Funktionale Programmierung
Sommersemester 2021

Prof. Dr. Jakob Rehof
TU Dortmund
LS XIV Software Engineering

Diese Vorlesung:

- λ- Kalkül (erste Schritte)
- DoberkatFP: Folien 29 – 57
  - Funktionsdefinitionen
  - Rekursive Funktionen
  - Pattern matching
  - Typklassen
  - Paare und Produkttyp

λ - Kalkül (erste Schritte)

Syntax (λ-Terme):

$$M ::= x \mid (M\ M) \mid (\lambda x.M)$$

Semantik (Reduktion):

$$((\lambda x.M)\ N) \longrightarrow_\beta M[x := N]$$

Sei Λ die Menge aller wohlgeformten λ-Terme

Freie und gebundene Variable

$$
\begin{aligned}
\mathrm{FV}(x) &= \{x\};\\
\mathrm{FV}(\lambda x.P) &= \mathrm{FV}(P)\backslash\{x\};\\
\mathrm{FV}(P\ Q) &= \mathrm{FV}(P) \cup \mathrm{FV}(Q).
\end{aligned}
$$

Wenn $FV(M) = \emptyset$, sagen wir, dass M *geschlossen* ist (eng. *closed*).
Wenn $x \in FV(M)$, sagen wir, dass x *frei in* M ist.
Wenn $x \in FV(M)$, dann ist x in $\lambda x.M$ *gebunden* (eng. *bound*).

Substitution

$$
\begin{array}{lcll}
x[x := N] & = & N; & \\
y[x := N] & = & y, & \text{if } x \neq y; \\
(P\ Q)[x := N] & = & P[x := N]\ Q[x := N]; & \\
(\lambda y.P)[x := N] & = & \lambda y.P[x := N], & \text{if } x \neq y, \text{ where } y \notin \mathrm{FV}(N).
\end{array}
$$

Beispiel:

(i) $(\lambda x.x\ y)[x := \lambda z.z] = \lambda x.x\ y;$

(ii) $(\lambda x.x\ y)[y := \lambda z.z] = \lambda x.x\ \lambda z.z.$


LEONARDO FIBONACCI
A
LEONARDO FIBONACCI
INSIGNE MATEMATICO
PISANO
DEL
SECOLO XII

Fibonacci Rekursionsbaum