

# PROGRAMMIERUNG

## ÜBUNG 6: $\lambda$ -KALKÜL — TEIL 1

---

Eric Kunze

`eric.kunze@tu-dresden.de`

1. Funktionale Programmierung
  - 1.1 Einführung in Haskell: Listen
  - 1.2 Algebraische Datentypen
  - 1.3 Funktionen höherer Ordnung
  - 1.4 Typpolymorphie & Unifikation
  - 1.5 Beweis von Programmeigenschaften
  - 1.6  $\lambda$  - **Kalkül**
2. Logikprogrammierung
3. Implementierung einer imperativen Programmiersprache
4. Verifikation von Programmeigenschaften
5.  $H_0$  – ein einfacher Kern von Haskell

# Der $\lambda$ -Kalkül

---

- ▶ weitere funktionale Programmiersprache
- ▶ Programme =  $\lambda$ -Terme
- ▶ Vorstellung: *anonyme* Funktionen

- ▶ weitere funktionale Programmiersprache
- ▶ Programme =  $\lambda$ -Terme
- ▶ Vorstellung: *anonyme* Funktionen

Sei  $X$  eine Menge mit Variablen,  $\Sigma$  eine Menge mit Symbolen. Die gültigen  $\lambda$ -Terme sind *induktiv* definiert:

1. **Atome** (Variablen oder Symbole) sind gültige  $\lambda$ -Terme.

- ▶ weitere funktionale Programmiersprache
- ▶ Programme =  $\lambda$ -Terme
- ▶ Vorstellung: *anonyme* Funktionen

Sei  $X$  eine Menge mit Variablen,  $\Sigma$  eine Menge mit Symbolen. Die gültigen  $\lambda$ -Terme sind *induktiv* definiert:

1. **Atome** (Variablen oder Symbole) sind gültige  $\lambda$ -Terme.
2. **Abstraktion:** Ist  $t$  ein gültiger  $\lambda$ -Term und  $x \in X$  eine Variable, dann ist auch  $\lambda x.t$  ein gültiger  $\lambda$ -Term.

- ▶ weitere funktionale Programmiersprache
- ▶ Programme =  $\lambda$ -Terme
- ▶ Vorstellung: *anonyme* Funktionen

Sei  $X$  eine Menge mit Variablen,  $\Sigma$  eine Menge mit Symbolen. Die gültigen  $\lambda$ -Terme sind *induktiv* definiert:

1. **Atome** (Variablen oder Symbole) sind gültige  $\lambda$ -Terme.
2. **Abstraktion**: Ist  $t$  ein gültiger  $\lambda$ -Term und  $x \in X$  eine Variable, dann ist auch  $\lambda x.t$  ein gültiger  $\lambda$ -Term.
3. **Applikation**: Sind  $t_1$  und  $t_2$  gültige  $\lambda$ -Terme, dann ist auch  $(t_1 \ t_2)$  ein gültiger  $\lambda$ -Term.

## BEISPIELE (INFORMELL)

Vorstellung: Jeder  $\lambda$ -Term beschreibt eine anonyme Funktion.



## BEISPIELE (INFORMELL)

Vorstellung: Jeder  $\lambda$ -Term beschreibt eine anonyme Funktion.

- Abstraktion gibt das Argument an:

$$\lambda x.t \quad \leftrightarrow \quad f(x) = t$$

# BEISPIELE (INFORMELL)

Vorstellung: Jeder  $\lambda$ -Term beschreibt eine anonyme Funktion.

- ▶ Abstraktion gibt das Argument an:

$$\lambda x.t \quad \leftrightarrow \quad f(x) = t$$

- ▶ Applikation beschreibt Funktionsanwendung (Einsetzen):

$$((\lambda x.t) \ 2) \quad \leftrightarrow \quad f(2)$$

# BEISPIELE (INFORMELL)

Vorstellung: Jeder  $\lambda$ -Term beschreibt eine anonyme Funktion.

- ▶ Abstraktion gibt das Argument an:

$$\lambda x.t \quad \leftrightarrow \quad f(x) = t$$

- ▶ Applikation beschreibt Funktionsanwendung (Einsetzen):

$$((\lambda x.t) \ 2) \quad \leftrightarrow \quad f(2)$$

# BEISPIELE (INFORMELL)

Vorstellung: Jeder  $\lambda$ -Term beschreibt eine anonyme Funktion.

- ▶ Abstraktion gibt das Argument an:

$$\lambda x.t \quad \leftrightarrow \quad f(x) = t$$

- ▶ Applikation beschreibt Funktionsanwendung (Einsetzen):

$$((\lambda x.t) \ 2) \quad \leftrightarrow \quad f(2)$$

## Beispiel:

$$\begin{aligned}\text{quadriere} &= \lambda x. x * x \\ ((\lambda x. x * x) \ 2) &= 2 * 2 = 4\end{aligned}$$

↗  $\beta$ -Reduktion

- Applikation ist linksassoziativ:

$$((t_1 \ t_2) \ t_3) = t_1 \ t_2 \ t_3$$

- ▶ Applikation ist linksassoziativ:

$$((t_1 \ t_2) \ t_3) = t_1 \ t_2 \ t_3$$

- ▶ mehrfache Abstraktion:

$$(\lambda x_1. (\lambda x_2. (\lambda x_3. t))) = \lambda x_1 x_2 x_3. t$$

- ▶ Applikation ist linksassoziativ:

$$((t_1 \ t_2) \ t_3) = t_1 \ t_2 \ t_3$$

- ▶ mehrfache Abstraktion:

$$(\lambda x_1. (\lambda x_2. (\lambda x_3. t))) = \lambda x_1 x_2 x_3. t$$

- ▶ Applikation vor Abstraktion:

$$\begin{aligned} (\lambda x. x \ y) &= (\lambda x. (x \ y)) \\ &\neq ((\lambda x. x) \ y) \end{aligned}$$

# GEBUNDENE UND FREIE VORKOMMEN

Mengen  $FV(t)$  und  $GV(t)$  geben frei bzw. gebunden *vorkommende* Variablen von  $t$  an — induktive Definition



Mengen  $FV(t)$  und  $GV(t)$  geben frei bzw. gebunden *vorkommende* Variablen von  $t$  an — induktive Definition

- einzelne **Variablen** sind immer frei:

$$x \in X \Rightarrow FV(x) = \{x\}, GV(x) = \emptyset$$

Mengen  $FV(t)$  und  $GV(t)$  geben frei bzw. gebunden *vorkommende* Variablen von  $t$  an — induktive Definition

- ▶ einzelne **Variablen** sind immer frei:

$$x \in X \Rightarrow FV(x) = \{x\}, GV(x) = \emptyset$$

- ▶ **Symbole** sind weder frei noch gebunden

Mengen  $FV(t)$  und  $GV(t)$  geben frei bzw. gebunden *vorkommende* Variablen von  $t$  an — induktive Definition

- ▶ einzelne **Variablen** sind immer frei:

$$x \in X \Rightarrow FV(x) = \{x\}, GV(x) = \emptyset$$

- ▶ **Symbole** sind weder frei noch gebunden

- ▶ **Applikation:** Sei  $t = (t_1 \ t_2)$ . Dann

$$\Rightarrow FV(t) = FV(t_1) \cup FV(t_2), \quad GV(t) = GV(t_1) \cup GV(t_2)$$

Mengen  $FV(t)$  und  $GV(t)$  geben frei bzw. gebunden *vorkommende* Variablen von  $t$  an — induktive Definition

- ▶ einzelne **Variablen** sind immer frei:

$$x \in X \Rightarrow FV(x) = \{x\}, GV(x) = \emptyset$$

- ▶ **Symbole** sind weder frei noch gebunden

- ▶ **Applikation:** Sei  $t = (t_1 \ t_2)$ . Dann

$$\Rightarrow FV(t) = FV(t_1) \cup FV(t_2), \quad GV(t) = GV(t_1) \cup GV(t_2)$$

- ▶ **Abstraktion:**  $t = \lambda x.t'$

$$\Rightarrow FV(t) = FV(t') \setminus \{x\}, \quad GV(t) = GV(t') \cup \{x\}$$

## $\beta$ -Reduktion

Seien  $s, t \in \lambda(\Sigma)$  gültige  $\lambda$ -Terme und es gilt  $GV(t) \cap FV(s) = \emptyset$ .

$$(\lambda x. t) s \longrightarrow_{\beta} t[x/s]$$

## $\beta$ -Reduktion

Seien  $s, t \in \lambda(\Sigma)$  gültige  $\lambda$ -Terme und es gilt  $GV(t) \cap FV(s) = \emptyset$ .

$$(\lambda x. t) s \longrightarrow_{\beta} t[x/s]$$

- ▶ Bedeutung von  $t[x/s]$ : Ersetze jedes *freie* Vorkommen von  $x$  in  $t$  durch  $s$ .
- ▶ Erinnerung: Vorstellung der Applikation als „Einsetzen“ in Funktionen
- ▶ beachte: Abstraktion  $\lambda x$  entfällt

## $\beta$ -Reduktion

Seien  $s, t \in \lambda(\Sigma)$  gültige  $\lambda$ -Terme und es gilt  $GV(t) \cap FV(s) = \emptyset$ .

$$(\lambda x. t) s \longrightarrow_{\beta} t[x/s]$$

- Bedeutung von  $t[x/s]$ : Ersetze jedes *freie* Vorkommen von  $x$  in  $t$  durch  $s$ .
- Erinnerung: Vorstellung der Applikation als „Einsetzen“ in Funktionen
- beachte: Abstraktion  $\lambda x$  entfällt

**Bsp.:** Seien die Symbole gegeben durch  $\Sigma = \{3, a\}$ .

$$\underbrace{(\lambda x. +x3)}_{GV=\emptyset} \underbrace{(\lambda z. a)}_{FV=\emptyset} \longrightarrow_{\beta} + (\lambda z. a)3$$

- ▶ Was machen wir, wenn Voraussetzung  $FV(t) \cap GV(t) = \emptyset$  für  $\beta$ -Reduktion nicht erfüllt ist?
- ▶ einfacher Ausweg: entsprechende Variablen umbenennen, sodass Bedingung erfüllt ist



- ▶ Was machen wir, wenn Voraussetzung  $FV(t) \cap GV(t) = \emptyset$  für  $\beta$ -Reduktion nicht erfüllt ist?
- ▶ einfacher Ausweg: entsprechende Variablen umbenennen, sodass Bedingung erfüllt ist

### $\alpha$ -Konversion

Sei  $t \in \lambda(\Sigma)$  und  $z \notin GV(t) \cup FV(t)$ .

$$(\lambda x.t) \longrightarrow_{\alpha} \lambda z.t[x/z]$$

- ▶ Was machen wir, wenn Voraussetzung  $FV(t) \cap GV(t) = \emptyset$  für  $\beta$ -Reduktion nicht erfüllt ist?
- ▶ einfacher Ausweg: entsprechende Variablen umbenennen, sodass Bedingung erfüllt ist

## $\alpha$ -Konversion

Sei  $t \in \lambda(\Sigma)$  und  $z \notin GV(t) \cup FV(t)$ .

$$(\lambda x. t) \longrightarrow_{\alpha} \lambda z. t[x/z]$$

**Bsp.:** Seien die Symbole gegeben durch  $\Sigma = \{3, a\}$ .

$$\underbrace{(\lambda x. (\lambda y. + xy))}_{GV=\{y\}} \underbrace{(\underbrace{y}_{FV=\{y\}})}_{FV=\{y\}} \longrightarrow_{\alpha} \underbrace{(\lambda x. (\lambda z. + xz))}_{GV=\{z\}} \underbrace{(\underbrace{y}_{FV=\{y\}})}_{FV=\{y\}}$$

# Übungsblatt 6

## *Aufgabe 1*

---

## AUFGABE 1 — TEIL (A)

►  $t_1 = (\lambda x.xy) (\lambda y.y)$ :

## AUFGABE 1 — TEIL (A)

- ▶  $t_1 = (\lambda x.xy) (\lambda y.y)$ :
  - ▶  $FV(t_1) = \{y\}$

## AUFGABE 1 — TEIL (A)

- ▶  $t_1 = (\lambda x.xy) (\lambda y.y)$ :
  - ▶  $FV(t_1) = \{y\}$
  - ▶  $GV(t_1) = \{x, y\}$

## AUFGABE 1 — TEIL (A)

- ▶  $t_1 = (\lambda x.xy) (\lambda y.y)$ :
  - ▶  $FV(t_1) = \{y\}$
  - ▶  $GV(t_1) = \{x, y\}$
- ▶  $t_2 = (\lambda x.(\lambda y.z(\lambda z.z(\lambda x.y))))$

## AUFGABE 1 — TEIL (A)

- ▶  $t_1 = (\lambda x.xy) (\lambda y.y)$ :
  - ▶  $FV(t_1) = \{y\}$
  - ▶  $GV(t_1) = \{x, y\}$
- ▶  $t_2 = (\lambda x.(\lambda y.z(\lambda z.z(\lambda x.y))))$ 
  - ▶  $FV(t_2) = \{z\}$



## AUFGABE 1 — TEIL (A)

- ▶  $t_1 = (\lambda x.xy) (\lambda y.y)$ :
  - ▶  $FV(t_1) = \{y\}$
  - ▶  $GV(t_1) = \{x, y\}$
- ▶  $t_2 = (\lambda x.(\lambda y.z(\lambda z.z(\lambda x.y))))$ 
  - ▶  $FV(t_2) = \{z\}$
  - ▶  $GV(t_2) = \{x, y, z\}$

## AUFGABE 1 — TEIL (A)

- ▶  $t_1 = (\lambda x.xy) (\lambda y.y)$ :
  - ▶  $FV(t_1) = \{y\}$
  - ▶  $GV(t_1) = \{x, y\}$
- ▶  $t_2 = (\lambda x.(\lambda y.z(\lambda z.z(\lambda x.y))))$ 
  - ▶  $FV(t_2) = \{z\}$
  - ▶  $GV(t_2) = \{x, y, z\}$
- ▶  $t_3 = (\lambda x.(\lambda y.z(yz)))(\lambda x.y(\lambda y.y))$

## AUFGABE 1 — TEIL (A)

- ▶  $t_1 = (\lambda x.xy) (\lambda y.y)$ :
  - ▶  $FV(t_1) = \{y\}$
  - ▶  $GV(t_1) = \{x, y\}$
- ▶  $t_2 = (\lambda x.(\lambda y.z(\lambda z.z(\lambda x.y))))$ 
  - ▶  $FV(t_2) = \{z\}$
  - ▶  $GV(t_2) = \{x, y, z\}$
- ▶  $t_3 = (\lambda x.(\lambda y.z(yz)))(\lambda x.y(\lambda y.y))$ 
  - ▶  $FV(t_3) = \{y, z\}$

## AUFGABE 1 — TEIL (A)

- ▶  $t_1 = (\lambda x.xy) (\lambda y.y)$ :
  - ▶  $FV(t_1) = \{y\}$
  - ▶  $GV(t_1) = \{x, y\}$
- ▶  $t_2 = (\lambda x.(\lambda y.z(\lambda z.z(\lambda x.y))))$ 
  - ▶  $FV(t_2) = \{z\}$
  - ▶  $GV(t_2) = \{x, y, z\}$
- ▶  $t_3 = (\lambda x.(\lambda y.z(yz)))(\lambda x.y(\lambda y.y))$ 
  - ▶  $FV(t_3) = \{y, z\}$
  - ▶  $GV(t_3) = \{x, y\}$

## AUFGABE 2 — TEIL (B)

$$\begin{aligned}
 & (\lambda x. \underbrace{(\lambda y. x \ z \ (y \ z))}_{\text{GV}=\{y\}}) \underbrace{(\lambda x. y \ (\lambda y. y))}_{\text{FV}=\{y\}} \\
 \Rightarrow_{\alpha} & (\lambda x. \underbrace{(\lambda y_1. x \ z \ (y_1 \ z))}_{\text{GV}=\{y\}}) \underbrace{(\lambda x. y \ (\lambda y. y))}_{\text{FV}=\{y\}} \\
 \Rightarrow_{\beta} & (\lambda y_1. \underbrace{(\lambda x. y \ (\lambda y. y))}_{\text{GV}=\{y\}}) \underbrace{z}_{\text{FV}=\{z\}} (y_1 \ z) \\
 \Rightarrow_{\beta} & (\lambda y_1. (y \ (\lambda y. y)) (y_1 \ z)) \\
 & = (\lambda y_1. y \ (\lambda y. y) (y_1 \ z))
 \end{aligned}$$

## AUFGABE 2 — TEIL (B)

$$\begin{aligned} & (\lambda x. (\underbrace{\lambda y. (\lambda z. z)}_{\text{GV}=\{y,z\}})) \underbrace{x}_{\text{FV}=\{x\}} (+ y 1) \\ \Rightarrow_{\beta} & (\lambda y. (\underbrace{\lambda z. z}_{\text{GV}=\{z\}})) \underbrace{(+ y 1)}_{\text{FV}=\{y\}} \\ \Rightarrow_{\beta} & (\lambda z. z) \end{aligned}$$

## AUFGABE 2 — TEIL (B)

$$\begin{aligned}
 & (\lambda x. (\lambda y. x (\lambda z. y z))) (((\lambda x. (\lambda y. y)) \underbrace{8}_{\substack{\text{GV}=\{y\} \\ \text{FV}=\emptyset}}) (\lambda x. (\lambda y. y) x)) \\
 \Rightarrow_{\beta} & (\lambda x. (\lambda y. x (\lambda z. y z))) ((\lambda y. y)) (\lambda x. (\lambda y. y) x)) \\
 \Rightarrow_{\beta} & (\lambda x. (\lambda y. x (\lambda z. y z))) ((\lambda y. y) (\lambda x. (\lambda y. \underbrace{y}_{\substack{\text{GV}=\emptyset}} \underbrace{x}_{\{x\}}))) \\
 \Rightarrow_{\beta} & (\lambda x. (\lambda y. x (\lambda z. y z))) ((\lambda y. \underbrace{y}_{\substack{\text{GV}=\emptyset}}) (\lambda x. \underbrace{x}_{\substack{\text{FV}=\emptyset}})) \\
 \Rightarrow_{\beta} & (\lambda x. (\lambda y. x (\lambda z. y z))) (\lambda x. x) \\
 \Rightarrow_{\beta} & (\lambda y. (\lambda x. \underbrace{x}_{\substack{\text{GV}=\emptyset}}) (\lambda z. y z)) \\
 \Rightarrow_{\beta} & (\lambda y. (\lambda z. y z)) = (\lambda y z. y z)
 \end{aligned}$$

## AUFGABE 2 — TEIL (B)

$$(\lambda h. (\lambda x. h (x x)) (\lambda x. h (x x))) ((\lambda x. \underbrace{x}_{GV=\emptyset}) \underbrace{(+ 1 5)}_{FV=\emptyset})$$

$$\Rightarrow_{\beta} (\lambda h. (\lambda x. \underbrace{h (x x)}_{GV=\emptyset}) \underbrace{(\lambda x. h (x x))}_{FV=\{h\}}) (+ 1 5)$$

$$\Rightarrow_{\beta} (\lambda h. h ((\lambda x. \underbrace{h (x x)}_{GV=\emptyset}) \underbrace{(\lambda x. h (x x))}_{FV=\{h\}})) (+ 1 5)$$

$$\Rightarrow_{\beta} (\lambda h. h (h ((\lambda x. \underbrace{h (x x)}_{GV=\emptyset}) \underbrace{(\lambda x. h (x x))}_{FV=\{h\}}))) (+ 1 5)$$

→ endlose Rekursion, bei der  $h$  durch  $(+ 1 5)$  noch reduziert werden könnte

$$\Rightarrow_{\beta} (+ 1 5) ((+ 1 5) ((\lambda x. (+ 1 5) (x x)) (\lambda x. (+ 1 5) (x x))))$$



## AUFGABE 2 — TEIL (B)

$$\begin{aligned}
 & (\lambda f. \underbrace{(\lambda a. (\lambda b. f \ a \ b))}_{GV=\{a,b\}} \underbrace{(\lambda x. (\lambda y. x))}_{FV=\emptyset}) \\
 \Rightarrow_{\beta} & (\lambda a. (\lambda b. (\lambda x. \underbrace{(\lambda y. x)}_{GV=\{y\}} \underbrace{a}_{FV=\{a\}} \ b))) \\
 \Rightarrow_{\beta} & (\lambda a. (\lambda b. (\lambda y. \underbrace{a}_{GV=\emptyset} \ \underbrace{b}_{FV=\{b\}}))) \\
 \Rightarrow_{\beta} & (\lambda a. (\lambda b. a)) \\
 = & (\lambda a b. a)
 \end{aligned}$$

# Übungsblatt 6

## *Aufgabe 2*

---

## AUFGABE 2

(a)  $A$  mit  $A t s u \Rightarrow^* s$ :

## AUFGABE 2

(a)  $A$  mit  $A \text{ t s } u \Rightarrow^* s$ :  $A = (\lambda xyz . y)$

## AUFGABE 2

(a)  $A$  mit  $A \text{ t s } u \Rightarrow^* s$ :  $A = (\lambda xyz . y)$

(b)  $B$  mit  $B \text{ t s } \Rightarrow^* s \text{ t}$ :

## AUFGABE 2

(a)  $A$  mit  $A \text{ t s } u \Rightarrow^* s$ :  $A = (\lambda xyz . y)$

(b)  $B$  mit  $B \text{ t s } \Rightarrow^* s \text{ t}$ :  $B = (\lambda xy . yx)$

## AUFGABE 2

(a)  $A$  mit  $A t s u \Rightarrow^* s$ :  $A = (\lambda xyz . y)$

(b)  $B$  mit  $B t s \Rightarrow^* s t$ :  $B = (\lambda xy . yx)$

(c)  $C$  mit  $C C \Rightarrow^* C C$ :

## AUFGABE 2

(a)  $A$  mit  $A t s u \Rightarrow^* s$ :  $A = (\lambda xyz . y)$

(b)  $B$  mit  $B t s \Rightarrow^* s t$ :  $B = (\lambda xy . yx)$

(c)  $C$  mit  $C C \Rightarrow^* C C$ :  $C = (\lambda x . xx)$



## AUFGABE 2

(a)  $A$  mit  $A t s u \Rightarrow^* s$ :  $A = (\lambda x y z . y)$

(b)  $B$  mit  $B t s \Rightarrow^* s t$ :  $B = (\lambda x y . y x)$

(c)  $C$  mit  $C C \Rightarrow^* C C$ :  $C = (\lambda x . x x)$

denn:  $(\lambda x . \underbrace{xx}_{GV=\emptyset}) (\underbrace{\lambda x . xx}_{FV=\emptyset}) \Rightarrow^\beta (\lambda x . xx)(\lambda x . xx)$

## AUFGABE 2

(a)  $A$  mit  $A \text{ t s } u \Rightarrow^* s$ :  $A = (\lambda x y z . y)$

(b)  $B$  mit  $B \text{ t s } \Rightarrow^* s \text{ t}$ :  $B = (\lambda x y . y x)$

(c)  $C$  mit  $C C \Rightarrow^* C C$ :  $C = (\lambda x . x x)$

denn:  $(\lambda x . \underbrace{xx}_{GV=\emptyset}) (\underbrace{\lambda x . xx}_{FV=\emptyset}) \Rightarrow^\beta (\lambda x . xx)(\lambda x . xx)$

(d)  $D$  mit  $D \Rightarrow^* D$ :

## AUFGABE 2

(a)  $A$  mit  $A t s u \Rightarrow^* s$ :  $A = (\lambda x y z . y)$

(b)  $B$  mit  $B t s \Rightarrow^* s t$ :  $B = (\lambda x y . y x)$

(c)  $C$  mit  $C C \Rightarrow^* C C$ :  $C = (\lambda x . x x)$

denn:  $(\lambda x . \underbrace{xx}_{GV=\emptyset}) (\underbrace{\lambda x . xx}_{FV=\emptyset}) \Rightarrow^\beta (\lambda x . xx)(\lambda x . xx)$

(d)  $D$  mit  $D \Rightarrow^* D$ :  $D = (C C)$

## AUFGABE 2

(a)  $A$  mit  $A t s u \Rightarrow^* s$ :  $A = (\lambda x y z . y)$

(b)  $B$  mit  $B t s \Rightarrow^* s t$ :  $B = (\lambda x y . y x)$

(c)  $C$  mit  $C C \Rightarrow^* C C$ :  $C = (\lambda x . x x)$

denn:  $(\lambda x . \underbrace{xx}_{GV=\emptyset}) (\underbrace{\lambda x . xx}_{FV=\emptyset}) \Rightarrow^\beta (\lambda x . xx)(\lambda x . xx)$

(d)  $D$  mit  $D \Rightarrow^* D$ :  $D = (C C)$

(e)  $E$  mit  $E E t \Rightarrow^* E t E$ :

## AUFGABE 2

(a)  $A$  mit  $A t s u \Rightarrow^* s$ :  $A = (\lambda x y z . y)$

(b)  $B$  mit  $B t s \Rightarrow^* s t$ :  $B = (\lambda x y . y x)$

(c)  $C$  mit  $C C \Rightarrow^* C C$ :  $C = (\lambda x . x x)$

denn:  $(\lambda x . \underbrace{xx}_{GV=\emptyset}) (\underbrace{\lambda x . xx}_{FV=\emptyset}) \Rightarrow^\beta (\lambda x . xx)(\lambda x . xx)$

(d)  $D$  mit  $D \Rightarrow^* D$ :  $D = (C C)$

(e)  $E$  mit  $E E t \Rightarrow^* E t E$ :  $E = (\lambda x y . x y x)$

## AUFGABE 2

(a)  $A$  mit  $A t s u \Rightarrow^* s$ :  $A = (\lambda x y z . y)$

(b)  $B$  mit  $B t s \Rightarrow^* s t$ :  $B = (\lambda x y . y x)$

(c)  $C$  mit  $C C \Rightarrow^* C C$ :  $C = (\lambda x . x x)$

denn:  $(\lambda x . \underbrace{xx}_{GV=\emptyset}) (\underbrace{\lambda x . xx}_{FV=\emptyset}) \Rightarrow^\beta (\lambda x . xx)(\lambda x . xx)$

(d)  $D$  mit  $D \Rightarrow^* D$ :  $D = (C C)$

(e)  $E$  mit  $E E t \Rightarrow^* E t E$ :  $E = (\lambda x y . x y x)$

denn:

$$\begin{aligned} (\underbrace{\lambda x y . x y x}_{GV=\{y\}}) (\underbrace{\lambda x y . x y x}_{FV=\emptyset}) t &\Rightarrow^\beta (\lambda y . (\lambda x y . x y x) y (\lambda x y . x y x)) t \\ &\Rightarrow^\beta (\underbrace{\lambda x y . x y x}_{=E}) t (\underbrace{\lambda x y . x y x}_{=E}) \end{aligned}$$

## AUFGABE 2

(a)  $A$  mit  $A \ t \ s \ u \Rightarrow^* s$ :  $A = (\lambda x y z . y)$

(b)  $B$  mit  $B \ t \ s \Rightarrow^* s \ t$ :  $B = (\lambda x y . y x)$

(c)  $C$  mit  $C \ C \Rightarrow^* C \ C$ :  $C = (\lambda x . x x)$   
 denn:  $(\lambda x . \underbrace{xx}_{GV=\emptyset}) (\underbrace{\lambda x . xx}_{FV=\emptyset}) \Rightarrow^\beta (\lambda x . xx)(\lambda x . xx)$

(d)  $D$  mit  $D \Rightarrow^* D$ :  $D = (C \ C)$

(e)  $E$  mit  $E \ E \ t \Rightarrow^* E \ t \ E$ :  $E = (\lambda x y . x y x)$   
 denn:

$$\begin{aligned} & (\underbrace{\lambda x y . x y x}_{GV=\{y\}}) (\underbrace{\lambda x y . x y x}_{FV=\emptyset}) t \\ & \Rightarrow^\beta \underbrace{(\lambda x y . x y x)}_{=E} t \underbrace{(\lambda x y . x y x)}_{=E} \end{aligned}$$

## AUFGABE 2

(a)  $A$  mit  $A t s u \Rightarrow^* s$ :  $A = (\lambda x y z . y)$

(b)  $B$  mit  $B t s \Rightarrow^* s t$ :  $B = (\lambda x y . y x)$

(c)  $C$  mit  $C C \Rightarrow^* C C$ :  $C = (\lambda x . x x)$

denn:  $(\lambda x . \underbrace{xx}_{GV=\emptyset}) (\underbrace{\lambda x . xx}_{FV=\emptyset}) \Rightarrow^\beta (\lambda x . xx)(\lambda x . xx)$

(d)  $D$  mit  $D \Rightarrow^* D$ :  $D = (C C)$

(e)  $E$  mit  $E E t \Rightarrow^* E t E$ :  $E = (\lambda x y . x y x)$

denn:

$$\begin{aligned} (\underbrace{\lambda x y . x y x}_{GV=\{y\}}) (\underbrace{\lambda x y . x y x}_{FV=\emptyset}) t &\Rightarrow^\beta (\lambda y . (\lambda x y . x y x) y (\lambda x y . x y x)) t \\ &\Rightarrow^\beta \underbrace{(\lambda x y . x y x)}_{=E} t \underbrace{(\lambda x y . x y x)}_{=E} \end{aligned}$$