

**8. Übungsaufgabe zu
Fortgeschrittene funktionale Programmierung
Thema: Funktionale Perlen
ausgegeben: Di, 14.05.2013, fällig: Mi, 22.05.2013**

Für dieses Aufgabenblatt sollen Sie Haskell-Rechenvorschriften zur Lösung der im folgenden angegebenen Aufgabenstellungen entwickeln und für die Abgabe in einer Datei namens `AufgabeFFP8.hs` in Ihrem Gruppenverzeichnis ablegen, wie gewohnt auf oberstem Niveau. Kommentieren Sie Ihre Programme aussagekräftig und benutzen Sie, wo sinnvoll, Hilfsfunktionen und Konstanten.

Am 16.05.2013 besucht Donald E. Knuth die Fakultät für Informatik an der TU Wien und hält um 17:30 Uhr im Hörsaal EI7 einen Gastvortrag "All Questions Answered". Zu diesem Vortrag sind alle herzlich eingeladen!

Aus Anlass des Besuchs von Donald Knuth betrachten wir heute das Problem "Making a Century", das sich als Übung 122 in

- Donald E. Knuth. *The Art of Computer Programming, Volume 4, Fascicle 4: Generating All Trees*. Reading, MA, Addison-Wesley, 2006

findet.

Dabei geht es um folgendes Problem: Gesucht sind alle Möglichkeiten, wie die Operatoren `+` und `*` in die Liste der Ziffern `[1..9]` eingefügt werden können, so dass sich ein Wert von 100 ergibt. Dabei wird angenommen, dass `*` stärker bindet als `+` und dass keine Klammern verwendet werden.

Zwei mögliche Lösungen sind dann folgende:

$$\begin{aligned} 100 &= 12 + 34 + 5 * 6 + 7 + 8 + 9 \\ 100 &= 1 + 2 * 3 + 4 + 5 + 67 + 8 + 9 \end{aligned}$$

Eine offensichtliche, aber ineffiziente Lösung ist eine erschöpfende Suche, bei der alle Möglichkeiten generiert werden und auf Gültigkeit überprüft werden. Entwickeln Sie im Stil des *equational reasoning* daraus eine effizientere Variante. Dabei sollen statt 100 auch andere Zahlen als Zielwert möglich sein.

```
type TargetValue = Integer
type Digit       = Integer -- ausschliesslich Werte 1,2,...,9
type Digits     = [Digit]

digits = [1..9] :: Digits

data Operator = P | T deriving (Eq,Show) -- P fuer plus, T fuer times
data Expr = Opd Digit -- hier jede natuerliche Zahl zulaessig
          | Opr Operator Expr Expr deriving (Eq,Show)

mkTV :: Digits -> TargetValue -> [Expr]
```

Der Aufruf `mkTV digits 100` löst dann das ‘klassische’ “Making a Century”-Problem, für andere Zielwerte ergeben sich Varianten davon. Die Funktion `mkTV` wird für den Abgabetest stets mit dem ersten Argument `digits` aufgerufen bzw. mit Anfangslisten davon (d.h. `take i digits` für $2 \leq i \leq 9$) für kleinere Probleminstanzen; der Wert des zweiten Argument kann beliebig variiert werden. Gibt es keine Lösung, so ist das Resultat des Funktionsaufrufs die leere Liste. Die Reihenfolge der Ausdrücke in einer nichtleeren Resultatliste ist unerheblich.