FGI-2 – Formale Grundlagen der Informatik II
Prozesse und Nebenläufigkeit

Aufgabenblatt 11 : Parallele Algorithmen

Abgabe am 22.1.07 Besprechung am 24.1.07.

Präsenzaufgabe 11 : (PRAM)

Erfordert eine direkte Umsetzung des Algorithmus 7.2 (paralleles Suchen) von Seite 291 eine EREW, eine CREW oder eine CRCW-PRAM?

Übungsaufgabe 11.1 :

Maximumssuche auf der PRAM. Betrachten Sie den Maximum-Algorithmus in Beispiel 7.15 für n^2 Prozessoren auf Seite 280.

von
5

a) Ersetzen Sie die Zeile 1 und die Zeile 3 durch Anweisungen, die das gleichzeitige Schreiben vermeiden (und natürlich den Algorithmus korrekt lassen)!

b) Fügen Sie zwischen Zeile 2 und 3 eine Zeile ein, die bewirkt, dass weiterhin das maximale Element in Zeile 3 angezeigt, aber zusätzlich in einer Zeile 4 die kleinste Position eines solchen Elementes angezeigt wird!

c) Wie groß ist die Zeitkomplexität $T(x)$ des Algorithmus im logarithmischen Maß, wenn x die Länge der Kodierung des Feldes $[x_1, x_2, \cdots, x_n]$ ist.

Übungsaufgabe 11.2 :

Präfixsumme. Der Algorithmus in Beispiel 7.18 (Seite 283) ist auch für andere Probleme so wichtig, dass er als Hardware realisiert wird. Er kann sogar für beliebige assoziative binäre Operationen $*$ formuliert werden, d.h. $b_{n+j} = a_n * \cdots * a_{n+j}$.

von
5

a) Beschreiben Sie sein Verhalten anhand des Beispiels $[a_n, a_{n+1}, \cdots, a_{2n-1}] = [3, 2, 4, 7, 6, 8, 1, 9]$, indem Sie einen vollständigen Binärbaum mit Blättern $(a_n, b_n), (a_{n+1}, b_{n+1}), \cdots, (a_{2n-1}, b_{2n-1})$ konstruieren und die darüberliegenden Knoten entsprechend anlegen (d.h. bis zur Wurzel (a_1, b_1)).

b) Zeichnen Sie die Werte für das Beispiel ein, und zeigen Sie, dass immer $b_{12} = a_8 + \cdots + a_{12}$ gilt!

c) Begründen Sie, warum $T(n) = O(\log n)$ ist!

d) Zeigen Sie, dass der Algorithmus durch die Anwendung von Korollar 7.13 optimal wird!

Bisher erreichbare Punktzahl: 115