

## 7. Übung zu “Parallelität in funktionalen Sprachen”, SS 2006

Abgabe schriftlicher Aufgaben: Do 8.Juni 2006 (vor der Vorlesung)

---

### Hinweise:

Wegen des Feiertags findet am Montag, dem 1.6. kein Tutorium statt.

---

## Aufgaben

### 7.1 Parallelisierung und Optimierung

7 Punkte

- (a) Parallelisieren Sie Ihr Programm `summePhi` aus Aufgabe 2.2 (oder die Version von der Vorlesungsseite) durch GpH-Annotationen `par` und `seq` (oder passende Strategien). Strukturieren Sie die Berechnung so um, dass die parallelen Teilaufgaben nicht zu klein werden und möglichst unabhängig voneinander bleiben.
- (b) Simulieren Sie Ihr Programm mit GRANSIM auf verschiedenen Rechnerplattformen. Optimieren Sie das Programm speziell für ein System mit verteiltem Speicher und 8 CPUs,<sup>1</sup> etwa indem sie die Parallelität begrenzen und Aufgaben zusammenfassen.
- (c) Übersetzen Sie Ihr Programm nun mit dem Eden-Compiler (mit der GpH-Option `-parallel` sowie Optimierung `-O2`). Führen Sie das Programm im Fachbereichsnetz mit unterschiedlicher Problemgröße und Rechneranzahl aus, messen Sie die Laufzeit mit dem Linux-Befehl `time` und ermitteln Sie aus den Messungen den *relativen Speedup* für bis zu 8 Rechner.

### 7.2 Abstrakte Interpretation

5 Punkte

- (a) Drücken Sie die Funktion `filter`

```
filter :: (a -> Bool) -> [a] -> [a]
filter p xs = [ x | x <- xs, p x ]
```

in Mini-Haskell aus.

- (b) Bestimmen Sie die abstrakte Interpretation `filter#` sowie die sicheren Auswertungsgrade der Argumente (*Evaluation Transformer*)  $ET(\text{filter})$ .

---

<sup>1</sup>Informieren Sie sich online in der GRANSIM-Anleitung über entsprechende Optionen. Die Anleitung enthält auch einige typische Parametersätze.