

Paralelné algoritmy, část č. 10

František Mráz

Kabinet software a výuky informatiky, MFF UK, Praha

Paralelné algoritmy, 2011/2012

1 P-úplnosť a ťažko paralelizovateľné úlohy

- Paralelizovateľnosť
- Problém generovateľnosti
- P-úplnosť problému generovateľnosti
- Problém CVP
- CFE – problém prázdneho slova v bezkontextovom jazyku

Kedy je problém paralelizovateľný?

- (1) doba paralelného výpočtu klesá s rastúcim počtom procesorov p v istom rozsahu $1 \leq p \leq a(n)$, kde $a(n)$ je rastúca funkcia
 - v predchádzajúcich prednáškach väčšinou $1 \leq p \leq \frac{n}{\log^k n}$, pre pevné k
 - používať $p > a(n)$ procesorov nič neprináša
 - malé $a(n)$ ruší výhody paralelizácie
- (2) problém je možné riešiť extrémne rýchle s rozumným počtom procesorov – polylogaritmický čas a polynomiálny počet procesorov – **NC**

budeme používať (2)

Pozor algoritmus z NC neznamená vždy zrýchlenie

- určenie stromu prehľadávania do **šírky** pre neorientovaný graf sa dá urobiť s použitím algoritmu na výpočet najkratších vzdialeností medzi všetkými dvojicami vrcholov – čas $O(\log^2 n)$ s $O(\frac{n^3}{\log n})$ procesormi na COMMON
- čas $O(\frac{n^3 \log n}{p} + \log^2 n)$ s p procesormi
- najlepší sekvenčný algoritmus s časovou zložitou $O(m + n)$ je však rýchlejší pre $p = o(n \log n)$!

Triedy zložitosti

obvykle definované cez jazyky; rozhodovacie problémy $w \stackrel{?}{\in} L$

NC trieda všetkých jazykov L ; \exists algoritmus pre PRAM, ktorý pre dané slovo w dĺžky n rozhoduje či $w \in L$ v čase $O(\log^k n)$ s polynomiálnym počtom procesorov, k je konštanta nezávislá na n .

P trieda všetkých jazykov rozpoznateľných (sekvenčným) Turingovým strojom v polynomiálnom čase

- **P** je robustná trieda – môže sa definovať rovnako i pre RAM a iné sekvenčné modely algoritmov
- triviálne $NC \subseteq P$ – sekvenčná simulácia NC-algoritmu
- $P \subseteq NC$ – otvorený problém
- usudzuje sa, že $P \not\subseteq NC$, ale nikto to nevie dokázať; aspoň boli nájdení kandidáti na problémy z $NC \setminus P$

NC-prevediteľnosť

L_1, L_2 jazyky

- L_1 je NC-prevediteľný na L_2 ($L_1 \leq_{NC} L_2$) \equiv_{df} \exists NC-algoritmus, ktorý transformuje ľubovoľné slovo u_1 na slovo u_2 také, že

$$u_1 \in L_1 \Leftrightarrow u_2 \in L_2$$

- ak $L_1 \leq_{NC} L_2$, tak každý algoritmus rozhodujúci L_2 sa dá transformovať na algoritmus rozhodujúci L_1

Lemma

Nech L_1, L_2 sú jazyky a $L_1 \leq_{NC} L_2$. Potom

$$L_2 \in \text{NC} \Rightarrow L_1 \in \text{NC} .$$

log-space prevediteľnosť

- $L_1 \leq_{\log} L_2 - L_1$ je transformované na L_2 v logaritmickom priestore –
 $\exists f: \forall u$ sa dá spočítať $f(u)$ v priestore $O(\log n)$ tak, že

$$u \in L_1 \Leftrightarrow f(u) \in L_2$$

- paralelná téza zaručuje:
 \leq_{\log} -prevediteľnosť $\Rightarrow \leq_{NC}$ -prevediteľnosť
- problém L je P-úplný ak
 - 1 $L \in P$
 - 2 Každý $A \in P$ sa dá previesť na L pomocou polynomiálneho počtu procesorov v polylogaritmickom čase
- budeme robiť NC-prevediteľnosť, ale v skutočnosti to pôjde log-space prevediteľnosťou, t.j. vždy, keď ukážeme, že problém je P-úplný, tak to bude platiť aj vzhľadom k log-space prevediteľnosti.

Problém generovateľnosti – GEN

Zadanie: Bázová množina X a binárny operátor \circ , iniciálna množina $T \subset X$ a cieľový prvok $x \in X$.

Otázka: Patrí x do uzáveru T vzhľadom k operácii \circ ?

- Výpočet uzáveru ($S \circ S = \{a \circ b \mid a \in S, b \in S\}$):

function closure(T)

begin $T' := T$

while $T' \neq T' \cup T' \circ T'$ do $T' := T' \cup T' \circ T'$

closure := T'

end;

- while-cyklus maximálne $|X \setminus T|$ -krát a $|\text{closure}(T)| \leq |X|$
 $\Rightarrow \text{GEN} \in \text{P}$

Problém generovateľnosti – GEN'

- ak by operácia \circ bola asociatívna, tak by sme to vedeli vyriešiť v logaritmickej čase s polynomiálnym počtom procesorov
- P-úplnosť ukážeme najprv pre problém GEN':

Zadanie: $(X, next)$, X množina, $next$ ternárny operátor na X ,
 $T \subset X$ a $x \in X$.

Otázka: Patrí x do uzáveru T vzhľadom k operácii $next$?

- zrejme $GEN' \in P$

Páskový model

- Nech Turingov stroj M rieši problém L (rozhoduje jazyk L) v čase $P(n)$, kde n je veľkosť vstupu. Počas výpočtu použije maximálne $P(n) + n$ políčok pásky. Nech $T(n) = P(n) + n$
- **Páskový model TS**
 - 1 dĺžka pásky je $T(n) + 2$, políčka $0, 1, \dots, T(n) + 1$
 - 2 políčko obsahuje jeden znak, 0-té a $(T(n) + 1)$ -vé obsahujú zarážku $\$$
 - 3 pre $1 \leq p, t \leq T(n)$ je $c(p, t)$ obsah políčka p v čase t
 - 4 $c(p, t + 1)$ je jednoznačne určené prechodovou funkciou TS

$c(p-1, t)$	$c(p, t)$	$c(p+1, t)$
	$c(p, t+1)$	

$trans : X \times X \times X \rightarrow X$

- 5 vstup je v $c(1, 0), \dots, c(n, 0)$; stav je kódovaný spolu so znakom na 1 políčku
- 6 výstup je

$$c(1, T(n)) = \begin{cases} \# & \text{áno} \\ \text{iný znak} & \text{nie} \end{cases}$$

Dôkaz P-úplnosti GEN'

Veta

GEN' je P-úplný problém.

Dôkaz:

- nech $L \in P \Rightarrow$ páskový model riešiaci L v čase $p(n)$ s abecedou Σ a množinou stavov Q
- GEN' $\in P$ už máme
- nech (t, p, sym) označuje, že v p -tom políčku v čase t je symbol sym
- bazová množina $X = \{ (t, p, sym) \mid \begin{array}{l} 0 \leq t \leq p(n), \\ 0 \leq p \leq p(n) + n + 2, \\ sym \in \Sigma \cup (\Sigma \times Q) \end{array} \}$
- iniciálna množina $T = \{(0, p, sym) \in X\}$ zodpovedajúca počiatočnému stavu pásky pre daný vstup
- cieľový prvok $(T(n), 1, \#)$

Dôkaz P-úplnosti GEN'

Lemma

(t, p, sym) sa dá vygenerovať z $T \Leftrightarrow$ v čase t políčko p obsahuje symbol sym .

- naša transformácia je NC redukcia, pretože $next(u, v, w)$ sa dá spočítať paralelne v polylogaritmickej čase s polynomiálnym počtom procesorov

Veta

Nech A je P-úplný problém, $B \in P$ a $A \leq_{NC} B$. Potom B je P-úplný problém.

Dôkaz P-úplnosti GEN

Veta

GEN je P-úplný problém.

Dôkaz:

- $GEN \in P$ máme
- GEN' je P-úplný; zadanie pre GEN': $(X', next), T', x'$
skonštruujeme zadanie pre GEN
 - $X := X' \cup (X')^2$
 - binárny operátor \circ :

$$\forall u, v, w \in X: \begin{array}{ll} u \circ v & := (u, v) \\ (u, v) \circ w & := next(u, v, w) \end{array}$$
 - $T := T'$
 - $x := x'$
- x je generovateľné z T v $X \Leftrightarrow x'$ je generovateľné z T' v X'

Problém výstupnej hodnoty obvodu – CVP

Circuit Value Problem

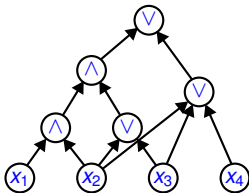
Zadanie: logický obvod (acyklický graf, vo vrchoch hradlá) a vstupné hodnoty

Otázka: Je výstup logická 1?

B je báza, z ktorej berieme hradlá. Napr.

- $B =$ množina všetkých booleovských funkcií dvoch premenných
- $B = \{\wedge, \vee\}$ – monotónna báza (bez negácie) **MonotoneCVP**

- $B = \left\{ \begin{array}{c} \wedge \\ \dots \end{array}, \begin{array}{c} \vee \\ \dots \end{array} \right\}$ – **UnboundedCVP**



P-úplnosť CVP

Veta

CVP je P-úplný problém.

Dôkaz:

- Nech (X, T, \circ, s) je zadanie problému GEN. Logický obvod bude obsahovať X tak, že T budú vstupné uzly so vstupnou hodnotou 1 a ostatné budú mať vstup 0.
- obvod bude mať $|X|$ vrstiev, prepojené budú iba po sebe idúce vrstvy
- pre všetky $x \in X \setminus T$ nájdeme všetky páry $(y_1, z_1), \dots, (y_k, z_k)$ také, že $y_i \circ z_i = x$ a postavíme obvod realizujúci funkciu $(y_1 \wedge z_1) \vee (y_2 \wedge z_2) \vee \dots \vee (y_k \wedge z_k)$ do uzlu x
- do uzlu x sa dostane 1 \Leftrightarrow existuje aspoň jedna dvojica ohodnotená 1
- celkovo uzol s vo vrstve $|X|$ dostane 1 $\Leftrightarrow s$ je generovateľné z T .
- konštrukcia je NC-redukcia

P-úplnosť CVP

Dôsledok

- MCVP je P-úplný problém*
- MCVP s topologicky usporiadanými uzlami je P-úplný problém (vstupné uzly majú menšie číslo, než uzly, do ktorých vedú vstupy – máme topologicky utriedený acyklický graf logického obvodu)*

Dôkaz:

- zrejmé z predchádzajúcej konštrukcie
- môžeme predpokladať, že prvky X sú očísľované a v dôkaze pre GEN môže očísľovanie zodpovedať lexikografickému usporiadaniu trojíc (t, p, sym)
⇒ z toho môžeme urobiť očísľovanie hradiel.

Pozn.: CVP je P-úplný aj pre planárne logické obvody, ale **CVP** nie je P-úplný pre planárne a zároveň monotónne obvody.

Prázdne slovo v bezkontextovom jazyku

Context-Free Language Empty Word Generation – CFE

Zadanie: bezkontextová gramatika G .

Otázka: Obsahuje $L(G)$ prázdne slovo?

Veta

CFE je P-úplný problém.

Dôkaz:

- U CFE je veľkosť vstupu rovná veľkosti gramatiky. $CFE \in P$.
- Nech (X, T, \circ, x) je zadanie GEN.
- Skonstruujeme bezkontextovú gramatiku $G = (\Pi, \Sigma, S, P)$:
 - $\Pi = X$,
 - $\Sigma = \emptyset$,
 - $S = x$,
 -

$$\begin{aligned} x \rightarrow yz &\equiv_{df} y \circ z = x \\ x \rightarrow \lambda &\equiv_{df} x \in T \end{aligned}$$

- G generuje prázdne slovo $\Leftrightarrow (X, T, \circ, x)$ má riešenie.

Prázdne slovo v bezkontextovom jazyku

- Ak gramatika G neobsahuje pravidlá s prázdnu pravou stranou a za veľkosť vstupu berieme súčet dĺžky vstupného slova w a veľkosti gramatiky, tak je problém $w \in L(G) ?$ v NC.
- Ak sa dovoľia λ -pravidlá, tak je to P-úplné.
- problém $L(G) \stackrel{?}{=} \emptyset$:
 - pre každé $x \in T$ pridáme pravidlo $x \rightarrow a$, kde a je nejaký terminál
- POZOR: Rozpoznávanie **pevne daného** bezkontextového jazyka je v NC!