

## 11. Aufgabenblatt – Haskell

Ausgabe: 24. Juni 2009 Abgabe: 30. Juni 2009 Revision: 3388

### 1 Haskell: Eine pure funktionale Sprache

Haskell ist eine pure funktionale Sprache. Pur bedeutet, dass es keinen Zustandsbegriff und somit keine Seiteneffekte gibt. Haskell hat daher eine Reihe von Eigenschaften, die es von anderen Sprachen abhebt. Hauptsächlich sind das:

**Faule Auswertung (lazy evaluation)** Wird ein Ausdruck an einen Bezeichner gebunden, so wird er erst bei der Verwendung des Bezeichners berechnet; wird der Bezeichner nie verwendet, wird der Ausdruck nie ausgewertet. Sprachen wie ML, oder auch C und Java werten Ausdrücke prinzipiell bei der Bindung aus. Man kann daher in Haskell auch Code schreiben, der in strikten Sprachen nicht terminieren würde:

```
foo      = foo  -- defines a function foo with no argument
              -- that potetially has an endless recursion
bar x    = 1    -- defines a function that ignores its argument
              -- and returns 1

bar foo   -- applies foo to bar; would not terminate
          -- in strict languages but in Haskell
```

**Typklassen** Ist ein eleganter Mechanismus, um Überladung von Funktionen zu kontrollieren. Typklassen ähneln Vererbung, sind jedoch flexibler.

**Monaden** werden verwendet, um zustandsbehaftete Programmierung in Haskell zu integrieren. Bei Ein-/Ausgabe ist es zum Beispiel wichtig, dass man eine Reihenfolge von E/A-Operationen angeben kann, sonst könnte man nie kontrolliert in eine Datei schreiben.

### 2 Ihre Aufgabe

Implementieren Sie Sortieren durch Mischen (Mergesort) in Haskell. Erstellen Sie zwei Versionen, eine mit der Signatur

```
msortBy :: (a -> a -> Ordered) -> [a] -> [a]
```

bei der man die Vergleichsfunktion mit übergeben kann und eine, die eine entsprechende Typklasse `Comparable` heranzieht:

```
msortBy :: (Comparable a) => [a] -> [a]
```

Implementieren Sie zusätzlich die Typklasse `Comparable` und eine Instanz für Ganzzahlen `Integer`.

### 3 Diskussion

Arbeiten Sie sich durch folgendes Beispiel:

```
ones           = 1 : ones
numsFrom n    = n : numsFrom (n+1)
squares       = map (^2) (numsFrom 0)
```

- Was ist der Typ von `squares`?
- Wie kommt hier faule Auswertung zum Einsatz?
- Wie bekommt man die ersten `n` Quadratzahlen?
- Überlegen Sie weiterhin, wie faule Auswertung implementiert sein könnte. Was sind die Vor- und Nachteile?

### 4 Abgabe Ihrer Lösung

Drucken Sie Ihre Lösung aus und werfen Sie den Ausdruck bis zum **30. Juni 2009 um 9:00 Uhr** in den Briefkasten des Lehrstuhls für Softwaretechnik<sup>1</sup>. Vergessen Sie dabei nicht, Ihren Ausdruck mit Ihrem Namen und Ihrer Matrikelnummer zu versehen.

Bringen Sie bitte zum Proseminar ebenfalls einen Ausdruck mit, so dass Sie ihn bei etwaigen Diskussionen vorliegen haben.

---

<sup>1</sup>Gebäude E1 1, neben dem InfoPoint des Rechenzentrums.

## 5 Links und Hinweise

Zum Erstellen und Bearbeiten Ihres Programms können Sie einen gewöhnlichen Texteditor benutzen. Der Glasgow-Haskell-Compiler (ghc: [www.haskell.org/ghc](http://www.haskell.org/ghc)) ist Bestandteil vieler Open Source-Distributionen.

Auf den Sun Compute Servern<sup>2</sup> finden Sie das Binary unter `/usr/local/bin/ghc`.

<http://www.haskell.org/tutorial/> bietet eine gute Einführung in Haskell.

---

<sup>2</sup>[appsrv1.studcs.uni-saarland.de](http://appsrv1.studcs.uni-saarland.de) oder [appsrv2.studcs.uni-saarland.de](http://appsrv2.studcs.uni-saarland.de)