Lehrstuhl für Programmiersprachen und Übersetzerkonstruktion
CAU Kiel, Informatik und Prakt. Mathematik
Dr. Frank Huch

24118 Kiel
Christan-Albrechts-Platz 4
Tel.: 0431 / 880-7277

10. Übung „Funktionale Programmierung“

Abgabe am 22. Juni 2005 vor der Vorlesung

Wichtig: Denken Sie bei allen definierten Funktionen/Typen an die auf dem 1. Übungsblatt formulierten Regeln (z.B. Typsignaturen, Beispielanwendungen und Beispielwerte).

Aufgabe 1 6 Punkte

Wir betrachten die kontextfreie Grammatik $G = \langle \{S\}, \Sigma, P, S \rangle$ mit $\Sigma = \{a, b, c\}$ und

$$P \ : \ S \longrightarrow SaS \mid SbS \mid ScS \mid \varepsilon$$

Erweitern Sie die Grammatik um drei Attributierungen, die jeweils folgende Eigenschaften für ein Wort $w \in L(G)$ überprüft:

a) Die Anzahl der as, bs und cs sind gleich: $|w|_a = |w|_b = |w|_c$

b) Jedes Präfix der Eingabe enthält mindestens so viele as wie bs:
$\forall v \in \Sigma^*$ mit $w = vu$ gilt: $|v|_a \geq |v|_b$

c) Jedes Suffix der Eingabe enthält mindestens so viele as wie bs:
$\forall v \in \Sigma^*$ mit $w = uv$ gilt: $|v|_a \geq |v|_b$

Die Eigenschaften sollen durch ein boolesches, synthetisches Attribut des Wurzelsymbols S des Ableitungsbaumes zu w angezeigt werden. Geben Sie jeweils die Art der Attributierung an (S,L oder sonstige).
Implementieren Sie Ihre Attributierungen mit Hilfe des in der Vorlesung vorgestellten Verfahrens (ohne CPS!) in Haskell.

Aufgabe 2 7 Punkte

a) Implementieren Sie einen Parser für die in der Vorlesung vorgestellten binären Gleitkommazahlen mit Hilfe von Parserkombinatoren. Als Ausgabe soll Ihr Parser einen Wert vom Typ N liefern. Beachten Sie hierbei, dass die in der Vorlesung vorgestellte Grammatik linksrekursiv ist.

b) Können Sie alle drei in der Vorlesung vorgestellten Attributberechnungen direkt in den Parsevorgang integrieren?
Passen Sie hierzu zunächst die attributierte Grammatik an den von Ihnen implementierten Parser an und verwenden Sie die veränderten Attributierungen zur Berechnung der Parser-Ergebnisse.

Aufgabe 3 4 Punkte

a) In der Vorlesung wurde gezeigt, wie eine S-Attributierung im CPS implementiert werden kann. Überlegen Sie, wie auch L-Attributierungen im CPS implementiert werden können. Verdeutlichen Sie ihre Überlegungen mit Hilfe einer CPS-Implementierung der in der Vorlesung vorgestellten L-Attributierung für ganzzahlige Binärzahlen.

b) Implementieren Sie Quicksort im CPS.

c) Implementieren Sie `unzipC` als CPS-Variante von `unzip`. Wie lautet der allgemeinste Typ von `unzipC`? Geben Sie eine (sinnvolle) Verwendung von `unzipC` an, welche kein Tupel als Ergebnis liefert.

Aufgabe 4 3 Punkte

a) Implementieren Sie Quicksort mit funktionalen Listen.

b) Implementieren Sie `foldr` mit funktionalen Listen.
Sollte dies nicht möglich sein, begründen Sie warum.

c) Reimplementieren Sie die Funktion `showEvalTree` des `Observe`-Moduls mit funktionalen Listen.