

Übungen zu „Parallelität in funktionalen Programmiersprachen“, Sommer 2009

Nr. 2, Abgabe der Aufgaben: 5. Mai 2009 vor der Vorlesung

Aufgaben

2.1 Inhärente Parallelität

5 Punkte

Skizzieren Sie funktionale Implementierungen für folgende Problemstellungen:

- Matrixmultiplikation
- Sortierung von Objekten
- Wortsuche in Texten

Diskutieren Sie jeweils die inhärente Parallelität und Strategien zur Parallelisierung.

2.2 Summierte φ -Funktion

7 Punkte

Die Eulersche oder φ -Funktion gibt zu einer Zahl n an, wie viele Zahlen zwischen 0 und n zu n teilerfremd sind:

$$\varphi(n) := |\{k \in \mathbb{N} \mid k < n \wedge \text{ggT}(k, n) = 1\}|$$

Schreiben Sie ein Haskellprogramm `summePhi.hs`, welches bei Aufruf einen Parameter erwartet, als ganze Zahl n interpretiert und die Zahl $\sum_{k=1}^n \varphi(k)$ berechnet.

An welchen Stellen lassen sich Berechnungen (ggf. durch geeignete Umstrukturierung) parallelisieren?

Geben Sie eine aussichtsreiche Parallelisierung mit Hilfe der parallelen Kombinatoren `par` und `pseq` an.