

Paralelné algoritmy, část č. 3

František Mráz

Kabinet software a výuky informatiky, MFF UK, Praha

Paralelné algoritmy, 2011/2012

- 1 Simulácia sekvenčných výpočtov na PRAMe v čase $O(1)$
- 2 Simulácie Turingových strojov na PRAMe
- 3 Simulácie PRAMu na Turingových strojoch

Úvod

Otázky:

- 1 Aké najvyššie zrýchlenie sa dá dosiahnuť na PRAMe?

Úvod

Otázky:

- 1 Aké najvyššie zrýchlenie sa dá dosiahnuť na PRAMe?
- 2 Je možné všetky problémy z \mathbb{P} zrýchliť exponenciálne na “rozumnom paralelnom modele”, tj. na modele spĺňajúcom paralelnú tézu?

Úvod

Otázky:

- 1 Aké najvyššie zrýchlenie sa dá dosiahnuť na PRAMe?
- 2 Je možné všetky problémy z \mathbb{P} zrýchliť exponenciálne na “rozumnom paralelnom modele”, tj. na modele spĺňajúcom paralelnú tézu?

Úvod

Otázky:

- 1 Aké najvyššie zrýchlenie sa dá dosiahnuť na PRAMe?
- 2 Je možné všetky problémy z \mathbb{P} zrýchliť exponenciálne na “rozumnom paralelnom modele”, tj. na modele spĺňajúcom paralelnú tézu?

Definícia

Hovoríme, že funkcia $T : \mathbb{N} \rightarrow \mathbb{N}$ je *ws-konstruovateľná* (word size), ak PRAM so šírkou slova $O(T(n))$ môže počítať $T(n)$ v konštantnom čase zo vstupu n .

Úvod

Otázky:

- 1 Aké najvyššie zrýchlenie sa dá dosiahnuť na PRAMe?
- 2 Je možné všetky problémy z \mathbb{P} zrýchliť exponenciálne na “rozumnom paralelnom modele”, tj. na modele spĺňajúcom paralelnú tézu?

Definícia

Hovoríme, že funkcia $T : \mathbb{N} \rightarrow \mathbb{N}$ je *ws-konstruovateľná* (word size), ak PRAM so šírkou slova $O(T(n))$ môže počítať $T(n)$ v konštantnom čase zo vstupu n .

- mnoho dôležitých funkcií je *ws-konstruovateľných*

Simulácia Turingovho stroja na PRAMe

Lemma (Parberry, Schnitger)

Nech $T(n)$ je ws -konštruovateľná. Potom PRAM s obmedzenou sadu aritmetických inštrukcií môže simulovať k -páskový (deterministický) $T(n)$ časovo obmedzený Turingov stroj v konštantnom čase a so šírkou slova $O(T(n))$.

Simulácia Turingovho stroja na PRAMe

Lemma (Parberry, Schnitger)

Nech $T(n)$ je *ws*-konštruovateľná. Potom PRAM s obmedzenou sadu aritmetických inštrukcií môže simulovať k -páskový (deterministický) $T(n)$ časovo obmedzený Turingov stroj v konštantnom čase a so šírkou slova $O(T(n))$.

Dôkaz:

- je daný Turingov stroj $M = (Q, \Sigma, \delta, q_0, F)$; jeho inštrukcie očísľujeme

Simulácia Turingovho stroja na PRAMe

Lemma (Parberry, Schnitger)

Nech $T(n)$ je *ws*-konštruovateľná. Potom PRAM s obmedzenou sadu aritmetických inštrukcií môže simulovať k -páskový (deterministický) $T(n)$ časovo obmedzený Turingov stroj v konštantnom čase a so šírkou slova $O(T(n))$.

Dôkaz:

- je daný Turingov stroj $M = (Q, \Sigma, \delta, q_0, F)$; jeho inštrukcie očísľujeme
- procesory rozdelíme do $2^{O(T(n))}$ tímov – jeden tím pre každú postupnosť inštrukcií $r_0, r_1, \dots, r_{T(n)-1}$ Turingovho stroja M . Každý tím bude mať $2^{O(T(n))}$ procesorov.

Simulácia Turingovho stroja na PRAMe

Lemma (Parberry, Schnitger)

Nech $T(n)$ je *ws*-konštruovateľná. Potom PRAM s obmedzenou sadu aritmetických inštrukcií môže simulovať k -páskový (deterministický) $T(n)$ časovo obmedzený Turingov stroj v konštantnom čase a so šírkou slova $O(T(n))$.

Dôkaz:

- je daný Turingov stroj $M = (Q, \Sigma, \delta, q_0, F)$; jeho inštrukcie očísľujeme
- procesory rozdelíme do $2^{O(T(n))}$ tímov – jeden tím pre každú postupnosť inštrukcií $r_0, r_1, \dots, r_{T(n)-1}$ Turingovho stroja M . Každý tím bude mať $2^{O(T(n))}$ procesorov.
- Každý tím najprv určí polohy hláv na všetkých k páskach vo všetkých krokoch výpočtu M – čiastočné súčty pohybov hlavy podľa postupnosti inštrukcií.

Simulácia Turingovho stroja na PRAMe

- overiť:

Simulácia Turingovho stroja na PRAMe

- overiť:

- 1 je postupnosť stavov určená postupnosťou inštrukcií uskutočniteľná? Inštrukcia r_0 začína v q_0 , ak inštrukcia r_{i-1} prevedie M do stavu q_i , tak inštrukcia r_i začína v stave q_i .

Simulácia Turingovho stroja na PRAMe

- overiť:

- 1 je postupnosť stavov určená postupnosťou inštrukcií uskutočniteľná? Inštrukcia r_0 začína v q_0 , ak inštrukcia r_{i-1} prevedie M do stavu q_i , tak inštrukcia r_i začína v stave q_i .
- 2 pre každú z k pásov a každý krok t , $0 \leq t < T(n)$:

Simulácia Turingovho stroja na PRAMe

- overiť:

- 1 je postupnosť stavov určená postupnosťou inštrukcií uskutočniteľná? Inštrukcia r_0 začína v q_0 , ak inštrukcia r_{i-1} prevedie M do stavu q_i , tak inštrukcia r_i začína v stave q_i .
- 2 pre každú z k pásov a každý krok t , $0 \leq t < T(n)$:
 - 1 Ak v kroku t hlava navštívila políčko prvýkrát, tak symbol čítaný inštrukciou r_t musí byť symbol, ktorý tam bol v počiatočnej konfigurácii TS M .

Simulácia Turingovho stroja na PRAMe

- overiť:

- 1 je postupnosť stavov určená postupnosťou inštrukcií uskutočniteľná? Inštrukcia r_0 začína v q_0 , ak inštrukcia r_{i-1} prevedie M do stavu q_i , tak inštrukcia r_i začína v stave q_i .
- 2 pre každú z k pásov a každý krok t , $0 \leq t < T(n)$:
 - 1 Ak v kroku t hlava navštívila políčko prvý krát, tak symbol čítaný inštrukciou r_t musí byť symbol, ktorý tam bol v počiatočnej konfigurácii TS M .
 - 2 Ak hlava navštívila toto políčko pred okamžikom t naposledy v čase s , tak symbol zapísaný inštrukciou r_s sa musí zhodovať so symbolom čítaným inštrukciou r_t .

Simulácia Turingovho stroja na PRAMe

- overiť:

- 1 je postupnosť stavov určená postupnosťou inštrukcií uskutočniteľná? Inštrukcia r_0 začína v q_0 , ak inštrukcia r_{i-1} prevedie M do stavu q_i , tak inštrukcia r_i začína v stave q_i .
- 2 pre každú z k pásov a každý krok t , $0 \leq t < T(n)$:
 - 1 Ak v kroku t hlava navštívila políčko prvýkrát, tak symbol čítaný inštrukciou r_t musí byť symbol, ktorý tam bol v počiatočnej konfigurácii TS M .
 - 2 Ak hlava navštívila toto políčko pred okamžikom t naposledy v čase s , tak symbol zapísaný inštrukciou r_s sa musí zhodovať so symbolom čítaným inštrukciou r_t .

Simulácia Turingovho stroja na PRAMe

- overiť:

- 1 je postupnosť stavov určená postupnosťou inštrukcií uskutočniteľná? Inštrukcia r_0 začína v q_0 , ak inštrukcia r_{i-1} prevedie M do stavu q_i , tak inštrukcia r_i začína v stave q_i .
- 2 pre každú z k pásov a každý krok t , $0 \leq t < T(n)$:
 - 1 Ak v kroku t hlava navštívila políčko prvý krát, tak symbol čítaný inštrukciou r_t musí byť symbol, ktorý tam bol v počiatočnej konfigurácii TS M .
 - 2 Ak hlava navštívila toto políčko pred okamžikom t naposledy v čase s , tak symbol zapísaný inštrukciou r_s sa musí zhodovať so symbolom čítaným inštrukciou r_t .

Ako spočítať – funkcia [prev](#).

Simulácia Turingovho stroja na PRAMe

- overiť:

- 1 je postupnosť stavov určená postupnosťou inštrukcií uskutočniteľná? Inštrukcia r_0 začína v q_0 , ak inštrukcia r_{i-1} prevedie M do stavu q_i , tak inštrukcia r_i začína v stave q_i .
- 2 pre každú z k pásov a každý krok t , $0 \leq t < T(n)$:
 - 1 Ak v kroku t hlava navštívila políčko prvý krát, tak symbol čítaný inštrukciou r_t musí byť symbol, ktorý tam bol v počiatočnej konfigurácii TS M .
 - 2 Ak hlava navštívila toto políčko pred okamžikom t naposledy v čase s , tak symbol zapísaný inštrukciou r_s sa musí zhodovať so symbolom čítaným inštrukciou r_t .

Ako spočítať – funkcia [prev](#).

- práve jeden tím zistí, že jeho postupnosť inštrukcií je správna, zistí, či stav dosiahnutý v r_t je koncový. Výsledok zapíše manažér tímu do spoločnej pamäte.

Simulácia Turingovho stroja na PRAMe

- | Počet proc. | Čas | Šírka slova | Model |
|---------------|--------|-------------|--------|
| $2^{O(T(n))}$ | $O(1)$ | $O(T(n))$ | COMMON |

Simulácia Turingovho stroja na PRAMe

- | Počet proc. | Čas | Šírka slova | Model |
|---------------|--------|-------------|--------|
| $2^{O(T(n))}$ | $O(1)$ | $O(T(n))$ | COMMON |

- ľahko sa dá zovšeobecniť pre TS počítajúce funkciu – využije sa `last()`

Simulácia Turingovho stroja na PRAMe

- | Počet proc. | Čas | Šírka slova | Model |
|---------------|--------|-------------|--------|
| $2^{O(T(n))}$ | $O(1)$ | $O(T(n))$ | COMMON |

- ľahko sa dá zovšeobecniť pre TS počítajúce funkciu – využije sa `last()`
- zrejme nereálny výsledok

Simulácia Turingovho stroja na PRAMe

- | Počet proc. | Čas | Šírka slova | Model |
|---------------|--------|-------------|--------|
| $2^{O(T(n))}$ | $O(1)$ | $O(T(n))$ | COMMON |

- ľahko sa dá zovšeobecniť pre TS počítajúce funkciu – využije sa `last()`
- zrejme nereálny výsledok
- ak šírka slova $W(n) = T(n)^{O(1)}$, tak by PRAM mohol spĺňať paralelnú tézu – BUDE NA PREDNÁŠKE NESKÔR! Ukážeme, že potom je možné dosiahnuť ľubovoľné polynomické zrýchlenie.

Simulácia Turingovho stroja na PRAMe s obmedzenou šírkou slova

Veta

Nech funkcia $B : \mathbb{N} \rightarrow \mathbb{N}$ je *ws-konstruovateľná* a PRAM so šírkou slova $W(n)$ môže simulovať ľubovoľný $B(n)$ časovo obmedzený deterministický TS v čase $C(n)$. Potom PRAM so šírkou slova $O(W(n) + B(n) + \log T(n))$ môže simulovať ľubovoľný $T(n)$ časovo obmedzený deterministický TS v čase

$$O\left(\frac{T(n)}{B(n)} + C(n)\right).$$

Simulácia Turingovho stroja na PRAMe s obmedzenou šírkou slova

Veta

Nech funkcia $B : \mathbb{N} \rightarrow \mathbb{N}$ je *ws-konstruovateľná* a PRAM so šírkou slova $W(n)$ môže simulovať ľubovoľný $B(n)$ časovo obmedzený deterministický TS v čase $C(n)$. Potom PRAM so šírkou slova $O(W(n) + B(n) + \log T(n))$ môže simulovať ľubovoľný $T(n)$ časovo obmedzený deterministický TS v čase

$$O\left(\frac{T(n)}{B(n)} + C(n)\right).$$

Dôkaz:

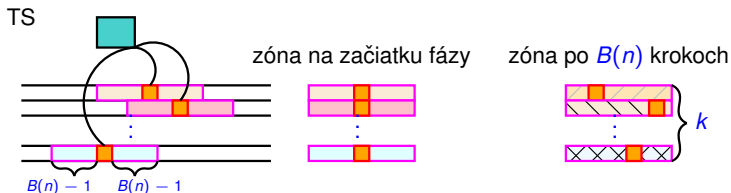
- Nech je daný $T(n)$ časovo obmedzený Turingov stroj $M = (Q, \Sigma, \delta, q_0, F)$

Simulácia Turingovho stroja na PRAMe s obmedzenou šírkou slova

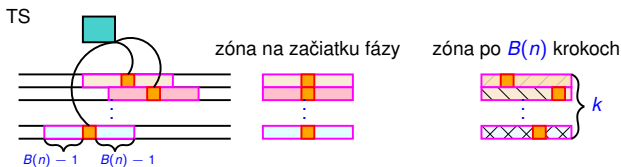
- Simulácia bude mať $O(\lceil \frac{T(n)}{B(n)} \rceil)$ fáz, z ktorých každá bude zodpovedať $B(n)$ krokom výpočtu stroja M . Budeme počítat konfigurácie na konci každej fázy.

Simulácia Turingovho stroja na PRAME s obmedzenou šírkou slova

- Simulácia bude mať $O(\lceil \frac{T(n)}{B(n)} \rceil)$ fáz, z ktorých každá bude zodpovedať $B(n)$ krokom výpočtu stroja M . Budeme počítat konfigurácie na konci každej fázy.
- **zóna** – časť pásky, ktorá sa môže zmeniť počas jednej fázy, tj. $k(2B(n) - 1)$ políčok pásky, ktoré sú vo vzdialenosti max. $B(n)$ od pozície hlavy na začiatku fázy.

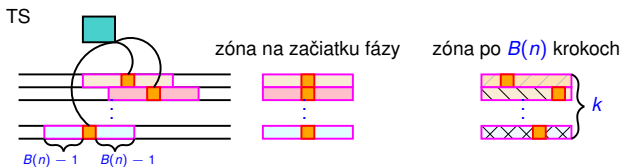


Simulácia Turingovho stroja na PRAMe s obmedzenou šírkou slova



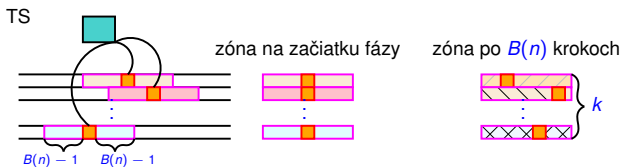
- V priebehu fázy sa simulácia riadi iba obsahom zóny. Na konci fázy sa obsah zóny aktualizuje.

Simulácia Turingovho stroja na PRAMe s obmedzenou šírkou slova



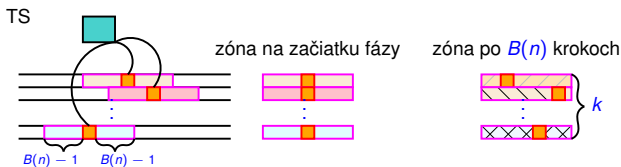
- V priebehu fázy sa simulácia riadi iba obsahom zóny. Na konci fázy sa obsah zóny aktualizuje.
- *predvýpočet*: tabuľka s $2^{O(B(n))}$ položkami.

Simulácia Turingovho stroja na PRAMe s obmedzenou šírkou slova



- V priebehu fázy sa simulácia riadi iba obsahom zóny. Na konci fázy sa obsah zóny aktualizuje.
- *predvýpočet*: tabuľka s $2^{O(B(n))}$ položkami.
 Index – obsah zóny na začiatku fázy,

Simulácia Turingovho stroja na PRAMe s obmedzenou šírkou slova

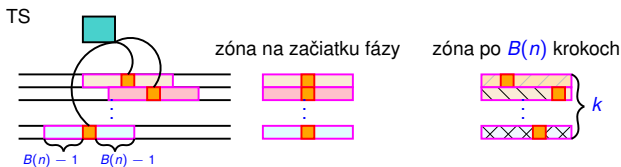


- V priebehu fázy sa simulácia riadi iba obsahom zóny. Na konci fázy sa obsah zóny aktualizuje.
- *predvýpočet*: tabuľka s $2^{O(B(n))}$ položkami.

Index – obsah zóny na začiatku fázy,

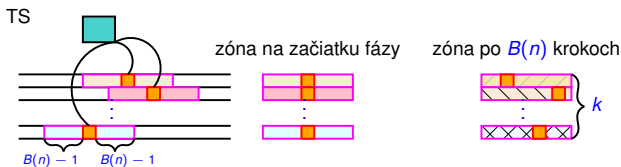
obsah políčka – obsah zóny na konci fázy (tj. po $B(n)$ krokoch výpočtu TS M).

Simulácia Turingovho stroja na PRAMe s obmedzenou šírkou slova



- V priebehu fázy sa simulácia riadi iba obsahom zóny. Na konci fázy sa obsah zóny aktualizuje.
- *predvýpočet*: tabuľka s $2^{O(B(n))}$ položkami.
 - Index** – obsah zóny na začiatku fázy,
 - obsah políčka** – obsah zóny na konci fázy (tj. po $B(n)$ krokoch výpočtu TS M).
- Na predvýpočet je treba šírka slova $O(W(n))$ bitov a čas $C(n)$.

Simulácia Turingovho stroja na PRAMe s obmedzenou šírkou slova



- V priebehu fázy sa simulácia riadi iba obsahom zóny. Na konci fázy sa obsah zóny aktualizuje.
- *predvýpočet*: tabuľka s $2^{O(B(n))}$ položkami.

Index – obsah zóny na začiatku fázy,

obsah políčka – obsah zóny na konci fázy (tj. po $B(n)$ krokoch výpočtu TS M).

- Na predvýpočet je treba šírka slova $O(W(n))$ bitov a čas $C(n)$.
- dá sa zovšeobecniť na TS počítajúci funkciu: využijeme funkciu **last**

Simulácia Turingovho stroja na PRAMe s obmedzenou šírkou slova

- Na začiatku fázy určíme obsah zóny “vykrojením” úsekov pásov do vzdialenosti $B(n) - 1$ od polôh hláv. Z tabuľky zistíme obsah zóny po $B(n)$ krokoch. Aktualizujeme obsah zóny a polohy hláv na konci fázy.

Simulácia Turingovho stroja na PRAMe s obmedzenou šírkou slova

- Na začiatku fázy určíme obsah zóny “vykrojením” úsekov pásov do vzdialenosti $B(n) - 1$ od polôh hláv. Z tabuľky zistíme obsah zóny po $B(n)$ krokoch. Aktualizujeme obsah zóny a polohy hláv na konci fázy.
- $T(n)$ krokov stroja M vyžaduje $\lceil \frac{T(n)}{B(n)} \rceil$ fáz. časová zložitosť:

Celkom $O(C(n) + \frac{T(n)}{B(n)})$.

Simulácia Turingovho stroja na PRAMe s obmedzenou šírkou slova

- Na začiatku fázy určíme obsah zóny “vykrojením” úsekov pásov do vzdialenosti $B(n) - 1$ od polôh hláv. Z tabuľky zistíme obsah zóny po $B(n)$ krokoch. Aktualizujeme obsah zóny a polohy hláv na konci fázy.
- $T(n)$ krokov stroja M vyžaduje $\lceil \frac{T(n)}{B(n)} \rceil$ fáz. časová zložitosť:
 - Predvýpočet: $O(C(n))$.

Celkom $O(C(n) + \frac{T(n)}{B(n)})$.

Simulácia Turingovho stroja na PRAMe s obmedzenou šírkou slova

- Na začiatku fázy určíme obsah zóny “vykrojením” úsekov pásov do vzdialenosti $B(n) - 1$ od polôh hláv. Z tabuľky zistíme obsah zóny po $B(n)$ krokoch. Aktualizujeme obsah zóny a polohy hláv na konci fázy.
- $T(n)$ krokov stroja M vyžaduje $\lceil \frac{T(n)}{B(n)} \rceil$ fáz. časová zložitosť:
 - Predvýpočet: $O(C(n))$.
 - Každá fáza: $O(1)$ (obmedzená sada aritmetických inštrukcií).

Celkom $O(C(n) + \frac{T(n)}{B(n)})$.

Simulácia Turingovho stroja na PRAMe s obmedzenou šírkou slova

- Na začiatku fázy určíme obsah zóny “vykrojením” úsekov pásov do vzdialenosti $B(n) - 1$ od polôh hláv. Z tabuľky zistíme obsah zóny po $B(n)$ krokoch. Aktualizujeme obsah zóny a polohy hláv na konci fázy.
- $T(n)$ krokov stroja M vyžaduje $\lceil \frac{T(n)}{B(n)} \rceil$ fáz. časová zložitosť:
 - Predvýpočet: $O(C(n))$.
 - Každá fáza: $O(1)$ (obmedzená sada aritmetických inštrukcií).
 Celkom $O(C(n) + \frac{T(n)}{B(n)})$.
- Šírka slova je maximum z:

Simulácia Turingovho stroja na PRAMe s obmedzenou šírkou slova

- Na začiatku fázy určíme obsah zóny “vykrojením” úsekov pások do vzdialenosti $B(n) - 1$ od polôh hláv. Z tabuľky zistíme obsah zóny po $B(n)$ krokoch. Aktualizujeme obsah zóny a polohy hláv na konci fázy.
- $T(n)$ krokov stroja M vyžaduje $\lceil \frac{T(n)}{B(n)} \rceil$ fáz. časová zložitosť:
 - Predvýpočet: $O(C(n))$.
 - Každá fáza: $O(1)$ (obmedzená sada aritmetických inštrukcií).

Celkom $O(C(n) + \frac{T(n)}{B(n)})$.

- Šírka slova je maximum z:
 - $O(W(n) + B(n))$ – pri predvýpočte,

Simulácia Turingovho stroja na PRAMe s obmedzenou šírkou slova

- Na začiatku fázy určíme obsah zóny “vykrojením” úsekov pások do vzdialenosti $B(n) - 1$ od polôh hláv. Z tabuľky zistíme obsah zóny po $B(n)$ krokoch. Aktualizujeme obsah zóny a polohy hláv na konci fázy.
- $T(n)$ krokov stroja M vyžaduje $\lceil \frac{T(n)}{B(n)} \rceil$ fáz. časová zložitosť:
 - Predvýpočet: $O(C(n))$.
 - Každá fáza: $O(1)$ (obmedzená sada aritmetických inštrukcií).
 Celkom $O(C(n) + \frac{T(n)}{B(n)})$.
- Šírka slova je maximum z:
 - $O(W(n) + B(n))$ – pri predvýpočte,
 - $O(B(n))$ – na identifikačné čísla procesorov pre všetky fázy.

Simulácia Turingovho stroja na PRAMe s obmedzenou šírkou slova

- Na začiatku fázy určíme obsah zóny “vykrojením” úsekov pások do vzdialenosti $B(n) - 1$ od polôh hláv. Z tabuľky zistíme obsah zóny po $B(n)$ krokoch. Aktualizujeme obsah zóny a polohy hláv na konci fázy.
- $T(n)$ krokov stroja M vyžaduje $\lceil \frac{T(n)}{B(n)} \rceil$ fáz. časová zložitosť:
 - Predvýpočet: $O(C(n))$.
 - Každá fáza: $O(1)$ (obmedzená sada aritmetických inštrukcií).

Celkom $O(C(n) + \frac{T(n)}{B(n)})$.

- Šírka slova je maximum z:
 - $O(W(n) + B(n))$ – pri predvýpočte,
 - $O(B(n))$ – na identifikačné čísla procesorov pre všetky fázy.
 - $O(\log T(n))$ – na uloženie polôh hláv v konfigurácii.

Simulácia Turingovho stroja na PRAMe spíňajúcom paralelnú tézu

Simulácia Turingovho stroja na PRAMe spíňajúcom paralelnú tézu

Veta

Nech funkcia $B : \mathbb{N} \rightarrow \mathbb{N}$ je *ws-konstruovateľná* funkcia. $T(n)$ časovo obmedzený det. TS sa dá simulovať v čase $O\left(\frac{T(n)}{B(n)}\right)$ s PRAMom so šírkou slova $O(B(n) + \log T(n))$.

Simulácia Turingovho stroja na PRAMe spíňajúcom paralelnú tézu

Veta

Nech funkcia $B : \mathbb{N} \rightarrow \mathbb{N}$ je *ws-konstruovateľná* funkcia. $T(n)$ časovo obmedzený det. TS sa dá simulovať v čase $O\left(\frac{T(n)}{B(n)}\right)$ s PRAMom so šírkou slova $O(B(n) + \log T(n))$.

Dôkaz: Ľubovoľný $B(n)$ časovo obmedzený TS sa dá simulovať v konštantnom čase na PRAMe so šírkou slova $O(B(n)) \Rightarrow$ dosadíme do predchádzajúcej vety. ■

Ľubovoľné polynomicke zrýchlenie je možné na PRAMe spíňajúcom paralelnú tézu

- Ak $T(n)$ je ws -konštruovateľná, tak aj $B(n) = T(n)^\varepsilon$ je ws -konštruovateľná.

Ľubovoľné polynomicke zrýchlenie je možné na PRAMe spíňajúcom paralelnú tézu

- Ak $T(n)$ je ws -konštruovateľná, tak aj $B(n) = T(n)^\varepsilon$ je ws -konštruovateľná.

Ľubovoľné polynomicke zrýchlenie je možné na PRAMe spĺňajúcom paralelnú tézu

- Ak $T(n)$ je *ws*-konštruovateľná, tak aj $B(n) = T(n)^\varepsilon$ je *ws*-konštruovateľná.

Dôsledok

*Ak je $T(n)$ *ws*-konštruovateľná funkcia, potom $T(n)$ časovo obmedzený det. TS sa dá simulovať v čase $O(T(n)^{1-\varepsilon})$, pre ľubovoľné reálne číslo $\varepsilon > 0$, PRAMom spĺňajúcom paralelnú tézu.*

Ľubovoľné polynomicke zrýchlenie je možné na PRAMe spĺňajúcom paralelnú tézu

- Ak $T(n)$ je *ws*-konštruovateľná, tak aj $B(n) = T(n)^\varepsilon$ je *ws*-konštruovateľná.

Dôsledok

*Ak je $T(n)$ *ws*-konštruovateľná funkcia, potom $T(n)$ časovo obmedzený det. TS sa dá simulovať v čase $O(T(n)^{1-\varepsilon})$, pre ľubovoľné reálne číslo $\varepsilon > 0$, PRAMom spĺňajúcom paralelnú tézu.*

- (Pozn. To že PRAM so šírkou slova $W(n) = T(n)^\varepsilon$ počítajúci v čase $O(T(n)^{1-\varepsilon})$ spĺňa paralelnú tézu – BUDE NA PREDNÁŠKE NESKÔR!)

Ľubovoľné polynomicke zrýchlenie je možné na PRAMe spĺňajúcom paralelnú tézu

- Ak $T(n)$ je *ws*-konštruovateľná, tak aj $B(n) = T(n)^\varepsilon$ je *ws*-konštruovateľná.

Dôsledok

Ak je $T(n)$ *ws*-konštruovateľná funkcia, potom $T(n)$ časovo obmedzený det. TS sa dá simulovať v čase $O(T(n)^{1-\varepsilon})$, pre ľubovoľné reálne číslo $\varepsilon > 0$, PRAMom spĺňajúcom paralelnú tézu.

- (Pozn. To že PRAM so šírkou slova $W(n) = T(n)^\varepsilon$ počítajúci v čase $O(T(n)^{1-\varepsilon})$ spĺňa paralelnú tézu – BUDE NA PREDNÁŠKE NESKÔR!)
- Nevieme, či je možné exponenciálne zrýchlenie (polylogaritmický čas) pre úlohy z \mathbb{P} , ale **ľubovoľné polynomicke zrýchlenie je možné!**

Paralelná téza

“paralelný čas” \approx “sekvenčný priestor”

- 1 Aby PRAM spĺňal paralelnú tézu, tak je ho treba obmedziť

Paralelná téza

“paralelný čas” \approx “sekvenčný priestor”

- 1 Aby PRAM spĺňal paralelnú tézu, tak je ho treba obmedziť
 - šírku slova

Paralelná téza

“paralelný čas” \approx “sekvenčný priestor”

- 1 Aby PRAM spĺňal paralelnú tézu, tak je ho treba obmedziť
 - šírku slova
 - sadu aritmetických inštrukcií

Paralelná téza

“paralelný čas” \approx “sekvenčný priestor”

- 1 Aby PRAM spĺňal paralelnú tézu, tak je ho treba obmedziť
 - šírku slova
 - sadu aritmetických inštrukcií
 - počet procesorov

Paralelná téza

“paralelný čas” \approx “sekvenčný priestor”

- 1 Aby PRAM spĺňal paralelnú tézu, tak je ho treba obmedziť
 - šírku slova
 - sadu aritmetických inštrukcií
 - počet procesorov

- 2 Paralelná téza ohraničuje, aké zrýchlenie sa dá dosiahnuť na rozumnom modele.

Napr. to čo sa dá vyriešiť na “rozumnom modele” v čase $\log^{O(1)} n$, tak sa dá vyriešiť sekvenčne v polylogaritmickej priestore (a naopak)

Tranzitívny uzáver grafu

GAP – Graph Accessibility Problem

Vstup: orientovaný graf $G = (V, E)$, $V = \{1, \dots, n\}$ je množina vrcholov, E je množina orientovaných hrán zadaná maticou susednosti $A[i, j] = \text{true} \Leftrightarrow (i, j) \in E$.

Tranzitívny uzáver grafu

GAP – Graph Accessibility Problem

Vstup: orientovaný graf $G = (V, E)$, $V = \{1, \dots, n\}$ je množina vrcholov, E je množina orientovaných hrán zadaná maticou susednosti $A[i, j] = \text{true} \Leftrightarrow (i, j) \in E$.

Výstup: graf tranzitívneho uzáveru $G' = (V, E')$, kde $(x, y) \in E' \Leftrightarrow$ v G existuje cesta z x do y .

Tranzitívny uzáver grafu

GAP – Graph Accessibility Problem

Vstup: orientovaný graf $G = (V, E)$, $V = \{1, \dots, n\}$ je množina vrcholov, E je množina orientovaných hrán zadaná maticou susednosti $A[i, j] = \text{true} \Leftrightarrow (i, j) \in E$.

Výstup: graf tranzitívneho uzáveru $G' = (V, E')$, kde $(x, y) \in E' \Leftrightarrow$ v G existuje cesta z x do y .

Tranzitívny uzáver grafu

GAP – Graph Accessibility Problem

Vstup: orientovaný graf $G = (V, E)$, $V = \{1, \dots, n\}$ je množina vrcholov, E je množina orientovaných hrán zadaná maticou susednosti $A[i, j] = \text{true} \Leftrightarrow (i, j) \in E$.

Výstup: graf tranzitívneho uzáveru $G' = (V, E')$, kde $(x, y) \in E' \Leftrightarrow$ v G existuje cesta z x do y .

Alg.: • n^3 procesorov

PID:

i	j	k
-----	-----	-----

Tranzitívny uzáver grafu

GAP – Graph Accessibility Problem

Vstup: orientovaný graf $G = (V, E)$, $V = \{1, \dots, n\}$ je množina vrcholov, E je množina orientovaných hrán zadaná maticou susednosti $A[i, j] = \text{true} \Leftrightarrow (i, j) \in E$.

Výstup: graf tranzitívneho uzáveru $G' = (V, E')$, kde $(x, y) \in E' \Leftrightarrow$ v G existuje cesta z x do y .

Alg.: • n^3 procesorov

PID:

i	j	k
-----	-----	-----

- $i := \lfloor \frac{PID}{n^2} \rfloor$;
- $j := \lfloor \frac{PID \bmod n^2}{n} \rfloor$;
- $k := PID \bmod n$;
- $A[i, i] := \text{true}$;
- $L := 1$;
- while $L < n$ do
 - begin
 - if $A[j, k]$ and $A[k, j]$ then $A[j, j] := \text{true}$;
 - $L := L * 2$;
 - end;

Tranzitívny uzáver grafu

GAP – Graph Accessibility Problem

- Na konci t -tej iterácie cyklu je $L = 2^t$ a $\forall 0 \leq i, j < n$ platí
 $A[i, j] = \text{true} \Leftrightarrow \exists$ v G cesta z i do j dĺžky max. L
(dokáže sa indukciou podľa t)

Tranzitívny uzáver grafu

GAP – Graph Accessibility Problem

- Na konci t -tej iterácie cyklu je $L = 2^t$ a $\forall 0 \leq i, j < n$ platí
 $A[i, j] = \text{true} \Leftrightarrow \exists$ v G cesta z i do j dĺžky max. L
 (dokáže sa indukciou podľa t)

Počet proc.	Čas	Šírka slova	Model
n^3	$O(\log n)$	$O(\log n)$	COMMON

Tranzitívny uzáver grafu

GAP – Graph Accessibility Problem

- Na konci t -tej iterácie cyklu je $L = 2^t$ a $\forall 0 \leq i, j < n$ platí
 $A[i, j] = \text{true} \Leftrightarrow \exists$ v G cesta z i do j dĺžky max. L
 (dokáže sa indukciou podľa t)

Počet proc.	Čas	Šírka slova	Model
n^3	$O(\log n)$	$O(\log n)$	COMMON

- dokonca s minimálnou sadou aritmetických inštrukcií (+, =, div2)**

Simulácia TS na PRAMe

Veta (Goldschlager)

*Predpokladajme, že $S(n) = \Omega(\log n)$ sa dá spočítať PRAMom s minimálnou sadou aritmetických inštrukcií, šírkou slova $O(S(n))$ v čase $O(S(n))$ z jediného vstupu n . Potom $S(n)$ priestorovo obmedzený **nedeterministický** TS môže byť simulovaný v čase $O(S(n))$ na PRAMe s minimálnou sadou aritmetických inštrukcií a šírkou slova $O(S(n))$.*

Simulácia TS na PRAMe

Veta (Goldschlager)

*Predpokladajme, že $S(n) = \Omega(\log n)$ sa dá spočítať PRAMom s minimálnou sadou aritmetických inštrukcií, šírkou slova $O(S(n))$ v čase $O(S(n))$ z jediného vstupu n . Potom $S(n)$ priestorovo obmedzený **nedeterministický** TS môže byť simulovaný v čase $O(S(n))$ na PRAMe s minimálnou sadou aritmetických inštrukcií a šírkou slova $O(S(n))$.*

Dôkaz:

- Predpokladáme, že TS M rozhoduje nejaký jazyk a má jedinú prijímaciu konfiguráciu.

Simulácia TS na PRAMe

Veta (Goldschlager)

Predpokladajme, že $S(n) = \Omega(\log n)$ sa dá spočítať PRAMom s minimálnou sadou aritmetických inštrukcií, šírkou slova $O(S(n))$ v čase $O(S(n))$ z jediného vstupu n . Potom $S(n)$ priestorovo obmedzený **nedeterministický** TS môže byť simulovaný v čase $O(S(n))$ na PRAMe s minimálnou sadou aritmetických inštrukcií a šírkou slova $O(S(n))$.

Dôkaz:

- Predpokladáme, že TS M rozhoduje nejaký jazyk a má jedinú prijímaciu konfiguráciu.
- $S(n)$ priestorovo obmedzený TS má maximálne $2^{O(S(n))+\log n} = 2^{O(S(n))}$ konfigurácií. Skonstruujeme graf G možných výpočtov: uzly budú konfigurácie, z konfigurácie K_1 do konfigurácie K_2 vedie hrana práve vtedy, keď $K_1 \vdash_M K_2$.

Simulácia TS na PRAMe

Veta (Goldschlager)

Predpokladajme, že $S(n) = \Omega(\log n)$ sa dá spočítať PRAMom s minimálnou sadou aritmetických inštrukcií, šírkou slova $O(S(n))$ v čase $O(S(n))$ z jediného vstupu n . Potom $S(n)$ priestorovo obmedzený **nedeterministický** TS môže byť simulovaný v čase $O(S(n))$ na PRAMe s minimálnou sadou aritmetických inštrukcií a šírkou slova $O(S(n))$.

Dôkaz:

- Predpokladáme, že TS M rozhoduje nejaký jazyk a má jedinú prijímaciu konfiguráciu.
- $S(n)$ priestorovo obmedzený TS má maximálne $2^{O(S(n))+\log n} = 2^{O(S(n))}$ konfigurácií. Skonstruujeme graf G možných výpočtov: uzly budú konfigurácie, z konfigurácie K_1 do konfigurácie K_2 vedie hrana práve vtedy, keď $K_1 \vdash_M K_2$.
- M prijíma vstup $x \Leftrightarrow$ v G existuje cesta z počiatkovej konfigurácie do koncovkej konfigurácie – algoritmus na riešenie GAP.

Simulácia TS na PRAMe

Veta (Goldschlager)

Predpokladajme, že $S(n) = \Omega(\log n)$ sa dá spočítať PRAMom s minimálnou sadou aritmetických inštrukcií, šírkou slova $O(S(n))$ v čase $O(S(n))$ z jediného vstupu n . Potom $S(n)$ priestorovo obmedzený **nedeterministický** TS môže byť simulovaný v čase $O(S(n))$ na PRAMe s minimálnou sadou aritmetických inštrukcií a šírkou slova $O(S(n))$.

Dôkaz:

- Predpokladáme, že TS M rozhoduje nejaký jazyk a má jedinú prijímaciu konfiguráciu.
- $S(n)$ priestorovo obmedzený TS má maximálne $2^{O(S(n))+\log n} = 2^{O(S(n))}$ konfigurácií. Skonstruujeme graf G možných výpočtov: uzly budú konfigurácie, z konfigurácie K_1 do konfigurácie K_2 vedie hrana práve vtedy, keď $K_1 \vdash_M K_2$.
- M prijíma vstup $x \Leftrightarrow$ v G existuje cesta z počiatkovej konfigurácie do koncovkej konfigurácie – algoritmus na riešenie GAP.
- Časová zložitosť: $\log 2^{O(S(n))} = O(S(n))$, šírka slova tá istá.

Simulácia TS na PRAMe

Keď funkciu priestorovej zložitosti nevieme odhadnúť

Dôsledok

$S(n)$ priestorovo obmedzený **deterministický** TS sa dá simulovať PRAMom s minimálnou sadou aritmetických inštrukcií a šírkou slova $O(S(n))$ v čase $O(S(n) \log S(n))$

Simulácia TS na PRAMe

Keď funkciu priestorovej zložitosti nevieme odhadnúť

Dôsledok

$S(n)$ priestorovo obmedzený **deterministický** TS sa dá simulovať PRAMom s minimálnou sadou aritmetických inštrukcií a šírkou slova $O(S(n))$ v čase $O(S(n) \log S(n))$

Dôkaz:

- Funkcia $S(n)$ sa nemusí dať dobre počítať. Stačí, že poznáme, keď TS dopočíta. Skúšame postup z predchádzajúceho dôkazu pre $S(n) = 1, 2, 4, 8, \dots$

Simulácia PRAMu na Turingovom stroji

Veta (Goldschlager)

$T(n)$ časovo obmedzený PRAM M so šírkou slova $W(n)$ môže byť simulovaný deterministickým TS v priestore $T(n)(W(n) + \log T(n)) + S(n)$, kde $S(n)$ je priestor, ktorý potrebuje TS na simuláciu jednej inštrukcie jedného procesoru PRAM.

Simulácia PRAMu na Turingovom stroji

Veta (Goldschlager)

$T(n)$ časovo obmedzený PRAM M so šírkou slova $W(n)$ môže byť simulovaný deterministickým TS v priestore $T(n)(W(n) + \log T(n)) + S(n)$, kde $S(n)$ je priestor, ktorý potrebuje TS na simuláciu jednej inštrukcie jedného procesoru PRAM.

Dôkaz:

- použije sa PRIORITY PRAM

Simulácia PRAMu na Turingovom stroji

Veta (Goldschlager)

$T(n)$ časovo obmedzený PRAM M so šírkou slova $W(n)$ môže byť simulovaný deterministickým TS v priestore $T(n)(W(n) + \log T(n)) + S(n)$, kde $S(n)$ je priestor, ktorý potrebuje TS na simuláciu jednej inštrukcie jedného procesoru PRAM.

Dôkaz:

- použije sa PRIORITY PRAM
- procesory nebudú používať lokálne pamäte

Simulácia PRAMu na Turingovom stroji

Veta (Goldschlager)

$T(n)$ časovo obmedzený PRAM M so šírkou slova $W(n)$ môže byť simulovaný deterministickým TS v priestore $T(n)(W(n) + \log T(n)) + S(n)$, kde $S(n)$ je priestor, ktorý potrebuje TS na simuláciu jednej inštrukcie jedného procesoru PRAM.

Dôkaz:

- použije sa PRIORITY PRAM
- procesory nebudú používať lokálne pamäte
- inštrukcie očíslované $1, 2, 3, \dots$

Simulácia PRAMu na Turingovom stroji

Veta (Goldschlager)

$T(n)$ časovo obmedzený PRAM M so šírkou slova $W(n)$ môže byť simulovaný deterministickým TS v priestore $T(n)(W(n) + \log T(n)) + S(n)$, kde $S(n)$ je priestor, ktorý potrebuje TS na simuláciu jednej inštrukcie jedného procesoru PRAM.

Dôkaz:

- použije sa PRIORITY PRAM
- procesory nebudú používať lokálne pamäte
- inštrukcie očíslované $1, 2, 3, \dots$
- 0-tý register je nulový počas výpočtu a je do neho vložené číslo

Potom sa už hodnota registru 0 nemení dokiaľ nedopočítajú všetky procesory.

Simulácia PRAMu na Turingovom stroji

Veta (Goldschlager)

$T(n)$ časovo obmedzený PRAM M so šírkou slova $W(n)$ môže byť simulovaný deterministickým TS v priestore $T(n)(W(n) + \log T(n)) + S(n)$, kde $S(n)$ je priestor, ktorý potrebuje TS na simuláciu jednej inštrukcie jedného procesoru PRAM.

Dôkaz:

- použije sa PRIORITY PRAM
- procesory nebudú používať lokálne pamäte
- inštrukcie očíslované $1, 2, 3, \dots$
- 0-tý register je nulový počas výpočtu a je do neho vložené číslo 1 ak M vstup prijíma, alebo

Potom sa už hodnota registru 0 nemení dokiaľ nedopočítajú všetky procesory.

Simulácia PRAMu na Turingovom stroji

Veta (Goldschlager)

$T(n)$ časovo obmedzený PRAM M so šírkou slova $W(n)$ môže byť simulovaný deterministickým TS v priestore $T(n)(W(n) + \log T(n)) + S(n)$, kde $S(n)$ je priestor, ktorý potrebuje TS na simuláciu jednej inštrukcie jedného procesoru PRAM.

Dôkaz:

- použije sa PRIORITY PRAM
- procesory nebudú používať lokálne pamäte
- inštrukcie očíslované $1, 2, 3, \dots$
- 0-tý register je nulový počas výpočtu a je do neho vložené číslo
 - 1 ak M vstup prijíma, alebo
 - 1 ak M vstup neprijíma.

Potom sa už hodnota registru 0 nemení dokiaľ nedopočítajú všetky procesory.

Simulácia PRAMu na Turingovom stroji

```

function instruction( $i, t$ )
  {vracia hodnotu PC  $i$ -teho procesoru v čase  $t$ }
  begin if  $t = 0$  then return 1
    else begin
       $h :=$  instruction( $i, t - 1$ );
      case  $h$ -ta inštrukcia of
        “goto  $m$  if  $\bar{r}_j > 0$ ”: if register( $j, t - 1$ )  $> 0$  then return  $m$ 
                                     else return  $h + 1$ ;
        “halt” : return  $h$ ;
        otherwise : return  $h + 1$ ;
      end
    end
  end;

function input( $i$ )
  {posuň čítaciu hlavu na  $i$ -te políčko a vráť znak, ktorý tam je}

```

Simulácia PRAMu na Turingovom stroji

```

function register( $j, t$ )
  {vracia obsah registru  $j$  v čase  $t$ }
  begin if  $t = 0$  then return (input( $j$ ))
    else {skontroluj, či tam niekto nezapisoval}
      begin winner :=  $P(n)$ ;
        for  $w := P(n) - 1$  downto 0 do
          begin
             $h :=$  instruction( $w, t - 1$ );
            if  $h$ -ta inštrukcia je zápis do  $\bar{r}_j$ 
              (( $\bar{r}_j := d$ ) or [( $\bar{r}_{\bar{r}_x} := d$ ) and (register( $r_x, t - 1$ )= $j$ )])
            then
              begin winner :=  $w$ ; value :=  $d$ ; end
            end
          end;
        if winner <  $P(n)$  then return value
          else return register( $j, t - 1$ )
      end;
  end;

```

Simulácia TS na PRAMe

- M prijíma vstup $\Leftrightarrow \text{register}(0, T(n))=1$.

Simulácia TS na PRAMe

- M prijíma vstup $\Leftrightarrow \text{register}(0, T(n))=1$.
- $T(n)$ sa možno ťažko počíta, tak skúšame $T(n) = 1, 2, 3, \dots$

Simulácia TS na PRAMe

- M prijíma vstup $\Leftrightarrow \text{register}(0, T(n))=1$.
- $T(n)$ sa možno ťažko počíta, tak skúšame $T(n) = 1, 2, 3, \dots$.
- výpočet $\text{register}(0, T(n))=1$ – rekurzia hĺbky $T(n)$; počet premenných použitých v jednej procedúre je konštantný, každá so šírkou slova buď $O(W(n))$ alebo $O(\log T(n))$,

Simulácia TS na PRAMe

- M prijíma vstup $\Leftrightarrow \text{register}(0, T(n))=1$.
- $T(n)$ sa možno ťažko počíta, tak skúšame $T(n) = 1, 2, 3, \dots$
- výpočet $\text{register}(0, T(n))=1$ – rekurzia hĺbky $T(n)$; počet premenných použitých v jednej procedúre je konštantný, každá so šírkou slova buď $O(W(n))$ alebo $O(\log T(n))$,
- $O(\log n)$ potrebujeme ako pointer na vstup. Celkovo pracovný priestor

$$T(n)(W(n) + \log T(n)) + S(n) + \log n,$$

kde $S(n)$ je priestor, v ktorom Turingov stroj nasimuluje vykonanie jednej inštrukcie jedného procesora. $\log n$ môžeme zanedbať, pretože $W(n) = \Omega(\log n)$ – taká šírka slova je nutná na adresovanie n vstupov.

Obmedzenie sily PRAMu do “rozumných medzí”

Jednotková cena inštrukcie:

- môžeme používať jednotkovú cenu takých inštrukcií, ktoré sa dajú simulovať TS v čase $T(n)^{O(1)}$, kde $T(n)$ je čas výpočtu PRAMu

Obmedzenie sily PRAMu do “rozumných medzí”

Jednotková cena inštrukcie:

- môžeme používať jednotkovú cenu takých inštrukcií, ktoré sa dajú simulovať TS v čase $T(n)^{O(1)}$, kde $T(n)$ je čas výpočtu PRAMu
 - ak to prijmeme, tak PRAM spĺňa paralelnú tézu, ak $W(n) = T(n)^{O(1)} \Rightarrow$ to umožňuje stále PRAM s $2^{T(n)^{O(1)}}$ procesormi

Obmedzenie sily PRAMu do “rozumných medzí”

Jednotková cena inštrukcie:

- môžeme používať jednotkovú cenu takých inštrukcií, ktoré sa dajú simulovať TS v čase $T(n)^{O(1)}$, kde $T(n)$ je čas výpočtu PRAMu
 - ak to prijmeme, tak PRAM spĺňa paralelnú tézu, ak $W(n) = T(n)^{O(1)} \Rightarrow$ to umožňuje stále PRAM s $2^{T(n)^{O(1)}}$ procesormi
 - preto zástancovia lenivej aktivácie trvajú na $P(n) = 2^{O(T(n))}$ alebo dokonca na $P(n) = n^{O(1)}$

Obmedzenie sily PRAMu do “rozumných medzí”

Používané obmedzenia:

- 1 Obmedzenia množiny inštrukcií (často preto, aby sa dala používať jednotková cena inštrukcií):

Obmedzenie sily PRAMu do “rozumných medzí”

Používané obmedzenia:

- 1 Obmedzenia množiny inštrukcií (často preto, aby sa dala používať jednotková cena inštrukcií):
 - a) Jednotlivé procesory nesmú používať inštrukcie, ktoré sa nedajú simulovať na det. TS v čase $T(n)^{O(1)}$, kde $T(n)$ je doba výpočtu na PRAMe.

Obmedzenie sily PRAMu do “rozumných medzí”

Používané obmedzenia:

- 1 Obmedzenia množiny inštrukcií (často preto, aby sa dala používať jednotková cena inštrukcií):
 - a) Jednotlivé procesory nesmú používať inštrukcie, ktoré sa nedajú simulovať na det. TS v čase $T(n)^{O(1)}$, kde $T(n)$ je doba výpočtu na PRAMe.
 - b) každá inštrukcia procesora sa dá simulovať TS v priestore $T(n)^{O(1)}$ – slabšie obmedzenie než a).

Obmedzenie sily PRAMu do “rozumných medzí”

Používané obmedzenia:

- 1 Obmedzenia množiny inštrukcií (často preto, aby sa dala používať jednotková cena inštrukcií):
 - a) Jednotlivé procesory nesmú používať inštrukcie, ktoré sa nedajú simulovať na det. TS v čase $T(n)^{O(1)}$, kde $T(n)$ je doba výpočtu na PRAMe.
 - b) každá inštrukcia procesora sa dá simulovať TS v priestore $T(n)^{O(1)}$ – slabšie obmedzenie než a).
- 2 Obmedzenie počtu procesorov a paralelného času (zabrániť simulácii v konštantnom čase):

Obmedzenie sily PRAMu do “rozumných medzí”

Používané obmedzenia:

- 1 Obmedzenia množiny inštrukcií (často preto, aby sa dala používať jednotková cena inštrukcií):
 - a) Jednotlivé procesory nesmú používať inštrukcie, ktoré sa nedajú simulovať na det. TS v čase $T(n)^{O(1)}$, kde $T(n)$ je doba výpočtu na PRAMe.
 - b) každá inštrukcia procesora sa dá simulovať TS v priestore $T(n)^{O(1)}$ – slabšie obmedzenie než a).
- 2 Obmedzenie počtu procesorov a paralelného času (zabrániť simulácii v konštantnom čase):
 - a) $P(n) < 2^{O(T(n))}$ – dôsledok lenivej aktivácie.

Obmedzenie sily PRAMu do “rozumných medzí”

Používané obmedzenia:

- 1 Obmedzenia množiny inštrukcií (často preto, aby sa dala používať jednotková cena inštrukcií):
 - a) Jednotlivé procesory nesmú používať inštrukcie, ktoré sa nedajú simulovať na det. TS v čase $T(n)^{O(1)}$, kde $T(n)$ je doba výpočtu na PRAMe.
 - b) každá inštrukcia procesora sa dá simulovať TS v priestore $T(n)^{O(1)}$ – slabšie obmedzenie než a).
- 2 Obmedzenie počtu procesorov a paralelného času (zabrániť simulácii v konštantnom čase):
 - a) $P(n) < 2^{O(T(n))}$ – dôsledok lenivej aktivácie.
 - b) $P(n) = n^{O(1)}$ a $T(n) = \log^{O(1)} n$ – NC.

Obmedzenie sily PRAMu do “rozumných medzí”

Používané obmedzenia:

- 1 Obmedzenia množiny inštrukcií (často preto, aby sa dala používať jednotková cena inštrukcií):
 - a) Jednotlivé procesory nesmú používať inštrukcie, ktoré sa nedajú simulovať na det. TS v čase $T(n)^{O(1)}$, kde $T(n)$ je doba výpočtu na PRAMe.
 - b) každá inštrukcia procesora sa dá simulovať TS v priestore $T(n)^{O(1)}$ – slabšie obmedzenie než a).
- 2 Obmedzenie počtu procesorov a paralelného času (zabrániť simulácii v konštantnom čase):
 - a) $P(n) < 2^{O(T(n))}$ – dôsledok lenivej aktivácie.
 - b) $P(n) = n^{O(1)}$ a $T(n) = \log^{O(1)} n$ – NC.
- 3 Obmedzenie šírky slova:

Obmedzenie sily PRAMu do “rozumných medzí”

Používané obmedzenia:

- 1 Obmedzenia množiny inštrukcií (často preto, aby sa dala používať jednotková cena inštrukcií):
 - a) Jednotlivé procesory nesmú používať inštrukcie, ktoré sa nedajú simulovať na det. TS v čase $T(n)^{O(1)}$, kde $T(n)$ je doba výpočtu na PRAMe.
 - b) každá inštrukcia procesora sa dá simulovať TS v priestore $T(n)^{O(1)}$ – slabšie obmedzenie než a).
- 2 Obmedzenie počtu procesorov a paralelného času (zabrániť simulácii v konštantnom čase):
 - a) $P(n) < 2^{O(T(n))}$ – dôsledok lenivej aktivácie.
 - b) $P(n) = n^{O(1)}$ a $T(n) = \log^{O(1)} n$ – NC.
- 3 Obmedzenie šírky slova:
 - a) $W(n) = O(T(n))$.

Obmedzenie sily PRAMu do “rozumných medzí”

Používané obmedzenia:

- 1 Obmedzenia množiny inštrukcií (často preto, aby sa dala používať jednotková cena inštrukcií):
 - a) Jednotlivé procesory nesmú používať inštrukcie, ktoré sa nedajú simulovať na det. TS v čase $T(n)^{O(1)}$, kde $T(n)$ je doba výpočtu na PRAMe.
 - b) každá inštrukcia procesora sa dá simulovať TS v priestore $T(n)^{O(1)}$ – slabšie obmedzenie než a).
- 2 Obmedzenie počtu procesorov a paralelného času (zabrániť simulácii v konštantnom čase):
 - a) $P(n) < 2^{O(T(n))}$ – dôsledok lenivej aktivácie.
 - b) $P(n) = n^{O(1)}$ a $T(n) = \log^{O(1)} n$ – NC.
- 3 Obmedzenie šírky slova:
 - a) $W(n) = O(T(n))$.
 - b) $W(n) = T(n)^{O(1)}$ – simulácia jednej inštrukcie viz. 1b).

Obmedzenie sily PRAMu do “rozumných medzí”

Používané obmedzenia:

- 1 Obmedzenia množiny inštrukcií (často preto, aby sa dala používať jednotková cena inštrukcií):
 - a) Jednotlivé procesory nesmú používať inštrukcie, ktoré sa nedajú simulovať na det. TS v čase $T(n)^{O(1)}$, kde $T(n)$ je doba výpočtu na PRAMe.
 - b) každá inštrukcia procesora sa dá simulovať TS v priestore $T(n)^{O(1)}$ – slabšie obmedzenie než a).
- 2 Obmedzenie počtu procesorov a paralelného času (zabrániť simulácii v konštantnom čase):
 - a) $P(n) < 2^{O(T(n))}$ – dôsledok lenivej aktivácie.
 - b) $P(n) = n^{O(1)}$ a $T(n) = \log^{O(1)} n$ – NC.
- 3 Obmedzenie šírky slova:
 - a) $W(n) = O(T(n))$.
 - b) $W(n) = T(n)^{O(1)}$ – simulácia jednej inštrukcie viz. 1b).
 - c) $W(n) = n^{O(1)}$ – širšie slovo – veľkosť vstupu nie je dobrou mierou, ak vstupné čísla môžu mať viac než polynomický počet bitov.