

Real World Haskell

Blatt 6

Julian Fleischer, Alexander Steen

Mittwoch, den 31. 07. 2013

Monad Transformers Step by Step

Lesen Sie das auf der Homepage verlinkte Paper (VPN only oder sie sind in der Uni) "Monad Transformers Step by Step" um die heutige Vorlesung nachzuvollziehen.

Parsec

Machen Sie sich ein wenig mit der Parsec-API vertraut. Schreiben Sie eine Funktion

```
1 parse :: String -> [[String]]
```

Die ein CSV-Dokument (Comma Separated Values) parst. Schreiben sie zunächst eine Funktion die nur anhand von Zeilenumbrüchen und Kommas trennt. Erweitern Sie dann ihren Parser, sodass Zelleninhalte in Anführungszeichen eingefasst sein können. Anführungszeichen sollen in diesen Strings durch doppelte Anführungszeichen escaped werden können.

Stack-basierte virtuelle Maschine

Schreiben Sie eine virtuelle Maschine (eigentlich ein Taschenrechner, aber die JVM funktioniert tatsächlich genauso) mit Hilfe von `ParsecT` und `IO`. Benutzen Sie den `user state` (Parameter `u` von `ParsecT`) als Stack (siehe auch `getParserState` und `setParserState` in `Text.Parsec.Prim`).

Die Maschine soll folgenden Befehlssatz unterstützen:

`PUSH <ZAHL>ADD` – addiert die beiden Werte auf dem Stack – und pusht das Ergebnis
`MUL DIV` – ganzzahlige Division wenn sie Integer verwenden, – Floating point wenn sie Float verwenden
`PRINT` – Gibt das aktuelle Ergebnis auf der Konsole aus – (hierfür benötigen sie `IO`)

Kurzum: Schreiben Sie einen Parser der beim parsen das geparste Script auswertet und ggf. ausgibt.

Der Zettel wird noch aktualisiert, neuladen!