}_3)}_{\text{vždy pravdivé}}\rangle,$$

a zrejme platí: $\langle\phi\rangle \in TQBF$ iff ϕ je pravdivá iff $\exists y_1 \forall x_1 \exists x_2 \forall y_2 \exists x_3 \forall x_4 \exists y_3 \forall x_5 \varphi'(y_1, x_1, x_2, y_2, x_3, x_4, y_3 x_5)$ je pravdivá iff H_1 má víť. stratégiu pre φ' iff $\langle\varphi'\rangle \in BF$. □

Pr.: HRA GG

Daný je orientovaný graf G a jeho vrchol b.
Hru hrajú 2 hráči H_1, H_2; začína H_1 vo vrchole b.
Hráči postupne a striedavo navštevujú ešte nenavštívené susedné vrcholy.
Prehrá hráč, ktorý už nemôže navštíviť žiaden ešte nenavštívený vrchol.



H_1: 0 → 1
H_2: 1 → 2
H_1: 2 → 3
H_2: 3 → 4
H_1 prehral, H_2 vyhral

Def: Jazyk $GG = \{\langle G, b\rangle \mid G$ je orientovaný graf, b jeho vrchol, H_1 má víťaznú stratégiu pre $(G, b)\}$

Veta: Jazyk GG je PSPACE-úplný.

26.11.2014

VÝPOČTY S ORÁKULOM

Def: TS s orákulom $A \subseteq \Sigma^*$ je "bežný" TS, ktorý má naviac orákulovú pásku, na ktorú môže v priebehu rozpoznávania vstupu zapísať ľub. slovo $w \in \Sigma^*$ a potom dostane (v 1 kroku) informáciu, či $w \in A$. Na orák. pásku môže postupne zapísať viacero slov.

Def: $P^A = \{L \mid L$ akceptuje DTS s orákulom A v pol. čase$\}$
$NP^A = \{L \mid$ — " — NTS — " — $\}$

Veta: (a) $P^{TQBF} = NP^{TQBF}$
(b) $\exists$ orákulum B: $P^B \neq NP^B$

(Idea dôkazu b: B je veľmi riedky jazyk (z každej dĺžky doň patrí len 1 slovo a pod.) - NTS to slovo uhádne, DTS ho nestihne v pol. čase nájsť.)

Dôkaz: (a) Platí $NP^{TQBF} \underset{(i)}{\subseteq} NPSPACE \underset{(ii)}{\subseteq} PSPACE \underset{(iii)}{\subseteq} P^{TQBF}$

(i) Výpočet bez orákula prebieha rovnako ako s orákulom, len namiesto orákula odpovedá na otázky, či $w \in TQBF$, podprogram (DTS) rozpoznávajúci TQBF s pol. pamäťou $S(n)$. Dĺžka otázok na vstupe x je $\leq p(|x|)$, kde $p(n)$ je pol. čas (aj pamäť) pre výpočet s orákulom.
$\Rightarrow$ Použitá pamäť na vstupe x (výpočet bez orákula) je $\leq p(|x|) + S(p(|x|)) =$ polynóm

(ii) Zo Savitchovej vety

(iii) Ľub. $L \in PSPACE$ je pol. transf. na PSPACE-úplný TQBF. L možno rozpoznať strojom M, ktorý realizuje túto transformáciu a na vstupe x zapíše na orák. pásku slovo $M(x)$, tj. výsledok pol. transf. pre x. Platí $x \in L$ iff $M(x) \in TQBF$. □

REKURZÍVNE FUNKCIE

Uvažujme len fcie $f: \mathbb{N}^n \to \mathbb{N}$ $(\mathbb{N} = \{0,1,2,\dots\})$
(f nemusí byť totálna)

Označ: (a) Pre fcie I^n_m $(1 \le m \le n)$ platí:
$$I^n_m(x_1,\dots,x_n) = x_m \quad \forall x_1,\dots,x_n$$
(b) Pre fciu S platí:
$$S(x) = x+1 \quad \forall x \qquad \text{(nasledovník)}$$
(c) 0 označujeme nulárnu fciu s konšt. hodnotou 0.

Def: Nech f je n-árna fcia, $g, f_1, \dots, f_n$ sú m-árne fcie.
Fcia g vznikne <u>skladaním</u> z $f, f_1, \dots, f_n$, ak:
$$\forall x_1,\dots,x_m: \; g(x_1,\dots,x_m) = f(f_1(x_1,\dots,x_m),\dots,f_n(x_1,\dots,x_m))$$

Def: Nech h je n-árna fcia, g je $(n+2)$-árna fcia a f je $(n+1)$-árna fcia. Fcia f vznikne z g a h operáciou <u>primitívnej rekurzie</u>, ak $\forall x_1,\dots,x_n, y$ platí:
$$f(0, x_1,\dots,x_n) = h(x_1,\dots,x_n)$$
$$f(y+1, x_1,\dots,x_n) = g(y, f(y,x_1,\dots,x_n), x_1,\dots,x_n)$$

Pr. 1: $f(y,x) = y+x$ vznikne operáciou prim. rek. z fcií
$g(y,z,x) = z+1$ a $h(x) = x$
Keď $y=0$: $f(0,x) = x = h(x)$ ✓
$$f(y+1,x) = x+y+1 = f(y,x)+1 = g(y, f(y,x), x)$$

Def: Fcia f je <u>primitívne rekurzívna</u>, ak vznikne z fcií $0, S$ a I^n_m konečným počtom operácií skladania a prim. rekurzie.

Pr. 2: Fcie h a g z Pr. 1 sú prim. rekurzívne, lebo
- $h(x) = x = I^1_1(x)$
- $g(y,z,x) = z+1 = S(I^3_2(y,z,x))$

$\Rightarrow$ f z Pr. 1 je tiež prim. rek.

Lema 1: Každá konštantná fcia $K_j(x) = j$ $\forall x$ je PR. ($j \in \mathbb{N}$)

D.: Zrejmé. □

Veta 2: Fcie F_1 až F_7 sú PR:

$F_1(x,y) = x+y$

$F_2(x,y) = x \cdot y$

$F_3(x,y) = x^y$

$F_4(x) = sg(x) = \begin{cases} 0, & \text{ak } x = 0 \\ 1, & \text{inak} \end{cases}$ (vie -1, lebo počítame v $\mathbb{N}$)

$F_5(x) = \overline{sg}(x) = 1 - sg(x)$

$F_6(x,y) = x \dot{-} y = \begin{cases} 0, & \text{ak } x < y \\ x-y, & \text{ak } x \geq y \end{cases}$

$F_7(x,y) = |x-y| = (x \dot{-} y) + (y \dot{-} x)$

D.: F_1 - v Pr. 2.

F_2: $g(x,z,y) = F_1(I_2^3(x,z,y), I_3^3(x,z,y)) = z+y$ je PR.

$F_2(0,y) = 0 = K_0(y)$ ✓ (nestačí tam dať len 0, lebo fcia na pravej strane musí byť unárna)

$F_2(x+1,y) = (x+1)\,y = xy + y = F_2(x,y) + y = g(x, F_2(x,y), y)$ ✓

$\Rightarrow F_2$ je PR.

F_4: $g(x,y) = K_1(I_1^2(x,y)) = 1$ (konštanta na 2 argumentoch)

↳ je PR.

$sg(0) = 0$ ✓ (tiež stačí 0 - je nulárna)

$sg(x+1) = 1 = g(x, sg(x))$ ✓

$\Rightarrow sg$ je PR.

Lema 3: Nech $h(y, x_1, \dots, x_n)$ je PR. Potom aj:

$$f_1(y, z, x_1, \dots, x_n) = \sum_{i=z}^{y+z} h(i, x_1, \dots, x_n)$$

$$f_2(y, z, x_1, \dots, x_n) = \prod_{i=z}^{y+z} h(i, x_1, \dots, x_n)$$

sú PR.

Def: Čiastočná fcia f vznikne z čiastočnej fcie g minimalizáciou, píšeme $f(x_1, \dots, x_n) = \mu y (g(y, x_1, \dots, x_n) = 0)$ ak $\forall x_1, \dots, x_n$ platí: $f(x_1, \dots, x_n) = y$ iff $\forall z < y$ je $g(z, x_1, \dots, x_n)$ definovaná a kladná a $g(y, x_1, \dots, x_n) = 0$.

Def: Ak naviac f aj g sú totálne, hovoríme, že f vznikne z g regulárnou minimalizáciou.

Pozn: Operácia minimalizácie nezachováva ani totálnosť, ani PR.

P1.: $f(0) = 0$, $f(x)$ nie je def. pre $x > 0$

$f(x) = \mu_y(\underbrace{I_2^2(y,x)}_{\text{toto} = 0 \text{ len pre } x=0} = 0)$

Def: Fcia f je rekurzívna, ak f vznikne z fcií $0, s, I_m^n$ konečným počtom operácií skladania a prim. rekurzie a reg. minimalizácie. (Rekurzívne fcie sú totálne.)

Pozn.: Každá PR fcia je R.

Def: Čiastočná fcia f je čiastočne rekurzívna, ak f vznikne z $0, s, I_m^n$ kon. počtom operácií skladania, prim. rek. a minimalizácie.

Pozn: Každá R fcia je ČR.

Veta 4: Nech $g(y, x_1, \dots, x_n)$, $h(x_1, \dots, x_n)$ sú PR, a nech $\forall x_1, \dots, x_n \; \exists z \le h(x_1, \dots, x_n): g(z, x_1, \dots, x_n) = 0$. Potom fcia $f(x_1, \dots, x_n) = \mu_y(g(y, x_1, \dots, x_n) = 0)$ je PR.

D.: (Idea: prechádzame všetky $z \le h(x_1, \dots, x_n)$, lebo vieme, že aspoň 1 z nich musí "fungovať")

Veta 5: Fcie F_8 až F_{15} sú PR:

$F_8(x,y) = \lfloor x/y \rfloor$ $(\lfloor x/y \rfloor = 0$ pre $y=0)$ $(\text{lebo } \lfloor x/y \rfloor = sg(y) \sum_{i=1}^{x} \bar{sg}(iy \dot{-} x))$

$F_9(x,y) = x \bmod y$ $(x \bmod y = x$ pre $y=0)$ $(\text{lebo } x \bmod y = x \dot{-} y \cdot \lfloor x/y \rfloor)$

$F_{10}(x,y) = div(x,y) = \begin{cases} 1, & y \mid x \\ 0, & y \nmid x \vee y = 0 \end{cases}$ $(\text{lebo } div(x,y) = \bar{sg}(x \bmod y))$

$F_{11}(x) = nd(x) = \#$ deliteľov $x \neq 0$, inak 0 $(\text{lebo } nd(x) = \sum_{i=1}^{x} div(x,i))$

$F_{12}(x) = chp(x) = \begin{cases} 0, & x \in \mathbb{P} \\ 1, & x \notin \mathbb{P} \end{cases}$ $(\text{lebo } chp(x) = sg(|nd(x) - 2|)$

$F_{13}(x) = \pi(x) = \#y \in \mathbb{P}, y \le x$ $(\text{lebo } \pi(x) = \sum_{i=0}^{x} \bar{sg}(chp(i)))$

$F_{14}(x) = p(x) = x$-té prvočíslo $(p(0)=2, p(1)=3, \dots)$ $(\text{lebo } p(x) = \mu_y(|\pi(y) \dot{-} (x+1)| = 0))$

↳ $p(x) \le 2^{2^x} \leftarrow$ PR

$F_{15}(x,y) = ex(x,y) =$ exp. prvoč. $p(x)$ v rozklade y

$(\text{lebo } ex(x,y) = \mu_u(sg(y) \cdot div(y, (p(x))^{u+1}) = 0)$

03.12.2014

REGISTROVÉ STROJE

Registrový stroj má:

- konečne veľa stavov q_0 (koncový), q_1 (počiatočný), $q_2, \dots, q_k$
- konečne veľa registrov $R_0, R_1, \dots, R_m$ (obsah $R_i \in \mathbb{N} = \{0, 1, \dots\}$)
- konečne veľa inštrukcií typu

  (q_i, R_j, q_e, q_m) - q_i: if $R_j = 0$ goto q_e else goto q_m

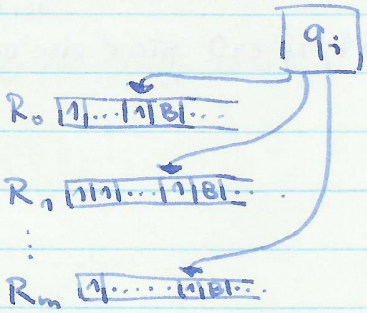
  (q_i, R_j^+1, q_m) - q_i: $R_j \leftarrow R_j + 1$ & goto q_m

  (q_i, R_j^-1, q_m) - q_i: $R_j \leftarrow R_j - 1$ & goto q_m

Def: Registrový stroj M **počíta fciu** $f(x_1, \dots, x_n)$ ak:

- na začiatku výpočtu: $R_i = x_i$, $i = 1, \dots, n$ a M je v stave q_1
- na konci výpočtu: $R_0 = f(x_1, \dots, x_n)$, a M je v stave q_0

Teda: RS je špeciálny TS majúci $m+1$ prac. pások (ale bez vstup. pásky), na každej páske je vždy nejaké slovo 1^i (unárne kódované $i \in \mathbb{N}$).



Def: TS (o 1 vstupnom a viacerými prac. páskami) **počíta fciu** $f(x_1, \dots, x_n)$, ak má na začiatku na vstupe slovo $01^{x_1}01^{x_2}0 \dots 01^{x_n}0$ a na konci výpočtu na 1. prac. páske slovo $1^{f(x_1, \dots, x_n)}$.

Lema A: Každý RS možno simulovať nejakým TS.

D.: Zrejmé. (Podľa obrázku, jediný rozdiel je vo vstupe/výstupe.)

Lema B: Každý TS možno simulovať nejakým RS.

D.: M_1 - ľubov. TS (pre jednoduchosť bez prac. pásky a s páskovou abecedou $\{0, 1, B\}$). Takýto stroj má na štart. výpočtu fcie $f(x_1, \dots, x_n)$ na vstup. páske slovo $01^{x_1}01^{x_2}0 \dots 01^{x_n}0$ a na konci výpočtu slovo $01^{f(x_1,\dots,x_n)}0$.

Nech $C_0, C_1, \dots, C_t$ je výpočet (post. konfigurácií) stroja M_1 na vstupe w (napr. nech $C_i = \dots BB\overbrace{0100}^{u_i}\overbrace{1101}^{v_i}B\dots$, q_i pod 1. 1 v v_i)

Cieľ: simulovať M_1 vhodným RS M_2 tak, aby $\forall i$ platilo:

(a) Ak je M_1 v konfigurácii C_i, potom M_2 je v takej konf., že jeho register R_{n+1} obsahuje číslo s bin. zápisom $u_i' = 1u_i$ (napr. $u_i = 0100 \rightsquigarrow u_i' = 10100$) a R_{n+2} obsahuje č. s bin. zápisom $v_i' = 1v_i^R$

Aby platilo (a), stačí, aby M_2 simuloval prechod z C_i do C_{i+1} (napr. nech $C_{i+1} = \dots B\overbrace{01011}^{u_{i+1}}\overbrace{101}^{v_{i+1}}B\dots$, q_i pod 1. 1 v u_{i+1}, tj. $u_{i+1}' = 10101$, $v_{i+1}' = 1101$)

tak, aby z čísel u_i', v_i' M_2 vypočítal vhodné čísla u_{i+1}', v_{i+1}', použijúc len príkazy: $z \leftarrow z+1$; $z \leftarrow z-1$; if $z=0$ goto ℓ else goto h.

Pr.: pre spomenuté C_i, C_{i+1}:

$u_i' = 10100$, $v_i' = 11011$

$\Rightarrow u_{i+1}' = 2(u_i'+1)+1 = 2u_i'+3$

$v_{i+1}' = (v_i'-1)/2$

- Ako zistiť najmenej významný bit pre u_i', v_i'?
  - Podľa parity. (Odpočítavať v cykle po 2 jednotky, zistiť, či to kleslo na 0 na konci alebo v strede tela cyklu; Pri odpoč. si zničíme obsah registra - treba si ho niekde pri odpočítavaní aj pripočítavať (do nejakého "pomocného, záložného" registra).)
- Ako zničiť obsah registra na polovicu?
  - V cykle odpočítavať z R 2 jednotky a do pomoc. registra R' pripočítavať 1 jednotku. Keď bude $R=0$, tak $R' = \lfloor R/2 \rfloor$. Následne presunúť obsah R' do R (tiež cyklom).

- Ako zvýšiť obsah R na 2-násobok?
- Podobne ako znížiť na 1/2.

...už len nejak doriešiť okrajové prípady (ako U_i končiace blankom...) □

EKVIVALENCIA R FCIÍ A TS

Lema:

(a) Pre každú fciu O, S, I^n_m existuje TS, ktorý ju počíta.

(b) Ak pre každú fciu $f, f_1, \dots, f_m$ ex. TS, ktorý ju počíta, a fcia g vznikne z $f, f_1, \dots, f_m$ skladaním, potom aj pre g ∃ TS, ktorý ju počíta.

(c) Ak pre fcie g a h ∃TS, čo ich počíta a fcia f vznikne z g a h prim. rekurziou, potom aj pre f ∃TS, čo ju počíta.

(d) Ak pre fciu g ∃TS, čo ju počíta, a fcia f vznikne z g minimalizáciou, potom aj pre f ∃TS, čo ju počíta.

D.: (a) - (d): zrejmé.

Lema C: Pre každú čiast. rek. fciu ∃TS, ktorý ju počíta.

D.: Vyplýva z predošlej lemy.

Lema D: Ak pre fciu $f: \mathbb{N}^n \to \mathbb{N}$ ∃TS, ktorý ju počíta, potom f je ČR.

D.: Číslovanie inštrukcií TS:

č. inštrukcie $\delta(q_i, a_j) = (q_m, a_k, L)$ je $p_{c(i,j)}^{3c(k,m)+1}$

kde p_t je t-te prvočíslo: $p_0 = 2, p_1 = 3, \dots$

a $c(i,j)$ je poradové číslo dvojice (i,j) v usp. postupnosti $(0,0), (0,1), (1,0), (0,2), (1,1), (2,0), (0,3), \dots$

L = -1 ... 1
N = 0 ... 0
R = 1 ... 2

Číslovanie TS: č. TS je súčin čísel jeho inštrukcií

Číslovanie konf. TS:

Príklad: $\Sigma = \{\overset{B}{a_0}, a_1, a_2, a_3\}$, $Q = \{q_0, q_1, q_2, q_3, q_4\}$

konf. $a_2 a_1 \underset{0}{q_3} a_2 a_3$ má č. $2^{2\cdot 2} \cdot 3^{2\cdot 0} \cdot 5^{2\cdot 3} \cdot 7^{2\cdot 1} \cdot 11^{2\cdot 2} \cdot 13^{2\cdot 3}$

↳ párne exponenty: páskové symboly

↳ nepárne: stav a pozícia hlavy

Všeobecne: konf. $a_{l_0} a_{l_1} \dots a_{l_{i-1}} q_j a_{l_{i+1}} \dots a_{l_n}$ má číslo

$2^{2 \cdot l_0} \cdot 3^{2 \cdot l_1} \cdot \dots \cdot p_{i-1}^{2 l_{i-1}} \cdot p_i^{2 \cdot j+1} \cdot p_{i+1}^{2 l_{i+1}} \cdot \dots \cdot p_n^{2 l_n}$

Jeden krok výpočtu TS kontrolujú fcie $prech_L$, $prech_N$, $prech_R$, kde:

$$prech_L(x,y,z) = \begin{cases} 0, & \text{ak platí (a) - v tomto prípade prejde TS č. } z \text{ v 1 kroku (o pred. hlavy č.) z konf. č. } x \text{ do konf. č. } y \\ 1, & \text{inak} \end{cases}$$

kde (a) $((x$ je č. nejakej konf.$) \wedge (y$ je č. nejakej konf.$) \wedge (z$ je č. nejakého TS$)$

$\wedge (\exists m_1, m_2, v, i, j, k, m, n \leq x+y : (x = m_1 p_i^{2v} p_{i+1}^{2j+1} p_{i+2}^{2k} \cdot m_2 \vee (\dots))$

$\wedge (y = m_1 p_i^{2m+1} p_{i+1}^{2v} p_{i+2}^{2n} \cdot m_2) \wedge p_i \nmid m_1 m_2 \wedge p_{i+1} \nmid m_1 m_2 \wedge p_{i+2} \nmid m_1 m_2$

$\wedge$ (TS č. z obsahuje inštr. $\delta(q_j, a_k) = (q_m, a_n, L))))$

(th len potrebujeme dosť veľké čísla (nehľadáme presne $x+y$))

(* = podmienka, keď je hlava celkom vľavo)

Pr.: konf. $C = \dots \underbrace{a_{l_0} \dots a_{l_{i-1}}}_{\text{toto doľava sa nemení, má č. } m_1} a_v \overset{q_j}{\uparrow} a_k \underbrace{a_{l_{i+3}} \dots a_{l_t}}_{\text{toto doprava sa nemení, má č. } m_2} \dots$

$\Rightarrow$ konf. $C' = \dots \underbrace{a_{l_0} \dots a_{l_{i-1}}}_{m_1} \overset{q_m}{\uparrow} a_v a_n \underbrace{a_{l_{i+3}} \dots a_{l_t}}_{m_2}$

C má číslo $x = \underbrace{2^{2l_0} \dots p_{i-1}^{2l_{i-1}}}_{=m_1} p_i^{2v} p_{i+1}^{2j+1} p_{i+2}^{2k} \underbrace{p_{i+3}^{2l_{i+3}} \dots p_t^{2l_t}}_{=m_2}$

C' má číslo $y = m_1 \cdot p_i^{2m+1} p_{i+1}^{2v} p_{i+2}^{2n} \cdot m_2$

Konštrukcia E_x: E_x obsahuje boolovské premenné $y_{j,s}$, kde $1 \leq j \leq (p(n)+1)$ a $s \in \Sigma' \cup Q$. ($y_{j,s} = 1 \Leftrightarrow$ j-ty symbol reťazca D je symbol s) | $(p(n)$ - dĺžka D

Pr. $D = acbc$

$y_{1,a} = 1$ $\quad y_{2,a} = 0$ $\quad y_{3,a} = 0$ $\quad y_{4,a} = 0$

$y_{1,b} = 0$ $\quad y_{2,b} = 0$ $\quad y_{3,b} = 1$ $\quad y_{4,b} = 0$

$y_{1,c} = 0$ $\quad y_{2,c} = 1$ $\quad y_{3,c} = 0$ $\quad y_{4,c} = 1$

Zostrojme F, G, H, I tak, aby:

(a) F bol splniteľný iff pre každé j, $(1 \leq j \leq (p(n)+1)(p(n)+2))$ práve 1 premenná $y_{j,s}$ $(s \in \Sigma' \cup Q)$ má hodnotu 1.

(b) výraz G resp. H resp. I kontroloval či D, určený hodnotami $y_{j,s}$ splňujúcimi F má vlastnosť (1), resp (2), resp. (3).

F: $F = F_1 \wedge F_2 \wedge F_3 \wedge F_4 \wedge \dots \wedge F_h$ $\quad h = (p(n)+1)(p(n)+2)$

$$F_j = \Big(\bigvee_{s \in \Sigma \cup Q} y_{j,s}\Big) \wedge \neg\Big(\bigvee_{\substack{s \neq r \\ s,r \in \Sigma \cup Q}} (y_{j,s} \wedge y_{j,r})\Big) \quad j = 1, 2, \dots, h$$

↑ aspoň 1 jednotka $\qquad$ ↑ najviac 1 jednotka

G: Nech $x = x_1 x_2 \dots x_n$, $x_i \in \Sigma$ $\forall i$, $D_0 = \underbrace{¢\, q_0\, x_1\, x_2 \dots x_n \overbrace{BB \dots B}^{p(n)-n}}_{p(n)}$

$$G = y_{1,¢} \wedge y_{2,q_0} \wedge y_{3,x_1} \wedge y_{4,x_2} \wedge \dots \wedge y_{n+2,x_n} \wedge y_{n+3,B} \wedge \dots y_{p(n)+2,B}$$

I:

$$I = y_{t,q_f} \vee y_{t+1,q_f} \vee \dots \vee y_{t+p(n)+1,q_f}$$

$t = p(n)(p(n)+2)+1$

→ H zabezpečí aby posledná konf. bola platná

D: Analýza. ¢ [0 1 1] [0 q 1] 0 B

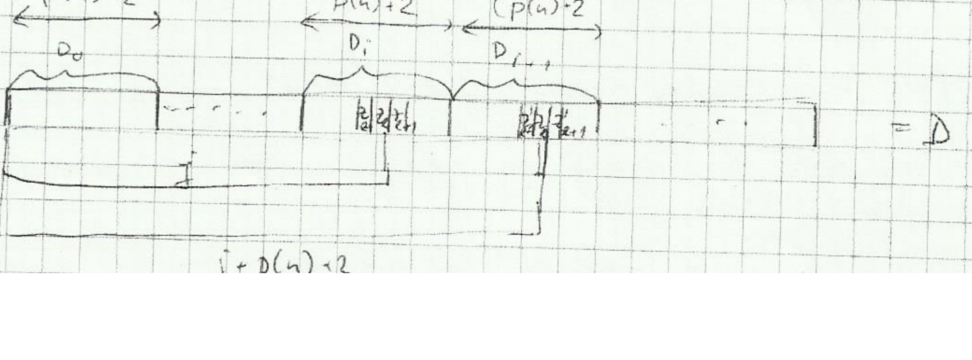
¢ 0 [0] 1 0 1 q' q" B

↳ zle → ak v okne nevidíme stav → stredné sa nesmie meniť

¢ 0 1 1 [q' 1 q"] 0 B → musí splniť δ-f-ciu

¢ 0 1 1 [0 0 q'] 0 B → iné okienka nepotrebujeme

↳ chyba



$$H_i^{*} = \bigwedge_{i(p(n)+2)+1 \le j \le (i+1)(p(n)+2)} \left(\bigwedge_{r,s,t \in \Sigma'} \left(y_{j-1,r} \wedge y_{j,s} \wedge y_{j+1,t} \Rightarrow y_{j+p(n)+2,s} \right) \right)$$

$$H_i^{**} = \bigwedge_{i(p(n)+2)+1 \le j \le (i+1)(p(n)+2)} \left(\left(\bigwedge_{r,s \in \Sigma',\ q \in Q} \left(y_{j-1,r} \wedge y_{j,q} \wedge y_{j+1,s} \Rightarrow \bigvee_{tuv \in K_{rqs}} \left(y_{j+p(n)+1,t} \wedge y_{j+p(n)+2,u} \wedge y_{j+p(n)+3,v} \right) \right) \right) \right)$$

$K_{rqs} = \{ tuv \mid t,u,v \in \Sigma' \cup Q$, reťazec tuv môže vzniknúť z reťazca rqs v 1 kroku podľa δ f-cie $\}$

$i = 0, 1, \ldots, p(n)-1$

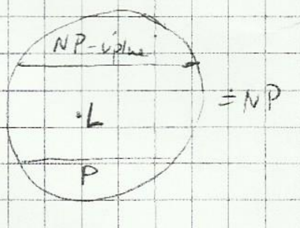
$$H = H_0^{*} \wedge H_1^{*} \wedge \ldots \wedge H_{p(n)-1}^{*} \wedge H_0^{**} \wedge H_1^{**} \wedge \ldots \wedge H_{p(n)-1}^{**}$$

(pozn. pozor na indexy → môžu sa vyskytnúť problémy s tým, že v okienku čítame v strede q a pred ním čítame ešte 1 znak z predchádzajúcej konf. ale to nevadí, ak si zoberieme ešte o 1 dlhšiu pásku alebo iné riešenie)

~~Všetkých premenn~~ Každá premenná je v E_x konštantne veľa ~~krát~~ vzhľadom na veľkosť abecedy.

E_x je len polynomiálne dlhé od dĺžky x.
a $E_x \in SAT \iff x \in L$ □

Veta: Ak $P \neq NP$, potom $\exists L \in NP$ s.t. $L \notin P$ a L nie je NP-úplný.



(# slová dĺžky n)

Def. Jazyk $L \subseteq \Sigma^*$ je riedky, ak existuje polynóm p s.t. $|L \cap \Sigma^n| \le p(n)$ $\forall n$

Veta. [Mahaney]
Žiaden riedky jazyk nie je NP-úplný, ak $P \neq NP$

15. 10. 2012

Veta: Nech $P \neq NP$. Potom žiadny unárny jazyk $L \subseteq a^*$ nie je NP-úplný.

Dôkaz: φ - ľub. bool. výraz v konjunktívnom normálnom tvare s premennými $x_1 \ldots x_m$
$t = t_1 t_2 \ldots t_r$, $r \le m$, $t_i \in \{0,1\}$ $\forall i$

$\varphi(t)$ - vznikne z φ po nahradení $t_i \to x_i$ $\forall i$

Nech by $L \in a^*$ bol NP-úplný $\Rightarrow$ CSAT (konjunktívny variant SAT) je polynomiálne transformovateľný na L

$R(\emptyset)$ – výsledok polynomiálnej transformácie CSAT na L

$R(\emptyset) = a^j \quad j \in N \quad [R(\emptyset(t)) = a^l]$

Cieľ: Zostrojiť det. polynomiálny alg pre CSAT $\Rightarrow$ P = NP (bude to spor s predp.)

procedure TEST $(\emptyset, t)$, { TEST vráti "áno", ak $\emptyset(t) \in$ CSAT, inak vráti "nie"
MEM[ ] – pamäť pre medzivýsledky ; MEM[.] $\in$ {áno, nie, neviem}

(1) MEM[$R(\emptyset(t))$] $= v \in$ {áno, nie}, potom return v

Nech MEM[$R(\emptyset(t))$] = neviem
(2) Ak $\emptyset(t)$ nemá premenné, potom return v = áno [nie] ak $\emptyset(t) \in$ [ak $\emptyset(t) \notin$ CSAT]

MEM[$R(\emptyset(t))$] $\leftarrow v$

(3) Ak $\emptyset(t)$ má premenné, potom return v = (TEST$(\emptyset, t0)$ $\vee$ TEST$(\emptyset, t1)$)

MEM[$R(\emptyset(t))$] $\leftarrow v$

end

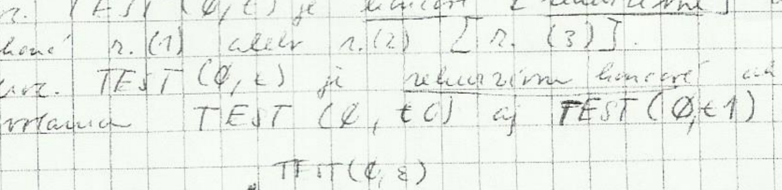
v v (3) je pravdivý pre: $\emptyset(t) \in$ CSAT iff $\emptyset(t0) \in$ CSAT $\vee$ $\emptyset(t1) \in$ CSAT

Pr. $\emptyset(x_1, x_2, x_3, x_4)$: $\emptyset(0, 1, x_3, x_4) \in$ CSAT iff $\emptyset(0,1,0,x_4) \in$ CSAT alebo $\emptyset(0,1,1,x_4) \in$ CSAT

Deterministický polynomiálny algoritmus pre CSAT:

$\forall x$: MEM[x] $\leftarrow$ neviem
TEST$(\emptyset, \varepsilon)$
end

Volanie pr. TEST$(\emptyset, t)$ je koncové [rekurzívne] ak sa v ňom vykoná r. (1) alebo r. (2) [r. (3)].
Volanie pr. TEST$(\emptyset, t)$ je rekurzívne koncové ak je rekurzívne a obe volania TEST$(\emptyset, t0)$ aj TEST$(\emptyset, t1)$ sú koncové.



Počet rek. konc. volaní:
Pri rek. koncového volania TEST$(\emptyset, t)$ sa "neviem" v MEM[$R(\emptyset(t))$] zmení na "áno"/"nie" $\Rightarrow$ ďalšie volanie TEST$(\emptyset, t')$, kde $R(\emptyset(t)) = R(\emptyset(t'))$, nemôže byť rekurzívne koncové (musí sa v ňom vykonať r. (1)).

$R(\emptyset(t)) = R(\emptyset(t'))$

①

$\Rightarrow$ Počet rekurzívnych koncových volaní je najviac počet rôznych hodnôt $R(\emptyset(t)) \le p(n)+1$, kde $n = |\emptyset|$, $t \in \{0,1\}^{\le m}$, m je počet premenných v φ

lebo $0 \le |R(\emptyset(t))| \le p(|\emptyset(t)|) \le p(|\emptyset|) = p(n)$

Počet rekurzívnych volaní $\le$ (počet rek. konc. volaní) x (výška stromu volaní) $\le$
$\le (p(n)+1)n$
lebo každé rek. volanie sa vyskytne niekde na ceste z niektorého rek. koncového volania do koreňa a výška stromu je najviac počet premenných v $\varphi \le |\emptyset| = n$.

Počet koncových volaní $\le 2n(p(n)+1)$, lebo každé rek. volanie môže priamo spôsobiť najviac 2 koncové volania.

$\Rightarrow$ počet všetkých volaní $\le$ (počet rekurz. vol.) + (počet konc. volaní) $\le$

$\le 3n(p(n)+1)$

$\Rightarrow$ Časová zložitosť algoritmu $\le 3n(p(n)+1) \cdot q(n)$, kde $q(n)$ je polynomiálny čas potrebný na 1 volanie fce. TEST (okrem času na prípadné následné rek. volanie)

□

NP-optimalizačné problémy

a) cieľ (min / max)
b) relácia $R \subseteq \Sigma^* \times \Sigma^*$, $(x,y) \in R \Rightarrow |y| \le p(|x|)$, p - polynóm.
c) f-cia $m: \Sigma^* \times \Sigma^* \to N$ (m - hodnotiaca f-cia)

$\{x \# y \mid (x,y) \in R\} \in P$
$m(x,y)$ je vypočítateľná deterministicky v polynomiálnom čase $q(|x|+|y|)$, q - polynóm

R je relácia: problém (vstup) x, riešenie (výstup) y
$m(x,y)$ - hodnota (kvalita) riešenia.

Pr. NP-opt. problém obchod. cestujúceho:
a) cieľ: min
b) $R = \{(x,y) \mid$ x je kompletný graf s ohodnotenými hranami, y je hamiltonovská kružnica grafu x $\}$
c) $m(x,y)$ = cena hamilt. kružnice y grafu x.

1) **Konštrukčný** problém: Pre X nájdi / vráť (ak ex.) y t. ž. $(x,y) \in R$ a zároveň $m(x,y) = \min_{y' \in \Sigma^*} \{ m(x,y') \mid (x,y') \in R\}$ pre min
opt. riešenie — vyhovujúce riešenie

$m(x,y) = \max_{y' \in \Sigma^*} \{ m(x,y') \mid (x,y') \in R\}$ pre max

2) **Hodnotový problém**: Pre x nájsť (ak ex.) $\min_{y' \in \Sigma^*} \{ m(x,y') \mid (x,y') \in R \}$ pre min, podobne pre max.

3) **Rozhodovací problém**: Pre x a $k \in \mathbb{N}$ určiť, či $\exists y\ (x,y) \in R$ &
dolná/horná ohraničenie
zvyšok
$m(x,y) \le k$ pre min.
$L = \{ x\#d(k) \mid x \in \Sigma^*, k \in \mathbb{N}, \exists y : (x,y) \in R \ \&\ m(x,y) \le k \}$ pre min
$m(x,y) \ge k$ pre max.
$L = \{ x\#d(k) \mid x \in \Sigma^*, k \in \mathbb{N}, \exists y : (x,y) \in R \ \&\ m(x,y) \ge k \}$ pre max

Lema: Rozhodovací problém každého NP-opt. problému patrí do NP.

Dôkaz: Stačí nedeterministicky uhádnuť y a overiť vlastnosti. □

Veta: Nech A je ľub. NP optimalizačný problém s cieľom min, relácion R a hodnotovou f-ciou m, ktorého rozhod. problém $L = \{ x\#d(k) \mid \ \}$ je NP-úplný. Nech možno L akceptovať det. v čase $T(n)$. Potom konštrukčný problém pre A možno deterministicky v čase $r(n)\, T(s(n))$ pre vhodné polynómy r, s.

Dôkaz. 1) Pre x nájdeme (ak ex.) $k_x = \min_y \{ m(x,y) \mid (x,y) \in R \}$

Nech f-ciu m možno vypočítať v polynom. čase q.
Po $q(|x|+|y|)$ krokoch výpočtu má hodnota $m(x,y)$ zapísaná v binárnom tvare $\le q(|x|+|y|)$ číslic, t.j.

$$m(x,y) \le 2^{q(|x|+|y|)} - 1 \le 2^{q'(|x|)}$$ pre vhodný polynóm q'
lebo $|y| \le p(|x|)$

Binárnym prehľadávaním intervalu $\langle 0, 2^{q'(|x|)} \rangle$ nájdeme minimálnu hodnotu $\min\{ i \mid x\#d(i) \in L \} = k_x$

2) **Pre** x a k_x zostrojíme (nejaké) opt. riešenie y_x t.z. $(x,y_x) \in R$ & $m(x,y_x) = k_x$ takto:

$L' = \{ x\#d(k)\#z \mid x \in \Sigma^*, k \in \mathbb{N}, \exists y : (x,y) \in R, m(x,y) \le k$ a z je nejaký prefix reťazca $y \}$

$\Rightarrow L' \in NP$ (stačí uhádnuť y ~~d(k) ...~~ a všetky podmienky overíme)
$\Rightarrow L'$ je polynomiálne transf. na NP-úplný L $\Rightarrow$ Každý $x\#d(k)\#z$ je polynomiálne transformovateľný na nejaké w, t.z.
$x\#d(k)\#z \in L'$ iff $w \in L$.

Pre jednoduchosť nech $\Sigma = \{0,1\}$. Hľadané y_x zostrojíme postupným predlžovaním už nájdeného prefixu z takto:

Ak $u_0 = x\#d(k_x)\#z0$ resp. $u_1 = x\#d(k_x)\#z1$ patrí do L', potom $z0$ resp. $z1$ je ďalší nájdený prefix.

Ako zistiť, či $u_i \in L'$? Transformujeme u_i na w_i a využijeme fakt:
$u_i \in L'$ iff $w_i \in L$
overíme alg. pre L.

Veta. Nech A je ľub. NP-opt-problém, ktorého rozhodovací problém L možno akceptovať det. v čase $T(n)$.
(L nemusí byť NP-úplný)
Potom konštrukčný problém pre A možno riešiť det. v čase $r(n)T(s(n))$ pre vhodné polynómy r, s.

Dôkaz. Je uvedený v časti (1) dôkazu predošlej vety. □

Veta.
(a) Nech A je ľub. NP-opt. problém a nech L je ľubovoľný NP-úplný problém, ktorý možno akceptovať deterministicky v čase $T(n)$. Potom konštrukčný problém pre A možno riešiť deterministicky v čase $r(n)T(s(n))$ pre vhodné polynómy r, s.
(b) Dôsledok: Ak $P=NP$, potom konštrukčný problém každého NP-opt. problému možno riešiť deterministicky v polynomiálnom čase.

Dôkaz pre (b):
(a) $\Rightarrow T(n)$ je polynóm, lebo $L \in NP = P$.
$\Rightarrow$ konštrukčný problém vieme riešiť v čase:
$r(n).T(s(n))$ (polynóm) $\Rightarrow$ (b) □

Pr. Pre daný graf G nájdi (ak existuje) najlacnejšiu hamiltonovskú kružnicu.
(Rozhod., hodnotový, konštr. problém?)
Ak si domyslíme ohodnotenie hrán a každé všetci budú mať rovnaké hodnoty $\Rightarrow$ konštrukčný problém
$R = \{(x,y) \mid$ graf x má hamiltonovskú kružnicu $y\}$
$m(x,y) = 1 \quad \forall x,y$, cieľ je min / max

Aproximovateľnosť NP-opt. problémov

Def. Nech A je NP-optimalizačný problém s cieľom min (max), relácia R a hodnotovou f-ciou m.
Nech $m^*(x) = \min \{m(x,y) \mid (x,y) \in R\}$ pre cieľ min.
$m^*(x) = \max \{m(x,y) \mid (x,y) \in R\}$ pre cieľ max.

Def. Problém A je α-aproximovateľný, $\alpha > 1$ ak existuje deterministický polynomiálny algoritmus M, ktorý pretransformuje vstup x na výstup y (t.j. $M(x)=y$), pričom $(x,y) \in R$ a $\alpha m^*(x) \geq m(x, M(x))$ (pre skoro všetky x) pre cieľ min; $m^*(x) \leq \alpha m(x, M(x))$ (pre skoro všetky x) pre cieľ max

Def. Problém A je dobre aproximovateľný, ak je α-aproximovateľný pre každé $\alpha > 1$.

Def. Problém A je neaproximovateľný, ak nie je α-aproximovateľný pre žiadne α. (ľub. veľké

Problém: NP-opt. problém 0-1 knapsack
cieľ: max
$R = \{(x,y) \mid x = d(v_1)\#d(v_2)\# \dots \#d(v_n)\#d(w_1)\#d(w_2)\# \dots \#d(w_n)\#d(W),$
$y = d(S), \subseteq \{1,2,\dots,n\}, \sum_{i \in S} w_i \leq W\}$

v_i - ceny, w_i - váhy, W - nosnosť
$d(u)$ - dekadický / binárny zápis čísla / množiny u

$m(x,y) = \sum_{i \in S} v_i$

Účastník A posle správu M účastníkovi B:

$$M \to \boxed{P_B}^{A} \xrightarrow{P_B(M)} \boxed{S_B}^{B} \to S_B(P_B(M)) = M$$

Vytvorenie P_u, S_u:

1.) Výber náhodné veľké (dek. 100 cif.) prvočísla p, q.
2.) Vypočítaj $n = pq$ $\quad \phi(n) = (p-1)(q-1)$
3.) Výber nepárne číslo e nesúdeliteľné s $\phi(n)$ [Pomocou algo. Euklid nájdi $e \in \{3,5,7$ a celé x, y t.ž. $1 = \phi(n)\,x + e\,y$]
4.) Vypočítaj $d = e^{-1} \bmod \phi(n)$ [$d \leftarrow y$; ak $y < 0$ tak $d \leftarrow y + \phi(n)$] Pre d platí: $1 = (e\,d) \pmod{\phi(n)}$, $d > 0$]
5.) Nech $D = \mathbb{Z}_n = \{0, 1, \dots, n-1\}$. Potom

$$P_u(M) = M^e \bmod n \quad \Big\} \forall M \in D$$
$$S_u(M) = M^d \bmod n$$

(Na výpočet $P_u(M)$ a $S_u(M)$ použijeme algo. Mod-Exp)

Bezpečnosť RSA

- Ako získať S_u? Vedieť P_u (t.j. e a n) nie ťažké, stačí získať $d \Rightarrow$ potrebujeme S_u (t.j. d a n?)
- Ako získať d (z hodnôt e a n)?
  Vieme, že známy bezpečný spôsob, ako z n získať p, q, potom $\phi(n)$ a potom d (z $\phi(n)$ a e)
- Ako získať p, q z n? Ťažké! :)

Digitálne podpísaný dokument M je dvojica $(M, S_A(M))$.
Každý pomocou P_A = t.j. môže získať istotu pravosti podpisu overením rovnosti $M = P_A(S_A(M))$

Odoslaný list s digitálnym podpisom od A pre B je tvaru
$P_B(M, S_A(M))$.

B dešifruje správu kľúčom S_B

$$S_B(P_B(M, S_A(M))) = (M, S_A(M))$$

overí podpis. $M \stackrel{?}{=} P_A(S_A(M))$

Def F-cia $f: \Sigma^* \to \Sigma^*$ je jednosmerná, ak
1.) $|w| = |f(w)| \quad \forall w \in \Sigma^*$
2.) f je vypočítateľná deterministickým polynomiálnym časom
3.) Pre každý pravdep. polynomiálny alg. A, pre každé k a dosť veľké n platí:

$$Pr[A(f(w)) = y, \text{ kde } f(y) = f(w)] \le n^{-k}$$

pre každé náhodne vybrané $w \in \Sigma^*$, $|w| = n$.

Pozn. a) Nie je známa žiadna jednosmerná fcia
b) Kandidátom je fcia násobenia dvoch prir. čísel $f(w) = w_1 \cdot w_2$ ($|w_1| = |w_2|$)

$w = \underbrace{w_1 w_2}_{\text{zreťazenie}}$

RSA $\quad f: n = p \cdot q$
$f(\langle p \rangle \langle q \rangle) = n$

Pre $0 < \varepsilon, \delta < 1$ platí: $|\text{integral} - \int_a^b f(x)\,dx| < \varepsilon$ s pravdep. $\geq 1-\delta$ keď $n \geq \lceil \frac{1}{4\varepsilon^2\delta} \rceil$

Testovanie veľkých prvočísel

12. 11. 2014

Miller-Rabinov test / vieš, či b je / nie je prvočíslo s pravdepodobnosťou omylu $\leq 2^{-r}$

vstup: nepárne $b > 2$, $r \in \mathbb{N}$
nech $2^t/(b-1)$ & $2^{t+1}/(b-1)$
for i from 1 to r do begin
  vyber náhodné celé číslo "a" $(1 \leq a \leq b-1)$
1. if $x^2 \equiv 1 \pmod b$ & $x \neq \pm 1 \pmod b$ pre nejaké $x \in \{a^{\frac{(b-1)}{2}}, a^{\frac{(b-1)}{4}}, \dots, a^{\frac{(b-1)}{2^t}}\}$
   then return "b určite nie je prvočíslo"
2. if $a^{(b-1)} \not\equiv 1 \pmod b$ then return "b určite nie je prvočíslo"
end
3. return "b je prvočíslo (s pravdep. omylu 2^{-r})"

Veta: Ak b nie je prvočíslo, potom náhodne vygenerované číslo "a" $(1 \leq a \leq b-1)$ spĺňa podmienku v príkaze 1. alebo 2. s pravdep. $\geq 1/2$.

Dôsledok: Iterácie v príkaze 3. je zbytočné.

Dôvod: Ak b nie je prvočíslo, potom pravdep. toho, že ani jedno z r náhodne vygenerených čísel "a" nespĺňa podmienku v príkaze 1 ani 2 je $\leq (1/2)^r$.

Pozn. Existuje implementácia Miller-Rabin. testu (založená na proc. Mod-Exp) s časovou zlož. $O(rm^3)$, ak b je najviac m-bitové číslo.

Pozn. Existuje deterministický algoritmus so zlož. $O(m^6)$.

Pravdepodobnostné generovanie náhodných (100 ciferných) prvočísel.

(1) Generuj náhodných čísel tým 100 cif číslo b (generuj náhodným binárnym postupnosť dĺžky 340)
(2) Zisti, či b je prvočíslo (Miller-Rabin)
(3) Ak nie, opakuj (1) a (2) až dok nájdeme prvočíslo.

Pozn. Algoritmus vyhovie (1) a (2) v priemere 230-krát, teda v priemere hľadá 230-te číslo medzi 100 cif je prvočíslo, hľadáme

$$\lim_{n\to\infty} \frac{P(n)}{n/\log_2 n} \quad a \quad \log_2 10^{100} \approx 230, \text{ kde } P(n) \text{ je počet prvočísel v } \langle 2, n\rangle \quad (n = 10^{100})$$

RSA - šifrovací systém

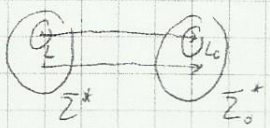
Ron Rivest, Adi Shamir, Len Adleman

Každý účastník U má:
- verejný kľúč P_U } vytvára si ich sám
- tajný kľúč S_U
D - množina správ

Platí $S_U(P_U(M)) = M = P_U(S_U(M))$ $\forall M \in D$

$$PSPACE = \bigcup_{k>0} DSPACE(n^k)$$

Def. $L \subseteq \Sigma^*$ je polynomiálně transf. na $L_0 \subseteq \Sigma_0^*$ ak existuje det. T-stroj s polynom. časovou zložit., ktorý vstup $w \in \Sigma^*$ prekarbonformuje na výstup $w' \in \Sigma_0^*$ t.ž. $w \in L$ iff $w' \in L_0$



Def: L_0 je PSPACE úplný, ak $L_0 \in PSPACE$ a každý jazyk $L \in PSPACE$ je polynom. transformovateľný na L_0.

Def. Bool. formula s kvantifikátormi je úplne kvantifikovaná, ak každá jej premenná je v oblasti pôsobnosti niektorého kvantifikátora.

Pr. $\forall x \exists y [(x \vee y) \wedge (\neg x \vee \neg y)]$ - úplne kvantifikovaná formula.

$TQBF = \{ \langle \Phi \rangle \mid \Phi$ je pravdivá úplne kvantifik. bool. formula $\}$
kód formuly Φ

Veta. Jazyk TQBF je PSPACE-úplný.

Dôkaz. (a): $TQBF \in PSPACE$:

Algoritmus pre TQBF:

(1) Ak Φ nemá premenné, (t.j. má len konštanty), obodnoť Φ a akceptuj, ak Φ je pravdivá, inak zamietni
(2) Ak Φ je tvaru $\exists x \psi$, rekurzívne 2-krát volaj alg. na ψ - raz pre $x=0$, raz pre $x=1$. Ak jeden z výsledkov akceptuje, potom akceptuj Φ, inak zamietni.
(3) Ak Φ je tvaru $\forall x \psi$, postupuj podobne ako v (2), ale musia sa akceptovať oba prípady.

Pamäť. zlož.: hĺbka rekurzie $\leq$ počet premenných $\leq$ dĺžka kódu formuly = dĺžka vstupu.
Na každej úrovni rekurzie stačí v pásobornom bloku rozsahu $O(1)$
$\Rightarrow$ celkom stačí lineárna pamäť $\Rightarrow TQBF \in PSPACE$.

19. 11. 2014

(b): Každý $L \in PSPACE$ je polynomiálne transf. na TQBF, t.zn.
Nech $L \in PSPACE$, $L \subseteq \Sigma^* \Rightarrow$ Pre L $\exists$ det. T-stroj M akceptujúci L v polynomiálnej pamäti $S(n)$ (M nemá pracovné pásky, ale má čítaciu hlavu na vstupnej páske - pozri Dôkaz Cook-Levin. Vety).
Čas výpočtu M na vstupe w ($|w|=n$) je $\leq$ # rôznych konfig. v pamäti $S(n)$, čo je $\leq 2^{rS(n)}$ pre vhodné $r \geq 1$.

Cieľ: Pre $\ell = 1, 2, 4, 8, \ldots, 2^{rS(n)}$ zostrojiť bool. formulu

$\Phi^\ell(\bar{x}, \bar{y})$ t.ž. $\Phi^\ell(\bar{C}_1, \bar{C}_2)$ je úplne kvantif. a je pravdivá iff
M prejde z konfig. C_1 do konf. C_2 počas $\leq \ell$ krokov.
$\bar{C}_1$ (resp. $\bar{C}_2$) je bool. kód konf. C_1 (resp. C_2); C_1, C_2 sú kódované podobne