PHILIPPS-UNIVERSITÄT MARBURG D-35032 Marburg
Fachbereich Mathematik und Informatik Hans Meerwein Straße
Prof. Dr. R. Loogen
J.Berthold

22.Juni 2006

9. Übung zu "Parallelität in funktionalen Sprachen", SS 2006

Abgabe schriftlicher Aufgaben: Do 29.Juni 2006 (vor der Vorlesung)

Aufgaben

9.1 Parallele `map`-Implementierungen — 4 Punkte

(a) Parallelisieren Sie das Programm zur Visualisierung von Juliamengen mit Hilfe der explizit-parallelen Sprache Eden. (Erstellen Sie ein Programm zum Vergleich der verschiedenen `map`-Implementierungen).

(b) Vergleichen Sie die Laufzeiten Ihrer Eden-Programme miteinander und mit der GpH-Implementierung und erzeugen Sie Laufzeitprofile der Eden-Versionen.

9.2 Paralleles `map-fold` in Eden — 5 Punkte

Wie kann eine Faltung über zuvor elementweise transformierte Listen (`fold . map`) günstig parallelisiert werden? Beschreiben Sie eine möglichst allgemeine Lösung und wenden Sie Ihre Parallelisierung auf das `summePhi`-Programm an.

9.3 Erläutern bzw. widerlegen Sie anhand von Eden-Beispielprogrammen: — 3 Punkte

- Ein Eden-Programm ist deadlockfrei, falls sein sequenzielles Äquivalent (Funktionen statt Prozesse, Applikation statt Instanzierung) deadlockfrei ist.
- Eden erlaubt wechselseitig rekursive Prozessnetze, welche in strikt auswertenden Sprachen nicht möglich sind.
- Eden erhält die referenzielle Transparenz der Sprache Haskell.

Übersetzung und Ausführung von Eden-Programmen: Der bereits für GpH-Programme verwendete Compiler `/app/lang/functional/bin/eden` übersetzt mit der Option `-eden` Eden-Programme. Jedes Eden-Programm muss das Modul `Eden` importieren und mindestens eine Prozessinstanzierung enthalten.
Eden-Programme arbeiten wie GpH-Programme mit PVM und verwenden ebenfalls die RTS-Option `-qp<N>` für die Prozessorzahl. Weitere RTS-Optionen liefert der "usage"-Text (`+RTS -?`.

Runtime-Tracing von Eden-Programmen: Übersetzt man ein Eden-Programm statt mit `-eden` mit der Option `-edentrace`, so werden während der Programmausführung Tracedateien für eine post-mortem-Analyse geschrieben. Auf den Webseiten zu Eden finden Sie ein Analyse- und Anzeigewerkzeug *EdenTV* für diese Trace-Dateien. Damit bei Übersetzung und Ausführung die Pablo-Bibliotheken gefunden werden, muss eine Umgebungsvariable gesetzt werden:

```
#>echo "setenv PABLO_ROOT /app/lang/functional/ghc-5.02.3-eden/Pablo" >> .tcshrc
```