

5. Übung zu „Grundlagen des Compilerbaus“, WS 2007/08

Abgabe der Aufgaben: Mi, 21.November 2007 (vor der Vorlesung)

Aufgaben

5.1 Recursive-Descent-Parsing

4 Punkte

Definieren Sie mit Parserkombinatoren und Hilfsfunktionen einen *recursive-descent*-Parser in Haskell, der den Zahlwert der dargestellten Summe berechnet.

Verwenden Sie die nebenstehenden Hilfsfunktionen.

$S \rightarrow EE'$	
$E' \rightarrow +E$	<code>lchoice :: [[tkn]] -> [Parser tkn b]</code>
ε	-> Parser tkn b
$E \rightarrow (S)$	<code>digit :: Parser Char Char</code>
D	<code>tok :: Eq a => [a] -> Parser a [a]</code>
$D \rightarrow ZD$	<code>readInt :: String -> Int</code>
Z	
$Z \rightarrow 0 1 2 3 4 5 6 7 8 9$	

5.2 Parserkombinatoren: bind vs. seqp

2 Punkte

Zur Sequentialisierung von Parsern wurde in der Vorlesung der Parserkombinator `bind` vorgestellt. Alternativ kann eine Sequentialisierung aber auch über `seqp` erfolgen:

```
seqp :: Parser tok a -> Parser tok b -> Parser tok (a,b)
seqp p1 p2 = \input -> [((v,w),input'') | (v,input') <- p1 input
                                , (w,input'') <- p2 input']
```

Drücken Sie `seqp` durch `bind` aus. Welches Problem besteht, wenn `bind` mit Hilfe von `seqp` ausgedrückt werden soll?

Aufgabe 5.3 hat eine Bearbeitungszeit von 2 Wochen. Sie ist am **28.11.**, zusammen mit den Lösungen zu Blatt 6 abzugeben.

SDDF ist ein flexibles Meta-Dateiformat, in dem sowohl Datensatz-Strukturen definiert, als auch Datensatz-Instanzen abgelegt werden können. Es wurde zur Speicherung von Event- und Leistungsdaten paralleler Programmabläufe entwickelt. Anbei ist ein Auszug aus dem Dokument „The Pablo **S**elf **D**efining **D**ata **F**ormat“¹ (S.24 und 25) zu finden, dem die Grammatik von SDDF Dateien in ASCII-Fassung zu entnehmen ist.

Sie sollen Programme schreiben, mit denen die Lexikalische und die Syntaktische Analyse von SDDF-Dateien durchgeführt werden kann. Trace-Beispieldateien zum Testen ihrer Programme sind auf der Vorlesungsseite zu finden.

- (a) Schreiben Sie mit dem Scannergenerator Alex einen SDDF-Scanner. Achten Sie bei der Scannerdefinition darauf, für die syntaktische Analyse sinnvolle Tokentypen mit den zur Weiterverarbeitung notwendigen Attributen zu definieren.
- (b) Benutzen Sie die Kombinatorenbibliothek Parsec² um einen SDDF-Parser zu erstellen.

¹Quelle: <http://www.renci.org/publications/documents.php#SDDF>

²siehe: <http://legacy.cs.uu.nl/daan/parsec.html>