

Übungen zur „Deskriptiven Programmierung“ Blatt 3

Viele listenverarbeitende Funktionen folgen dem gleichen Rekursionsschema. Dieses Schema kann durch eine Funktion höherer Ordnung eingefangen werden.

$$\begin{aligned} fold & :: \beta \rightarrow (\alpha \rightarrow \beta \rightarrow \beta) \rightarrow [\alpha] \rightarrow \beta \\ fold\ e\ f\ [] & = e \\ fold\ e\ f\ (a : x) & = f\ a\ (fold\ e\ f\ x) \end{aligned}$$

Die Funktion *fold* hat die bemerkenswerte Eigenschaft, dass sie die *einzigste* Lösung der obigen Gleichungen (aufgefaßt als Gleichungen in der Unbekannten *fold*) ist.

$$fold\ e\ f = \varphi \iff \begin{cases} \varphi\ [] = e \\ \varphi\ (a : x) = f\ a\ (\varphi\ x) \end{cases}$$

Dieses Gesetz heißt auch die „universelle Eigenschaft“ von *fold* (Englisch: universal property).

Aufgabe 5. Definiere die Funktionen

$$\begin{aligned} sum\ [] & = 0 \\ sum\ (a : x) & = a + sum\ x \\ length\ [] & = 0 \\ length\ (a : x) & = 1 + length\ x \\ reverse\ [] & = [] \\ reverse\ (a : x) & = reverse\ x \# [a] \end{aligned}$$

ohne explizite Rekursion mit Hilfe von *fold*.

Aufgabe 6. Zeige den folgenden Zusammenhang

$$plus\ n \cdot sum = fold\ n\ plus$$

mit Hilfe der universellen Eigenschaft von *fold*. Die Funktion *plus* ist durch $plus\ a\ b = a + b$ definiert.

Aufgabe 7. Das folgende Gesetz (Englisch: fusion) läßt sich oft zur Programmoptimierung einsetzen. Es zeigt, wie ein nachgeschalteter Funktionsaufruf in *fold* hineingezogen werden kann: *h* und *fold e f* werden zu *fold e' f'* fusioniert.

$$h \cdot fold\ e\ f = fold\ e'\ f' \iff \begin{cases} h\ e = e' \\ h\ (f\ a\ x) = f'\ a\ (h\ x) \end{cases}$$

Leite das Gesetz aus der universellen Eigenschaft von *fold* ab. Zeige den Zusammenhang aus Aufgabe 6 erneut, diesmal mit Hilfe von Fusion.