IB107 Vyčíslitelnost a složitost

časová složitost algoritmu, časové složitostní třídy

Jan Strejček

Fakulta informatiky
Masarykova univerzita

vztah teorie vyčíslitelnosti a teorie složitosti

časová složitost algoritmu

- počet “kroků” výpočtu
- závisí na vstupu a výpočetním modelu
- jako základní model použijeme Turingův stroj
- zkoumáme nejhorší případ, tedy maximální počet kroků v závislosti na délce vstupu
- lze zkoumat i průměrný případ

časová složitost deterministického Turingova stroje

Definice (časová složitost TM)

Nechť $\mathcal{M}$ je úplný deterministický (jednopáskový nebo vícepáskový) Turingův stroj se vstupní abecedou Σ. Pro každé $w \in \Sigma^*$ definujeme $t_{\mathcal{M}}(w)$ jako počet kroků výpočtu stroje $\mathcal{M}$ na vstupu w. Časová složitost stroje $\mathcal{M}$ je pak funkce $T_{\mathcal{M}} : \mathbb{N} \to \mathbb{N}$ definovaná vztahem

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

Říkáme, že $\mathcal{M}$ pracuje v čase $T_{\mathcal{M}}(n)$.

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		$(q_{rej}, -, -)$	$(q_1, 1, R)$	$(q_{acc}, -, -)$

příklad

$\mathcal{M} = (\{q_0, q_1, r, s_0, s_1, q_{acc}, q_{rej}\}, \{0,1\}, \{0,1,X,\triangleright,\sqcup\}, \triangleright, \sqcup, \delta, q_0, q_{acc}, q_{rej})$

δ	$\triangleright$	0	1	X	$\sqcup$
q_0	$(q_0, \triangleright, R)$	$(q_0, 0, R)$	$(q_1, 1, R)$		$(r, \sqcup, L)$
q_1		$(q_{rej}, -, -)$	$(q_1, 1, R)$		$(r, \sqcup, L)$
r	$(s_0, \triangleright, R)$	$(r, 0, L)$	$(r, 1, L)$	(r, X, L)	
s_0		(s_1, X, R)	$(q_{rej}, -, -)$	(s_0, X, R)	$(q_{acc}, -, -)$
s_1		$(s_1, 0, R)$	(r, X, L)	(s_1, X, R)	$(q_{rej}, -, -)$

$\triangleright$	0	0	0	1	1	1	$\sqcup$	$\sqcup$	...

motivace pro asymptotickou analýzu

- jaká složitost je lepší: $6n$ nebo $n^2 + 5$?

- na delších vstupech jsou výpočty obvykle delší
- zajímá nás chování pro $n \to \infty$
- konstantní faktor neuvažujeme (výpočet lze zrychlit)

věta o zrychlení

Věta (o zrychlení)

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

$$T_{\mathcal{M}'}(n) \leq \frac{T_{\mathcal{M}}(n)}{m} + n + 1.$$

Důkaz:

$\mathcal{O}$-notace

Definice (asymptotická horní závora)

Nechť $f, g : \mathbb{N} \to \mathbb{R}^+$ jsou funkce. Řekneme, že g je asymptotická horní závora pro f, a píšeme $f \in \mathcal{O}(g)$ nebo $f = \mathcal{O}(g)$, jestliže existují konstanty $c, n_0 \in \mathbb{N}$ takové, že pro každé $n \geq n_0$ platí

$$f(n) \leq c \cdot g(n).$$

Příklad: $15n^3 + 3n^2 + 11n + 7$

počítání s $\mathcal{O}$-notací

- logaritmy: $\mathcal{O}(\log n)$
- sčítání: $\mathcal{O}(n^3) + \mathcal{O}(n)$
- mocniny: $2^{\mathcal{O}(n)}$

příklad

TM $\mathcal{M}$ rozhodující $\{0^k 1^k \mid k \geq 0\}$ lze popsat i takto:

1. Zjistí, zda vstup obsahuje nějakou 0 za 1. Pokud ano, zamítne.
2. Dokud je na pásce nějaká 0, projíždí pásku a vždy škrtne jednu 0 a jednu 1. Pokud se nepovede k nějaké 0 najít 1, zamítne.
3. Pokud po vyškrtání všech 0 zbude na pásce nějaká 1, zamítne. Jinak akceptuje.

deterministické časové složitostní třídy problémů

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

Definice (časová složitostní třída problémů)

Každá funkce $f : \mathbb{N} \to \mathbb{R}^+$ *definuje (deterministickou) časovou složitostní třídu problémů*

$$TIME(f(n)) = \{L \mid L \text{ je rozhodovaný nějakým deterministickým jedno- nebo vícepáskovým TM } \mathcal{M} \text{ s časovou složitostí } T_{\mathcal{M}}(n) = \mathcal{O}(f(n))\}.$$

příklad

$L = \{0^k 1^k \mid k \geq 0\}$

- ukázali jsme jednopáskový det. TM rozhodující L v čase $\mathcal{O}(n^2)$
- existuje jednopáskový det. TM rozhodující L v čase $\mathcal{O}(n \log n)$
  1. Zjistí, zda vstup obsahuje nějakou 0 za 1. Pokud ano, zamítne.
  2. Dokud páska obsahuje nějakou 0 i 1, opakuje následující: Pokud je celkový počet 0 a 1 n pásce lichý, zamítne. Pokud je sudý, škrtne každý lichý výskyt 0 a každý lichý výskyt 1.
  3. Jestliže na pásce zůstane nějaká 0 nebo 1, zamítne. Jinak akceptuje.

příklad

$L = \{0^k 1^k \mid k \geq 0\}$

- neexistuje jednopáskový det. TM rozhodující L s menší složitostí než $\mathcal{O}(n \log n)$
  (platí, že každý jazyk rozhodnutelný jednopáskovým det. TM v čase $o(n \log n)$ je regulární)
- existuje dvoupáskový deterministický TM rozhodující L v čase $\mathcal{O}(n)$, tedy L je ve třídě TIME(n)

vliv výpočetního modelu

- na rozdíl od vyčíslitelnosti, ve složitosti na výpočetním modelu záleží
- rozdíl je i mezi jednopáskovým a dvoupáskovým deterministickým TM
- jaký model zvolit?
- je volba deterministického vícepáskového TM správná?
- rozdíly jsou u běžných sekvenčních deterministických výpočetních modelů poměrně malé
- např. RAM (random-access machine) pracující v čase $f(n)$ lze převést na vícepáskový deterministický TM pracující v čase $\mathcal{O}(f^3(n) \cdot (f(n) + n)^2)$
- nedeterminismus přináší výrazný rozdíl

převod vícepáskového TM na jednopáskový

Věta

Pro každý vícepáskový deterministický TM pracující v čase $f(n) \geq n$ lze sestrojit ekvivalentní jednopáskový deterministický TM pracující v čase $\mathcal{O}(f^2(n))$.

Důkaz:

1. neprázdný obsah k pásek a polohy hlav zapíšeme za sebe na 1 pásku $\rightarrow \mathcal{O}(n)$
2. simulace jednoho kroku
   - zjistit informace pod hlavami = projít pásku, každá původní páska má max. $f(n)$ neprázdných polí $\rightarrow \mathcal{O}(f(n))$
   - provést krok, zapsat nové symboly a posunout hlavy (případně přidat další políčka na původní pásky odsunutím obsahu dalších pásek, max. k políček) $\rightarrow \mathcal{O}(f(n))$
3. simulujeme $f(n)$ kroků $\rightarrow$ celkem $\mathcal{O}(n) + \mathcal{O}(f^2(n))$ ■

časová složitost nedeterministického Turingova stroje

Definice (časová složitost nedeterministického TM)

Nechť $\mathcal{M}$ je úplný nedeterministický Turingův stroj se vstupní abecedou Σ. Pro každé $w \in \Sigma^*$ definujeme $t_{\mathcal{M}}(w)$ jako počet kroků nejdelšího výpočtu stroje $\mathcal{M}$ na vstupu w. Časová složitost stroje $\mathcal{M}$ je pak funkce $T_{\mathcal{M}} : \mathbb{N} \to \mathbb{N}$ definovaná vztahem

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

nedeterministické časové složitostní třídy problémů

Definice (nedeterministická časová složitostní třída problémů)

Každá funkce $f : \mathbb{N} \to \mathbb{R}^+$ *definuje (nedeterministickou) časovou složitostní třídu problémů*

$$NTIME(f(n)) = \{L \mid L \text{ je rozhodovaný nějakým nedeterministickým jedno- nebo vícepáskovým TM } \mathcal{M} \text{ s časovou složitostí } T_{\mathcal{M}}(n) = \mathcal{O}(f(n))\}.$$

Z definic plyne $\text{TIME}(f(n)) \subseteq \text{NTIME}(f(n))$.

převod nedeterministického TM na deterministický

Věta

Pro každý nedeterministický jednopáskový TM pracující v čase $f(n) \geq n$ lze sestrojit ekvivalentní deterministický jednopáskový TM pracující v čase $2^{\mathcal{O}(f(n))}$.

Důkaz: Nedet. TM $\mathcal{M}$, který má z každé konfigurace max. k přechodů, simulujeme 3-páskovým deterministickým strojem, který prohledává strom výpočtů $\mathcal{M}$ do šířky. Pro každý uzel provedeme znovu výpočet z iniciální konfigurace.
Strom výpočtů má hloubku nejvýše $f(n)$ a tudíž má nejvýše $k^{f(n)}$ listů a méně než $2 \cdot k^{f(n)}$ uzlů. $\rightarrow \mathcal{O}(k^{f(n)})$ uzlů
Simulace výpočtu do jednoho uzlu zabere nejvýše $\mathcal{O}(f(n))$ kroků.
3-páskový stroj tedy pracuje v čase $\mathcal{O}(f(n) \cdot k^{f(n)}) = 2^{\mathcal{O}(f(n))}$.

Převod na jednopáskový det. stroj:
$(2^{\mathcal{O}(f(n))})^2 = 2^{\mathcal{O}(2f(n))} = 2^{\mathcal{O}(f(n))}$. ■

nejvýznamnější časové složitostní třídy

deterministické	nedeterministické
$\mathrm{P} = \mathrm{PTIME} = \bigcup_{k \in \mathbb{N}} \mathrm{TIME}(n^k)$	$\mathrm{NP} = \bigcup_{k \in \mathbb{N}} \mathrm{NTIME}(n^k)$
$\mathrm{EXPTIME} = \bigcup_{k \in \mathbb{N}} \mathrm{TIME}(2^{n^k})$	$\mathrm{NEXPTIME} = \bigcup_{k \in \mathbb{N}} \mathrm{NTIME}(2^{n^k})$

$$\mathrm{P} \subseteq \mathrm{NP} \subseteq \mathrm{EXPTIME} \subseteq \mathrm{NEXPTIME}$$

- běžné deterministické sekvenční modely výpočtu lze mezi sebou převádět s polynomiálním nárůstem časové složitosti $\Longrightarrow$ definice P a EXPTIME nejsou citlivé na volbu modelu
- EXPTIME je obvykle složitost algoritmů řešících problém hrubou silou

Cook-Karpova teze

P obsahuje právě všechny prakticky řešitelné problémy.

vztah P a NP

$$\mathrm{P} \stackrel{?}{=} \mathrm{NP}$$

- asi nejznámější otevřený problém teoretické informatiky
- věří se, že platí $P \subsetneq \mathrm{NP}$
- důsledky do počítačové bezpečnosti
- Clay Mathematics Institute (CMI) vypsal 1.000.000 USD za řešení