

III.1 Prinzipien der funktionalen Programmierung

Dienstag, 12. Dezember 2017 09:00

Listen-Längen-Algorithmus

imperativ:

$x = [15, 70, 36]$

$n = 0$

↓

$x = [70, 36]$

$n = 1$

↓

$x = [36]$

$n = 2$

↓

$x = []$

$n = 3$

Bsp: Als Seiteneffekt
wird die Liste x gelöscht.

Listen-Länge in Haskell

Listenschreibweise:

Kopf : rest . . . steht für die

Liste, die aus
der Liste *rest* entsteht,
wenn man vorne den
Wert *Kopf* einfügt.

$$15 : [70, 36] = [15, 70, 36]$$

$$15 : 70 : 36 : [] = [15, 70, 36]$$

steht für

$$15 : (70 : (36 : []))$$

[] steht für die leere
Liste

Klammern um Argumente
können weggelassen werden,
wenn dies nicht zu Uneindeu-
tigkeiten führt.

$$\left(\begin{array}{l} \text{len } [] = 0 \quad \text{statt} \\ \text{len } ([]) = 0 \end{array} \right)$$

Compiler:

übersetzt Prog. zunächst
in Maschinensprache.

Ausschließend kann Maschinensprache
Prog. ausgeführt werden.