Lambda-Kalkül und Kombinatorische Logik

Peter Schroeder-Heister

Sommersemester 1997

Skriptum von Michael Arndt

1997, 2012, 2014
Universität Tübingen
Wilhelm-Schickard-Institut für Informatik

Inhaltsverzeichnis

Vorwort

Dies ist ein Skriptum zu einer Vorlesung, die ich zuletzt im Sommersemester 1997 gehalten habe. In den ersten beiden Teilen orientiert es sich im wesentlichen am klassischen Lehrbuch von Hindley und Seldin, in den letzten beiden Teilen an Barendregts Kapitel über den getypten λ-Kalkül im *Handbook of Logic in Computer Science* (Band II). Das Skriptum soll zur Orientierung über das technische Gerüst des Themas dienen. Dementsprechend ist es nicht bis in alle Einzelheiten ausgearbeitet. So wurde auf Stilfragen wenig Rücksicht genommen. Auch wurden elementare, aber langwierige Beweise häufig weggelassen. Erläuternde Passagen zu Sinn und Zweck des λ-Kalküls sowie einzelner Begriffsbildungen sind ebenfalls nicht aufgezeichnet. Hierzu seien Leser auf die genannten Texte verwiesen.

Ich danke Michael Arndt für die Erstellung des Skriptums. Frau Natali Alt und Herrn Reinhard Kahle danke ich für eine kritische Durchsicht des Textes. Alle verbleibenden inhaltlichen Fehler gehen natürlich zu meinen Lasten.

Peter Schroeder-Heister

1 Der ungetypte λ-Kalkül

1.1 Syntax und operationelle Semantik

Gegeben sei eine unendliche Folge von Variablen. (Es ist wichtig, daß eine feste Reihenfolge angenommen wird.) Die metasprachlichen Zeichen dafür seien $x, y, z, x_1, x_2, x_3, \ldots$
Man unterscheidet zwischen zwei Varianten des ungetypten λ-Kalküls:

- dem reinen λ-Kalkül, bei dem keine Konstanten gegeben sind.
- dem angewandten λ-Kalkül, bei dem zusätzlich eine endliche oder unendliche Menge von Konstanten gegeben ist.

(In den ersten zwei Kapiteln wird nur der ungetypte λ-Kalkül behandelt. Daher wird die Bezeichnung "ungetypt" immer weggelassen.)

Definition 1.1 (Syntax)**.**

- *Alle Variablen und Konstanten sind λ-Terme (*"Atome"*)*
- *Mit M und N ist auch (MN) ein λ-Term (*"Applikation"*) mit M und N als unmittelbaren Teiltermen*
- *Mit M ist auch $(\lambda x.M)$ ein λ-Term (*"Abstraktion"*) mit x und M als unmittelbaren Teiltermen*

Die Länge *eines Termes M ist die Anzahl der Vorkommen von Atomen in M.*
Teilterme *eines Terms sind dieser Term selbst, sowie die Teilterme seiner unmittelbaren Teilterme. Alle Teilterme eines Termes mit Ausnahme seiner selbst sind dessen* echte *Teilterme.*
Man schreibt $M[P]$, wenn P an einer bestimmten Stelle als Teilterm in M vorkommt.
Im Kontext von $M[P]$ bedeute $M[Q]$, daß man das in $M[P]$ gemeinte Vorkommen von P in M durch Q ersetzt.
Ein Vorkommen einer Variable x in einem Term M ist gebunden*, falls es zu einem Teilterm $\lambda x.P$ von M gehört, ansonsten ist es* frei.
Falls x ein freies Vorkommen in M hat, heißt x freie Variable *von M. Die Menge dieser freien Variablen sei $FV(M)$.*
M heißt geschlossen*, wenn $FV(M) = \emptyset$.*
Ein geschlossener Term ohne Konstanten heißt Kombinator.
Metasprachliche Variablen: $M, N, P, Q, R, S, T, \ldots$ *für λ-Terme; $a, b, c, \ldots$ für Atome.*
Außenklammern können wegfallen. Bei Klammerung gilt Linksassoziation, d.h. $MNPQ$ meint $((MN)P)Q$. Ferner steht $\lambda x.MN$ für $(\lambda x.(MN))$, $\lambda x_1 \ldots\ x_n.M$ für $\lambda x_1.\lambda x_2.\ldots\lambda x_n.M$.
$M \doteq N$ bezeichne die syntaktische Identität von M und N.

Beispiel (Grammatik für Terme).
Die Terme des reinen λ-Kalküls können durch folgende kontextfreie Grammatik charakterisiert werden, wenn Variablen die Form $v'^{\cdots}{}'$ haben:

- *Terminalalphabet:* $\{\lambda, ., (,), v, '\}$
- *Nichtterminalalphabet:* $\{T, V\}$
- *Startsymbol:* T
- *Produktionen:* $\begin{array}{lll} T & \longrightarrow & V \mid (TT) \mid (\lambda V.T) \\ V & \longrightarrow & v \mid V' \end{array}$

Beispiel (Kombinatoren).

- $\mathbf{I} \simeq_{\mathrm{def}} \lambda x.x$
- $\mathbf{K} \simeq_{\mathrm{def}} \lambda xy.x$
- $\mathbf{S} \simeq_{\mathrm{def}} \lambda xyz.xz(yz)$

Definition 1.2 (Substitution).

1. $x[N/x] \simeq_{\mathrm{def}} N$
2. $a[N/x] \simeq_{\mathrm{def}} a$, *falls* $x \neq a$
3. $(PQ)[N/x] \simeq_{\mathrm{def}} (P[N/x]Q[N/x])$
4. $(\lambda x.P)[N/x] \simeq_{\mathrm{def}} \lambda x.P$
5. $(\lambda y.P)[N/x] \simeq_{\mathrm{def}} \lambda y.P[N/x]$, *falls* $x \neq y$ *und nicht:* $y \in FV(N)$ *und* $x \in FV(P)$
6. $(\lambda y.P)[N/x] \simeq_{\mathrm{def}} \lambda z.P[z/y][N/x]$, *falls* $x \neq y$ *und* $y \in FV(N)$ *und* $x \in FV(P)$, *wobei* z *die erste Variable (in der Aufzählung aller Variablen) mit* $z \notin FV(NP)$

Beispiel.
$(\lambda y.x)[y/x] \simeq \lambda z.y$ *(falls* z *erste von* x *und* y *verschiedene Variable)*

Lemma 1.3. *Für alle* λ*-Terme*M, N *und alle Variablen* x *gilt:*

1. $M[x/x] \simeq M$
2. *Wenn* $x \notin FV(M)$, *dann* $M[N/x] \simeq M$.
3. *Wenn* $x \in FV(M)$, *dann* $FV(M[N/x]) = FV(N) \cup (FV(M) \setminus \{x\})$.

Lemma 1.4. *Es sei keine im λ-Term M gebundene Variable frei in λ-Termen P, Q und z. Dann gilt:*

1. *Wenn $z \notin FV(M)$, dann $M[z/x][P/z] \simeq M[P/x]$.*
2. *Wenn $z \notin FV(M)$, dann $M[z/x][x/z] \simeq M$.*
3. $M[Q/x][P/x] \simeq M[(Q[P/x])/x]$
4. *Wenn $y \notin FV(P)$, dann $M[Q/y][P/x] \simeq M[P/x][(Q[P/x])/y]$.*
5. *Wenn $y \notin FV(P)$ und $x \notin FV(Q)$, dann $M[Q/y][P/x] \simeq M[P/x][Q/y]$.*

Definition 1.5 (α-Konversion, Kongruenz)**.**
Falls $y \notin FV(M)$, so sei $P[\lambda x.M] \equiv_{1\alpha} P[\lambda y.M[y/x]]$ "gebundene Umbenennung"
$P \equiv_\alpha Q$, *falls* $P \simeq P_1 \equiv_{1\alpha} P_2 \equiv_{1\alpha} \cdots \equiv_{1\alpha} P_n \simeq Q$ "α-Konversion", "Kongruenz"

Lemma 1.6.

1. *Wenn $P \equiv_\alpha Q$, dann $FV(P) = FV(Q)$.*
2. *Für jeden Term P und alle Variablen $x_1, \ldots, x_n$ existiert ein Term P' mit $P \equiv_\alpha P'$, wobei kein $x_1, \ldots, x_n$ gebunden in P' vorkommt.*
3. *$\equiv_\alpha$ ist Äquivalenzrelation*

Beweis.
Übung □

Lemma 1.7 (Kongruenz von $\equiv_\alpha$)**.**
Wenn $M \equiv_\alpha M'$ und $N \equiv_\alpha N'$, dann $M[N/x] \equiv_\alpha M'[N'/x]$.

Definition 1.8 (β-Kontraktion, β-Reduktion, β-Konversion)**.**
$P[\underbrace{(\lambda x.M)N}_{\textit{Redex}}] \rhd_{1\beta} P[\underbrace{M[N/x]}_{\textit{Kontraktum}}]$ "β-Kontraktion"

$P \rhd_\beta Q$, *falls* $P \simeq P_1 \begin{smallmatrix}\rhd_{1\beta}\\ \equiv_{1\alpha}\end{smallmatrix} P_2 \begin{smallmatrix}\rhd_{1\beta}\\ \equiv_{1\alpha}\end{smallmatrix} \cdots \begin{smallmatrix}\rhd_{1\beta}\\ \equiv_{1\alpha}\end{smallmatrix} P_n \simeq Q$ "β-Reduktion"

Falls $P \simeq P_1 \rhd_{1\beta} P_2 \rhd_{1\beta} P_3 \rhd_{1\beta} \cdots$, dann heißt $(P_1, P_2, P_3, \ldots)$ β-Reduktionsfolge *von P.*

$P =_\beta Q$, *falls* $P \simeq P_1 \begin{smallmatrix}\equiv_{1\alpha}\\ \rhd_{1\beta}\\ \lhd_{1\beta}\end{smallmatrix} P_2 \begin{smallmatrix}\equiv_{1\alpha}\\ \rhd_{1\beta}\\ \lhd_{1\beta}\end{smallmatrix} \cdots \begin{smallmatrix}\equiv_{1\alpha}\\ \rhd_{1\beta}\\ \lhd_{1\beta}\end{smallmatrix} P_n \simeq Q$ "β-Konversion", "β-Gleichheit"

P ist in β-Normalform, *falls P kein β-Redex enthält.*
Falls $P \rhd_\beta Q$ und Q in β-Normalform ist, dann heißt Q eine β-Normalform von P.

P heißt (schwach) normalisierbar*, wenn es eine β-Normalform von P gibt.*
P heißt stark normalisierbar*, wenn es keine unendliche β-Reduktionsfolge von P gibt.*

Beispiel.

- $(\lambda x.(\lambda y.yx)z)v \rhd_{1\beta} (\lambda x.zx)v \rhd_{1\beta} zv$
  zv ist β-Normalform von $(\lambda x.(\lambda y.yx)z)v$.

- $\mathbf{\Omega} \simeq_{\text{def}} (\lambda x.xx)(\lambda x.xx)$ *hat keine β-Normalform:*
  $(\lambda x.xx)(\lambda x.xx) \rhd_{1\beta} (\lambda x.xx)(\lambda x.xx) \rhd_{1\beta} \cdots$
  Allerdings können λ-Terme, die den $\mathbf{\Omega}$-Kombinator enthalten, eine β-Normalform besitzen:
  $(\lambda x.y)\mathbf{\Omega} \rhd_{1\beta} y$. *Der Term $(\lambda x.y)\mathbf{\Omega}$ ist also schwach normalisierbar, jedoch nicht stark normalisierbar, da es eine unendliche Reduktionsfolge gibt:* $(\lambda x.y)\mathbf{\Omega} \rhd_{1\beta} (\lambda x.y)\mathbf{\Omega} \rhd_{1\beta} \ldots$

Lemma 1.9.

- *Wenn $P \equiv_\alpha P'$, $Q \equiv_\alpha Q'$, $P \overset{\rhd_\beta}{=_\beta} Q$, dann $P' \overset{\rhd_\beta}{=_\beta} Q'$*

- *Wenn $M \overset{\rhd_\beta}{=_\beta} N$ und $P \overset{\rhd_\beta}{=_\beta} Q$, dann $P[M/x] \overset{\rhd_\beta}{=_\beta} Q[N/x]$*

Lemma 1.10.
Die Klasse aller β-Normalformen läßt sich induktiv definieren durch folgende Regeln:

1. *Jedes Atom ist eine β-Normalform.*

2. *Mit $M_1, \ldots, M_n$ ist auch $aM_1 \ldots M_n$ eine β-Normalform.*

3. *Mit M ist auch $\lambda x.M$ eine β-Normalform.*

Das heißt, eine β-Normalform hat die Form $\lambda x_1 \ldots x_n.aM_1 \ldots M_m$, wobei die M_i dieselbe Form haben.

Beweis.
Sei M eine β-Normalform. Falls $M \simeq a$, so läßt sich M nach *1.* erzeugen. Falls $M \simeq (PQ)$, dann sind nach Induktionsvoraussetztung P und Q mit den Regeln *1.–3.* zu erzeugen, wobei P keine Abstraktion ist. Also ist $P \simeq a$ oder $P \simeq aM_1 \ldots M_k$. Damit ist $M \simeq aQ$ oder $M \simeq aM_1 \ldots M_kQ$. Das läßt sich nach *2.* erzeugen. Falls $M \simeq \lambda x.P$, dann läßt sich P nach Induktionsvoraussetztung aus *1.–3.* erzeugen, daraus also auch M mit *3.*.
Falls umgekehrt M mit den Regeln *1.–3.* erzeugt ist, dann ist offensichtlich, daß M in β-Normalform ist. □

Lemma 1.11.
Ein beliebiger λ-Term hat entweder die in Lemma 1.10 genannte Form, oder er enthält einen Teilterm der Form $\lambda x_1 \ldots x_n.\underbrace{(\lambda x.M)N}_{Kopfredex} M_1 \ldots M_m$ $(m, n \geq 0)$.

Beweis.
Betrachte den ausgeschlossenen Unterfall für $M \bumpeq (PQ)$ im Beweis des vorigen Lemmas. □

Theorem 1.12 (Church-Rosser)**.**

1. *Wenn $P \rhd_\beta M$ und $P \rhd_\beta N$, dann existiert ein Term T, so daß $M \rhd_\beta T$ und $N \rhd_\beta T$.*
2. *Wenn $M =_\beta N$, so existiert ein Term T, so daß $M \rhd_\beta T$ und $N \rhd_\beta T$.*

Beispiel.

$$\begin{tikzcd}
 & (\lambda x.(\lambda y.yx)z)v \arrow[dl, "\rhd_{1\beta}"'] \arrow[dr, "\rhd_{1\beta}"] & \\
(\lambda x.zx)v \arrow[dr, "\rhd_{1\beta}"'] & & (\lambda y.yv)z \arrow[dl, "\rhd_{1\beta}"] \\
 & zv &
\end{tikzcd}$$

Beweis von *2.*
Der Beweis erfolgt durch Induktion über der Anzahl der Schritte von M nach N.

Anzahl $=0$: trivial

Anzahl $=n+1$:

$$\begin{tikzcd}
M \bumpeq P_1 \arrow[dr, "\rhd_\beta"'] & \cdots & P_n \arrow[dl, "\rhd_\beta"] \arrow[r, phantom, "\substack{\equiv_{1\alpha} \\ \rhd_{1\beta} \\ \lhd_{1\beta}}"] & P_{n+1} \bumpeq N \\
 & T' & &
\end{tikzcd}$$

Fall 1 ($\equiv_{1\alpha}$): $T \bumpeq_{\text{def}} T'$, da $P_{n+1} \equiv_{1\alpha} P_n \rhd_\beta T'$.

Fall 2 ($\lhd_{1\beta}$): $T \bumpeq_{\text{def}} T'$, da $P_{n+1} \rhd_{1\beta} P_n \rhd_\beta T'$.

Fall 3 ($\rhd_{1\beta}$): Da $P_n \rhd_\beta T'$ und $P_n \rhd_{1\beta} P_{n+1}$, existiert nach *1.* ein T mit $T' \rhd_\beta T$, und $P_{n+1} \rhd_\beta T$, also $P_1 \rhd_\beta T$ und $P_n \rhd_\beta T$.

□

Für den Beweis von *1.* werden noch einige Definitionen und Lemmata benötigt.

Definition 1.13.
Seien R und S mit β-Redexe in P mit $R \neq S$, wobei $R \rhd_{1\beta} R'$.
Das Residuum $\mathrm{Res}(S,R)$ *von S bezüglich R ist wie folgt definiert:*

1. *R ist (echter) Teilterm von S. Dann sei* $\mathrm{Res}(S,R) =_{\mathrm{def}} S[R']$.

2. *R ist kein Teilterm von S. Dann sei* $\mathrm{Res}(S,R) =_{\mathrm{def}} S$.

Bemerkung.
$\mathrm{Res}(S,R)$ ist die Gestalt von S nach Kontraktion von R. Aufgrund der Verwendung von $\mathrm{Res}(S,R)$ können einige Fälle unberücksichtigt bleiben.

Definition 1.14.
Sei $\mathcal{R} = \{R_1, \ldots, R_n\}$ eine Menge von Redexen in P. R_i heißt minimal*, falls kein R_j echter Teilterm von R_i ist.*
$P \rhd_{\mathrm{mcd}} Q$, falls Q aus P durch folgendes (nichtdeterministisches) Verfahren hervorgeht:

1. *Es wird ein minimales Element R_i in $\mathcal{R}$ gewählt.*

2. *R_i wird in P β-kontrahiert: $P \rhd_{1\beta} P'$.*

3. *Es sei $\mathcal{R}' = \bigcup_{j \neq i} \mathrm{Res}(R_j, R_i)$ ($\mathcal{R}'$ hat damit $n-1$ Elemente).*

4. *Falls $\mathcal{R}' \neq \emptyset$, dann beginne bei Schritt* 1. *mit $\mathcal{R} =_{\mathrm{def}} \mathcal{R}'$ und $P \doteq_{\mathrm{def}} P'$.*

5. *Falls $\mathcal{R}' = \emptyset$, dan sei $Q \equiv_\alpha P'$.*

Bemerkungen.

1. “mcd” steht für “minimal complete development”.

2. $\rhd_{\mathrm{mcd}}$ ist stets relativ zu einer gewählten Menge $\mathcal{R}$ von Redexen zu verstehen, wobei $\mathcal{R}$ für jede $\rhd_{\mathrm{mcd}}$-Beziehung verschieden sein kann. Man müßte also genaugenommen von $\rhd^{\mathcal{R}}_{\mathrm{mcd}}$ sprechen. Allerdings würde dies die Darstellungen der folgenden Beweise erschweren. Es wird im Folgenden davon ausgegangen, daß die Menge von Redexen jeweils geeignet gewählt ist, d.h. daß alle Redexe des jeweiligen Termes in $\mathcal{R}$ enthalten sind.

3. Aus diesem Grund ist $\rhd_{\mathrm{mcd}}$ nicht transitiv. (Siehe folgendes Beispiel.)

4. Für eine leere Menge von Redexen ergibt sich als Grenzfall $P \rhd_{\mathrm{mcd}} P$.

Beispiel.
$(\lambda x.xy)(\lambda x.x) \rhd_{\mathrm{mcd}} (\lambda x.x)y$ und $(\lambda x.x)y \rhd_{\mathrm{mcd}} y$, aber $(\lambda x.xy)(\lambda x.x) \not\rhd_{\mathrm{mcd}} y$.

Lemma 1.15.
Wenn $P \triangleright_{\mathrm{mcd}} Q$ und $P \equiv_\alpha P'$, dann $P' \triangleright_{\mathrm{mcd}} Q$.

Lemma 1.16.
Wenn $M \triangleright_{\mathrm{mcd}} M'$ und $N \triangleright_{\mathrm{mcd}} N'$, dann $M[N/x] \triangleright_{\mathrm{mcd}} M'[N'/x]$.

Lemma 1.17.
Wenn $P \triangleright_{\mathrm{mcd}} Q$ und $P \triangleright_{\mathrm{mcd}} R$, so existiert ein Term T, so daß $Q \triangleright_{\mathrm{mcd}} T$ und $R \triangleright_{\mathrm{mcd}} T$.

Beweis.
Aufgrund von *Lemma 1.15* können wir annehmen, daß in den gegebenen mcd-Reduktionen keine α-Schritte vorkommen. Der Beweis erfolgt durch Induktion über der Struktur von P:

1. $P \bumpeq a$: $P \bumpeq Q \bumpeq R \bumpeq T$

2. $P \bumpeq \lambda x.P_1$, d.h. $Q \bumpeq \lambda x.Q_1$ und $R \bumpeq \lambda x.R_1$ (keine α-Schritte!)
   $P_1 \triangleright_{\mathrm{mcd}} Q_1$ und $P_1 \triangleright_{\mathrm{mcd}} R_1$. Nach Induktionsvoraussetzung existiert T_1, so daß $Q_1 \triangleright_{\mathrm{mcd}} T_1$ und $R_1 \triangleright_{\mathrm{mcd}} T_1$. Setze $T \bumpeq_{\mathrm{def}} \lambda x.T_1$.

3. $P \bumpeq (P_1P_2)$ und alle Redexe von $\mathcal{R}$ sind in P_1 und P_2, dh. P selbst, wird nicht reduziert.
   Dann gilt nach Induktionsvoraussetzung mit

$$
\begin{tikzcd}
 & P_1 \arrow[dl, "\triangleright_{\mathrm{mcd}}"'] \arrow[dr, "\triangleright_{\mathrm{mcd}}"] & \\
Q_1 \arrow[dr, "\triangleright_{\mathrm{mcd}}"'] & & R_1 \arrow[dl, "\triangleright_{\mathrm{mcd}}"] \\
 & T_1 &
\end{tikzcd}
\quad \text{und} \quad
\begin{tikzcd}
 & P_2 \arrow[dl, "\triangleright_{\mathrm{mcd}}"'] \arrow[dr, "\triangleright_{\mathrm{mcd}}"] & \\
Q_2 \arrow[dr, "\triangleright_{\mathrm{mcd}}"'] & & R_2 \arrow[dl, "\triangleright_{\mathrm{mcd}}"] \\
 & T_2 &
\end{tikzcd}
\quad \text{auch} \quad
\begin{tikzcd}
 & (P_1P_2) \arrow[dl, "\triangleright_{\mathrm{mcd}}"'] \arrow[dr, "\triangleright_{\mathrm{mcd}}"] & \\
(Q_1Q_2) \arrow[dr, "\triangleright_{\mathrm{mcd}}"'] & & (R_1R_2) \arrow[dl, "\triangleright_{\mathrm{mcd}}"] \\
 & (T_1T_2) &
\end{tikzcd}
$$

   Setze $T \bumpeq_{\mathrm{def}} (T_1T_2)$.

4. $P \bumpeq ((\lambda x.M)N)$ und das Residuum von P wird bei $P \triangleright_{\mathrm{mcd}} Q$, nicht jedoch bei $P \triangleright_{\mathrm{mcd}} R$ kontrahiert, d.h.
   $P \bumpeq (\lambda x.M)N \triangleright_{\mathrm{mcd}} (\lambda x.M')N' \triangleright_{1\beta} M'[N'/x] \bumpeq Q \qquad (M \triangleright_{\mathrm{mcd}} M'$ und $N \triangleright_{\mathrm{mcd}} N')$
   $P \bumpeq (\lambda x.M)N \triangleright_{\mathrm{mcd}} (\lambda x.M'')N'' \bumpeq R \qquad (M \triangleright_{\mathrm{mcd}} M''$ und $N \triangleright_{\mathrm{mcd}} N'')$
   Nach Induktionsvoraussetzung existieren M^+ und N^+ derart, daß $M', M'' \triangleright_{\mathrm{mcd}} M^+$ und $N', N'' \triangleright_{\mathrm{mcd}} N^+$.
   Setze $T \bumpeq_{\mathrm{def}} M^+[N^+/x]$, dann ist $Q \bumpeq M'[N'/x] \triangleright_{\mathrm{mcd}} M^+[N^+/x]$ nach Lemma 1.16.
   Ferner $(\lambda x.M'')N'' \triangleright_{\mathrm{mcd}} (\lambda x.M^*)N^* \triangleright_{1\beta} M^*[N^*/x] \equiv_\alpha M^+[N^+/x]$, wobei wir annehmen, daß ohne α-Schritte $M'' \triangleright_{\mathrm{mcd}} M^*$, $N'' \triangleright_{\mathrm{mcd}} N^*$, und $M^* \equiv_\alpha M^+$, $M^* \equiv_\alpha M^+$.

5. $P \bumpeq ((\lambda x.M)N)$ und beide mcd-Reduktionen kontrahieren das Residuum von P, d.h.
   $P \bumpeq (\lambda x.M)N \rhd_{\text{mcd}} (\lambda x.M')N' \rhd_{1\beta} M'[N'/x] \bumpeq Q$
   $P \bumpeq (\lambda x.M)N \rhd_{\text{mcd}} (\lambda x.M'')N'' \rhd_{1\beta} M''[N''/x] \bumpeq R$
   Wir argumentieren wie in Fall *4.* und setzen wieder $T \bumpeq_{\text{def}} M^+[N^+/x]$. Mit *Lemma 1.16* ergibt sich dann die Behauptung.

□

Bemerkung.
Der Beweis der lokalen Church-Rosser-Eigenschaft für $\rhd_{\text{mcd}}$ beruht wesentlich darauf, daß alle Redexe bereits im Ausgangsterm vorliegen und dadurch kontrolliert werden können.

Beweis von Theorem 1.12 (1.)
Zunächst ergibt sich aus dem vorhergehenden Lemma durch Induktion:
Wenn $P \rhd_{\text{mcd}} M$ und $P \rhd_\beta N$, dann existiert ein Term T, so daß $M \rhd_\beta T$ und $N \rhd_{\text{mcd}} T$.

$$
\begin{tikzcd}
 & P \arrow[dl, "\rhd_{\text{mcd}}"'] \arrow[dr, "\substack{\rhd_{1\beta}\\ \equiv_\alpha}"] & & & \\
M \arrow[dr, "\rhd_{\text{mcd}}"'] & & N_1 \arrow[dl, "\rhd_{\text{mcd}}"] \arrow[dr, "\substack{\rhd_{1\beta}\\ \equiv_\alpha}"] & & \\
 & T_1 \arrow[dr, "\rhd_{\text{mcd}}"'] & & N_2 \arrow[dl, "\rhd_{\text{mcd}}"] \arrow[dr, "\cdots"] & \\
 & & T_2 \arrow[dr, "\cdots"'] & & N_n \bumpeq N \arrow[dl, "\rhd_{\text{mcd}}"] \\
 & & & T_n \bumpeq T &
\end{tikzcd}
$$

Beachte, daß mit $\rhd_{1\beta}$ auch $\rhd_{\text{mcd}}$ gilt, und mit $\rhd_{\text{mcd}}$ auch $\rhd_\beta$, und daß $\rhd_\beta$ transitiv ist.
Daraus ergibt sich:
Wenn $P \rhd_{1\beta} M$ und $P \rhd_\beta N$, dann existiert ein Term T, so daß $M \rhd_\beta T$ und $N \rhd_\beta T$.
Hieraus folgt dann sofort durch Induktion die Behauptung des Theorems.

$$
\begin{tikzcd}
 & & & P \arrow[dl, "\substack{\rhd_{1\beta} \\ \equiv_\alpha}"'] \arrow[dr, "\rhd_\beta"] & \\
 & & P_1 \arrow[dl, "\substack{\rhd_{1\beta} \\ \equiv_\alpha}"'] \arrow[dr, "\rhd_\beta"'] & & N \arrow[dl, "\rhd_\beta"] \\
 & P_2 \arrow[dl, "\cdots"'] \arrow[dr, "\rhd_\beta"'] & & T_1 \arrow[dl, "\rhd_\beta"] & \\
M \arrow[dr, "\rhd_\beta"'] & & T_2 \arrow[dl, "\cdots"] & & \\
 & T & & &
\end{tikzcd}
$$

□

Korollar 1.18.

1. *Falls M und N β-Normalformen von P sind, dann $M \equiv_\alpha N$.*
2. *Wenn $M =_\beta N$ und N β-Normalform ist, dann $M \rhd_\beta N$.*
3. *Falls $M =_\beta N$, dann gilt: M und N haben beide keine, oder beide haben dieselbe β-Normalform (bis auf Kongruenz).*
4. *β-gleiche Terme in β-Normalform sind kongruent.*

Definition 1.19.
Eine L-Reduktionsfolge *ist eine β-Reduktionsfolge, bei der immer das am weitesten links stehende Redex im jeweiligen Term kontrahiert wird. Ein Redex $(\lambda x_1.M_1)N_1$ ist dabei* weiter links stehend *als $(\lambda x_2.M_2)N_2$ (im betrachteten Term), falls sich λx_1 links von λx_2 befindet.*
Eine QL-Reduktionsfolge *ist eine β-Reduktionsfolge $(M_1, M_2, M_3, \ldots)$, so daß es zu jedem M_i, das nicht letztes Glied der Folge ist, ein M_j und ein M_{j+1} mit $j \geq i$ gibt, so daß beim Übergang von M_j zu M_{j+1} das linkeste Redex in M_j kontrahiert wird.*

Bemerkung.
L steht für "leftmost", QL für "quasi-leftmost". Eine QL-Reduktionsfolge ist also eine β-Reduktionsfolge, bei der "immer wieder" das linkeste Redex kontrahiert wird.

Theorem 1.20.
Falls ein λ-Term M eine β-Normalform hat, dann terminiert jede mit M beginnende L-Reduktionsfolge (und damit auch jede QL-Reduktionsfolge).

Bemerkung.
Für die Beweise siehe Barendregt (1980), Abschnitt 13.2.

Theorem 1.21.
Es gibt Fixpunktkombinatoren Y*, d.h. Kombinatoren mit folgender Eigenschaft:*

1. $Yx =_\beta x(Yx)$

2. $Yx \rhd_\beta x(Yx)$

Beweis.

1. $\mathbf{\Upsilon} \doteq_{\text{def}} \lambda x.(\lambda y.x(yy))\underbrace{(\lambda y.x(yy))}_{M}$ (Curry)

2. $\mathbf{\Theta} \doteq_{\text{def}} (\lambda zx.x(zzx))\underbrace{(\lambda zx.x(zzx))}_{N}$ (Turing)

- $\mathbf{\Upsilon}x \rhd_\beta MM \rhd_\beta x(MM) \lhd_\beta x(\mathbf{\Upsilon}x)$
- $\mathbf{\Theta}x \rhd_\beta (\lambda x.x(NNx))x \rhd_\beta x(NNx) \doteq x(\mathbf{\Theta}x)$

$\mathbf{\Theta}$ erfüllt natürlich auch 1., $\mathbf{\Upsilon}$ erfüllt aber nicht 2. □

Korollar 1.22.
Für jedes N und $n \geq 0$ gibt es ein M, so daß $My_1 \ldots y_n =_\beta N[M/x]$.

Bemerkung.
Jede "intuitive" Gleichung der Form $xy_1 \ldots y_n = N$, die x durch einen Term N definiert, in dem x selbst wieder vorkommen kann (und die insofern "selbstbezüglich" ist) besitzt als Lösung einen Term M.

Beweis.
Setze $M \doteq_{\text{def}} Y(\lambda xy_1 \ldots y_n.N)$ für einen Fixpunktkombinator Y. □

Bemerkung.
Wählen wir $\mathbf{\Theta}$ für Y, d.h. $M \doteq_{\text{def}} \mathbf{\Theta}(\lambda xy_1 \ldots y_n.N)$, dann gilt sogar $My_1 \ldots y_n \rhd_\beta N[M/x]$.

Proposition 1.23.
M ist ein Fixpunktkombinator (d.h. $Mx =_\beta x(Mx)$) genau dann, wenn M Fixpunkt von **SI** *ist, d.h. $M =_\beta \mathbf{SI}M$.*

Beweis (Barendregt (1980), 6.5.3)
$\mathbf{SI} =_\beta \lambda yz.z(yz)$
Sei M Fixpunkt von SI, d.h. $M =_\beta \mathbf{SI}M$. Dann $MN =_\beta \mathbf{SI}MN =_\beta N(MN)$, d.h. M ist Fixpunktkombinator.
Sei $Mx =_\beta x(Mx)$. Dann ist Mx nicht in Normalform, da sonst Mx und $x(Mx)$ α-kongruent

wären. Damit gilt $Mx \rhd_\beta xP$ und $x(Mx) \rhd_\beta xP$ für ein P. Ferner gilt $M \rhd_\beta \lambda z.N$, da M als Kombinator nicht mit einer Variable beginnen kann. Damit gilt
$\lambda x.Mx =_\beta \lambda x.(\lambda z.N)x =_\beta \lambda x.N[x/z] =_\beta M$ (d.h. η-Konversion (s.u.) ist für M beweisbar).
Somit $M =_\beta \lambda x.Mx =_\beta \lambda x.x(Mx) =_\beta \mathbf{SI}M$. □

Definition 1.24.
$P[\lambda x.Mx] \rhd_{1\eta} P[M]$, falls $x \notin FV(M)$ "η-Kontraktion"

$$P \rhd_{\beta\eta} Q, \text{ falls } P \bumpeq P_1 \;\substack{\equiv_{1\alpha} \\ \rhd_{1\beta} \\ \rhd_{1\eta}}\; P_2 \;\substack{\equiv_{1\alpha} \\ \rhd_{1\beta} \\ \rhd_{1\eta}}\; \cdots \;\substack{\equiv_{1\alpha} \\ \rhd_{1\beta} \\ \rhd_{1\eta}}\; P_n \bumpeq Q \qquad \text{“}\beta\eta\text{-Reduktion”}$$

$$P =_{\beta\eta} Q \;,\text{ falls } P \bumpeq P_1 \;\substack{\equiv_{1\alpha} \\ \rhd_{1\beta} \\ \lhd_{1\beta} \\ \rhd_{1\eta} \\ \lhd_{1\eta}}\; P_2 \;\substack{\equiv_{1\alpha} \\ \rhd_{1\beta} \\ \lhd_{1\beta} \\ \rhd_{1\eta} \\ \lhd_{1\eta}}\; \cdots \;\substack{\equiv_{1\alpha} \\ \rhd_{1\beta} \\ \lhd_{1\beta} \\ \rhd_{1\eta} \\ \lhd_{1\eta}}\; P_n \bumpeq Q \qquad \text{“}\beta\eta\text{-Gleichheit”}$$

Bemerkung.
$\beta\eta$-Gleichheit besagt intuitiv, daß es für die Bedeutung eines Terms nur auf sein Verhalten bei Anwendung auf einen anderen Term ankommt (Extensionalität, vgl. Lemma 1.35).

Lemma 1.25.
Lemma 1.9 gilt auch für $\rhd_{\beta\eta}$.

Theorem 1.26.
$\beta\eta$-Reduktion genügt Church-Rosser.

1.2 λ-Definierbarkeit rekursiver Funktionen

Definition 1.27.
Sei $M^0 N \doteq_{\text{def}} N$ *und* $M^{n+1} N \doteq_{\text{def}} M(M^n N)$.
Dann sind die Church-Ziffern *wie folgt definiert:* $\underline{n} \doteq_{\text{def}} \lambda xy.x^n y$

Bemerkungen.

1. Vgl. Wittgenstein, Tractatus 6.021: "Die Zahl ist der Exponent einer Operation".

2. Falls $\underline{m} =_\beta \underline{n}$, dann $m = n$, da Church-Ziffern in β-Normalform sind.

Definition 1.28.
Der λ-Term P definiert die k-stellige zahlentheoretische Funktion f, falls für alle $m_1, \ldots, m_k$ gilt, daß $P\,\underline{m_1} \ldots \underline{m_k} \simeq_\beta \underline{f(m_1, \ldots, m_k)}$, *d.h.* $P\,\underline{\vec{m}} \simeq_\beta \underline{f(\vec{m})}$.
Dabei bedeutet $P\,\underline{\vec{m}} \simeq_\beta \underline{n}$, *daß* $\begin{cases} P\,\underline{\vec{m}} =_\beta \underline{n} \iff f(\vec{m}) = n & \textit{falls } f(\vec{m}) \textit{ definiert} \\ P\,\underline{\vec{m}} \textit{ hat keine } \beta\textit{-Normalform} & \textit{falls } f(\vec{m}) \textit{ nicht definiert} \end{cases}$

Lemma 1.29.
Es gibt Kombinatoren mit folgenden Eigenschaften:

1. $\mathbf{N}\,\underline{k} =_\beta \underline{k+1}$

2. $\mathbf{V}\,\underline{k+1} =_\beta \underline{k}$

3. $\mathbf{D}PQ\,\underline{0} =_\beta P$
   $\mathbf{D}PQ\,\underline{k+1} =_\beta Q$

4. $\mathbf{R}PQ\,\underline{0} =_\beta P$
   $\mathbf{R}PQ\,\underline{k+1} =_\beta Q\,\underline{k}(\mathbf{R}PQ\,\underline{k})$

Beweis.

1. $\mathbf{N} \doteq_{\text{def}} \lambda uxy.x(uxy)$

3. $\mathbf{D} \doteq_{\text{def}} \lambda xyz.z(\mathbf{K}y)x$

2. $\mathbf{V} \doteq_{\text{def}} \lambda x.x(\lambda z.\mathbf{D}(\mathbf{N}(z\,\underline{0}))(z\,\underline{0}))(\mathbf{D}\,\underline{0}\,\underline{0})\,\underline{1}$
   Beweis.
   Wir zeigen durch Induktion über k: $(\underbrace{\lambda z.\mathbf{D}(\mathbf{N}(z\,\underline{0}))(z\,\underline{0})}_{P})^{k+1}(\mathbf{D}\,\underline{0}\,\underline{0}) =_\beta \mathbf{D}\,\underline{k+1}\,\underline{k}$

   Induktionsanfang:

$$P^1 \mathbf{D}\,\underline{0}\,\underline{0} =_\beta \mathbf{D}(\mathbf{N}(\mathbf{D}\,\underline{0}\,\underline{0}\,\underline{0}))(\mathbf{D}\,\underline{0}\,\underline{0}\,\underline{0}) =_\beta \mathbf{D}(\mathbf{N}\,\underline{0})\,\underline{0} =_\beta \mathbf{D}\,\underline{1}\,\underline{0}$$

Induktionsschritt: Sei $P^{k+1}(\mathbf{D}\,\underline{0}\,\underline{0}) =_\beta \mathbf{D}\,\underline{k+1}\,\underline{k}$. Dann ist

$$\begin{aligned}
P^{k+2}(\mathbf{D}\,\underline{0}\,\underline{0}) &=_\beta P(P^{k+1}(\mathbf{D}\,\underline{0}\,\underline{0}))\\
&=_\beta P(\mathbf{D}\,\underline{k+1}\,\underline{k}) \quad \text{(Induktionsvoraussetzung)}\\
&=_\beta \mathbf{D}(\mathbf{N}(\mathbf{D}\,\underline{k+1}\,\underline{k}\,\underline{0}))(\mathbf{D}\,\underline{k+1}\,\underline{k}\,\underline{0})\\
&=_\beta \mathbf{D}(\mathbf{N}\,\underline{k+1})\,\underline{k+1}\\
&=_\beta \mathbf{D}\,\underline{k+2}\,\underline{k+1}
\end{aligned}$$

Damit ist

$$\begin{aligned}
\mathbf{V}\,\underline{k+1} &=_\beta \underline{k+1}P(\mathbf{D}\,\underline{0}\,\underline{0})\,\underline{1}\\
&=_\beta P^{k+1}(\mathbf{D}\,\underline{0}\,\underline{0})\,\underline{1}\\
&=_\beta \mathbf{D}\,\underline{k+1}\,\underline{k}\,\underline{1}\\
&=_\beta \underline{k}
\end{aligned}$$

□

4. $\mathbf{R} \simeq_{\text{def}} \mathbf{\Theta}(\lambda uxyz.\mathbf{D}x(y(\mathbf{V}z)(uxy(\mathbf{V}z)))z)$
   $\mathbf{R}$ ist nach Korollar 1.22 Lösung von $\mathbf{R}xyz =_\beta \mathbf{D}x(y(\mathbf{V}z)(\mathbf{R}xy(\mathbf{V}z)))z$

□

Theorem 1.30.
Jede primitiv-rekursive Funktion ist λ-definierbar.

Beweis

- $0 : \mathbb{N}^0 \longrightarrow \mathbb{N}$ ist λ-definiert durch den Term $\underline{0}$
- $s : \mathbb{N} \longrightarrow \mathbb{N}$ ist λ-definiert durch den Term $\mathbf{N}$.
- $\pi_i^n : \mathbb{N}^n \longrightarrow \mathbb{N}$ ist λ-definiert durch den Term $\lambda x_1 \dots x_n.x_i$
- Falls $h : \mathbb{N}^k \longrightarrow \mathbb{N}$ und $g_i : \mathbb{N}^n \longrightarrow \mathbb{N}$ durch P und Q_i λ-definiert sind $(1 \leq i \leq k)$, und und $f : \mathbb{N}^n \longrightarrow \mathbb{N}$ gegeben ist als $f(\vec{m}) =_{\text{def}} (h \circ [g_1; \dots; g_k])(\vec{m})$ für alle $\vec{m} = (m_1, \dots, m_n)$, dann wird die Funktion $f : \mathbb{N}^n \longrightarrow \mathbb{N}$ durch den Term $\lambda\vec{x}.P(Q_1\vec{x})\dots(Q_k\vec{x})$ λ-definiert, wobei $Q_i\vec{x} \simeq_{\text{def}} (\dots(Q_i x_1)\dots)x_n$.
- Falls $g : \mathbb{N}^k \longrightarrow \mathbb{N}$ und $h : \mathbb{N}^{k+2} \longrightarrow \mathbb{N}$ durch die λ-Terme P und Q λ-definiert sind, und $f : \mathbb{N}^{k+1} \longrightarrow \mathbb{N}$ gegeben ist durch:

  1. $f(0, \vec{m}) = g(\vec{m})$
  2. $f(n+1, \vec{m}) = h(n, f(n, \vec{m}), \vec{m})$

  dann wird f λ-definiert durch den Term $\lambda u\vec{x}.\mathbf{R}(P\vec{x})(\lambda uv.Quv\vec{x})u$

Beweis durch Induktion über n
Induktionsanfang:

$$\begin{aligned}(\lambda u\vec{x}.\mathbf{R}(P\vec{x})(\lambda uv.Quv\vec{x})u)\,\underline{0}\,\underline{\vec{m}} &=_\beta \mathbf{R}(P\,\underline{\vec{m}})(\lambda uv.Quv\,\underline{\vec{m}})\,\underline{0}\\ &=_\beta P\,\underline{\vec{m}}\\ &=_\beta \underline{g(\vec{m})}\ \text{ (nach Voraussetzung über } g)\end{aligned}$$

Induktionsschritt:

$$\begin{aligned}(\lambda u\vec{x}.\mathbf{R}(P\vec{x})(\lambda uv.Quv\vec{x})u)\,\underline{n+1}\,\underline{\vec{m}} &=_\beta \mathbf{R}(P\,\underline{\vec{m}})(\lambda uv.Quv\,\underline{\vec{m}})\,\underline{n+1}\\ &=_\beta (\lambda uv.Quv\,\underline{\vec{m}})\,\underline{n}(\mathbf{R}(P\,\underline{\vec{m}})(\lambda uv.Quv\,\underline{\vec{m}})\,\underline{n})\\ &=_\beta Q\,\underline{n}(\mathbf{R}(P\,\underline{\vec{m}})(\lambda uv.Quv\,\underline{\vec{m}})\,\underline{n})\,\underline{\vec{m}}\\ &=_\beta Q\,\underline{n}((\lambda u\vec{x}.\mathbf{R}(P\vec{x})(\lambda uv.Quv\vec{x})u)\,\underline{n}\,\underline{\vec{m}})\,\underline{\vec{m}}\\ &=_\beta Q\,\underline{n}\,\underline{f(n,\vec{m})}\,\underline{\vec{m}}\ \text{ (Induktionsvoraussetzung)}\\ &=_\beta \underline{h(n,f(n,\vec{m}),\vec{m})}\ \text{ (nach Voraussetzung über } h)\end{aligned}$$

□

Beispiel.
Die Funktion add : $\mathbb{N}^2 \to \mathbb{N}$ *mit* $add(n,m) = n+m$ *ist primitiv rekursiv wie folgt definiert:*

1. $add(0,m) = \pi_1^1(m)$

2. $add(n+1,m) = (s \circ \pi_2^3)(n, add(n,m), m)$

Die λ-Definition der Funktion π_1^1 ist der Term $\lambda x.x \doteq \mathbf{I}$, und die λ-Definition der Komposition $s \circ \pi_2^3$ ist der Term $\lambda y_1y_2y_3.\mathbf{N}((\lambda x_1x_2x_3.x_2)y_1y_2y_3)$. Somit ergibt sich nach dem Schema als λ-Definition der Funktion add der Term

$$\mathbf{Add} \doteq \lambda ux.\mathbf{R}(\mathbf{I}x)(\lambda uv.(\lambda y_1y_2y_3.\mathbf{N}((\lambda x_1x_2x_3.x_2)y_1y_2y_3))uvx)u$$

Man kann alle durch die schematische Übersetzung erzeugten Redexe kontrahieren, und erhält diesen vereinfachten Term:

$$\mathbf{Add} \triangleright_\beta \lambda ux.\mathbf{R}x(\lambda uv.\mathbf{N}v)u$$

Die Berechnung von 1+1 (unter Verwendung von Lemma 1.33, 4. und 1.):

$$\begin{aligned}\mathbf{Add}\,\underline{1}\,\underline{1} &\triangleright_\beta \mathbf{R}\,\underline{1}(\lambda uv.\mathbf{N}v)\,\underline{1}\\ &=_\beta (\lambda uv.\mathbf{N}v)\,\underline{0}(\mathbf{R}\,\underline{1}(\lambda uv.\mathbf{N}v)\,\underline{0})\\ &\triangleright_\beta \mathbf{N}(\mathbf{R}\,\underline{1}(\lambda uv.\mathbf{N}v)\,\underline{0})\\ &=_\beta \mathbf{N}\,\underline{1}\\ &=_\beta \underline{2}\end{aligned}$$

Theorem 1.31.
Jede partiell-rekursive Funktion ist λ-definierbar.

Beweis.
Jedes partiell-rekursive f läßt sich nach Kleene darstellen als $f(\vec{m}) =_{\text{def}} h(\mu k.g(\vec{m}, k) = 0)$, wobei g, h primitiv-rekursive Funktionen sind (Informatik III). Die Funktionen g und h seien durch Terme P und Q λ-definiert.
Betrachte die Gleichung

$$(\star) \quad M\vec{x}y =_\beta \mathbf{D}y(M\vec{x}(\mathbf{N}y))(P\vec{x}y)$$

Nach Korollar 1.22 ist $\mathbf{\Theta}\,\underbrace{(\lambda u\vec{x}y.\mathbf{D}y(u\vec{x}(\mathbf{N}y))(P\vec{x}y))}_{Z}$ eine Lösung der Gleichung.
Behauptung: f wird λ-definiert durch den Term $\lambda\vec{x}.Q(\mathbf{\Theta}Z\vec{x}\,\underline{0})$.
Dazu genügt es zu zeigen: $\mathbf{\Theta}Z\,\underline{\vec{m}}\,\underline{0} =_\beta \underline{k_1}$, falls k_1 kleinstes k mit $g(\vec{m}, k) = 0$.
Wir werden zeigen:

$$(\star\star) \quad \text{Falls } g(\vec{m}, k) \neq 0 \text{ für alle } k < k_1, \text{ dann } \mathbf{\Theta}Z\vec{m}\,\underline{0} =_\beta \mathbf{D}\,\underline{k_1}(\mathbf{\Theta}Z\,\underline{\vec{m}}\,\underline{k_1+1})(P\,\underline{\vec{m}}\,\underline{k_1})$$

Hieraus ergibt sich: Wenn k_1 kleinstes k mit $g(\vec{m}, k) = 0$, dann $\mathbf{\Theta}Z\vec{m}\,\underline{0} =_\beta \underline{k_1}$, da $P\,\underline{\vec{m}}\,\underline{k_1} =_\beta \underline{0}$.
Beweis von $(\star\star)$ durch Induktion über k_1:

- $k_1 = 0$: $\mathbf{\Theta}Z\,\underline{\vec{m}}\,\underline{0} =_\beta \mathbf{D}\,\underline{0}(\mathbf{\Theta}Z\,\underline{\vec{m}}\,\underline{1})(P\,\underline{\vec{m}}\,\underline{0})$ mit $(\star)$

- $k_1 > 0$:

$$\begin{aligned} \mathbf{\Theta}Z\,\underline{\vec{m}}\,\underline{0} &=_\beta \mathbf{D}\,\underline{k_1-1}(\mathbf{\Theta}Z\,\underline{\vec{m}}\,\underline{k_1})\underbrace{(P\,\underline{\vec{m}}\,\underline{k_1-1})}_{\substack{=_\beta\ \underline{l+1} \textbf{ für ein } l, \\ \textbf{da } g(\vec{m}, k_1-1) \neq 0}} \qquad \text{(Induktionsvoraussetzung)} \\ &=_\beta \mathbf{\Theta}Z\,\underline{\vec{m}}\,\underline{k_1} \\ &=_\beta \mathbf{D}\,\underline{k_1}(\mathbf{\Theta}Z\,\underline{\vec{m}}\,\underline{k_1+1})(P\,\underline{\vec{m}}\,\underline{k_1}) \quad \text{mit } (\star) \end{aligned}$$

Es bleibt zu zeigen: Wenn $f(\vec{m})$ undefiniert, d.h. wenn $g(\vec{m}, k) \neq 0$ für alle k bei gegebenem $\vec{m}$, dann hat $\mathbf{\Theta}Z\,\underline{\vec{m}}\,\underline{0}$ keine β-Normalform. Es gilt:

$$\begin{aligned} \mathbf{\Theta}Z\,\underline{\vec{m}}\,\underline{0} &\rhd_\beta \mathbf{D}\,\underline{0}(\mathbf{\Theta}Z\,\underline{\vec{m}}\,\underline{1})(P\,\underline{\vec{m}}\,\underline{0}) \rhd_\beta \mathbf{\Theta}Z\,\underline{\vec{m}}\,\underline{1} \\ &\rhd_\beta \mathbf{D}\,\underline{1}(\mathbf{\Theta}Z\,\underline{\vec{m}}\,\underline{2})(P\,\underline{\vec{m}}\,\underline{1}) \rhd_\beta \mathbf{\Theta}Z\,\underline{\vec{m}}\,\underline{2} \\ &\rhd_\beta \mathbf{D}\,\underline{2}(\mathbf{\Theta}Z\,\underline{\vec{m}}\,\underline{3})(P\,\underline{\vec{m}}\,\underline{2}) \rhd_\beta \mathbf{\Theta}Z\,\underline{\vec{m}}\,\underline{3} \\ &\rhd_\beta \cdots \end{aligned}$$

Diese Reduktionsfolge ist QL (quasi-leftmost), d.h. es kommt immer wieder vor, daß ein linkester Term kontrahiert wird, nämlich ein Term der Form $\mathbf{D}MN\,\underline{l+1}$. Wir haben also eine nichtterminierende QL-Reduktionsfolge. Also hat $\mathbf{\Theta}Z\,\underline{\vec{m}}\,\underline{0}$ keine β-Normalform (Theorem 1.20). Man beachte, daß wir $\mathbf{\Theta}$ als Fixpunktkombinator gewählt haben. Damit haben wir statt $=_\beta$ immer $\rhd_\beta$ in der Anwendung von $(\star)$ (vgl. die Bemerkung zu Korollar 1.22). □

Theorem 1.32.
Jede λ-definierbare Funktion ist partiell-rekursiv.

Beweisskizze
Sei f n-stellig und durch P λ-definiert. Dann gilt:

$f(k_1, \ldots, k_n)$ = dasjenige k, für das die Gleichung $P\,\underline{k_1} \ldots \underline{k_n} = \underline{k}$ die Endformel der kürzesten Ableitung in $\lambda\beta$ ist (siehe folgendes Kapitel), die mit einer Formel der Gestalt $P\,\underline{k_1} \ldots \underline{k_n} = \underline{m}$ endet, falls es eine solche Ableitung in $\lambda\beta$ gibt.
$f(k_1, \ldots, k_n)$ ist sonst undefiniert.

Nach geeigneter Gödelisierung erweist sich f als partiell-rekursive Funktion. □

1.3 Die formalen Theorien $\lambda\beta$ und $\lambda\beta\eta$

Definition 1.33.

Formeln der Systeme $\lambda\beta$ und $\lambda\beta\eta$ sind alle Gleichungen der Form $M = N$ für λ-Terme M, N. Die Axiome sind:

(ρ) $M = M$

(α) $\lambda x.M = \lambda y.M[{}^{y}\!/\!{}_{x}]$, *falls* $y \notin FV(M)$

(β) $(\lambda x.M)N = M[{}^{N}\!/\!{}_{x}]$

(η) $\lambda x.Mx = M$, *falls* $x \notin FV(M)$ (nur $\lambda\beta\eta$!)

Die Regeln sind:

$$(\sigma)\ \frac{M = N}{N = M}$$

$$(\tau)\ \frac{M = N \quad N = P}{M = P}$$

$$(\mu)\ \frac{M = M'}{MN = M'N}$$

$$(\nu)\ \frac{N = N'}{MN = MN'}$$

$$(\xi)\ \frac{M = M'}{\lambda x.M = \lambda x.M'} \qquad \textit{(schwache Extensionalität)}$$

$\lambda\beta \vdash M = N$ heißt, daß $M = N$ in $\lambda\beta$ ableitbar ist.
$\lambda\beta\eta \vdash M = N$ heißt, daß $M = N$ in $\lambda\beta\eta$ ableitbar ist.
$\lambda\beta_{\triangleright}$ und $\lambda\beta\eta_{\triangleright}$ sind Systeme ohne Regel (σ) (Symmetrie).
$\lambda\beta_{\triangleright} \vdash M = N$ heißt, daß $M = N$ in $\lambda\beta_{\triangleright}$ ableitbar ist.
$\lambda\beta\eta_{\triangleright} \vdash M = N$ heißt, daß $M = N$ in $\lambda\beta\eta_{\triangleright}$ ableitbar ist.

Beispiel.

$$(\tau)\ \dfrac{(\mu)\ \dfrac{(\xi)\ \dfrac{(\beta)\ \dfrac{}{(\lambda y.yx)z = zx}}{\lambda x.(\lambda y.yx)z = \lambda x.zx}}{(\lambda x.(\lambda y.yx)z)v = (\lambda x.zx)v} \qquad (\beta)\ \dfrac{}{(\lambda x.zx)v = zv}}{\lambda x.(\lambda y.yx)z) = zv}$$

Lemma 1.34.

1. *$M \rhd_\beta N$ genau dann, wenn $\lambda\beta_\rhd \vdash M = N$*
2. *$M \rhd_{\beta\eta} N$ genau dann, wenn $\lambda\beta\eta_\rhd \vdash M = N$*
3. *$M =_\beta N$ genau dann, wenn $\lambda\beta \vdash M = N$*
4. *$M =_{\beta\eta} N$ genau dann, wenn $\lambda\beta\eta \vdash M = N$*

Beweis.
Übung. □

Lemma 1.35.
Ersetzt man in der Definition von $\lambda\beta\eta$ das Axiom (η) durch

$$(\chi)\ \frac{MP = NP \text{ für alle } P}{M = N} \qquad \textit{oder}$$

$$(\zeta)\ \frac{Mx = Nx}{M = N}, \textit{ falls } x \notin FV(NM)$$

so sind in $\lambda\beta\eta$ dieselben Gleichungen wie vorher ableitbar.

Bemerkung.
Bei (χ) handelt es sich um eine sogenannte ω-Regel, d.h. eine Regel mit unendlich vielen Prämissen. Wir beschränken uns hier auf die Betrachtung der Regel (ζ).
Beweis.

- "$(\eta) \Longrightarrow (\zeta)$":

$$(\tau)\ \frac{(\sigma)\ \dfrac{(\eta)\ \dfrac{}{\lambda x.Mx = M}}{M = \lambda x.Mx} \quad (\xi)\ \dfrac{Mx = Nx}{\lambda x.Mx = \lambda x.Nx}}{(\eta) + (\tau)\ \dfrac{M = \lambda x.Nx}{M = N}}$$

- "$(\zeta) \Longrightarrow (\eta)$":

$$(\eta)\ \frac{(\beta)\ \dfrac{}{(\lambda x.Mx)x = Mx}}{\lambda x.Mx = M}$$

□

1.4 Entscheidbarkeit

Theorem 1.36 (Church 1936).
Die Menge $NF_\beta =_{\mathrm{def}} \{M : M \text{ hat } \beta\text{-Normalform}\}$ *ist nicht entscheidbar.*

Beweisskizze
Wir können die einstelligen partiell-rekursiven Funktionen so abzählen: $f_1, f_2, \ldots$, daß die Funktion u mit $u(m,n) \simeq_{\mathrm{def}} f_m(n)$ partiell rekursiv ist. Nun werde u durch P λ-definiert. Dann gilt: $P\,\underline{m}\,\underline{n}$ hat β-Normalform g.d.w. $u(m,n)$ ist definiert.
Wäre NF_β entscheidbar, wäre g mit

$$g(n) \quad =_{\mathrm{def}} \quad \begin{cases} u(n,n)+1 & \text{falls } u(n,n) \text{ definiert} \\ 1 & \text{sonst} \end{cases}$$

eine total-rekursive Funktion. Damit wäre $g = f_k$ für ein k, also

$$u(k,k) = f_k(k) = g(k) = u(k,k)+1$$

da f_k total. □

Theorem 1.37 (Church 1936).
Die Relation $=_\beta$ *ist nicht entscheidbar.*

Beweisskizze
Die zu einem Term β-konvertiblen Terme lassen sich rekursiv aufzählen. Sei

$$f(m,k) \quad =_{\mathrm{def}} \quad \text{Gödelnummer des } k\text{-ten Terms, der zum Term mit Gödelnummer } m\ \beta\text{-konvertibel ist}$$

$$h(m) \quad =_{\mathrm{def}} \quad \begin{cases} 0 & \text{falls } m \text{ eine Gödelnummer eines Terms in } \beta\text{-Normalform ist} \\ 1 & \text{sonst} \end{cases}$$

f und h sind primitiv rekursiv. Sie seien durch F und H λ-definiert.
Betrachte die Gleichung (in u): $Mxy =_\beta \mathbf{D}\,\underline{1}(Mx(\mathbf{N}y))(H(Fxy))$. Eine Lösung dieser Gleichung ist nach Korollar 1.22: $\boldsymbol{\Upsilon}\lambda gxy.\underbrace{\mathbf{D}\,\underline{1}(gx(\mathbf{N}y))(H(Fxy))}_{V}$. Es gilt dann:
$(\boldsymbol{\Upsilon}\lambda gxy.V)\,\underline{m}\,\underline{0} =_\beta \underline{1}$, falls m Gödelnummer eines Terms ist, der zu einem Term in β-Normalform konvertibel ist. $(\boldsymbol{\Upsilon}\lambda gxy.V)\,\underline{m}\,\underline{0}$ hat sonst keine β-Normalform.
Falls nun $=_\beta$ entscheidbar wäre, dann wäre $(\boldsymbol{\Upsilon}\lambda gxy.V)\,\underline{\ulcorner M \urcorner}\,\underline{0} =_\beta \underline{1}$ entscheidbar (wobei $\ulcorner M \urcorner$ Gödelnummer von M). Also ist NF_β entscheidbar. □

Theorem 1.38 (Church 1936)**.**
Die Prädikatenlogik erster Stufe $\mathcal{PL}$ ist nicht entscheidbar.

Beweisskizze
Da $=_\beta$ unentscheidbar, ist $\lambda\beta$ ein unentscheidbarer Kalkül. Nun gilt:

$$\lambda\beta \vdash M = N \text{ genau dann, wenn } \mathcal{PL} \vdash (F_1 \wedge \ldots \wedge F_8) \to E(\overline{\overline{\ulcorner M \urcorner}}, \overline{\overline{\ulcorner N \urcorner}})$$

Hierbei sei E ein ausgezeichnetes zweistelliges Prädikat und

$$\left.\begin{array}{lll} \overline{\overline{0}} & \hat{=}_{\text{def}} & z \\ \overline{\overline{1}} & \hat{=}_{\text{def}} & f(z) \\ \overline{\overline{2}} & \hat{=}_{\text{def}} & f(f(z)) \\ & \vdots & \end{array}\right\} \text{für ausgezeichnete } z \text{ und } f$$

F_i sei die prädikatenlogische Übersetzung der i-ten Regel von $\lambda\beta$ durch Gödelnummerierung.
Beispiele:
(σ) wird übersetzt als $E(\overline{\overline{\ulcorner M \urcorner}}, \overline{\overline{\ulcorner N \urcorner}}) \to E(\overline{\overline{\ulcorner N \urcorner}}, \overline{\overline{\ulcorner M \urcorner}})$
(τ) wird übersetzt als $E(\overline{\overline{\ulcorner M \urcorner}}, \overline{\overline{\ulcorner N \urcorner}}) \wedge E(\overline{\overline{\ulcorner N \urcorner}}, \overline{\overline{\ulcorner P \urcorner}}) \to E(\overline{\overline{\ulcorner M \urcorner}}, \overline{\overline{\ulcorner P \urcorner}})$
Wenn die Prädikatenlogik entscheidbar wäre, wäre somit $\lambda\beta$ entscheidbar. □

2 Kombinatorische Logik

Gegeben sei eine unendliche Folge von Variablen mit fester Reihenfolge. Der Einfachheit halber seien die Variablen der kombinatorischen Logik dieselben wie die des λ-Kalküls.
Im folgenden seien **K** und **S** vorgegebene Konstanten. Wenn außer diesen noch weitere Konstanten hinzu kommen, heißt das System *angewandt* (sonst *rein*).

Definition 2.1 (Syntax)**.**

- *Alle Variablen und Konstanten sind $\mathcal{CL}$-Terme (*Atome*)*
- *Mit X und Y ist auch (XY) ein $\mathcal{CL}$-Term (*Applikation*)*

Ein geschlossener *$\mathcal{CL}$-Term enthält keine Variablen.*
Ein Kombinator *enthält nur* **K** *und* **S** *als Atome.*
FV(X) sei die Menge der Variablen in X.
Die Substitution von Variablen $Y[^X/_z]$ ist in offensichtlicher Weise definiert, da es in $\mathcal{CL}$ keine gebundenen Variablen gibt.
Metasprachliche Variablen: $U, V, W, X, Y, Z, \ldots$ *für $\mathcal{CL}$-Terme; $a, b, c, \ldots$ für Atome.*
Außenklammern können wegfallen. Bei Klammerung gilt Linksassoziation, d.h. $UVWX$ meint $((UV)W)X$.
$X \doteq Y$ bezeichne die syntaktische Identität von X und Y.

Beispiel.

- $\mathbf{S}xy(\mathbf{K}y)(\mathbf{KKSS})$ *ist ein $\mathcal{CL}$-Term*
- $\mathbf{S}(\mathbf{KS})$ *ist ein $\mathcal{CL}$-Term*

Definition 2.2 (schwache Reduktion, schwache Konversion)**.**

$$\left.\begin{array}{rcl} U[\mathbf{K}XY] & \rhd_{1w} & U[X] \\ U[\mathbf{S}XYZ] & \rhd_{1w} & U[XZ(YZ)] \end{array}\right\} \text{(“schwache Kontraktion”)}$$

$X \rhd_w Y$*, falls* $X \doteq V_1 \rhd_{1w} V_2 \rhd_{1w} \ldots \rhd_{1w} V_n \doteq Y$ *(*“schwache Reduktion”*)*
$X =_w Y$*, falls* $X \doteq V_1 \overset{\rhd_{1w}}{\lhd_{1w}} V_2 \overset{\rhd_{1w}}{\lhd_{1w}} \ldots \overset{\rhd_{1w}}{\lhd_{1w}} V_n \doteq Y$ *(*“schwache Konversion”*)*

Bemerkung.
Schwache Reduktion $\rhd_w$ ist invariant gegenüber Substitution (vgl. Lemma 1.11), und es gilt die Church-Rosser-Eigenschaft (vgl. Theorem 1.14).

Definition 2.3.
Formeln des Systems $\mathcal{CL}w$ sind alle Gleichungen $X = Y$ für $\mathcal{CL}$-Terme X, Y.
Die Axiome sind:

(ρ) $X = X$

(K) $\mathbf{K}XY = X$

(S) $\mathbf{S}XYZ = XZ(YZ)$

Die Regeln sind:

(σ) $\dfrac{X = Y}{Y = X}$

(τ) $\dfrac{X = Y \quad Y = Z}{X = Z}$

(μ) $\dfrac{X = X'}{YX = YX'}$

(ν) $\dfrac{Y = Y'}{YX = Y'X}$

$\mathcal{CL}w \vdash X = Y$ *bedeutet, daß* $X = Y$ *in* $\mathcal{CL}w$ *ableitbar ist.*
$\mathcal{CL}w_{\triangleright} \vdash X = Y$ *bedeutet, daß* $X = Y$ *in* $\mathcal{CL}w$ *ohne* (σ) *ableitbar ist.*

Lemma 2.4.

- $X =_w Y$ *genau dann, wenn* $\mathcal{CL}w \vdash X = Y$.
- $X \triangleright_w Y$ *genau dann, wenn* $\mathcal{CL}w_{\triangleright} \vdash X = Y$.

Definition 2.5.
Für einen $\mathcal{CL}$*-Term* X *ist der* λ*-Term* X_λ *wie folgt definiert:*

1. $x_\lambda \doteqdot_{\text{def}} x$
2. $\mathbf{K}_\lambda \doteqdot_{\text{def}} \lambda xy.x$
3. $\mathbf{S}_\lambda \doteqdot_{\text{def}} \lambda xyz.xz(yz)$
4. $(XY)_\lambda \doteqdot_{\text{def}} X_\lambda Y_\lambda$

(Wir identifizieren dabei α*-kongruente Terme.)*

Lemma 2.6.

- *Wenn* $X \triangleright_w Y$, *dann* $X_\lambda \triangleright_\beta Y_\lambda$.
- *Wenn* $X =_w Y$, *dann* $X_\lambda =_\beta Y_\lambda$.

Beweis.
Benutze $\mathcal{CL}w$ bzw. $\lambda\beta$. □

Bemerkung.
Die Umkehrung gilt nicht! Es gilt z.B. $\mathbf{S}_\lambda\mathbf{K}_\lambda =_\beta \mathbf{K}_\lambda(\mathbf{S}_\lambda\mathbf{K}_\lambda\mathbf{K}_\lambda)$, nicht jedoch $\mathbf{SK} =_w \mathbf{K}(\mathbf{SKK})$. Keiner der $\mathcal{CL}$-Terme kontrahiert, während beide λ-Terme Redexe enthalten.

Definition 2.7.
Für einen λ-Term M ist der $\mathcal{CL}$-Term $M_{\mathcal{CL}}$ wie folgt definiert:

1. $x_{\mathcal{CL}} \doteq_{\text{def}} x$
2. $(MN)_{\mathcal{CL}} \doteq_{\text{def}} M_{\mathcal{CL}}N_{\mathcal{CL}}$
3. $(\lambda x.M)_{\mathcal{CL}} \doteq_{\text{def}} [x].M_{\mathcal{CL}}$

wobei $[x].X$ für $\mathcal{CL}$-Terme X wie folgt definiert ist:

1. $[x].x \doteq_{\text{def}} \mathbf{SKK}$ *(abgekürzt: $\mathbf{I} \doteq_{\text{def}} \mathbf{SKK}$)*
2. $[x].X \doteq_{\text{def}} \mathbf{K}X$ *falls $x \notin FV(X)$*
3. $[x].Xx \doteq_{\text{def}} X$ *falls $x \notin FV(X)$*
4. $[x].(XY) \doteq_{\text{def}} \mathbf{S}([x].X)([x].Y)$ *falls die vorherigen Fälle nicht zutreffen*

Beispiel.

$$\begin{aligned}[x].xxz &\doteq \mathbf{S}([x].xx)([x].z)\\ &\doteq \mathbf{S}(\mathbf{S}([x].x)([x].x))(\mathbf{K}z)\\ &\doteq \mathbf{S}(\mathbf{SII})(\mathbf{K}z)\end{aligned}$$

Bemerkungen.

1. $[x].X$ ist eine metasprachliche Operation, d.h. ein Term $[x].X$ ist kein $\mathcal{CL}$-Term sondern *repräsentiert* nur einen $\mathcal{CL}$-Term.
2. $x \notin FV([x].Y)$. Insofern verhält sich $[x]$ wie ein variablenbindender Operator.

Lemma 2.8.
$([x].Y)Z \rhd_w Y[Z/x]$

Beweis.
Induktion über der Struktur von Y:

1. $Y \doteq x$: $([x].x)Z \doteq \mathbf{I}Z \rhd_w Z \doteq x[Z/x]$

2. Y ist Atom, $Y \not\equiv x$: $([x].Y)Z \doteq \mathbf{K}YZ \rhd_w Y \doteq Y[Z/x]$

3. $Y \doteq (UV)$:

   - $x \notin FV(Y)$: $([x].Y)Z \doteq \mathbf{K}YZ \rhd_w Y \doteq Y[Z/x]$
   - $x \notin FV(U)$, $V \doteq x$: $([x].Y)Z \doteq UZ \doteq Ux[Z/x]$
   - keiner der vorherigen Fälle:

$$\begin{array}{rcll} ([x].Y)Z & \doteq & \mathbf{S}([x].U)([x].V)Z & \\ & \rhd_w & ([x].U)Z(([x].V)Z) & \\ & \rhd_w & (U[Z/x])(V[Z/x]) & \text{nach Induktionsvoraussetzung} \\ & \doteq & Y[Z/x] & \end{array}$$

□

Korollar 2.9 (Kombinatorische Vollständigkeit)**.**
Sei V ein Term mit $\{x_1, \ldots, x_n\} \subseteq FV(V)$. Dann gibt es einen Term U, in dem $x_1, \ldots, x_n$ nicht vorkommt, so daß $UX_1 \ldots X_n \rhd_w V[X_1/x_1] \ldots [X_n/x_n]$.

Beweis.
Setze $U \doteq_{\text{def}} [x_1]. \ldots [x_n].V$ □

Bemerkung.
Damit kann man jeden Kombinator U, der durch eine "neue" Kontraktion $UX_1 \ldots X_n \rhd_w W$ gegeben ist, wobei W nur aus $X_1, \ldots, X_n$ zusammengesetzt ist, in $\mathcal{CL}$ durch einen variablenfreien Term definieren. Mit Hilfe von $\mathbf{S}$ und $\mathbf{K}$ lassen sich also "alle" Kombinatoren ausdrücken.

Lemma 2.10.

(i) Für $\mathcal{CL}$-Terme X gilt: $(X_\lambda)_{\mathcal{CL}} \doteq X$

(ii) Für λ-Terme M gilt: $(M_{\mathcal{CL}})_\lambda =_{\beta\eta} M$

Beweis.

(i) Induktion über der Struktur von X:

   - $(x_\lambda)_{\mathcal{CL}} \doteq x_{\mathcal{CL}} \doteq x$
   - $(\mathbf{K}_\lambda)_{\mathcal{CL}} \doteq (\lambda xy.x)_{\mathcal{CL}} \doteq [x].([y].x) \doteq [x].\mathbf{K}x \doteq \mathbf{K}$
   - $$\begin{array}{rcl} (\mathbf{S}_\lambda)_{\mathcal{CL}} & \doteq & (\lambda xyz.xz(yz))_{\mathcal{CL}} \\ & \doteq & [x].([y].([z].xz(yz))) \\ & \doteq & [x].([y].\mathbf{S}([z].xz)([z].yz)) \\ & \doteq & [x].([y].\mathbf{S}xy) \\ & \doteq & [x].\mathbf{S}x \\ & \doteq & \mathbf{S} \end{array}$$

- $((XY)_\lambda)_{\mathcal{CL}} \doteq (X_\lambda Y_\lambda)_{\mathcal{CL}} \doteq (X_\lambda)_{\mathcal{CL}}(Y_\lambda)_{\mathcal{CL}} \doteq XY$ nach Induktionsvoraussetzung

(ii) Induktion über der Struktur von M:

- $(x_{\mathcal{CL}})_\lambda \doteq x_\lambda \doteq x$
- $((MN)_{\mathcal{CL}})_\lambda \doteq (M_{\mathcal{CL}}N_{\mathcal{CL}})_\lambda \doteq (M_{\mathcal{CL}})_\lambda(N_{\mathcal{CL}})_\lambda =_{\beta\eta} MN$ nach Induktionsvorauss.
- zu zeigen: $((\lambda x.M')_{\mathcal{CL}})_\lambda \doteq ([x].(M')_{\mathcal{CL}})_\lambda =_{\beta\eta} \lambda x.M'$
  Induktion über der Struktur von M':
  - $([x].x_{\mathcal{CL}})_\lambda \doteq \mathbf{I}_\lambda \doteq \mathbf{S}_\lambda\mathbf{K}_\lambda\mathbf{K}_\lambda =_\beta \lambda x.x$
  - falls $x \notin FV((UV)_{\mathcal{CL}})$:
  $$\begin{aligned} ([x].(UV)_{\mathcal{CL}})_\lambda &\doteq (\mathbf{K}(UV)_{\mathcal{CL}})_\lambda \\ &\doteq \mathbf{K}_\lambda((UV)_{\mathcal{CL}})_\lambda \\ &\doteq (\lambda xy.x)((UV)_{\mathcal{CL}})_\lambda \qquad \text{wo } y \notin FV(UV) \\ &=_\beta \lambda y.((UV)_{\mathcal{CL}})_\lambda \\ &=_{\beta\eta} \lambda y.(UV) \qquad \text{nach Induktionsvoraussetzung} \end{aligned}$$
  - falls $x \notin FV(U_{\mathcal{CL}})$ und $V_{\mathcal{CL}} \doteq x$:
  $$\begin{aligned} ([x].(UV)_{\mathcal{CL}})_\lambda &\doteq ([x].(U_{\mathcal{CL}}V_{\mathcal{CL}}))_\lambda \\ &\doteq ([x].(U_{\mathcal{CL}}x))_\lambda \\ &\doteq (U_{\mathcal{CL}})_\lambda \\ &=_{\beta\eta} U \qquad \text{nach Induktionsvoraussetzung} \\ &=_\eta \lambda x.(Ux) \\ &\doteq \lambda x.(UV) \end{aligned}$$
  - sonst:
  $$\begin{aligned} ([x].(UV)_{\mathcal{CL}})_\lambda &\doteq (\mathbf{S}([x].U_{\mathcal{CL}})([x].V_{\mathcal{CL}}))_\lambda \\ &\doteq \mathbf{S}_\lambda([x].U_{\mathcal{CL}})_\lambda([x].V_{\mathcal{CL}})_\lambda \\ &=_{\beta\eta} \mathbf{S}_\lambda(\lambda x.U)(\lambda x.V) \qquad \text{nach Induktionsvoraussetzung} \\ &\doteq (\lambda uvy.uy(vy))(\lambda x.U)(\lambda x.V) \\ &=_\beta \lambda y.(\lambda x.U)y((\lambda x.V)y) \\ &=_\beta \lambda x.(UV) \end{aligned}$$

□

Bemerkungen.

- Es gilt also mit Lemma 2.6: $M_{\mathcal{CL}} =_w N_{\mathcal{CL}}$ genau dann, wenn $M =_{\beta\eta} N$
- Es gilt aber *nicht*: $(M_{\mathcal{CL}})_\lambda =_\beta M$
  Zum Beispiel ist $((\lambda x.yx)_{\mathcal{CL}})_\lambda \doteq ([x].yx)_\lambda \doteq y_\lambda \doteq y \neq_\beta \lambda x.yx$
  Keine der folgenden Behauptungen gilt:

$M_{\mathcal{CL}} \rhd_w N_{\mathcal{CL}} \implies M \rhd_\beta N \qquad M_{\mathcal{CL}} \rhd_w N_{\mathcal{CL}} \impliedby M \rhd_\beta N$

$M_{\mathcal{CL}} =_w N_{\mathcal{CL}} \implies M =_\beta N \qquad M_{\mathcal{CL}} =_w N_{\mathcal{CL}} \impliedby M =_\beta N$

Man darf also zur Auswertung von λ-Termen, wenn es nur um β-Reduktion geht, *nicht* so verfahren: $M \rightsquigarrow M_{\mathcal{CL}} \rhd_w N_{\mathcal{CL}} \rightsquigarrow N$

Das Problem mit "$\implies$" besteht darin, daß (η) in $\mathcal{CL}w$ gilt. Denn es ist

$(\lambda x.Mx)_{\mathcal{CL}} \doteq [x].M_{\mathcal{CL}}x \doteq M_{\mathcal{CL}}$ falls $x \notin FV(M)$

Das Problem mit "$\impliedby$" besteht darin, daß (ξ) in $\mathcal{CL}w$ nicht gilt. Denn es ist

$$\begin{array}{rcl}
[x].\mathbf{S}xyz & \doteq & \mathbf{S}([x].\mathbf{S}xy)([x].z) \\
& \doteq & \mathbf{S}(\mathbf{S}([x].\mathbf{S}x)([x].y))(\mathbf{K}z) \\
& \doteq & \mathbf{S}(\mathbf{SS}(\mathbf{K}y))(\mathbf{K}z) \\
[x].xz(yz) & \doteq & \mathbf{S}([x].xz)([x].yz) \\
& \doteq & \mathbf{S}(\mathbf{S}([x].x)([x].z))(\mathbf{K}(yz)) \\
& \doteq & \mathbf{S}(\mathbf{SI}(\mathbf{K}z))(\mathbf{K}(yz))
\end{array}$$

Also ist zwar $\mathbf{S}xyz =_w xz(yz)$, aber nicht $[x].\mathbf{S}xyz =_w [x].xz(yz)$.

- Die Hinzunahme von (ξ) zu $\mathcal{CL}w$ bewirkt *volle* Extensionalität:

  $$(\xi)\dfrac{\dfrac{Xx = Yx}{[x].Xx = [x].Yx}}{X = Y}$$

  Während also im $\lambda\beta$-Kalkül die Hinzunahme von (η) Extensionalität zur Folge hat, wohingegen (ξ) eo ipso gilt, bewirkt in der Kombinatorischen Logik die Hinzunahme von (ξ) Extensionalität, während (η) eo ipso gilt. Das zeigt die Disparatheit beider Systeme.

- Wir definieren $\succ\!\!-$ durch Erweiterung von $\mathcal{CL}w_{\rhd}$ um

  $$(\xi)\ \dfrac{X \succ\!\!- Y}{[x].X \succ\!\!- [x].Y}$$

  Dann gilt für λ-Terme:

  $M \rhd_{\beta\eta} N \implies M_{\mathcal{CL}} \succ\!\!- N_{\mathcal{CL}}$

  $M =_{\beta\eta} N \impliedby M_{\mathcal{CL}} \succ\!\!- N_{\mathcal{CL}}$

  η-Konversion ist jetzt Fall von (ρ) und gilt in beliebiger Richtung. Daraus ergibt sich:

  $M =_{\beta\eta} N \iff M_{\mathcal{CL}} \succ\!\!\prec N_{\mathcal{CL}}$, wobei $\succ\!\!\prec$ der symmetrische Abschluß von $\succ\!\!-$ ist.

  Damit gilt für $\mathcal{CL}$-Terme X, Y:

  $X \succ\!\!\prec Y \iff X_\lambda =_{\beta\eta} Y_\lambda$

- Um β-Gleichheit in $\mathcal{CL}w$ repräsentieren zu können, kann man die Definition von $[x].Y$ so abschwächen, daß (η) nicht automatisch gilt, z.B. durch Weglassen der Klausel $[x].Xx \doteq X$ (falls $x \notin FV(X)$) in der Definition von $[x]$.

Jedoch gilt selbst dann (ξ) nicht:
$[x].\mathbf{S}xyz \simeq \mathbf{S}(\mathbf{S}(\mathbf{S}(\mathbf{KS})\mathbf{I})(\mathbf{K}y))(\mathbf{K}z)$
$[x].xz(yz) \simeq \mathbf{S}(\mathbf{SI}(\mathbf{K}z))(\mathbf{K}(yz))$
Man kann mit dieser neuen Definition von $[x]$ jedoch zeigen, daß
$\lambda\beta \vdash M = N \iff (\mathcal{CL}w + \otimes) \vdash M_{\mathcal{CL}} = N_{\mathcal{CL}}$
$(\mathcal{CL}w + \otimes) \vdash X = Y \iff \lambda\beta \vdash X_\lambda = Y_\lambda$
wobei $\otimes$ eine Erweiterung von $\mathcal{CL}w$ um ein bestimmtes Regelschema bzw. eine endliche Menge von Axiomen ist (vgl. Hindley/Seldin Ch. 9).
$(\mathbf{S}_\lambda)_{\mathcal{CL}} =_w \mathbf{S}$ gilt bei der modifizierten Definition von $[x]$ nicht. Eine weitergehende Modifikation ist jedoch möglich, so daß statt Lemma 2.10 jetzt gilt:

(i) $(X_\lambda)_{\mathcal{CL}} \simeq X$

(ii) $(M_{\mathcal{CL}})_\lambda =_\beta M$

Damit ist die Auswertung von übersetzten λ-Termen in $\mathcal{CL}w$ korrekt, was in der funktionalen Programmierung oft ausgenutzt wird.

- Bezüglich Vollständigkeit gilt folgendes: wir betrachten eine erweiterte Sprache, in der zusätzliche Funktionen ausgewertet werden können (sog. δ-Regeln). Die entsprechenden Reduktionen sind $\rhd_{1\beta\delta}, \rhd_{\beta\delta}, \rhd_{1l\beta\delta}$ ("l" für leftmost). Dann gilt:
Wenn $M \rhd_{1l\beta\delta} N$, mit M abgeschlossen und nicht von der Form $[x].P$, so ist $M_{\mathcal{CL}} \rhd_{1w\delta} N_{\mathcal{CL}}$.
Falls ein getyptes System 2. Stufe mit `Int`, `Bool`, `Char` als Grundtypen gegeben ist, in dem ein Fixpunktoperator $\mathbf{\Upsilon}$ existiert und M vom Grundtyp ist, dann gilt:
$M \rhd_{l\beta\delta} N \implies M_{\mathcal{CL}} \rhd_{w\delta} N_{\mathcal{CL}}$
Das bedeutet, daß man *alles*, was man in $\lambda\beta$ finden kann, durch Übersetzung in $\mathcal{CL}w$ finden kann. Dies wird zum Beispiel in der funktionalen Programmiersprache Miranda ausgenutzt (Turner 1979).
Es ergibt sich daraus für eine Konstante c eines Grundtyps (die per definitionem in β-Normalform ist):
$\lambda\delta \vdash M = c \implies M =_{\beta\delta} c \implies M \rhd_{l\beta\delta} c \implies M_{\mathcal{CL}} \rhd_{w\delta} c \implies \mathcal{CL}w\delta \vdash M_{\mathcal{CL}} = c$
sowie umgekehrt
$\mathcal{CL}w\delta \vdash M_{\mathcal{CL}} = c \implies M_{\mathcal{CL}} =_{w\delta} c \implies M_{\mathcal{CL}} \rhd_{w\delta} c \implies \underbrace{(M_{\mathcal{CL}})_\lambda}_{=_\beta M} \rhd_{\beta\delta} c \implies \lambda\beta \vdash M = c$
Jede Berechnung eines Wertes für M kann also in $\mathcal{CL}w$ durchgeführt werden.

3 Der getypte λ-Kalkül

Es gibt zwei Versionen der Typisierung des λ-Kalküls:

- *Curry-Typisierung*: Terme sind die Terme der ungetypten Theorie. Jeder Term hat eine Menge möglicher Typen (*implizite* Typisierung, "type assignment").
- *Church-Typisierung*: Terme haben assoziierte Typen, die damit in der Regel eindeutig sind (*explizite* Typisierung).

Wir behandeln die Curry-Typisierung, und zwar für die einfachste Form, die nur Funktionstypen enthält (Bezeichnung des Kalküls: $\lambda{\rightarrow}$). Wir folgen dabei der Darstellung in Barendregt (1992).

Bemerkung.
Für den getypten λ-Kalkül ($\lambda{\rightarrow}$) gilt starke Normalisierung. Daher sind nicht alle rekursiven Funktionen definierbar — die partiellen Funktionen sowieso nicht, aber auch nicht alle totalen.
Definiere dazu die Funktion F wie folgt:
$F(n,m) = k \iff$ der n-te getypte Term angewandt auf Argument $\underline{m}$ hat die β-Normalform $\underline{k}$,
$F(n,m) = 0$ sonst (bei geeigneter Aufzählung der getypten Terme).
Dann kann die (totale) Funktion $g(n) =_{\text{def}} F(n,n)+1$ nicht in $\lambda{\rightarrow}$ definierbar sein:
Sei g in $\lambda{\rightarrow}$ definiert durch den p-ten getypten Term. Dann ist $g(p) = F(p,p)$; aber nach Definition ist $g(p) = F(p,p)+1$, ein Widerspruch.

3.1 Implizite Typisierung

Definition 3.1.
Die Menge der Typen T *von* $\lambda{\rightarrow}$ *ist wie folgt definiert:*

1. Typvariable $\alpha, \alpha', \alpha'', \alpha''', \ldots$ *sind Typen.*
2. *Mit* σ *und* τ *ist* $(\sigma \rightarrow \tau)$ *ein Typ.*

Es steht $\sigma_1 \rightarrow \sigma_2 \rightarrow \cdots \rightarrow \sigma_{n-1} \rightarrow \sigma_n$ *für* $\sigma_1 \rightarrow (\sigma_2 \rightarrow (\cdots (\sigma_{n-1} \rightarrow \sigma_n) \cdots))$
Ein Urteil *hat die Form* $M : \sigma$ *für einen* λ*-Term* M *und einen Typ* σ*. Dabei heißt* M *das* Subjekt *des Urteils.*
Eine Deklaration *ist ein Urteil, dessen Subjekt eine Termvariable ist.*
Eine Basis Γ *ist eine endliche Menge von Deklarationen, deren Subjekte paarweise verschieden sind. Eine* Sequenz *hat die Form* $\Gamma \vdash M : \sigma$ *für eine Basis* Γ *und ein Urteil* $M : \sigma$*.*

Definition 3.2.
Sequenzen $\Gamma \vdash M:\sigma$, die ausdrücken, daß das Urteil $M:\sigma$ in der Basis Γ gilt, kann man mit folgenden Regeln im Kalkül $\lambda\!\rightarrow$ herleiten:

- *(Id)* $\Gamma, x:\sigma \vdash x:\sigma$
- *(→I)* $\dfrac{\Gamma, x:\sigma \vdash M:\tau}{\Gamma \vdash (\lambda x.M):\sigma\to\tau}$
- *(→E)* $\dfrac{\Gamma \vdash M:\sigma\to\tau \quad \Gamma \vdash N:\sigma}{\Gamma \vdash (MN):\tau}$

Ist $\Gamma \vdash M:\sigma$ in $\lambda\!\rightarrow$ herleitbar, schreiben wir $\Gamma \vdash_{\lambda\to} M:\sigma$ oder auch $\Gamma \vdash M:\sigma$. (Wir identifizieren also häufig Sequenzen mit der Behauptung ihrer Herleitbarkeit. Aus dem Kontext ergibt es sich dann, was gemeint ist.)

Beispiel.

- $\vdash \lambda xy.x:\sigma\to\tau\to\sigma$

$$\dfrac{\dfrac{\dfrac{}{x:\sigma, y:\tau \vdash x:\sigma}}{x:\sigma \vdash \lambda y.x:\tau\to\sigma}}{\vdash \lambda x.\lambda y.x:\sigma\to\tau\to\sigma}$$

- $\vdash \lambda xyz.xz(yz):(\sigma\to\tau\to\tau')\to(\sigma\to\tau)\to\sigma\to\tau'$ *(S-Kombinator)*

Bemerkung.
Konstanten können hinzukommen. Entsprechende Konstantendeklarationen gehören dann zu jeder Basis dazu. Ein Beispiel ist der Fixpunktkombinator $\mathbf{Y}:(\sigma\to\sigma)\to\sigma$ für alle σ in der Programmiersprache ML.

Definition 3.3.
Ein geschlossener Term M heißt typbar, *falls $\vdash M:\sigma$ für ein σ. Ein Term M mit freien Variablen $x_1,\dots,x_n$ heißt* typbar, *falls $\Gamma \vdash M:\sigma$ für ein σ, wobei $\Gamma = \{x_1:\sigma_1,\dots,x_n:\sigma_n\}$ für gewisse $\sigma_1,\dots,\sigma_n$.*
Sei $\Gamma = \{x_1:\sigma_1,\dots,x_n:\sigma_n\}$ eine Basis. Dann sei $\Gamma(x_i) =_{\text{def}} \sigma_i$.
Sei V eine Menge von Variablen. Dann sei $\Gamma|_V =_{\text{def}} \{x:\sigma \mid x \in V, \Gamma(x) = \sigma\}$
$dom\,\Gamma =_{\text{def}} \{x_1,\dots,x_n\}$
Substitution von Typen: $\sigma[\tau/\alpha]$ bedeutet eine gleichzeitige Ersetzung aller in σ vorkommenden α durch τ.

Lemma 3.4.

1. *Wenn* $\Gamma \subseteq \Gamma' \implies (\Gamma \vdash M\!:\!\sigma,$ *dann* $\Gamma' \vdash M\!:\!\sigma)$. *(Monotonie)*
2. *Wenn* $\Gamma \vdash M\!:\!\sigma$, *dann* $FV(M) \subseteq dom\,\Gamma$.
3. *Wenn* $\Gamma \vdash M\!:\!\sigma$, *dann* $\Gamma|_{FV(M)} \vdash M\!:\!\sigma$.
4. *Wenn* $\Gamma \vdash x\!:\!\sigma$, *dann* $(x\!:\!\sigma) \in \Gamma$.
5. *Wenn* $\Gamma \vdash MN\!:\!\sigma$, *dann* $\Gamma \vdash M\!:\!\tau \to \sigma$ *und* $\Gamma \vdash N\!:\!\tau$ *für ein* τ.
6. *Wenn* $\Gamma \vdash (\lambda x.M)\!:\!\sigma$, *dann* $\sigma \doteq \sigma_1 \to \sigma_2$ *für gewisse* σ_1, σ_2 *und* $\Gamma, x\!:\!\sigma_1 \vdash M\!:\!\sigma_2$.
7. *Wenn* M' *Teilterm von* M *ist und* $\Gamma \vdash M\!:\!\sigma$, *dann* $\Gamma' \vdash M'\!:\!\sigma'$ *für gewisse* Γ', σ'.
8. *Wenn* $\Gamma \vdash M\!:\!\sigma$, *dann* $\Gamma[\tau/\alpha] \vdash M\!:\!\sigma[\tau/\alpha]$.
9. *Wenn* $\Gamma, x\!:\!\sigma \vdash M\!:\!\tau$ *und* $\Gamma \vdash N\!:\!\sigma$, *dann* $\Gamma \vdash M[N/x]\!:\!\tau$.
10. *Wenn* $M \rhd_\beta M'$ *und* $\Gamma \vdash M\!:\!\sigma$, *dann* $\Gamma \vdash M'\!:\!\sigma$. *(Subjektreduktion)*

Bemerkung.
Umgekehrt (zu Lemma 3.1 (10)) gilt nicht Invarianz gegenüber Expansion $\lhd_\beta$:
Im Falle von $M \lhd_\beta M'$ gilt i.a. nicht: wenn $\Gamma \vdash M\!:\!\sigma$, dann $\Gamma \vdash M'\!:\!\sigma$.
Beispiel: $\vdash I\!:\!\sigma \to \sigma$, aber $\not\vdash KI(\lambda x.xx)\!:\!\sigma \to \sigma$, obwohl $I \lhd_\beta KI(\lambda x.xx)$.

Wir zeigen jetzt, daß alle typbaren Terme stark normalisierbar sind. (Die schwache Normalisierbarkeit wurde schon von Turing gezeigt; die starke Normalisierbarkeit geht auf Tait zurück.) Die Umkehrung dieser Behauptung gilt nicht, wie das Beispiel des nicht typbaren Terms $\lambda x.xx$ zeigt.

Definition 3.5.
Sei $\mathcal{SN}$ *die Menge der stark normalisierbaren* λ*-Terme.*
Für Mengen A, B *von* λ*-Termen sei* $A \to B =_{\text{def}} \{M \mid$ *für alle* $N \in A$ *ist* $MN \in B\}$.
Die Interpretation *von Typen ist wie folgt induktiv erklärt:*

- $[\![\alpha]\!] =_{\text{def}} \mathcal{SN}$ *für alle Typvariablen* α
- $[\![\sigma \to \tau]\!] =_{\text{def}} [\![\sigma]\!] \to [\![\tau]\!]$

Bemerkung.
Die Interpretation eines Pfeiltyps ist eine Menge von λ-Termen, welche hinsichtlich des vorgegebenen Vor- und Nachbereichs die gewünschte Überführungseigenschaft haben.

Definition 3.6. *Eine Menge A von Termen heißt* saturiert*, wenn gilt:*

a) $A \subseteq \mathcal{SN}$

b) $xR_1 \dots R_n \in A$*, falls x Termvariable und* $R_1, \dots, R_n \in \mathcal{SN}$ $(n \geq 0)$

c) $(\lambda x.M)NR_1 \dots R_n \in A$*, falls* $(M[{}^N\!/\!_x])R_1 \dots R_n \in A$ *für* $N, R_1, \dots, R_n \in \mathcal{SN}$ $(n \geq 0)$

$\mathcal{SAT} =_{\mathrm{def}} \{A \mid A \text{ saturiert}\}$

Lemma 3.7.
Für jeden Typ σ von $\lambda\!\to$ gilt: $[\![\sigma]\!]$ ist saturiert.

Beweis.

1. σ ist Typvariable. Zu zeigen: $\mathcal{SN}$ ist saturiert.

   a) $\mathcal{SN} \subseteq \mathcal{SN}$

   b) $xR_1 \dots R_n \in \mathcal{SN}$, falls $R_1, \dots, R_n \in \mathcal{SN}$

   c) Sei $(M[{}^N\!/\!_x])R_1 \dots R_n \in \mathcal{SN}$ mit $N, R_1, \dots, R_n \in \mathcal{SN}$
   Dann gilt auch: $M \in \mathcal{SN}$, sonst könnte $(M[{}^N\!/\!_x])R_1 \dots R_n$ nicht stark normalisierbar sein. Wir betrachten $(\lambda x.M)NR_1 \dots R_n$. Jede Reduktionfolge sieht dann so aus:
   $$\begin{array}{lll} (\lambda x.M)NR_1 \dots R_n & \rhd_\beta & \dots \\ & \rhd_\beta & (\lambda x.M')N'R'_1 \dots R'_n \\ & \rhd_{1\beta} & M'[{}^{N'}\!/\!_x]R_1 \dots R_n \\ & \rhd_\beta & \dots \end{array}$$
   wobei $M \rhd_\beta M'$, $N \rhd_\beta N'$ und $R_i \rhd_\beta R'_i$ für alle i.
   Damit erhält man: $(M[{}^N\!/\!_x])R_1 \dots R_n \rhd_\beta (M'[{}^{N'}\!/\!_x])R'_1 \dots R'_n \rhd_\beta \cdots$. Da diese Folge terminiert, terminiert auch die erste, d.h. $(\lambda x.M)NR_1 \dots R_n$ ist stark normalisierbar.

2. Mit A und B ist $A \to B$ saturiert:

   a) Sei $M \in A \to B$. Wegen Definition 3.6 b) gilt $x \in A$ für alle Variablen x. Also $Mx \in B$. Da Mx stark normalisierbar, ist M stark normalisierbar.

   b) Sei $M \in A$. Dann ist $M \in \mathcal{SN}$. Dann ist $xR_1 \dots R_n M \in B$ für $R_i \in \mathcal{SN}$. Damit ist auch $xR_1 \dots R_n \in A \to B$.

   c) Sei $M \in A$. Dann ist $M \in \mathcal{SN}$.
   Dann $(\lambda x.P)NR_1 \dots R_n M \in B$, falls $(P[{}^N\!/\!_x])R_1 \dots R_n M \in B$.
   Dann $(\lambda x.P)NR_1 \dots R_n \in A \to B$, falls $(P[{}^N\!/\!_x])R_1 \dots R_n \in A \to B$.

$\square$

Definition 3.8.
Eine Bewertung ρ *ist eine Abbildung* ρ : *Variablen* $\longrightarrow$ λ*-Terme.*
$[\![M]\!]_\rho = M[{}^{\rho(x_1)}/x_1, \ldots, {}^{\rho(x_n)}/x_n]$, *wobei* $\{x_1, \ldots, x_n\}$ *Menge* aller *freien Variablen in* M*.*
$\rho \models M:\sigma$ *("*ρ erfüllt $M:\sigma$*"), falls* $[\![M]\!]_\rho \in [\![\sigma]\!]$*.*
$\rho \models \Gamma$*, falls* $\rho \models x:\sigma$ *für alle* $(x:\sigma) \in \Gamma$*.*
$\Gamma \models M:\sigma$*, falls für alle* ρ *gilt: wenn* $\rho \models \Gamma$*, dann* $\rho \models M:\sigma$*.*

Lemma 3.9.
Wenn $\Gamma \vdash M:\sigma$*, dann* $\Gamma \models M:\sigma$*.*

Beweis.
Induktion über der Struktur der Ableitung von $M:\sigma$.

1. $\Gamma', x:\sigma \vdash x:\sigma$ $\surd$

2. $M \mathrel{\hat{=}} PQ$: Dann $\Gamma \vdash P:\tau\to\sigma$ und $\Gamma \vdash Q:\tau$.
   Nach Induktionsvoraussetzung $\Gamma \models P:\tau\to\sigma$ und $\Gamma \models Q:\tau$.
   D.h., falls $\rho \models \Gamma$, dann $[\![P]\!]_\rho \in [\![\tau\to\sigma]\!]$ und $[\![Q]\!]_\rho \in [\![\tau]\!]$.
   Dann $[\![PQ]\!]_\rho \in [\![\sigma]\!]$ nach Definition von $[\![\ ]\!]$, da $[\![PQ]\!]_\rho = [\![P]\!]_\rho[\![Q]\!]_\rho$.

3. $M \mathrel{\hat{=}} \lambda x.N$: Dann ist $\sigma \mathrel{\hat{=}} \sigma_1\to\sigma_2$ und $\Gamma, x:\sigma_1 \vdash N:\sigma_2$.
   Nach Induktionsvoraussetzung: $\Gamma, x:\sigma_1 \models N:\sigma_2$.
   D.h., falls $\rho \models \Gamma$ und $\rho(x) \in [\![\sigma_1]\!]$, dann $[\![N]\!]_\rho \in [\![\sigma_2]\!]$.
   Dann ist $[\![(\lambda x.N)x]\!]_\rho \in [\![\sigma_2]\!]$, da $[\![\sigma_2]\!]$ saturiert.
   D.h., falls $\rho \models \Gamma$ und $\rho(x) = Q \in [\![\sigma_1]\!]$, dann $[\![\lambda x.N]\!]_\rho Q = [\![(\lambda x.N)x]\!]_\rho \in [\![\sigma_2]\!]$.
   Also $[\![\lambda x.N]\!]_\rho \in [\![\sigma_1\to\sigma_2]\!]$.

$\square$

Theorem 3.10. *Wenn* $\Gamma \vdash M:\sigma$*, dann* M *ist stark normalisierbar.*

Beweis.
Wenn $\Gamma \vdash M:\sigma$, dann $\Gamma \models M:\sigma$, d.h. für alle Bewertungen ρ gilt: wenn $\rho \models \Gamma$, dann $\rho \models M:\sigma$.
Wir wählen $\rho = id$. $id \models \Gamma$ gilt, da $x \in [\![\sigma]\!]$ für jedes σ (da σ saturiert).
Also gilt $M \in [\![\sigma]\!]$. $[\![\sigma]\!]$ ist saturiert und damit stark normalisierbar. $\square$

3.2 Der Typisierungsalgorithmus

Im folgenden stellen wir einen Algorithmus zur Typisierung von Termen dar. Dieser Algorithmus liefert einen Typ, falls der Term typisierbar ist, und gibt `fail` aus, falls dieser Term nicht typisierbar ist.

Damit lassen sich sofort die beiden wichtigsten Entscheidbarkeitsfragen der impliziten Typisierung lösen:

1. Gegeben M und σ, gilt $\vdash M\!:\!\sigma$? (Typprüfung, "type checking")
2. Gegeben M, gibt es σ mit $\vdash M\!:\!\sigma$? (Typbarkeit, "typability")

Die dritte Entscheidbarkeitsfrage in diesem Kontext

3. Gegeben σ, gibt es M mit $\vdash M\!:\!\sigma$? ("inhabitation")

wird durch den Curry-Howard-Isomorphismus am Schluß dieses Kapitels gelöst.

Zunächst sind einige Vorbereitungen über Substitution und Unifikation von Typvariablen und Typen notwendig.

Definition 3.11.
Eine Substitution (in Typen) *ist eine Abbildung*
s : *Typvariablen* $\longrightarrow$ *Typen,*
wobei $s(\alpha) \not\equiv \alpha$ *nur für endlich viele* α. *Wir schreiben* s *auch als* $\{\alpha_1 = \sigma_1, \ldots, \alpha_n = \sigma_n\}$, *falls* $s(\alpha_i) = \sigma_i$.
Offenbar determiniert s *eine Abbildung* $\bar{s}$ *von Typen in Typen:*

$$\begin{array}{rcl} \bar{s}(\alpha) & \simeq_{\text{def}} & s(\alpha) \\ \bar{s}(\sigma \to \tau) & \simeq_{\text{def}} & \bar{s}(\sigma) \to \bar{s}(\tau) \end{array}$$

Wir identifizieren s *und* $\bar{s}$ *und schreiben* $s(\sigma)$ *oder* $\sigma[{}^{\sigma_1}\!/\!_{\alpha_1}, \ldots, {}^{\sigma_n}\!/\!_{\alpha_n}]$.
Für Substitutionen s_1 *und* s_2 *ist* $s_1 \circ s_2$ *(kurz:* $s_1 s_2$*) in natürlicher Weise definiert. Entsprechend ist* $s_1 \circ s_2(\sigma) \simeq s_1(s_2(\sigma))$.
Ein Unifikator *für* σ *und* τ *ist ein* s *mit* $s(\sigma) \simeq s(\tau)$, *ein Unifikator für eine Menge von Gleichungen* $E = \{\sigma_1 = \tau_1, \ldots, \sigma_n = \tau_n\}$ *ist ein* s *mit* $s(\sigma_i) \simeq s(\tau_i)$ *für alle* i $(1 \leq i \leq n)$.
Ein allgemeinster Unifikator (mgu — "most general unifier") *von* σ *und* τ *(bzgl.* E*) ist ein Unifikator* s, *so daß für jeden anderen Unifikator* s' *von* σ *und* τ *(bzgl.* E*) gilt:* $s' = s_1 \circ s$ *für eine Substitution* s_1.
Wir schreiben: $s = mgu(\sigma, \tau)$ *bzw.* $s = mgu(E)$.
τ *ist* Variante *von* σ, *falls es* s_1 *und* s_2 *gibt mit* $s_1(\tau) \simeq \sigma$ *und* $s_2(\sigma) \simeq \tau$.

Beispiel.

- $\alpha \to (\beta \to \alpha)$ *und* $\alpha \to (\beta \to \alpha)$ *haben* $\{{}^{\alpha}\!/\!_{\beta}\}$ *als mgu.*
- $\beta \to (\alpha \to \beta)$ *und* $(\gamma \to \gamma) \to \delta$ *haben* $\{{}^{\gamma \to \gamma}\!/\!_{\beta}, {}^{\alpha \to (\gamma \to \gamma)}\!/\!_{\delta}\}$ *als mgu.*

Bemerkung.
Zwei *mgu*s bezüglich derselben Menge sind immer Varianten voneinander. In diesem Sinn sind also *mgu*s eindeutig.

Theorem 3.12.
Es gibt einen Algorithmus, der zu jedem Gleichungssystem E einen mgu von E liefert, falls E unifizierbar ist, und `fail` liefert, falls E nicht unifizierbar ist.

BEWEIS.
Siehe Logikprogrammierung. Ein Verfahren, das auf Herbrand zurückgeht, besteht aus Regeln zur Umformung von Gleichungssystemen $E = \{\sigma_1 = \tau_1, \ldots, \sigma_n = \tau_n\}$:

- (*id*) $\dfrac{E \mathbin{\dot\cup} \{\sigma = \sigma\}}{E}$
- (*sym*) $\dfrac{E \mathbin{\dot\cup} \{\sigma = \alpha\}}{E \cup \{\alpha = \sigma\}}$ falls σ keine Variable ist
- (*fail*) $\dfrac{E \mathbin{\dot\cup} \{\alpha = \sigma\}}{\texttt{fail}}$ falls α in σ vorkommt
- (*subst*) $\dfrac{E \mathbin{\dot\cup} \{\alpha = \sigma\}}{E[\sigma/\alpha] \cup \{\alpha = \sigma\}}$ falls α nicht in σ vorkommt und α in E vorkommt
- (*func*) $\dfrac{E \mathbin{\dot\cup} \{\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2\}}{E \cup \{\tau_1 = \sigma_1, \tau_2 = \sigma_2\}}$

Es gilt: die Anwendung dieser Reduktionsregeln auf eine Gleichungsmenge terminiert. Das Resultat ist entweder `fail` oder ist $\{\alpha_1 = \sigma_1, \ldots, \alpha_n = \sigma_n\}$, wobei $\{\alpha_1 = \sigma_1, \ldots, \alpha_n = \sigma_n\}$ der *mgu* von E ist. □

Gleichungssystemen dieser Art ordnen wir nun Sequenzen zu.

Definition 3.13.
Das mit der Sequenz $\Gamma \vdash M:\sigma$ assoziierte Gleichungssystem $E(\Gamma \vdash M:\sigma)$ ist wie folgt definiert:

1. $E(\Gamma \vdash x:\sigma) =_{\text{def}} \{\sigma = \Gamma(x)\}$
2. $E(\Gamma \vdash \lambda x.M:\sigma) =_{\text{def}} \{\sigma = \alpha \to \beta\} \cup E(\Gamma, x:\alpha \vdash M:\beta)$ *für neue Typvariablen α, β*
3. $E(\Gamma \vdash MN:\sigma) =_{\text{def}} E(\Gamma \vdash M:\alpha \to \sigma) \cup E(\Gamma \vdash N:\alpha)$ *für neue Typvariable α*

Beispiel.

- $E(x:\alpha \vdash x:\beta) = \{\beta = \alpha\}$

- $$\begin{aligned} E(\vdash \lambda xy.x : \alpha \to \beta) &= \{\alpha \to \beta = \alpha_1 \to \alpha_2\} \cup E(x : \alpha_1 \vdash \lambda y.x : \alpha_2) \\ &= \{\alpha \to \beta = \alpha_1 \to \alpha_2, \alpha_2 = \alpha_3 \to \alpha_4\} \cup E(x : \alpha_1, y : \alpha_3 \vdash x : \alpha_4) \\ &= \{\alpha \to \beta = \alpha_1 \to \alpha_2, \alpha_2 = \alpha_3 \to \alpha_4, \alpha_4 = \alpha_1\} \end{aligned}$$

  Lösung:

  $$\cfrac{\cfrac{\cfrac{\{\alpha \to \beta = \alpha_1 \to \alpha_2, \alpha_2 = \alpha_3 \to \alpha_4, \alpha_4 = \alpha_1\}}{\{\alpha = \alpha_1, \beta = \alpha_2, \alpha_2 = \alpha_3 \to \alpha_4, \alpha_4 = \alpha_1\}}\,(func)}{\{\alpha = \alpha_1, \beta = \alpha_3 \to \alpha_4, \alpha_2 = \alpha_3 \to \alpha_4, \alpha_4 = \alpha_1\}}\,(subst)}{\{\alpha = \alpha_1, \beta = \alpha_3 \to \alpha_1, \alpha_2 = \alpha_3 \to \alpha_1, \alpha_4 = \alpha_1\}}\,(subst)$$

  Es gilt also $\vdash \lambda xy.x : \alpha_1 \to (\alpha_3 \to \alpha_1)$

Lemma 3.14.

(i) Sei s Lösung von $E(\Gamma \vdash M : \sigma)$. Dann gilt: $s(\Gamma) \vdash M : s(\sigma)$

(ii) Wenn $s(\Gamma) \vdash M : s(\sigma)$, dann gilt: es gibt ein s', das die Typvariablen in Γ und σ wie s interpretiert, und s' ist Lösung von $E(\Gamma \vdash M : \sigma)$. Dabei können die Typvariablen, die von s und s' verschieden interpretiert werden, aus einer festen Typvariablenmenge V mit $V \cap FV(\Gamma \cup \{\sigma\}) = \emptyset$ gewählt werden.

Beweis.

(i) Induktion über der Struktur von M

- s ist Lösung von $E(\Gamma \vdash x : \sigma)$, d.h. $s(\sigma) = s(\Gamma(x))$. Also kommt $x : s(\sigma)$ in Γ vor. Somit $s(\Gamma) \vdash x : s(\sigma)$.
- Wenn s Lösung von $E(\Gamma \vdash \lambda x.M : \sigma)$ ist, dann ist s Lösung von $\{\sigma = \alpha \to \beta\} \cup E(\Gamma, x : \alpha \vdash M : \beta)$.
  Nach Induktionsvoraussetzung ist $s(\Gamma), x : s(\alpha) \vdash M : s(\beta)$, damit $s(\Gamma) \vdash \lambda x.M : (s(\alpha) \to s(\beta)) = s(\sigma)$
- Wenn s Lösung von $E(\Gamma \vdash MN : \sigma)$ ist, dann ist s Lösung von $E(\Gamma \vdash M : \alpha \to \sigma)$ und von $E(\Gamma \vdash N : \alpha)$
  Nach Induktionsvoraussetzung ist $s(\Gamma) \vdash M : s(\alpha \to \sigma)$ und $s(\Gamma) \vdash N : s(\alpha)$, damit $s(\Gamma) \vdash MN : s(\sigma)$

(ii) Induktion über der Struktur von M

- Es ist $s(\Gamma) \vdash x : s(\sigma)$, d.h. $(x : s(\sigma)) \in s(\Gamma)$. Daher $s(\sigma) = s(\Gamma(x))$.
  Somit ist s Lösung von $E(\Gamma \vdash x : \sigma)$.

- Es ist $s(\Gamma) \vdash \lambda x.M : s(\sigma)$, d.h. $s(\sigma) = \sigma_1 \to \sigma_2$ für gew. σ_1, σ_2 und $s(\Gamma), x : \sigma_1 \vdash M : \sigma_2$. Sei s' wie s, jedoch erweitert um $\alpha_1 \mapsto \sigma_1$ und $\alpha_2 \mapsto \sigma_2$ mit α_1, α_2 neu. Dann ist $s'(\Gamma), x : \alpha_1 \vdash M : s'(\alpha_2)$. Damit stimmen s und s' auf den in Γ und σ vorkommenden Typvariablen überein, und s' ist Lösung von $\{\sigma = \alpha_1 \to \alpha_2\}$. Nach Induktionsvoraussetzung gibt es s'' als Lösung von $E(\Gamma, x : \alpha_1 \vdash M : \alpha_2)$ derart, daß s'' mit s' auf den Typvariablen in $\Gamma, \alpha_1, \alpha_2$ übereinstimmt. Ferner kann angenommen werden, daß die Typvariablen, die von s' und s'' verschieden interpretiert werden, nicht in σ vorkommen, also $s'(\sigma) = s''(\sigma)$.
  Damit ist s'' also Lösung von $\{\sigma = \alpha_1 \to \alpha_2\} \cup E(\Gamma, x : \alpha_1 \vdash M : \alpha_2)$, das heißt von $E(\Gamma \vdash \lambda x.M : \sigma)$, und stimmt auf den Typvariablen in Γ und σ mit s überein.
- Es ist $s(\Gamma) \vdash MN : \sigma$, d.h. $s(\Gamma) \vdash M : \tau \to s(\sigma)$ und $s(\Gamma) \vdash N : \tau$ für ein τ.
  Wir definieren s' als s, erweitert um $\alpha \mapsto \tau$, wobei α neu. Dann stimmt s' mit s auf den Typvariablen in Γ und σ überein. Wir haben also $s'(\Gamma) \vdash M : s'(\alpha) \to s'(\sigma)$ und $s'(\Gamma) \vdash N : s'(\alpha)$. Nach Induktionsvoraussetzung gibt es Lösungen s''_1 und s''_2 von $E(\Gamma \vdash M : \alpha \to \sigma)$ bzw. $E(\Gamma \vdash N : \alpha)$, die mit s' auf den Typvariablen in Γ, σ, α übereinstimmen. Ferner können wir annehmen, daß die bei der Konstruktion von $E(\Gamma \vdash M : \alpha \to \sigma)$ und $E(\Gamma \vdash N : \alpha)$ neu eingeführten Typvariablen voneinander verschieden sind.
  Dann ist $s''_1 \cup s''_2$ Lösung von $E(\Gamma \vdash M : \alpha \to \sigma) \cup E(\Gamma \vdash N : \alpha)$, das heißt also von $E(\Gamma \vdash MN : \sigma)$, und stimmt mit s auf den Typvariablen in Γ und σ überein.

$\square$

Definition 3.15.
σ heißt Haupttyp *für einen geschlossenen Term M, falls $\vdash M : \sigma$ und ferner:*
falls $\vdash M : \sigma$ und $\vdash M : \sigma'$, dann gibt es eine Substitution s, so daß $\sigma' = s(\sigma)$.
(Γ, σ) heißt Hauptpaar *für M, falls $\Gamma \vdash M : \sigma$ und ferner:*
falls $\Gamma \vdash M : \sigma$ und $\Gamma' \vdash M : \sigma'$, dann gibt es eine Substitution s, so daß $\sigma' = s(\sigma)$ und $\Gamma' \supseteq s(\Gamma)$.

Theorem 3.16.
Es gibt einen Algorithmus, der zu jedem geschlossenen Term M einen Haupttyp und zu jedem M ein Hauptpaar liefert, falls M überhaupt einen Typ hat, und ansonsten `fail` ausgibt.

Beweis.
Sei $\Gamma_0 =_{\text{def}} \{x_1 : \alpha_1, \ldots, x_n : \alpha_n\}$ und $\sigma_0 =_{\text{def}} \beta$, wobei $x_1, \ldots, x_n$ die freien Variablen in M sind. Jede *mgu*-Lösung von $E(\Gamma_0 \vdash M : \sigma_0)$ ist Hauptpaar von M, falls es eine Lösung gibt, sonst wird `fail` ausgegeben.

1. M hat Typ genau dann, wenn $\Gamma \vdash M : \sigma$ für gewisse Γ, σ
   genau dann, wenn $s(\Gamma_0) \vdash M : s(\sigma_0)$ für gewisses s
   genau dann, wenn $E(\Gamma_0 \vdash M : \sigma_0)$ ist lösbar (Lemma 3.14 (i))

2. Sei s *mgu*-Lösung von $E(\Gamma_0 \vdash M : \sigma_0)$, dann ist $s(\Gamma_0) \vdash M : s(\sigma_0)$. Sei nun $\Gamma' \vdash M : \sigma'$. Dann ist $\tilde{\Gamma} \vdash M : \sigma'$, wobei $\tilde{\Gamma} =_{\text{def}} \Gamma'|_{FV(M)}$. Sei s' so gewählt, daß $s'(\Gamma_0) = \tilde{\Gamma}$, sowie $s'(\sigma_0) = \sigma'$. Dann $s'(\Gamma_0) \vdash M : s'(\sigma_0)$. Dann gilt für ein s'', das die Typvariablen in Γ_0 und σ_0 wie s' interpretiert: s'' ist Lösung von $E(\Gamma_0 \vdash M : \sigma_0)$ (Lemma 3.14 (ii)). Da s *mgu* ist, gilt $s'' = S_1 \circ S$ für ein s_1, das heißt $\sigma' = s''(\sigma_0) = S_1(s(\sigma_0))$. Ferner haben wir, daß $\Gamma \supseteq \tilde{\Gamma}$ mit $\tilde{\Gamma} = s''(\Gamma_0) = S_1(s(\Gamma_0))$. $(s(\Gamma_0), s(\sigma_0))$ ist also Hauptpaar.

□

Theorem 3.17.
Typprüfung und Typisierbarkeit sind entscheidbar.

Beweis.

- Sei geschlossenes M gegeben. Der Algorithmus aus Theorem 3.16 liefert einen Haupttyp σ, wenn M typisierbar ist, sonst `fail`.
- Zur Prüfung von $\vdash M : \sigma'$ für gegebenes σ' muß noch geprüft werden, ob $\sigma' = s(\sigma)$ für ein s. Das ist leicht algorithmisch möglich.

□

Theorem 3.18.
Das Problem, ob es zu einem gegebenen Typ σ einen Term gibt, ist entscheidbar.

Beweis.
Es gibt M mit $\vdash M : \sigma$ genau dann, wenn es einen Beweis für σ (als Formel) in der positiven Implikationslogik gibt. Dies ergibt sich aus der im folgenden dargestellten sog. "Curry-Howard-Interpretation" (Theorem 3.23). Die positive Implikationslogik ist entscheidbar. □

3.3 Der Curry-Howard-Isomorphismus

Definition 3.19.
Typvariablen heißen auch Aussagevariablen, *Typen auch* (implikationslogische) Formeln.
Eine endliche Menge von Formeln heißt Kontext. *Metasprachliche Variablen für Kontexte sind* $\Delta, \Delta', \ldots$
Die Axiome und Regeln der positiven Implikationslogik $P\!\rightarrow$ sind:

- *(Id)* $\Delta, \sigma \vdash \sigma$

- *(→I)* $\dfrac{\Delta, \sigma \vdash \tau}{\Delta \vdash \sigma \to \tau}$

- *(→E)* $\dfrac{\Delta \vdash \sigma \to \tau \quad \Delta \vdash \sigma}{\Delta \vdash \tau}$

$\Delta \vdash_{P\to} \sigma$ bedeutet, daß $\Delta \vdash \sigma$ in $P\to$ herleitbar ist.
Für ein Urteil $M{:}\sigma$ sei $(M{:}\sigma)^\circ \simeq \sigma$.
Für eine Basis $\Gamma = \{x_1{:}\sigma_1, \ldots, x_n{:}\sigma_n\}$ sei Γ° der Kontext $\{\sigma_1, \ldots, \sigma_n\}$.

Lemma 3.20.
Wenn $\Gamma \vdash_{\lambda\to} M{:}\sigma$, dann $\Gamma^\circ \vdash_{P\to} \sigma$.

Beweis.
Lasse Terme weg. □

Lemma 3.21.
Es gibt einen Algorithmus, der zu jedem typbaren Term M eine Ableitung von $\Delta \vdash \sigma$ in $P\to$ liefert derart, daß $\Delta = \Gamma^\circ$ und $\Gamma \vdash_{\lambda\to} M{:}\sigma$.

Beweis.
Theorem 3.16 □

D.h.: Jeder typbare Term M kodiert eine Ableitung in $P\to$. Aus dieser Ableitung lassen sich durch Substitution alle Ableitungen von $\Gamma^\circ \vdash \sigma$ in $P\to$ gewinnen, die Ableitungen von $\Gamma \vdash M{:}\sigma$ in $\lambda\to$ entsprechen.

Lemma 3.22.
Zu jeder Ableitung von $\Delta \vdash \sigma$ in $P\to$ läßt sich ein Term M und eine Ableitung von $\Gamma \vdash M{:}\sigma$ in $\lambda\to$ konstruieren mit $\Gamma^\circ = \Delta$.

Beweis.
Induktion über dem Aufbau der Ableitung von $\Delta \vdash \sigma$ in $P\to$.

- Alle in Anfangssequenzen vorkommenden Formeln σ werden durch Typdeklarationen $x{:}\sigma$ ersetzt, wobei die Variable x so gewählt ist, daß allen Vorkommen einer Formel σ *dieselbe* Deklaration $x{:}\sigma$ und *verschiedenen* Formeln σ und τ Deklarationen $x{:}\sigma$ und $y{:}\tau$ mit *verschiedenen* Variablen x und y entsprechen.

- Sei in $P\to$ der Schritt
$$(\to\mathrm{I})\ \frac{\sigma_1, \ldots \sigma_n, \sigma \vdash \tau}{\sigma_1, \ldots \sigma_n \vdash \sigma \to \tau}$$

angewendet, wobei für $\sigma_1, \dots \sigma_n, \sigma \vdash \tau$ schon eine Ableitung in $\lambda\!\rightarrow$ von

$$x_1 \!:\! \sigma_1, \dots x_n \!:\! \sigma_n, x \!:\! \sigma \vdash M \!:\! \tau$$

vorliegt. Dann verlängern wir diese unter Verwendung von ($\rightarrow$I) in $\lambda\!\rightarrow$ zu

$$x_1 \!:\! \sigma_1, \dots x_n \!:\! \sigma_n \vdash \lambda x.M \!:\! \sigma \rightarrow \tau.$$

- Sei in $P\!\rightarrow$ der Schritt

$$(\rightarrow\mathrm{E})\ \frac{\sigma_1, \dots \sigma_n \vdash \sigma \rightarrow \tau \qquad \sigma_1, \dots \sigma_n \vdash \sigma}{\sigma_1, \dots \sigma_n \vdash \tau}$$

angewendet, wobei in $\lambda\!\rightarrow$ für $\sigma_1, \dots \sigma_n \vdash \sigma \rightarrow \tau$ und $\sigma_1, \dots \sigma_n \vdash \sigma$ schon Ableitungen von

$$x_1 \!:\! \sigma_1, \dots x_n \!:\! \sigma_n \vdash M \!:\! \sigma \rightarrow \tau$$

und

$$x_1 \!:\! \sigma_1, \dots x_n \!:\! \sigma_n \vdash N \!:\! \sigma$$

vorliegen. Man beachte, daß einem Typ σ_i genau eine Variable x_i zugeordnet ist. Damit erhalten wir mit ($\rightarrow$E) eine Ableitung in $\lambda\!\rightarrow$ von

$$x_1 \!:\! \sigma_1, \dots x_n \!:\! \sigma_n \vdash MN \!:\! \tau.$$

□

Theorem 3.23 (Curry-Howard-Isomorphismus).
Sei M_P die gemäß Lemma 3.21 zu einem in $\lambda\!\rightarrow$ typbaren Term M gehörende Ableitung in $P\!\rightarrow$. Sei Π_λ der gemäß Lemma 3.22 zu einer Ableitung in $P\!\rightarrow$ gehörendeTerm von $\lambda\!\rightarrow$. Dann gilt:

(i) $(\Pi_\lambda)_P$ *ist eine Ableitung in $P\!\rightarrow$, aus der sich* Π *durch Substitution von Formeln für Aussagevariablen gewinnen läßt.*

(ii) $(M_P)_\lambda$ *ist (bis auf Umbenennung von freien und/oder gebundenen Variablen) ein Term, der aus M durch Identifizierung von freien oder von gebundenen Variablen entsteht.*

Beweis.
Aus den Lemmas 3.21 und 3.22. □

Bemerkung.
Ein Beispiel für *(ii)* ist der λ-Term $u(zx)(zy)$, der nach dem Typisierungsalgorithmus durch

$$u \!:\! \alpha \rightarrow \alpha \rightarrow \beta, z \!:\! \gamma \rightarrow \beta, x \!:\! \gamma, y \!:\! \gamma \vdash_{\lambda\rightarrow} u(zx)(zy) \!:\! \beta$$

getypt wird. Die korrespondierende Ableitung in $P{\rightarrow}$ liefert

$$\alpha \rightarrow \alpha \rightarrow \beta, \gamma \rightarrow \beta, \gamma \vdash \beta.$$

Dieser Ableitung entspricht gemäß Lemma 3.22 ein λ-Term der Form $u(zx)(zx)$, in dem x und y identifiziert sind. Zur Identifizierung der Variablen kommt es dadurch, daß beim Übergang von $\lambda{\rightarrow}$ zu $P{\rightarrow}$ Information verlorengeht, die beim Übergang von $P{\rightarrow}$ zu $\lambda{\rightarrow}$ nicht mehr rekonstruiert werden kann. Das hängt damit zusammen, daß bei der Zuordnung von Variablen zu Formeln (= Typen) in Anfangssequenzen von $\lambda{\rightarrow}$ gemäß Lemma 3.22 keine zwei Variablen denselben Typ haben können, d.h. beim Übergang von $\lambda{\rightarrow}$ zu $P{\rightarrow}$ keine Sequenz der Form $\Gamma, x{:}\sigma, y{:}\sigma \vdash M{:}\tau$ auftreten kann.

Eine andere Fassung des aussagenlogischen Kalküls erlaubt es, ohne diesen Informationsverlust auszukommen. Dazu wird auf die Beweistheorie verwiesen (siehe Trolstra/Schwichtenberg 1996).

Der Curry-Howard-Isomorphismus induziert Reduktions- und Gleichheitsrelationen für Herleitungen, die $\rhd_\beta$ und $=_\beta$ entsprechen. Diese behandelt man ebenfalls in der Beweistheorie, genauer in der Theorie des natürlichen Schließens.

4 Der polymorph getypte λ-Kalkül

Der polymorph getypte λ-Kalkül heißt auch *Getypter λ-Kalkül 2. Stufe* oder *System F*.
Die Motivation: es soll zum Beispiel die Identitätsfunktion $id \simeq \lambda x.x : \alpha \to \alpha$ unabhängig von einem speziellen Typ α sein. Man will also, daß $id \simeq \lambda x.x : \forall\alpha.(\alpha \to \alpha)$

Definition 4.1.
Die Menge der Typen des polymorph getypten λ-Kalküls (λ2) ist wie folgt gegeben:

- *Jede Typvariable ist ein Typ*
- *Mit σ und τ ist auch $(\sigma \to \tau)$ ein Typ*
- *Mit α und σ ist auch $\forall\alpha.\sigma$ ein Typ*

Konvention: $\forall$ bindet stärker als $\to$.
$\sigma[{}^{\tau}/_{\alpha}]$ *ist nur erlaubt, falls τ frei für α in σ ist.*

Definition 4.2.
Die Typisierung in λ2 ist durch folgende Regeln definiert:

- (Id) $\Gamma, x:\sigma \vdash x:\sigma$
- $(\to I)$ $\dfrac{\Gamma, x:\sigma \vdash M:\tau}{\Gamma \vdash (\lambda x.M):\sigma \to \tau}$
- $(\to E)$ $\dfrac{\Gamma \vdash M:\sigma \to \tau \quad \Gamma \vdash N:\sigma}{\Gamma \vdash MN:\tau}$
- $(\forall I)$ $\dfrac{\Gamma \vdash M:\sigma}{\Gamma \vdash M:\forall\alpha.\sigma}$ *falls* $\alpha \notin FV(\Gamma)$
- $(\forall E)$ $\dfrac{\Gamma \vdash M:\forall\alpha.\sigma}{\Gamma \vdash M:\sigma[{}^{\tau}/_{\alpha}]}$

Beispiel.

- $\lambda x.x : \forall\alpha.\alpha \to \alpha$

$$\dfrac{\dfrac{\dfrac{}{x:\sigma \vdash x:\sigma}}{\vdash \lambda x.x:\alpha \to \alpha}}{\vdash \lambda x.x:\forall\alpha.\alpha \to \alpha}$$

- $\lambda xy.y : \forall\alpha\beta.\alpha \to (\beta \to \beta)$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{x:\alpha, y:\beta \vdash y:\beta}}{x:\alpha \vdash \lambda y.y:\beta\to\beta}}{\vdash \lambda xy.y:\alpha\to(\beta\to\beta)}}{\vdash \lambda xy.y:\forall\beta.\alpha\to(\beta\to\beta)}}{\vdash \lambda xy.y:\forall\alpha\beta.\alpha\to(\beta\to\beta)}$$

- $\lambda x.xx : \forall\beta.(\forall\alpha.\alpha) \to \beta$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{x:\forall\alpha.\alpha \vdash x:\forall\alpha.\alpha}}{x:\forall\alpha.\alpha \vdash x:\alpha\to\beta} \quad \dfrac{\dfrac{}{x:\forall\alpha.\alpha \vdash x:\forall\alpha.\alpha}}{x:\forall\alpha.\alpha \vdash x:\alpha}}{x:\forall\alpha.\alpha \vdash xx:\beta}}{\vdash \lambda x.xx:(\forall\alpha.\alpha)\to\beta}}{\vdash \lambda x.xx:\forall\beta.(\forall\alpha.\alpha)\to\beta}$$

Definition 4.3.

Es sei $\sigma \sqsupset \tau$, falls

- $\tau \mathrel{\hat{=}} \forall\alpha.\sigma$ *(τ ist* Generalisierung *von σ)*
- $\sigma \mathrel{\hat{=}} \forall\alpha.\sigma_1$ *und* $\tau \mathrel{\hat{=}} \sigma_1[\sigma_2/\alpha]$ *(τ ist* Spezialisierung *von σ)*

Es sei $\sigma \sqsupseteq \tau$ genau dann, wenn es $n \geq 0$ gibt, so daß $\sigma \mathrel{\hat{=}} \sigma_1 \sqsupset \cdots \sqsupset \sigma_n \mathrel{\hat{=}} \tau$.
Es sei σ^0 der Typ σ ohne Quantorenpräfix (d.h. führende Quantoren sind weggelassen).

Beispiel.

- $(\forall\alpha_1 \ldots \alpha_n.\sigma)^0 \mathrel{\hat{=}} \sigma$
- $(\forall\alpha.(\forall\beta.\beta) \to \alpha)^0 \mathrel{\hat{=}} (\forall\beta.\beta) \to \alpha$

Bemerkung.

- $\sqsupset$ ist nicht symmetrisch:
  Zwar gilt, daß $\sigma \sqsupset \forall\alpha.\sigma \implies \forall\alpha.\sigma \sqsupset \sigma$,
  aber $\forall\alpha.\sigma_1 \sqsupset \sigma_1[\sigma_2/\alpha] \not\Longrightarrow \sigma_1[\sigma_2/\alpha] \sqsupset \forall\alpha.\sigma_1$.
- Intuitiv bedeutet $\sigma \sqsupseteq \tau$, daß $M:\tau$ aus $M:\sigma$ alleine durch Anwendung von $\forall$-Regeln herleitbar ist, also

$$\left.\begin{array}{c} \Gamma \vdash M:\sigma \\ \vdots \\ \Gamma \vdash M:\tau \end{array}\right\} \text{nur } \forall\text{-Regeln}$$

Lemma 4.4.
Sei Γ gegeben. Dann ist $\sigma \sqsupseteq \tau$ mit $\sigma \doteq \sigma_1 \sqsupset \cdots \sqsupset \sigma_n \doteq \tau$, wobei keine in Γ frei vorkommende Typvariable bei einem Übergang von σ_i nach σ_{i+1} generalisiert wird, genau dann, wenn

$$\left.\begin{array}{c} \Gamma \vdash M : \sigma \\ \vdots \\ \Gamma \vdash M : \tau \end{array}\right\} \textit{nur } \forall\textit{-Regeln}$$

Beweis.

Definition von $\sqsupseteq$. □

Lemma 4.5.

- $\Gamma \vdash x : \sigma \implies (\exists \sigma' \sqsupseteq \sigma)\; x : \sigma' \in \Gamma$
- $\Gamma \vdash MN : \tau \implies (\exists \sigma)\; (\exists \tau' \sqsupseteq \tau)\; [\Gamma \vdash M : \sigma \to \tau'$ und $\Gamma \vdash N : \sigma]$
- $\Gamma \vdash \lambda x.M : \rho \implies (\exists \sigma)\; (\exists \tau)\; [\Gamma, x : \sigma \vdash M : \tau$ und $(\sigma \to \tau) \sqsupseteq \rho]$

Bemerkung.
Alle anderen Aussagen von Lemma 3.4 außer der Subjektreduktion gelten offensichtlich in $\lambda 2$. Die Subjektreduktion wird im folgenden gesondert bewiesen.

Lemma 4.6.
Wenn $(\sigma \to \tau) \sqsupseteq (\sigma' \to \tau')$, dann ist $(\sigma' \to \tau') \doteq s(\sigma \to \tau)$ für eine Substitution s, d.h. $\sigma' \to \tau'$ ist spezieller als $\sigma \to \tau$.

Beweis.
Sei $(\sigma \to \tau) \doteq \sigma_1 \sqsupset \cdots \sqsupset \sigma_n \doteq (\sigma' \to \tau')$.
Wir zeigen: $\sigma_i^0 \doteq s_i(\sigma \to \tau)$ für jedes s_i $(1 \leq i \leq n)$.
Damit folgt die Behauptung, da $(\sigma' \to \tau')^0 \doteq (\sigma' \to \tau')$.
Beweis durch Induktion über n:

- n=0: $\surd$
- n=m+1: Sei $\sigma_m^0 \doteq s_m(\sigma \to \tau)$.
  - Sei $\sigma_{m+1} \doteq \forall \alpha.\sigma_m$, dann ist $\sigma_{m+1}^0 = \sigma_m^0$, also $s_{m+1} =_{\text{def}} s_m$.
  - Sei $\sigma_m \doteq \forall \alpha.\rho$ und $\sigma_{m+1} \doteq \rho[\rho_1/\alpha]$. Dann setze $s_{m+1} =_{\text{def}} s_m[\rho_1/\alpha]$, wobei $s_m[\rho_1/\alpha]$ wie s_m ist, modifiziert um $\alpha \mapsto \rho_1$.

□

Theorem 4.7 (Subjektreduktion).
$M \rhd_\beta M' \implies (\Gamma \vdash M : \sigma \implies \Gamma \vdash M' : \sigma)$

Beweis.
Wir betrachten den Fall $M \doteq (\lambda x.P)Q \rhd_{1\beta} P[Q/x] \doteq M'$. Daraus ergibt sich der Rest.

$$\begin{aligned}
&\Gamma \vdash (\lambda x.P)Q : \sigma \\
&\overset{(4.5)}{\implies} \quad (\exists\tau)\,(\exists\sigma' \sqsupseteq \sigma)\,[\,\Gamma \vdash \lambda x.P : \tau \to \sigma' \text{ und } \Gamma \vdash Q : \tau] \\
&\overset{(4.5)}{\implies} \quad (\exists\tau)\,(\exists\sigma' \sqsupseteq \sigma)\,(\exists\tau')\,(\exists\sigma'')\,[\,\Gamma, x : \tau' \vdash P : \sigma'' \text{ und } \Gamma \vdash Q : \tau \text{ und } (\tau' \to \sigma'') \sqsupseteq (\tau \to \sigma')] \\
&\overset{(4.6)}{\implies} \quad (\exists\tau)\,(\exists\sigma' \sqsupseteq \sigma)\,[\,\underbrace{\Gamma, x : \tau \vdash P : \sigma'}_{\tau,\sigma' \text{ spezieller als } \tau',\sigma''} \quad \text{und } \Gamma \vdash Q : \tau] \\
&\implies \quad (\exists\sigma' \sqsupseteq \sigma)\ \Gamma \vdash P[Q/x] : \sigma' \\
&\overset{(4.4)}{\implies} \quad \Gamma \vdash P[Q/x] : \sigma
\end{aligned}$$

□

Definition 4.8.
Eine Bewertung *in $\mathcal{SAT}$ ist eine Abbildung ν : Typvariablen $\to \mathcal{SAT}$.*
Wir definieren eine Semantik von Typen relativ zu Bewertungen:

- $[\![\alpha]\!]_\nu =_{\text{def}} \nu(\alpha)$
- $[\![\sigma \to \tau]\!]_\nu =_{\text{def}} [\![\sigma]\!]_\nu \to [\![\tau]\!]_\nu$
- $[\![\forall\alpha.\sigma]\!]_\nu =_{\text{def}} \bigcap_{A \in \mathcal{SAT}} [\![\sigma]\!]_{\nu\left[A/\alpha\right]}$

Lemma 4.9.
Für jeden Typ σ und jede Bewertung ν gilt: $[\![\sigma]\!]_\nu$ ist saturiert.
(vgl. Lemma 3.7)

Beweis.
Analog zum Beweis von Lemma 3.7. Es wäre noch zu zeigen, daß $\mathcal{SAT}$ unter Durchschnitt abgeschlossen ist. Das ist aber trivial. □

Definition 4.10.
$\rho, \nu \models M : \sigma \iff_{\text{def}} [\![M]\!]_\rho \in [\![\sigma]\!]_\nu$
$\rho, \nu \models \Gamma \iff_{\text{def}} \rho, \nu \models x : \sigma$ *für alle* $(x : \sigma) \in \Gamma$
$\Gamma \models M : \sigma \iff_{\text{def}} (\forall\rho)\,(\forall\nu)\,[\rho, \nu \models \Gamma \implies \rho, \nu \models M : \sigma]$

Lemma 4.11.
$\Gamma \vdash M : \sigma \implies \Gamma \models M : \sigma$

Theorem 4.12.
$\Gamma \vdash M : \sigma \implies M$ *ist stark normalisierbar.*

Beweis.
Da $[\![\sigma]\!]_\nu$ für jedes ν saturiert, gilt $id, \nu \models \Gamma$ für jedes ν. Also $id, \nu \models M : \sigma$, dh. $M \in [\![\sigma]\!]_\nu$. □

Bemerkung.

- $\lambda 2$ hat mehr typbare Terme als $\lambda\!\rightarrow$. $\lambda x.xx$ ist ein Beispiel für einen stark normalisierbaren Term, der nicht in $\lambda\!\rightarrow$, aber in $\lambda 2$ typbar ist.

- Frage: Ist Typbarkeit in $\lambda 2$ = Normalisierbarkeit?
  Nein. Jedoch kann jeder Term in Normalform in $\lambda 2$ getypt werden, dh.
  $x_1 : \forall\alpha.\alpha, \ldots, x_n : \forall\alpha.\alpha \vdash M : \sigma$ für ein σ, wobei $x_1, \ldots, x_n$ freie Variablen in M sind.

- Es ist aber starke Normalisierbarkeit = Typbarkeit in $\lambda\cap$ (System D)