Funktionale Programmierung

Fixpunkte

Fixpunkte bieten einen konstruktiven Zugang zu rekursiv definierten Funktionen[1]. Sie formalisieren die Idee des "Aufbauens einer rekursiven Struktur von Unten"[2].

Definition

Als Fixpunkte einer Funktion $F : X \to Y$ werden Elemente $x \in X \cap Y$ mit

$$F(x) = x$$

bezeichnet.

[1] Das heisst zu verstehen, wieso die verschiedenen Rekursionsgleichungen eine Lösung (die zu definierende Funktion) besitzen und wie diese konstruiert wird.
[2] Stichwort: Fixpunktiteration

Unter geeigneten Umständen[3], lassen sich Definitionen als Fixpunktgleichungen darstellen:

$$\text{Das } x \text{ mit der Eigenschaft: } f(x) = x.$$

Man kann zum Beispiel die Zahl 0 durch die Gleichung

$$2x = x$$

definieren.

[3] Wenn der Fixpunkt existiert und eindeutig ist.

Wir betrachten nun Fixpunkte von Funktionen höherer Ordnung[4]. Wir beginnen mit dem Beispiel

$$\mathrm{expF}(f) = x \mapsto \begin{cases} 1 & \text{falls } x = 0 \\ 2 \cdot f(x-1) & \text{sonst.} \end{cases}$$

Oder äquivalent dazu

```haskell
expF f x
     | x == 0 = 1
     | otherwise = 2 * f (x - 1)
```

[4] In diesem Zusammenhang oft Funktionale genannt.

Wenn wir das Funktional expF mit der rekursiven Definition der Funktion $x \mapsto 2^x$ vergleichen, dann sehen wir deutliche Parallelen:

```
expF f x
    | x == 0 = 1
    | otherwise = 2 * f (x - 1)
```

```
exp x
    | x == 0 = 1
    | otherwise = 2 * exp (x - 1)
```

Konkret besteht der einzige Unterschied darin, dass in expF der Parameter f anstelle eines rekursiven Aufrufes steht.

Unsere Vermutung könte also sein, dass ein Fixpunkt von expF gerade der Funktion $x \mapsto 2^x$ entspricht (weil dann $f = \mathrm{expF} f$ gilt).
Gehen wir nun davon aus, dass wir einen Fixpunkt H von expF haben. Es gelte also

$$\mathrm{expF}(H) = H$$

und deshalb

$$\mathrm{expF}(H)(x) = H(x).$$

Wir untersuchen nun das Verhalten von H..

$$H(0) = \mathsf{expF}(H)(0) = 1$$
$$H(n+1) = \mathsf{expF}(H)(n+1) = 2 \cdot H(n)$$

Wir untersuchen nun das Verhalten von H..

$$\begin{aligned} H(0) &= \mathrm{expF}(H)(0) = 1 \\ H(n+1) &= \mathrm{expF}(H)(n+1) = 2 \cdot H(n) \end{aligned}$$

... und erkennen, dass $H(x) = 2^x$ gilt.

Natürlich ist das von uns beobachtete Phänomen nicht auf die Exponentialfunktion beschränkt, vielmehr handelt es sich dabei um eine Technik die es erlaubt, wenn die nötigen Voraussetzungen erfüllt sind, Lösungen für beliebige Reursionsgleichungen zu finden.

Wir können in Haskell direkt eine Funktion `fix` zum finden von Fixpunkten definieren:

```
fix f = f $ fix f
```

Aufgabe

Implementieren Sie ein Funktional