Funktionale Programmierung

Sommersemester 2025

Prof. Dr. Jakob Rehof
M.Sc. Felix Laarmann
TU Dortmund
LS XIV Software Engineering

Diese Vorlesung:

- Lambda-Kalkül:
  - Grundlegende Definitionen

λ - Kalkül: Reduktionstheorie

Syntax (λ-Terme):

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

Semantik (Reduktion):

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

Sei $\longrightarrow_\beta^*$ die reflexiv-transitive Hülle von $\longrightarrow_\beta$.
Also, $P \longrightarrow_\beta^* Q$ genau dann, wenn

$$\exists n \geq 0.\ P = P_0 \longrightarrow_\beta P_1 \longrightarrow_\beta \ldots \longrightarrow_\beta P_n = Q$$

λ - Kalkül: Kurzschreibweise

Damit wir λ-Terme leichter lesen können, führen wir folgende Regeln als Kurzschreibweise ein:

Wir schreiben

1. $(K\ L\ M)$ für $((K\ L)\ M)$;
2. $(\lambda x\,.\,\lambda y\,.\,M)$ für $(\lambda x\,.\,(\lambda y\,.\,M))$;
3. $(\lambda x\,.\,M\ N)$ für $(\lambda x\,.\,(M\ N))$;
4. $(M\ \lambda x\,.\,N)$ für $(M\ (\lambda x\,.\,N))$;
5. $(\lambda x_1\ x_2\ \ldots\ x_n\,.\,M)$ für $(\lambda x_1\,.\,(\lambda x_2\,.\ \ldots\ (\lambda x_n\,.\,M)\ldots))$;
6. und verzichten auf die äußersten Klammern.

Wir dürfen also z.B.

$$(\lambda z\ s\,.\,s\ (s\ z))(\lambda z\ s\,.\,z)\ \lambda n\ z\ s\,.\,s\ (n\ z\ s) \quad \text{für}$$

$$(((\lambda z\,.\,(\lambda s\,.\,s\ (s\ z)))(\lambda z\,.\,(\lambda s\,.\,z)))\ (\lambda n\,.\,(\lambda z\,.\,(\lambda s\,.\,(s\ ((n\ z)\ s))))))$$

schreiben.

Free variables, closed term

1.1.11. Definition. For $M \in \Lambda^-$ define the set $\mathrm{FV}(M) \subseteq V$ of *free variables* of M as follows.

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

If $\mathrm{FV}(M) = \{\}$ then M is called *closed*.

Beispiel

1.1.12. EXAMPLE. Let x, y, z denote distinct variables. Then

(i) $\mathrm{FV}(x\ y\ z) = \{x, y, z\}$;

(ii) $\mathrm{FV}(\lambda x.x\ y) = \{y\}$;

(iii) $\mathrm{FV}((\lambda x.x\ x)\ \lambda y.y\ y) = \{\}$.

Substitution

1.1.13. DEFINITION. For $M, N \in \Lambda^-$ and $x \in V$, the *substitution of* N *for* x *in* M, written $M[x := N] \in \Lambda^-$, is defined as follows, where $x \neq y$:

$$
\begin{array}{lll}
x[x := N] & = N; & \\
y[x := N] & = y; & \\
(P\,Q)[x := N] & = P[x := N]\,Q[x := N]; & \\
(\lambda x.P)[x := N] & = \lambda x.P; & \\
(\lambda y.P)[x := N] & = \lambda y.P[x := N], & \text{if } y \notin \mathrm{FV}(N) \text{ or } x \notin \mathrm{FV}(P); \\
(\lambda y.P)[x := N] & = \lambda z.P[y := z][x := N], & \text{if } y \in \mathrm{FV}(N) \text{ and } x \in \mathrm{FV}(P).
\end{array}
$$

where z is chosen as the $v_i \in V$ with minimal i such that $v_i \notin \mathrm{FV}(P) \cup \mathrm{FV}(N)$ in the last clause.

Beispiel

1.1.14. Example. If x, y, z are distinct variables, then for a certain variable u:

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

Alpha-equivalence (alpha-conversion)

1.1.15. DEFINITION. Let α-*equivalence*, written $=_\alpha$, be the smallest relation on Λ^-, such that

$$\begin{array}{ll} P =_\alpha P & \text{for all } P; \\ \lambda x.P =_\alpha \lambda y.P[x := y] & \text{if } y \notin \mathrm{FV}(P), \end{array}$$

and closed under the rules:

$$\begin{array}{lcll} P =_\alpha P' & \Rightarrow & \forall x \in V: & \lambda x.P =_\alpha \lambda x.P'; \\ P =_\alpha P' & \Rightarrow & \forall Z \in \Lambda^-: & P\ Z =_\alpha P'\ Z; \\ P =_\alpha P' & \Rightarrow & \forall Z \in \Lambda^-: & Z\ P =_\alpha Z\ P'; \\ P =_\alpha P' & \Rightarrow & P' =_\alpha P; & \\ P =_\alpha P' \ \&\ P' =_\alpha P'' & \Rightarrow & P =_\alpha P''. & \end{array}$$

Beispiel

1.1.16. Example. Let x, y, z denote different variables. Then

(i) $\lambda x.x =_\alpha \lambda y.y$;

(ii) $\lambda x.x\ z =_\alpha \lambda y.y\ z$;

(iii) $\lambda x.\lambda y.x\ y =_\alpha \lambda y.\lambda x.y\ x$;

(iv) $\lambda x.x\ y \neq_\alpha \lambda x.x\ z$.

λ - Kalkül: Reduktionstheorie

Vollständige Definition der β-Reduktion ($\longrightarrow_\beta$):

- $((\lambda x.P)Q) \longrightarrow_\beta P[x := Q]$
- $P \longrightarrow_\beta P' \Rightarrow (PQ) \longrightarrow_\beta (P'Q)$
- $Q \longrightarrow_\beta Q' \Rightarrow (PQ) \longrightarrow_\beta (PQ')$
- $P \longrightarrow_\beta P' \Rightarrow (\lambda x.P) \longrightarrow_\beta (\lambda x.P')$