

Reduktionsstrategien

Um in Haskell eine Funktion auszuwerten, gibt es unterschiedliche Strategien. Die Funktion `quadr` die dem Quadrieren einer Zahl dient, wird in anderen Funktionen verwendet, wie das Beispiel zur Berechnung des Abstandes zweier Punkte zeigt.

```

7 quadr :: Double -> Double
8 quadr a = a * a
9
10 distance :: Double -> Double -> Double -> Double -> Double
11 distance x1 y1 x2 y2 = sqrt (quadr (x2 - x1) + quadr (y2 - y1))

```

Um den Ausdruck auszuwerten, ist eine möglichst effiziente Vorgehensweise sinnvoll. Dafür gibt es jedoch verschiedene Möglichkeiten. Der folgende Ausdruck kann auf zwei Wegen reduziert werden.

`quadr (2 + 7)`

Wertet man den Ausdruck also bis zum Ergebnis aus (**Evaluation**) gibt es zwei mögliche Reduktionsfolgen, die im Folgenden dargestellt werden.

leftmost innermost Reduktion - LI - linkseitige innere Reduktion

`quadr (2 + 7) ↦ quadr 9 ↦ 9 x 9 ↦ 81`

leftmost outermost Reduktion - LO - linkseitige äußere Reduktion

`quadr (2 + 7) ↦ (2+7) x (2+7) ↦ (2+7) x 9 ↦ 9 x 9 ↦ 81`

Schaut man sich den Ausdruck

`fst (1, quadr (2 + 7))` ist die linksseitige innere Reduktion länger als die rechtsseitige, welche sofort den Wert 1 liefert.

Aus diesem Grund wird die Reduktion in Haskell unter bestimmten Bedingungen von Außen durchgeführt.

Aufgaben

- (a) Werten Sie den Ausdruck `square (square 4)` nach den Reduktionsstrategien LI und LO aus.
- (b) Welchen entscheidenden Nachteil hat die **leftmost outermost Reduktion - LO** ?
- (c) Erläutern Sie den Begriff **lazy evaluation** – Internet (!)