

Fortgeschrittene funktionale Programmierung in Haskell

Übungszettel 6

Aufgabe 6.1:

Für die Bearbeitung dieses Zettels benötigen Sie die Bibliothek `attoparsec` `>= 0.13.0.0 && < 0.14`. Richten Sie ein cabal-Projekt und eine cabal-Sandbox ein und installieren sie die Voraussetzungen.

Aufgabe 6.2:

In der Vorlesung haben wir an einem Beispiel gelernt, wie Parsing funktioniert. Nun haben wir aber amerikanische Freunde, die ein anderes Format für ihre Logdateien benutzen:

```
154.41.32.99 29/06/2013 15:32:23 4 internet
76.125.44.33 29/06/2013 16:56:45 3 noanswer
123.45.67.89 29/06/2013 18:44:29 4 friend
100.23.32.41 29/06/2013 19:01:09 1 internet
151.123.45.67 29/06/2013 20:30:13 2 internet
```

Hierbei sind die Geräte als Zahlen codiert:

- 1 entspricht Mouse
- 2 entspricht Keyboard
- 3 entspricht Monitor
- 4 entspricht Speakers

Passen Sie den gegebenen Code an, indem Sie eine zweite Funktion `parseAmericanLog` schreiben, die das amerikanische Logformat in dieselben Datenstrukturen überführt wie das bisher bekannte. Die weiteren Angaben (`internet`, `noanswer`, `friend`) sollen zunächst ignoriert werden.

Aufgabe 6.3:

Das Management hat entschieden, dass nun mehr Informationen zur Verfügung stehen sollen. Erweitern Sie Ihr Programm erneut um ein Feld `Maybe Source`, sodass die Definition der Datenstruktur nun wie folgt aussieht:

```
data Source = Internet | Friend | NoAnswer deriving Show

data LogZeile = LogZeile Datum IP Geraet (Maybe Source) deriving Show
```

Implementieren Sie die Änderung in beiden Parsern.

Hinweis: Die Funktion `option` aus `Data.Attoparsec.Combinator` könnte hilfreich sein.

Aufgabe 6.4:

Stellen Sie alles, was Sie bisher haben zu einer kompilierten Binärdatei zusammen, die als erstes Argument eine Datei im „normalen“ Format nimmt und als zweites Argument eine Datei im „amerikanischen“ Format.

Geben Sie dann die gesamte Datenstruktur aus, in der beide Logdateien nach Datum/Uhrzeit sortiert vorkommen.

Beispiel:

```
./parser "log.txt" "logAm.txt"
```

```
Right [LogZeile (Datum {tag = 2013-06-29, zeit = 11:16:23}) (IP 124 67 34 60) Keyboard Nothing,  
LogZeile (Datum {tag = 2013-06-29, zeit = 11:32:12}) (IP 212 141 23 67) Mouse Nothing,  
LogZeile (Datum {tag = 2013-06-29, zeit = 11:33:08}) (IP 212 141 23 67) Monitor Nothing,  
LogZeile (Datum {tag = 2013-06-29, zeit = 12:12:34}) (IP 125 80 32 31) Speakers Nothing,  
LogZeile (Datum {tag = 2013-06-29, zeit = 12:51:50}) (IP 101 40 50 62) Keyboard Nothing,  
LogZeile (Datum {tag = 2013-06-29, zeit = 13:10:45}) (IP 103 29 60 13) Mouse (Just NoAnswer)]
```

(Umbrüche Textbedingt)