IB107 Vyčíslitelnost a složitost

další NP-úplné problémy, prostorová složitost vztahy složitostních tříd

Jan Strejček

Fakulta informatiky
Masarykova univerzita

problém VERTEX-COVER

Definice (vrcholové pokrytí)

Nechť G je neorientovaný graf. Podmnožina jeho vrcholů se nazývá vrcholové pokrytí grafu G, pokud obsahuje alespoň jeden vrchol každé hrany grafu G.

Definice (problém VERTEX-COVER)

Problém VERTEX-COVER je problém rozhodnout, zda daný graf G má vrcholové pokrytí obsahující právě daný počet vrcholů k.

$$\mathit{VERTEX\text{-}COVER} = \{\langle G, k\rangle \mid G \text{ je graf s vrcholovým pokrytím obsahujícím právě } k \text{ vrcholů}\}$$

VERTEX-COVER je NP-úplný

Věta

VERTEX-COVER je NP-úplný.

Důkaz: *VERTEX-COVER* $\in$ NP:

VERTEX-COVER je NP-těžký: $3SAT \leq_p$ *VERTEX-COVER*

VERTEX-COVER je NP-úplný

Nechť $\varphi = (a_1 \vee b_1 \vee c_1) \wedge (a_2 \vee b_2 \vee c_2) \wedge \ldots \wedge (a_l \vee b_l \vee c_l)$ používá množinu proměnných X. Zkonstruujeme graf $G = (V, E)$, kde

- $V = \{a_1, b_1, c_1, a_2, b_2, c_2, \ldots, a_l, b_l, c_l\} \cup \{x, \neg x \mid x \in X\}$ a
- $E = \{\{a_i, b_i\}, \{a_i, c_i\}, \{b_i, c_i\} \mid 1 \leq i \leq l\} \cup \{\{x, \neg x\} \mid x \in X\} \cup$
  $\cup$ “hrany mezi literálem a shodným vrcholem x nebo $\neg x$”

$$\varphi = (x \vee y \vee z) \wedge (\neg x \vee \neg y \vee \neg z) \wedge (x \vee \neg y \vee \neg y)$$

φ je splnitelná $\iff$ G má vrcholové pokrytí s $k = 2l + |X|$ vrcholy
Zároveň $|G| = \mathcal{O}(|\varphi|)$ a tedy $3SAT \leq_p VERTEX\text{-}COVER$. ■

problém SUBSET-SUM

Definice (problém SUBSET-SUM)

Problém SUBSET-SUM je problém rozhodnout, zda pro danou konečnou multimnožinu S obsahující přirozená čísla existuje $\{y_1, y_2, \ldots, y_l\} \subseteq S$ splňující $y_1 + y_2 + \ldots + y_l = t$ pro dané t.

$$\mathit{SUBSET\text{-}SUM} = \{\langle S, t\rangle \mid S = \{x_1, \ldots, x_k\} \text{ a pro nějaké } \{y_1, \ldots, y_l\} \subseteq S \text{ platí } \sum_{i=1}^{l} y_i = t\}$$

SUBSET-SUM je NP-úplný

Věta

SUBSET-SUM je NP-úplný.

Důkaz: *SUBSET-SUM* $\in$ NP:

SUBSET-SUM je NP-těžký: $3SAT \leq_p$ *SUBSET-SUM*

VERTEX-COVER je NP-úplný

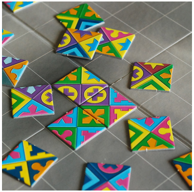
Nechť $\varphi = (a_1 \vee b_1 \vee c_1) \wedge (a_2 \vee b_2 \vee c_2) \wedge \ldots \wedge (a_k \vee b_k \vee c_k)$ proměnné $x_1, \ldots, x_l$. Zkonstruujeme S a t takto:

		1	2	3	4	...	l	1	2	...	k
S	x_1	1	0	0	0	...	0	1	0	...	0
	$\neg x_1$	1	0	0	0	...	0	0	0	...	0
	x_2	0	1	0	0	...	0	1	0	...	0
	$\neg x_2$	0	1	0	0	...	0	0	1	...	0
	$\vdots$					$\vdots$				$\vdots$	
	x_l	0	0	0	0	...	1	0	0	...	0
	$\neg x_l$	0	0	0	0	...	1	0	1	...	1
	g_1	0	0	0	0	...	0	1	0	...	0
	h_1	0	0	0	0	...	0	1	0	...	0
	g_2	0	0	0	0	...	0	0	1	...	0
	h_2	0	0	0	0	...	0	0	1	...	0
	$\vdots$					$\vdots$				$\vdots$	
	g_k	0	0	0	0	...	0	0	0	...	1
	h_k	0	0	0	0	...	0	0	0	...	1
t		1	1	1	1	...	1	3	3	...	3

φ je splnitelná $\iff \langle S, t\rangle \in$ *SUBSET-SUM*.
Zároveň $|S| = \mathcal{O}((l+c)^2)$ a tedy $3SAT \leq_p$ *SUBSET-SUM*. ■

další NP-úplné problémy

- problém hamiltonovské cesty *HAMPATH*
- problém obchodního cestujícího
  (rozhodovací problém: existuje cesta do dané velikosti?)
- problém okachličkování (tiling problem)



Eternity II

- 256 dílků (16×16)
- 2.000.000 dolarů za řešení
- během 3 a půl roku nenalezeno

Lemma

Pokud je nějaký NP-úplný problém v P, pak $P = NP$.

prostorová složitost algoritmu

- paměť použitá při výpočtu
- závisí na vstupu
- jako základní model použijeme Turingův stroj
- zkoumáme nejhorší případ, tedy maximální počet přečtených políček pásky v závislosti na délce vstupu
- lze zkoumat i průměrný případ

prostorová složitost Turingova stroje

Definice 2.1 (prostorová složitost deterministického TM)

Nechť $\mathcal{M}$ je úplný deterministický (jednopáskový nebo vícepáskový) Turingův stroj se vstupní abecedou Σ. Pro každé $w \in \Sigma^*$ definujeme $s_{\mathcal{M}}(w)$ jako počet políček pásky, které stroj $\mathcal{M}$ čte při výpočtu na vstupu w. Prostorová složitost stroje $\mathcal{M}$ je pak funkce $S_{\mathcal{M}} : \mathbb{N} \to \mathbb{N}$ definovaná vztahem

$$S_{\mathcal{M}}(n) = \max\{s_{\mathcal{M}}(w) \mid w \in \Sigma^n\}.$$

Definice 2.2 (prostorová složitost nedeterministického TM)

Definice prostorové složitosti úplného nedeterministického Turingova stroje se liší jen tím, že $s_{\mathcal{M}}(w)$ označuje maximální počet políček pásky, které stroj $\mathcal{M}$ čte při nějakém výpočtu na vstupu w.

Příklad

$$\mathcal{M} = (\{q_0, q_1, q_{acc}, q_{rej}\}, \{0,1\}, \{0,1,\triangleright,\sqcup\}, \triangleright, \sqcup, \delta, q_0, q_{acc}, q_{rej})$$

δ	$\triangleright$	0	1	$\sqcup$
q_0	$(q_0, \triangleright, R)$	$(q_0, 0, R)$	$(q_1, 1, R)$	$(q_{acc}, -, -)$
q_1		$(q_{rej}, -, -)$	$(q_1, 1, R)$	$(q_{acc}, -, -)$

komprese prostoru

Věta

Pro každý deterministický úplný TM $\mathcal{M}$ a pro každé $m > 1$ lze zkonstruovat deterministický úplný TM $\mathcal{M}'$ tak, že $L(\mathcal{M}) = L(\mathcal{M}')$ a

$$S_{\mathcal{M}'}(n) = \left\lceil \frac{S_{\mathcal{M}}(n)}{m} \right\rceil + n + 2.$$

Důkaz:

$\blacksquare$

Protože prostor lze komprimovat, používáme asymptotickou notaci.

prostorové složitostní třídy problémů

prostorová složitost problému = nejmenší prostorová složitost, s jakou lze daný problém rozhodnout

Definice 2.3 (prostorové složitostní třídy problémů)

Každá funkce $f : \mathbb{N} \to \mathbb{R}_0^+$ definuje prostorové složitostní třídy problémů:

$$\begin{aligned}SPACE(f(n)) = \{L \mid L \text{ je rozhodovaný nějakým det. TM } \mathcal{M}\\ \text{s prostorovou složitostí } S_{\mathcal{M}}(n) = \mathcal{O}(f(n))\}\end{aligned}$$

$$\begin{aligned}NSPACE(f(n)) = \{L \mid L \text{ je rozhodovaný nějakým nedet. TM } \mathcal{N}\\ \text{s prostorovou složitostí } S_{\mathcal{N}}(n) = \mathcal{O}(f(n))\}\end{aligned}$$

$SAT \in \mathrm{SPACE}(n)$

$SAT = \{\langle\varphi\rangle \mid \varphi$ je splnitelná výroková formule$\}$

SAT může být rozhodován deterministickým třípáskovým TM $\mathcal{M}$:

- na 2. pásku postupně zapisujeme všechna ohodnocení proměnných
- na 3. pásce pro dané ohodnocení ověříme, zda splňuje φ
- akceptujeme, pokud narazíme na splňující ohodnocení
- zamítneme, pokud žádné ohodnocení není splňující

Prostor lze použít opakovaně.

determinismus a nedeterminismus, čas a prostor

Věta

Pro každou funkci $f : \mathbb{N} \to \mathbb{R}_0^+$ *platí:*

1. $TIME(f(n)) \subseteq NTIME(f(n))$
2. $SPACE(f(n)) \subseteq NSPACE(f(n))$
3. $TIME(f(n)) \subseteq SPACE(f(n))$
4. $NTIME(f(n)) \subseteq NSPACE(f(n))$
5. $SPACE(f(n)) \subseteq \bigcup_{k\in\mathbb{N}} TIME(k^{f(n)})$
6. $NSPACE(f(n)) \subseteq \bigcup_{k\in\mathbb{N}} NTIME(k^{f(n)})$

Důkaz:

■

Savitchova věta

Věta 2.6 (Savitchova věta)

Pro každou funkci $f : \mathbb{N} \to \mathbb{R}_0^+$ *splňující* $f(n) \geq n$ *platí:*

$$NSPACE(f(n)) \subseteq SPACE(f^2(n))$$

Standardní převod nedet. TM na deterministický nefunguje:

Savitchova věta

Důkaz: Nechť $\mathcal{N}$ je nedet. TM s prostorovou složitostí $f(n)$. Stroj upravíme tak, aby před akceptováním smazal pásku a posunul hlavu zcela vlevo. Má tedy jen jednu akceptující konfiguraci c_{acc}. Výpočet stroje má maximálně $k^{f(n)}$ kroků.

Ekvivalentní deterministický stroj $\mathcal{M}$ implementuje proceduru $comp(c_1, c_2, t)$, která akceptuje, pokud lze ve stroji $\mathcal{N}$ během nejvýše t kroků přejít z konfigurace c_1 do c_2, jinak zamítá. Je-li c_w iniciální konfigurace stroje $\mathcal{N}$ pro w, stačí tedy spustit $comp(c_w, c_{acc}, k^{f(n)})$.

$comp(c_1, c_2, t)$ lze implementovat rekurzivně:

Savitchova věta

Algoritmus pro *comp*(c_1, c_2, t):

$t = 1$: Otestujeme, zda platí $c_1 = c_2$ nebo $c_1 \vdash_{\mathcal{N}} c_2$.

Pokud platí, akceptujeme, jinak zamítneme.

$t > 1$: Pro každou konfiguraci c' stroje $\mathcal{N}$ využívající nejvýše $f(n)$ políček

- spustíme $comp(c_1, c', \lceil \frac{t}{2} \rceil)$ a $comp(c', c_2, \lfloor \frac{t}{2} \rfloor)$,
- pokud obojí akceptuje, akceptujeme.

Pokud žádné c' nevedlo k akceptování, zamítneme.

Prostorová složitost:

- *comp* potřebuje prostor na c_1, c_2, c' a t (a něco konstantního):
- hloubka rekurzivního volání:
- celkem:

polynomiální prostorové složitostní třídy

$$\text{PSPACE} = \bigcup_{k \in \mathbb{N}} \text{SPACE}(n^k) \qquad \text{NPSPACE} = \bigcup_{k \in \mathbb{N}} \text{NSPACE}(n^k)$$

Věta

$PSPACE = NPSPACE$

Důkaz: Plyne přímo z definice ($\subseteq$) a ze Savitchovy věty ($\supseteq$). ∎

vztahy prostorových a časových tříd

$$P \subseteq \mathrm{NP} \subseteq \mathrm{NPSPACE} = \mathrm{PSPACE} \subseteq \mathrm{EXPTIME} \subseteq \mathrm{NEXPTIME}$$