APPLIKATIVE THEORIEN UND EXPLIZITE MATHEMATIK

Sommersemester 1996

Gerhard Jäger

Leitung und Durchführung des Übungsbetriebs:

Thomas Strahm

Ausarbeitung des Skripts:

Marc Wirz

Literaturverzeichnis

[1] BARENDREGT, H. P. *The Lambda Calculus*, revised ed. North Holland, Amsterdam, 1984.

[2] BEESON, M. J. *Foundations of Constructive Mathematics: Metamathematical Studies.* Springer, Berlin, 1985.

[3] CANTINI, A., AND MINARI, P. Uniform inseparability in explicit mathemathics. Preprint, Firenze, 1996.

[4] FEFERMAN, S. Definedness. *Erkenntnis*. To appear.

[5] FEFERMAN, S. A language and axioms for explicit mathematics. In *Algebra and Logic*, J. Crossley, Ed., vol. 450 of *Lecture Notes in Mathematics.* Springer, Berlin, 1975, pp. 87–139.

[6] FEFERMAN, S. Constructive theories of functions and classes. In *Logic Colloquium '78*, M. Boffa, D. van Dalen, and K. McAloon, Eds. North Holland, Amsterdam, 1979, pp. 159–224.

[7] FEFERMAN, S., AND JÄGER, G. Systems of explicit mathematics with non-constructive μ-operator. Part I. *Annals of Pure and Applied Logic 65*, 3 (1993), 243–263.

[8] FEFERMAN, S., AND JÄGER, G. Systems of explicit mathematics with non-constructive μ-operator. Part II. *Annals of Pure and Applied Logic 79*, 1 (1996).

[9] Girard, J.-Y. Une extension de l'interprétation fonctionelle de Gödel à l'analyse et son application à l'élimination des coupures dans l'analyse et la théorie des types. In *Proceedings Second Scandinavian Logic Symposium*. North Holland, Amsterdam, 1971, pp. 63–92.

[10] Girard, J.-Y. The system F of variables types, fifteen years later. *Theoretical Computer Science 45* (1986), 159–192.

[11] Girard, J.-Y., Taylor, P., and Lafont, Y. *Proofs and Types*. Cambridge University Press, 1989.

[12] Glass, T. *Standardstrukturen für Systeme Expliziter Mathematik*. PhD thesis, Westfälische Wilhelms-Universität Münster, 1993.

[13] Jäger, G. Power types in explicit mathematics. *Journal of Symbolic Logic*. To appear.

[14] Jäger, G. Induction in the elementary theory of types and names. In *Computer Science Logic '87*, E. Börger et al., Ed., vol. 329 of *Lecture Notes in Computer Science*. Springer, Berlin, 1988, pp. 118–128.

[15] Jansen, D. Zu den Potenztypen in expliziter Mathematik. Master's thesis, Mathematisches Institut, Universität Bern, 1996.

[16] Reynolds, J. Towards a theory of type structure. In *Paris Colloquium on Programming*, vol. 19 of *Lecture Notes in Computer Science*. Springer, 1974, pp. 408–425.

[17] Strahm, T. Theories with self-application of strength PRA. Master's thesis, Institut für Informatik und angewandte Mathematik, Universität Bern, 1992.

[18] Strahm, T. *On the Proof Theory of Applicative Theories*. PhD thesis, Institut für Informatik und angewandte Mathematik, Universität Bern, 1996.

[19] Troelstra, A. Notes on intuitionistic second order arithmetic. In *Cambridge Summer School in Mathematical Logic*, vol. 337 of *Lecture Notes in Mathematics*. Springer, 1973, pp. 171–203.

[20] Troelstra, A., and van Dalen, D. *Constructivism in Mathematics*, vol. I. North-Holland, Amsterdam, New York, 1988.

[21] Troelstra, A., and van Dalen, D. *Constructivism in Mathematics*, vol. II. North Holland, Amsterdam, New York, 1988.

Inhaltsverzeichnis

Kapitel 1

Logik der partiellen Terme

Die Logik der partiellen Terme in der hier eingeführten Form geht auf Beeson und Feferman zurück. Sie ist verwandt zur Logik mit Existenzprädikat im Sinne von Scott. Für weitere Details und Hintergrundinformationen sei auf folgende Arbeiten und Lehrbücher verwiesen: [2, 4, 17, 20, 21].

1.1 Die Syntax der Logik der partiellen Terme

1.1.1 Definition Eine Sprache $\mathcal{L}$ der Logik der partiellen Terme umfasst folgende Grundzeichen:

1. Abzählbar unendlich viele Variablen $v0, v1, \ldots, vi, \ldots (i \in \mathbb{N})$.

2. Die logischen Symbole $\neg$ (nicht), $\vee$ (oder) und $\exists$ (es gibt).

3. Das 1-stellige Symbol $\downarrow$ für Definiertheit und das 2-stellige Symbol $=$ für Gleichheit.

4. Für jede natürliche Zahl n eine (eventuell leere) Menge von n-stelligen Funktionssymbolen.

5. Für jede natürliche Zahl n eine (eventuell leere) Menge von n-stelligen Relationssymbolen.

6. Hilfszeichen.

Die 0-stelligen Funktionssymbole von $\mathcal{L}$ nennen wir die *Konstanten* von $\mathcal{L}$. Ausserdem setzen wir im folgenden voraus, dass die Grundzeichen verschiedener Art voneinander verschieden sind.

1.1.2 Definition Die $\mathcal{L}$-*Terme* werden induktiv definiert durch:

1. Jede Variable und jede Konstante von $\mathcal{L}$ ist ein $\mathcal{L}$-Term.

2. Sind $a_1, \ldots, a_n$ $\mathcal{L}$-Terme und ist f ein n-stelliges Funktionssymbol von $\mathcal{L}$ mit $n \geq 1$, so ist $\mathsf{f}(a_1, \ldots, a_n)$ ein $\mathcal{L}$-Term.

Sind a, b sowie $a_1, \ldots, a_n$ $\mathcal{L}$-Terme und ist R ein n-stelliges Relationssymbol von $\mathcal{L}$, so bezeichnen wir die Ausdrücke $a\downarrow$, $(a = b)$ sowie $\mathsf{R}(a_1, \ldots, a_n)$ als *Atomformeln* von $\mathcal{L}$.

1.1.3 Definition Die $\mathcal{L}$-Formeln werden induktiv definiert durch:

1. Jede Atomformel von $\mathcal{L}$ ist eine $\mathcal{L}$-Formel.

2. Ist A eine $\mathcal{L}$-Formel, so ist $\neg A$ eine $\mathcal{L}$-Formel.

3. Sind A und B $\mathcal{L}$-Formeln, so ist $(A \vee B)$ eine $\mathcal{L}$-Formel.

4. Ist A eine $\mathcal{L}$-Formel und x eine Variable von $\mathcal{L}$, so ist $\exists x A$ eine $\mathcal{L}$-Formel.

Anstelle von $\mathcal{L}$-Termen und $\mathcal{L}$-Formeln sprechen wir häufig nur von Termen und Formeln, sofern der Bezug auf die Sprache $\mathcal{L}$ klar oder unwichtig ist. Bei Formeln lassen wir oft Klammern fort, wenn dadurch keine Unklarheiten entstehen. Wir verwenden die Vektor-Notation zur Bezeichnung endlicher Folgen von Termen und schreiben z.B. $\vec{a}$ oder $\vec{b}$ für $a_1, \ldots, a_m$ oder $b_1, \ldots, b_n$. Die Länge einer solchen Folge ergibt sich aus dem Kontext.

Mitteilungszeichen (auch mit Indizes)

$u, v, w, x, y, z, f, g, h$	für Variablen;
a, b, c, r, s, t	für Terme;
A, B, C, D, E, F	für Formeln.

1.1.4 Definition

1. $(A \wedge B) \ := \ \neg(\neg A \vee \neg B)$.

2. $(A \to B) \ := \ (\neg A \vee B)$.

3. $(A \leftrightarrow B) \ := \ (A \to B) \wedge (B \to A)$.

4. $\forall x A \ := \ \neg \exists x \neg A$.

5. $(a \simeq b) \ := \ (a{\downarrow} \vee b{\downarrow} \to a = b)$.

6. $(a \neq b) \ := \ a{\downarrow} \wedge b{\downarrow} \wedge \neg(a = b)$.

1.1.5 Definition Für jeden $\mathcal{L}$-Term a wird die Menge $FV(a)$ der Variablen von a induktiv definiert durch:

1. Ist a eine Variable von $\mathcal{L}$, so ist $FV(a) := \{a\}$.

2. Ist a eine Konstante von $\mathcal{L}$, so ist $FV(a) := \emptyset$.

3. Sind $a_1, \ldots, a_n$ $\mathcal{L}$-Terme, ist f ein n-stelliges Funktionssymbol mit $n \geq 1$ und ist a von der Form $\mathsf{f}(a_1, \ldots, a_n)$, so ist

$$FV(a) \ := \ FV(a_1) \cup \ldots \cup FV(a_n).$$

Ein Term a heisst *geschlossen*, falls $FV(a) = \emptyset$ ist.

1.1.6 Definition Für jede $\mathcal{L}$-Formel A wird die Menge $FV(A)$ der freien Variablen von A induktiv definiert durch:

1. Ist A von der Form $a\downarrow$, so ist $FV(A) := FV(a)$.
2. Ist A von der Form $(a = b)$, so ist $FV(A) := FV(a) \cup FV(b)$.
3. Ist R ein n-stelliges Relationssymbol von $\mathcal{L}$ und ist A von der Form $\mathsf{R}(a_1, \ldots, a_n)$, so ist $FV(A) := FV(a_1) \cup \ldots \cup FV(a_n)$.
4. Ist A von der Form $\neg B$, so ist $FV(A) := FV(B)$.
5. Ist A von der Form $(B \vee C)$, so ist $FV(A) := FV(B) \cup FV(C)$.
6. Ist A von der Form $\exists x B$, so ist $FV(A) := FV(B) \setminus \{x\}$.

Eine Formel A heisst *geschlossen* (oder *Satz*), falls $FV(A) = \emptyset$ ist.

1.1.7 Definition Für jeden $\mathcal{L}$-Term t sowie alle Variablen $\vec{x} = x_1, \ldots, x_n$ von $\mathcal{L}$ und alle $\mathcal{L}$-Terme $\vec{a} = a_1, \ldots, a_n$ wird der $\mathcal{L}$-Term $t[\vec{a}/\vec{x}]$ induktiv definiert durch:

1. Ist t eine Variable x_i mit $1 \leq i \leq n$, so ist $t[\vec{a}/\vec{x}] := a_i$.
2. Ist t eine von $x_1, \ldots, x_n$ verschiedene Variable oder eine Konstante von $\mathcal{L}$, so ist $t[\vec{a}/\vec{x}] := t$.
3. Ist f ein ein m-stelliges Funktionssymbol von $\mathcal{L}$ mit $m \geq 1$ und ist t von der Form $\mathsf{f}(s_1, \ldots, s_m)$, so ist

$$t[\vec{a}/\vec{x}] \;:=\; \mathsf{f}(s_1[\vec{a}/\vec{x}], \ldots, s_m[\vec{a}/\vec{x}]).$$

1.1.8 Definition Für jede $\mathcal{L}$-Formel A sowie alle Variablen $\vec{x} = x_1, \ldots, x_n$ von $\mathcal{L}$ und alle $\mathcal{L}$-Terme $\vec{a} = a_1, \ldots, a_n$ wird die $\mathcal{L}$-Formel $A[\vec{a}/\vec{x}]$ induktiv definiert durch:

1. Ist A von der Form $t\downarrow$, so ist $A[\vec{a}/\vec{x}] := t[\vec{a}/\vec{x}]\downarrow$.

2. Ist A von der Form $(s = t)$, so ist $A[\vec{a}/\vec{x}] := (s[\vec{a}/\vec{x}] = t[\vec{a}/\vec{x}])$.

3. Ist R ein m-stelliges Relationssymbol von $\mathcal{L}$ und ist A von der Form $\mathsf{R}(t_1, \ldots, t_m)$, so ist $A[\vec{a}/\vec{x}] := \mathsf{R}(t_1[\vec{a}/\vec{x}], \ldots, t_m[\vec{a}/\vec{x}])$.

4. Ist A von der Form $\neg B$, so ist $A[\vec{a}/\vec{x}] := \neg B[\vec{a}/\vec{x}]$.

5. Ist A von der Form $(B \vee C)$, so ist $A[\vec{a}/\vec{x}] := (B[\vec{a}/\vec{x}] \vee C[\vec{a}/\vec{x}])$.

6. Ist A von der Form $\exists x B$ und ist $x_{i_1}, \ldots, x_{i_k}$ die Folge der Variablen aus $\{x_1, \ldots, x_n\} \cap FV(A)$, dann ist

$$A[\vec{a}/\vec{x}] \;:=\; \exists y B[a_{i_1}, \ldots, a_{i_k}, y/x_{i_1}, \ldots, x_{i_k}, x].$$

Dabei sei y die Variable x, falls x in $a_{i_1}, \ldots, a_{i_k}$ nicht auftritt; andernfalls sei y die in der Aufzählung $v0, v1, v2, \ldots$ erste Variable, die nicht in $B, a_{i_1}, \ldots, a_{i_k}$ vorkommt.

1.1.9 Bemerkung $A[a_1, \ldots, a_n/x_1, \ldots, x_n]$ ist diejenige Formel, die dadurch aus A entsteht, dass die freien Vorkommnisse der Variablen $x_1, \ldots, x_n$ simultan durch $a_1, \ldots, a_n$ ersetzt werden. Dabei müssen zur Vermeidung von Variablenkollisionen manchmal Umbenennungen von (gebundenen) Variablen durchgeführt werden.

Wird die Formel A durch $B(\vec{x})$ bezeichnet, so schreiben wir im folgenden häufig (etwas unpräzise) $B(\vec{a})$ anstelle von $A[\vec{a}/\vec{x}]$.

Im folgenden stellen wir nun die Axiome und Schlussregeln der Logik der partiellen Terme zusammen und wählen dafür die Form eines Hilbert-Kalküls. Ausserdem beschränken wir uns auf den klassischen Fall; es ist offensichtlich, wie diese Axiome und Schlussregeln modifiziert werden müssen, um die intuitionistische Logik der partiellen Terme zu erhalten.

I. Aussagenlogische Axiome und Regeln. Wie üblich.

II. Quantorenaxiome und Quantorenregeln. Für alle $\mathcal{L}$-Formeln A und B, alle $\mathcal{L}$-Terme a und alle Variablen x von $\mathcal{L}$:

$$A[a/x] \wedge a\!\downarrow \;\rightarrow\; \exists x A \qquad \text{und} \qquad \frac{A \rightarrow B}{\exists x A \rightarrow B}$$

Dabei wird im Fall der Regel gefordert, dass die Variable x kein Element von $FV(B)$ ist.

III. Defininertheitsaxiome. Für alle n-stelligen Funktionssymbole f und Relationssymbole R von $\mathcal{L}$ sowie alle $\mathcal{L}$-Terme a, b und $t_1, \ldots, t_n$:

(D1) $a\!\downarrow$, falls a eine Variable oder Konstante von $\mathcal{L}$ ist.

(D2) $\mathsf{f}(t_1, \ldots, t_n)\!\downarrow \;\rightarrow\; t_1\!\downarrow \wedge \ldots \wedge t_n\!\downarrow$.

(D3) $(a = b) \;\rightarrow\; a\!\downarrow \wedge\, b\!\downarrow$.

(D4) $\mathsf{R}(t_1, \ldots, t_n) \;\rightarrow\; t_1\!\downarrow \wedge \ldots \wedge t_n\!\downarrow$.

IV. Gleichheitsaxiome. Für alle n-stelligen Funktionssymbole f und Relationssymbole R von $\mathcal{L}$ sowie alle $\mathcal{L}$-Terme $a, b, c, s_1, \ldots, s_n$ und $t_1, \ldots, t_n$:

(E1) $(a = a)$, falls a eine Variable oder eine Konstante von $\mathcal{L}$ ist.

(E2) $(a = b) \;\rightarrow\; (b = a)$.

(E3) $(a = b) \wedge (b = c) \;\rightarrow\; (a = c)$.

(E4) $\mathsf{R}(s_1, \ldots, s_n) \land (s_1 = t_1) \land \ldots \land (s_n = t_n) \;\rightarrow\; \mathsf{R}(t_1, \ldots, t_n)$.

(E5) $(s_1 = t_1) \land \ldots \land (s_n = t_n) \;\rightarrow\; \mathsf{f}(s_1, \ldots, s_n) \simeq \mathsf{f}(t_1, \ldots, t_n)$.

Mit der Bezeichnung $\mathsf{LPT} \vdash A$ drücken wir aus, dass die Formel A im Hilbertkalkül, der durch die Axiome und Schlussregeln I – IV gegeben ist, hergeleitet werden kann. Daher bedeutet $\mathsf{LPT} \vdash A$, dass die Formel A in der Logik der partiellen Terme bewiesen werden kann.

Die Axiome (D2), (D3) und (D4) werden manchmal als *Striktheitsaxiome* bezeichnet. In einem gewissen Sinne entspricht (D2) einer Call-by-Value-Berechnungsstrategie. Ausserdem ist leicht zu sehen, dass sich aus den aussagenlogischen Axiomen und Regeln sowie den Quantorenaxiomen und Quantorenregeln folgende beiden Eigenschaften der Logik der partiellen Terme zeigen lassen.

(1) $\mathsf{LPT} \vdash \forall x A \land a{\downarrow} \rightarrow A[a/x]$.

(2) Ist $x \notin FV(A)$, so gilt: $\mathsf{LPT} \vdash A \rightarrow B \implies \mathsf{LPT} \vdash A \rightarrow \forall x B$.

1.1.10 Bemerkung Für alle $\mathcal{L}$-Terme a und b sowie alle Variablen x von $\mathcal{L}$ gilt:

$$\mathsf{LPT} \vdash a{\downarrow} \text{ und } \mathsf{LPT} \vdash b{\downarrow} \implies \mathsf{LPT} \vdash a[b/x]{\downarrow}.$$

Beweis Aus $\mathsf{LPT} \vdash a{\downarrow}$ erhalten wir durch einige Umformungen in der Logik der partiellen Terme auch $\mathsf{LPT} \vdash \forall x(a{\downarrow})$. Ausserdem gilt ganz allgemein

$$\mathsf{LPT} \vdash \forall x(a{\downarrow}) \land b{\downarrow} \;\rightarrow\; a[b/x]{\downarrow}.$$

Daraus folgt die Behauptung durch zweifache Anwendung des Modus Ponens. $\square$

1.2 Die Semantik der Logik der partiellen Terme

In diesem Abschnitt sei $\mathcal{L}$ weiterhin eine Logik der partiellen Terme. Wir führen nun eine 2-wertige Semantik für $\mathcal{L}$ ein, die auf den Wahrheitswerten $\mathbf{t}$ (wahr) und $\mathbf{f}$ (falsch) basiert. Im Gegensatz zur üblichen Semantik der klassischen Prädikatenlogik arbeiten wir mit Strukturen, bei denen die Funktionssymbole als partielle Funktionen interpretiert werden können.

1.2.1 Definition Eine *partielle $\mathcal{L}$-Struktur* ist ein 5-Tupel

$$\mathcal{M} = (M, I_0, I_1, I_2, \ell),$$

das folgende Bedingungen erfüllt:

1. M ist eine nichtleere Menge und ℓ ein Objekt, das nicht zu M gehört.
2. I_0 ist eine Abbildung, die jedem n-stelligen Relationssymbol R von $\mathcal{L}$ eine n-stellige Funktion $I_0(\mathsf{R})$ von M nach $\{\mathbf{f}, \mathbf{t}\}$ zuordnet.
3. I_1 ist eine Abbildung, die jeder Konstanten c von $\mathcal{L}$ ein Element $I_1(\mathsf{c})$ aus M zuordnet.
4. I_2 ist eine Abbildung, die jedem n-stelligen Funktionssymbol f von $\mathcal{L}$ mit $n \geq 1$ eine n-stellige partielle Funktion $I_2(\mathsf{f})$ von M nach M zuordnet.

Ist $\mathcal{M} = (M, I_0, I_1, I_2, \ell)$ eine partielle $\mathcal{L}$-Struktur, so schreiben wir häufig $|\mathcal{M}|$ für das Universum M von $\mathcal{M}$, $\bowtie_{\mathcal{M}}$ für das ausgezeichnete Objekt ℓ, das nicht zu $|\mathcal{M}|$ gehört sowie $\mathsf{R}^{\mathcal{M}}, \mathsf{c}^{\mathcal{M}}$ und $\mathsf{f}^{\mathcal{M}}$ für die Interpretationen $I_0(\mathsf{R}), I_1(\mathsf{c})$ und $I_2(\mathsf{f})$ der Relationssymbole R, Konstanten c und (mindestens 1-stelligen) Funktionssymbole f.

1.2.2 Definition

1. Eine *Variablenbelegung* α in der nichtleeren Menge M ist eine Abbildung, die jeder Variablen vi mit $i \in \mathbb{N}$ einen Wert $\alpha(vi) \in M$ zuordnet.

2. Gegeben seien eine Variablenbelegung α in der nichtleeren Menge M, ein Element m aus M und eine Variable u. Dann bezeichnen wir mit $\alpha[u{=}m]$ die Variablenbelegung in M, die u auf m abbildet und sonst mit α übereinstimmt, d.h.

$$\alpha[u{=}m](v) \; := \; \begin{cases} m, & \text{falls } v = u, \\ \alpha(v), & \text{sonst.} \end{cases}$$

1.2.3 Definition Für jede partielle $\mathcal{L}$-Struktur $\mathcal{M}$ und jede Variablenbelegung α in $|\mathcal{M}|$ wird der Wert $\mathcal{M}_\alpha(t) \in |\mathcal{M}| \cup \{\bowtie_\mathcal{M}\}$ eines $\mathcal{L}$-Terms induktiv definiert durch:

1. Ist t eine Variable von $\mathcal{L}$, so ist $\mathcal{M}_\alpha(t) := \alpha(t)$.

2. Ist t eine Konstante von $\mathcal{L}$, so ist $\mathcal{M}_\alpha(t) := t^\mathcal{M}$.

3. Nun sei t von der Form $\mathsf{f}(t_1, \ldots, t_n)$ für ein n-stelliges Funktionssymbol von $\mathcal{L}$ mit $n \geq 1$. Sind $\mathcal{M}_\alpha(t_1), \ldots, \mathcal{M}_\alpha(t_n)$ Elemente von $|\mathcal{M}|$ und ist $\mathsf{f}^\mathcal{M}$ für $(\mathcal{M}_\alpha(t_1), \ldots, \mathcal{M}_\alpha(t_n))$ definiert, so ist

$$\mathcal{M}_\alpha(t) \; := \; \mathsf{f}^\mathcal{M}(\mathcal{M}_\alpha(t_1), \ldots, \mathcal{M}_\alpha(t_n));$$

   sind andererseits $\mathcal{M}_\alpha(t_1), \ldots, \mathcal{M}_\alpha(t_n)$ Elemente von $|\mathcal{M}|$ und ist $\mathsf{f}^\mathcal{M}$ für $(\mathcal{M}_\alpha(t_1), \ldots, \mathcal{M}_\alpha(t_n))$ nicht definiert oder ist $\mathcal{M}_\alpha(t_i) = \bowtie_\mathcal{M}$ für eine natürliche Zahl i mit $1 \leq i \leq n$, so ist $\mathcal{M}_\alpha(t) := \bowtie_\mathcal{M}$.

1.2.4 Definition Für jede partielle $\mathcal{L}$-Struktur $\mathcal{M}$ und jede Variablenbelegung α in $|\mathcal{M}|$ wird der Wert $\mathcal{M}_\alpha(A) \in \{\mathbf{f}, \mathbf{t}\}$ einer $\mathcal{L}$-Formel A induktiv definiert durch:

1. Ist A von der Form $a\downarrow$, so ist

$$\mathcal{M}_\alpha(A) \ := \ \begin{cases} \mathbf{t}, & \text{falls } \mathcal{M}_\alpha(a) \in |\mathcal{M}|, \\ \mathbf{f}, & \text{sonst.} \end{cases}$$

2. Ist A von der Form $(a = b)$, so ist

$$\mathcal{M}_\alpha(A) \ := \ \begin{cases} \mathbf{t}, & \text{falls } \mathcal{M}_\alpha(a) \in |\mathcal{M}|, \mathcal{M}_\alpha(b) \in |\mathcal{M}| \\ & \qquad \text{und } \mathcal{M}_\alpha(a) = \mathcal{M}_\alpha(b), \\ \mathbf{f}, & \text{sonst.} \end{cases}$$

3. Ist A von der Form $\mathsf{R}(a_1, \ldots, a_n)$ für ein n-stelliges Relationssymbol R von $\mathcal{L}$, so ist

$$\mathcal{M}_\alpha(A) \ := \ \begin{cases} \mathbf{t}, & \text{falls } \mathcal{M}_\alpha(a_1) \in |\mathcal{M}|, \ldots, \mathcal{M}_\alpha(a_n) \in |\mathcal{M}| \\ & \qquad \text{und } \mathsf{R}^{\mathcal{M}}(\mathcal{M}_\alpha(a_1), \ldots, \mathcal{M}_\alpha(a_n)) = \mathbf{t}, \\ \mathbf{f}, & \text{sonst.} \end{cases}$$

4. Ist A von der Form $\neg B$, so ist

$$\mathcal{M}_\alpha(A) \ := \ \begin{cases} \mathbf{t}, & \text{falls } \mathcal{M}_\alpha(B) = \mathbf{f}, \\ \mathbf{f}, & \text{sonst.} \end{cases}$$

5. Ist A von der Form $(B \vee C)$, so ist

$$\mathcal{M}_\alpha(A) \ := \ \begin{cases} \mathbf{t}, & \text{falls } \mathcal{M}_\alpha(B) = \mathbf{t} \text{ oder } \mathcal{M}_\alpha(C) = \mathbf{t}, \\ \mathbf{f}, & \text{sonst.} \end{cases}$$

6. Ist A von der Form $\exists x B$, so ist

$$\mathcal{M}_\alpha(A) \ := \ \begin{cases} \mathbf{t}, & \text{falls } \mathcal{M}_{\alpha[x=m]}(B) = \mathbf{t} \text{ für ein } m \in |\mathcal{M}|, \\ \mathbf{f}, & \text{sonst.} \end{cases}$$

Ist $\mathcal{M}$ eine partielle $\mathcal{L}$-Struktur und A eine $\mathcal{L}$-Formel, so nennen wir A *gültig in* $\mathcal{M}$ und schreiben dafür $\mathcal{M} \models A$, falls $\mathcal{M}_\alpha(A) = \mathsf{t}$ für alle Variablenbelegungen α in $|\mathcal{M}|$ gilt. $\mathcal{M}$ heisst *Modell* einer Menge Th von $\mathcal{L}$-Formeln, in Zeichen $\mathcal{M} \models \mathsf{Th}$, falls alle Formeln aus Th in $\mathcal{M}$ gültig sind. Ist schliesslich A in allen $\mathcal{L}$-Strukturen gültig, so bezeichnen wir A als *gültig in der Logik der partiellen Terme*; als Notation dafür verwenden wir $\mathsf{LPT} \models A$.

Das folgende Theorem besagt, dass der Hilbert-Kalkül des vorhergehenden Abschnittes eine korrekte und vollständige Axiomatisierung der Logik der partiellen Terme im Sinne der eben eingeführten Semantik ist. Auf einen Beweis dieses Satzes wird verzichtet.

1.2.5 Theorem *Für jede $\mathcal{L}$-Formel A gilt:*

$$\mathsf{LPT} \vdash A \quad \Longleftrightarrow \quad \mathsf{LPT} \models A.$$

Kapitel 2

Applikative Theorien

Applikative Theorien bilden den erststufigen Teil von Systemen expliziter Mathematik, die im nächsten Kapitel eingeführt werden. Bei den hier dargestellten Resultaten handelt es sich im wesentlichen um Standardergebnisse, und wie früher wird darauf verzichtet, jeweils die genauen Literaturangaben anzuführen. Zentrale Texte sind etwa [2, 5, 7, 8, 18, 20, 21].

2.1 Die Theorie BON

Als erste applikative Theorie betrachten wir die Basic Theory of Operations and Numbers BON. Sie umfasst eine Axiomatisierung von partiellen kombinatorischen Algebren, Paarbildung und Projektionen sowie einige Grundeigenschaften natürlicher Zahlen.

BON ist formalisiert in der Sprache L_p der Logik der partiellen Terme, die die Konstanten k, s (Kombinatoren), $\mathsf{p}, \mathsf{p}_0, \mathsf{p}_1$ (pairing, unpairing), $\mathsf{0}$ (Zero), $\mathsf{s_N}$ (Numerical Successor), $\mathsf{p_N}$ (Numerical Predecessor), $\mathsf{d_N}$ (Definition by Numerical Cases) und $\mathsf{r_N}$ (Recursion) umfasst. Dazu kommen ein 2-stelliges

Funktionssymbol $\cdot$, das infix geschrieben wird, für die Applikation von Objekten sowie ein 1-stelliges Relationssymbol N für die Menge der natürlichen Zahlen.

Folglich werden die L_p-Terme generiert durch:

1. Jede Variable und jede Konstante von L_p ist ein L_p-Term.

2. Sind a und b L_p-Terme, so ist auch $(a \cdot b)$ ein L_p-Term.

Wir schreiben $(a \cdot b)$ oft nur als (ab) oder ab und vereinbaren Klammerung nach links, so dass $ab_1 \ldots b_n$ für $(\ldots(ab_1)\ldots b_n)$ steht. Ausserdem sei a' der Term $\mathsf{s_N}a$, 1 der Term $\mathsf{0}'$ und (a, b) der Term $\mathsf{p}ab$. Die n-Tupel $(a_1, \ldots, a_n)$ werden induktiv definiert durch $(a_1) := a_1$ und $(a_1, \ldots, a_{n+1}) := ((a_1, \ldots, a_n), a_{n+1})$. Im Zusammenhang mit dem Relationssymbol N verwenden wir folgende Abkürzungen:

$$\begin{aligned}
a \in \mathsf{N} &:= \mathsf{N}(a), \\
(\exists x \in \mathsf{N})A &:= \exists x(x \in \mathsf{N} \land A), \\
(\forall x \in \mathsf{N})A &:= \forall x(x \in \mathsf{N} \to A), \\
(a : \mathsf{N} \to \mathsf{N}) &:= (\forall x \in \mathsf{N})(ax \in \mathsf{N}), \\
(a : \mathsf{N}^{n+1} \to \mathsf{N}) &:= (\forall x \in \mathsf{N})(ax : \mathsf{N}^n \to \mathsf{N}).
\end{aligned}$$

Die Theorie BON umfasst die Axiome und Schlussregeln der Logik der partiellen Terme LPT formuliert für die Sprache L_p. Die *nichtlogischen Axiome* von BON lassen sich in die folgenden fünf Gruppen unterteilen.

I. Partial Combinatory Algebra

(1) $\mathsf{k}xy = x$.

(2) $\mathsf{s}xy\downarrow \ \land \ \mathsf{s}xyz \simeq (xz)(yz)$.

II. Pairing and Projections

(3) $\mathsf{p}_0 x\downarrow \;\wedge\; \mathsf{p}_1 x\downarrow$.

(4) $\mathsf{p}_0(x,y) = x \;\wedge\; \mathsf{p}_1(x,y) = y$.

III. Natural Numbers

(5) $\mathsf{0} \in \mathsf{N} \;\wedge\; (\forall x \in \mathsf{N})(x' \in \mathsf{N})$.

(6) $(\forall x \in \mathsf{N})(x' \neq \mathsf{0} \;\wedge\; \mathsf{p_N}(x') = x)$.

(7) $(\forall x \in \mathsf{N})(x \neq \mathsf{0} \;\rightarrow\; \mathsf{p_N} x \in \mathsf{N} \wedge (\mathsf{p_N} x)' = x)$.

IV. Definition by Numerical Cases

(8) $u \in \mathsf{N} \;\wedge\; v \in \mathsf{N} \;\wedge\; u = v \;\;\rightarrow\;\; \mathsf{d_N} xyuv = x$.

(9) $u \in \mathsf{N} \;\wedge\; v \in \mathsf{N} \;\wedge\; u \neq v \;\;\rightarrow\;\; \mathsf{d_N} xyuv = y$.

V. Primitive Recursion on N

(10) $(f : \mathsf{N} \to \mathsf{N}) \;\wedge\; (g : \mathsf{N}^3 \to \mathsf{N}) \;\;\rightarrow\;\; (\mathsf{r_N} fg : \mathsf{N}^2 \to \mathsf{N})$.

(11) $(f : \mathsf{N} \to \mathsf{N}) \;\wedge\; (g : \mathsf{N}^3 \to \mathsf{N}) \;\wedge\; x \in \mathsf{N} \;\wedge\; y \in \mathsf{N} \;\wedge\; h = \mathsf{r_N} fg$
$\quad\rightarrow\;\; hx\mathsf{0} = fx \;\wedge\; hx(y') = gxy(hxy)$.

Durch $\mathsf{BON} \vdash A$ drücken wir von nun an aus, dass sich die L_p-Formel A mit Hilfe der Logik der partiellen Terme LPT aus den nichtlogischen Axiomen von BON ableiten lässt. Ist K eine Klasse von L_p-Formeln, so wird $\mathsf{BON} + \mathsf{K} \vdash A$ analog definiert.

Manchmal betrachten wir auch applikative Theorien, bei denen die Applikationsoperation total ist. Das entsprechende *Totalitätsaxiom* lautet:

(Tot) $\qquad \forall x \forall y (xy\downarrow)$

2.1.1 Bemerkung Für alle L_p-Terme t gilt:

$$\mathsf{BON} + (\mathsf{Tot}) \vdash t\downarrow.$$

2.2 Kombinatorische Eigenschaften von BON

In diesem Abschnitt stellen wir einige kombinatorische Eigenschaften von BON zusammen. Dabei sind vor allem der Satz über λ-Abstraktion und das Rekursionstheorem hervorzuheben, die beide bereits aus den Axiomen (1) und (2) folgen.

2.2.1 Definition Für jeden L_p-Term t und jede Variable x von L_p definieren wir einen L_p-Term $(\lambda x.t)$ induktiv durch:

1. Ist t die Variable x, so ist $(\lambda x.t) := \mathsf{skk}$.

2. Ist t eine von x verschiedene Variable oder eine Konstante von L_p, so ist $(\lambda x.t) := \mathsf{k}t$.

3. Ist t von der Form (ab), so ist $(\lambda x.t) := \mathsf{s}(\lambda x.a)(\lambda x.b)$.

Das folgende Theorem, das durch Induktion nach dem Aufbau von t bewiesen wird, macht deutlich, dass in BON durch die Definition von $(\lambda x.t)$ die λ-Abstraktion des üblichen λ-Kalküls weitgehend simuliert werden kann. Bei dieser Definition tragen wir der Partialität Rechnung und stellen sicher, dass $(\lambda x.t)$ immer einen Wert besitzt.

2.2.2 Theorem ($\boldsymbol{\lambda}$-Abstraktion) *Für alle L_p-Terme a und t und alle Variablen x von L_p gilt:*

1. $FV(\lambda x.t) = FV(t) \setminus \{x\}$.

2. $\mathsf{BON} \vdash (\lambda x.t){\downarrow}$.

3. $\mathsf{BON} \vdash (\lambda x.t)x \simeq t$.

4. $\mathsf{BON} \vdash a{\downarrow} \rightarrow (\lambda x.t)a \simeq t[a/x]$.

2.2.3 Bemerkung Falls x und y verschiedene Variablen sind und x kein Element von $FV(a)$ ist, gilt im üblichen λ-Kalkül, dass die Terme $(\lambda x.t)[a/y]$ und $(\lambda x.t[a/y])$ gleich sind. Eine entsprechende Gleichung kann jedoch in BON nicht bewiesen werden. Betrachte den Fall $t := y$ und $a := zz$. Dann beweist BON

$$(\lambda x.t)[a/y] = \mathsf{k}(zz) \quad \text{und} \quad (\lambda x.t[a/y]) = \mathsf{s}(\mathsf{k}z)(\mathsf{k}z).$$

Es gibt aber auch ein Modell von BON, in dem $\mathsf{k}(zz)$ und $\mathsf{s}(\mathsf{k}z)(\mathsf{k}z)$ nicht gleich sind.

Es folgt nun eine Abschwächung des oben beschriebenen Substitutionsprinzips, die in BON bewiesen werden kann und in der Regel für alle Anwendungen ausreicht.

2.2.4 Lemma *Für alle L_p-Terme a und t sowie von einander verschiedenen Variablen x und y von L_p gilt:*

$$\mathsf{BON} \vdash (\lambda x.t)[a/y]x \simeq t[a/y].$$

Beweis: Wir zeigen die Behauptung durch Induktion nach dem Aufbau von t und arbeiten informell in BON. Wir unterscheiden mehrere Fälle.

1. t ist die Variable x. Dann gilt in BON:

$$(\lambda x.t)[a/y]x \simeq (\mathsf{skk})[a/y]x \simeq (\mathsf{k}x)(\mathsf{k}x) \simeq x \simeq t[a/y].$$

2. t ist die Variable y. Dann gilt in BON:

$$(\lambda x.t)[a/y]x \simeq (\mathsf{k}t)[a/y]x \simeq (\mathsf{k}a)x \simeq a \simeq t[a/y].$$

3. t ist eine von x und y verschiedene Variable oder eine Konstante. Dann gilt in BON:

$$(\lambda x.t)[a/y]x \simeq (\mathsf{k}t)[a/y]x \simeq (\mathsf{k}t)x \simeq t \simeq t[a/y].$$

4. t ist von der Form $(t_1 t_2)$. Dann erhalten wir mit Hilfe der Induktionsvoraussetzung in BON:

$$\begin{aligned} (\lambda x.t)[a/y]x &\simeq (\mathsf{s}(\lambda x.t_1)(\lambda x.t_2))[a/y]x \\ &\simeq ((\lambda x.t_1)[a/y]x)((\lambda x.t_2)[a/y]x) \\ &\simeq t_1[a/y]\,t_2[a/y] \simeq t[a/y]. \end{aligned}$$ $\square$

Aufgrund von Theorem 2.2.2 folgt aus diesem Lemma, dass für alle L_p-Terme a und t und von einander verschiedenen Variablen x und y in BON gezeigt werden kann:

$$(\lambda x.t)[a/y]x \;\simeq\; (\lambda x.t[a/y])x.$$

Die Bildung von λ-Termen kann kanonisch auf mehrere Argumente erweitert werden. Ausserdem lässt sich das vorhergehende Lemma entsprechend verallgemeinern.

2.2.5 Definition Für alle L_p-Terme und alle Variablen $x_1, \ldots, x_n$ von L_p setzen wir $(\lambda x_1 \ldots x_n.t) := (\lambda x_1.(\ldots(\lambda x_n.t)\ldots))$.

2.2.6 Lemma *Für alle L_p-Terme a und t sowie alle Variablen $x_1, \ldots, x_n$ und y von L_p, so dass alle $x_1, \ldots, x_n$ von y verschieden sind, gilt:*

$$\mathsf{BON} \vdash (\lambda x_1 \ldots x_n.t)[a/y]x_1 \ldots x_n \;\simeq\; t[a/y].$$

2.2.7 Theorem (Rekursionstheorem) *Es gibt einen geschlossenen L_p-Term $\underline{\mathrm{rec}}$, so dass gilt:*

$$\mathsf{BON} \vdash \underline{\mathrm{rec}}\, f\!\downarrow \;\wedge\; \underline{\mathrm{rec}}\, fx \simeq f(\underline{\mathrm{rec}}\, f)x.$$

Beweis: Wir definieren

$$t := (\lambda yx.f(yy)x) \quad \text{und} \quad \underline{\mathrm{rec}} := (\lambda f.tt)$$

und arbeiten nun informell in BON. Da t durch λ-Abstraktion eingeführt wird, hat t einen Wert. Ausserdem ergibt Theorem 2.2.2, dass

$$\underline{\mathrm{rec}}\, f \simeq tt \simeq (\lambda yx.f(yy)x)t \simeq (\lambda x.f(yy)x)[t/y].$$

Da $(\lambda x.f(yy)x)$ als λ-Term einen Wert besitzt, folgt aufgrund von Bemerkung 1.1.10 dass $\underline{\mathrm{rec}}\, f\!\downarrow$ gilt. Schliesslich erhalten wir mit Hilfe von Lemma 2.2.4 auch

$$\underline{\mathrm{rec}}\, fx \simeq (\lambda x.f(yy)x)[t/y]x \simeq f(tt)x \simeq f(\underline{\mathrm{rec}}\, f)x. \qquad \square$$

2.2.8 Lemma *Es gibt einen geschlossenen L_p-Term $\underline{\mathrm{nn}}$, so dass gilt:*

$$\mathsf{BON} \vdash \underline{\mathrm{nn}} \notin \mathsf{N}.$$

Beweis Wir definieren

$$t := (\lambda xy.\mathsf{d_N}\mathsf{1}\mathsf{0}(xy)\mathsf{0}) \quad \text{und} \quad \underline{\mathrm{nn}} := \underline{\mathrm{rec}}\, t\mathsf{0}$$

und arbeiten nun informell in BON. Mit Hilfe von Theorem 2.2.2, Theorem 2.2.7 und Lemma 2.2.4 erhalten wir

$$\begin{aligned}\underline{\mathrm{nn}} &\simeq \underline{\mathrm{rec}}\, t\mathsf{0} \simeq t(\underline{\mathrm{rec}}\, t)\mathsf{0} \simeq (\lambda y.\mathsf{d_N}\mathsf{10}(xy)\mathsf{0})[\underline{\mathrm{rec}}\, t/x]\mathsf{0}\\ &\simeq \mathsf{d_N}\mathsf{10}(\underline{\mathrm{rec}}\, t\mathsf{0})\mathsf{0} \simeq \mathsf{d_N}\mathsf{10}\underline{\mathrm{nn}}\mathsf{0}.\end{aligned}$$

Nun nehmen wir $\underline{\mathrm{nn}} \in \mathsf{N}$ an. Dann gelten die beiden Implikationen

$$\underline{\mathrm{nn}} = \mathsf{0} \to \underline{\mathrm{nn}} \simeq \mathsf{d_N}\mathsf{10}\underline{\mathrm{nn}}\mathsf{0} \simeq \mathsf{1} \quad \text{und} \quad \underline{\mathrm{nn}} \neq \mathsf{0} \to \underline{\mathrm{nn}} \simeq \mathsf{d_N}\mathsf{10}\underline{\mathrm{nn}}\mathsf{0} \simeq \mathsf{0},$$

so dass $\underline{\mathrm{nn}} = \mathsf{0}$ äquivalent zu $\underline{\mathrm{nn}} \neq \mathsf{0}$ ist. Dies ist ein Widerspruch und daher muss $\underline{\mathrm{nn}} \notin \mathsf{N}$ gelten. □

2.2.9 Bemerkung Wir können in BON weder $\underline{\mathrm{nn}}\downarrow \wedge \underline{\mathrm{nn}} \notin \mathsf{N}$ noch $\neg\underline{\mathrm{nn}}\downarrow$ beweisen. Es gibt Modelle von BON, in denen $\underline{\mathrm{nn}}$ keinen Wert besitzt, und Modelle, in denen $\underline{\mathrm{nn}}$ einen Wert besitzt, aber nicht zur Interpretation der natürlichen Zahlen gehört.

Wie bereits früher bemerkt, werden die Terme in der Logik der partiellen Terme — und damit auch in BON — aufgrund der Striktheitsaxiome im Sinne von Call-by-Value evaluiert. Die folgende Behandlung des `if-then-else`-Konstrukts deutet an, wie auch Call-by-Name in diesem Rahmen simuliert werden kann.

Es seien u und v beliebige natürliche Zahlen sowie a und b beliebige Terme. Dann kann der Ausdruck

$$\texttt{if } u{=}v \texttt{ then } a \texttt{ else } b$$

auf zwei verschiedene Arten interpretiert werden:

1. Strikte Interpretation. Dann besitzt er nur einen Wert, falls sowohl a als auch b einen Wert besitzen. Der Term $\mathsf{d_N}abuv$ ist eine geeignete Repräsentation in BON.

2. Liberale Interpretation. Falls $u = v$ ist, so evaluiert er zu a, unabhängig davon, ob b definiert ist; ist andererseits $u \neq v$, so evaluiert er zu b, unabhängig davon, ob a definiert ist. Nun wählen wir eine Variable x, die weder in a noch in b auftritt, und erhalten durch $\mathsf{d_N}(\lambda x.a)(\lambda x.b)uvx$ eine geeignete Repräsentation in BON.

2.3 Definition by Cases auf dem Universum

Manchmal ist es vorteilhaft, nicht nur Definition by Numerical Cases, sondern Definition by Cases auf dem ganzen Universum zur Verfügung zu haben. Um dies zu erreichen, kann man etwa zu L_p eine neue Konstante $\mathsf{d_V}$ hinzunehmen und BON um folgendes Axiom

$$(\mathsf{d_V}) \qquad (u = v \rightarrow \mathsf{d_V}xyuv = x) \ \wedge \ (u \neq v \rightarrow \mathsf{d_V}xyuv = y)$$

erweitern. Später werden wir sehen, dass $\mathsf{BON} + (\mathsf{d_V})$ konsistent ist. Allerdings ist Definition by Cases auf dem Universum nicht unproblematisch, da Inkonsistenzen im Zusammenhang mit der Extensionalität von Operationen und dem Totalitätsaxiom auftreten.

Unter der Extensionalität von Operationen verstehen wir die Aussage

$$(\mathsf{Ext}) \qquad \forall f \forall g[\forall x(fx \simeq gx) \ \rightarrow \ f = g].$$

Wir werden später auch Modelle von BON betrachten, in denen (Ext) gültig ist; daher ist $\mathsf{BON} + (\mathsf{Ext})$ konsistent. Aus dem folgenden Theorem folgt aber auch, dass die Modelle von $\mathsf{BON} + (\mathsf{Ext})$ Definition by Cases auf dem Universum verletzen.

2.3.1 Theorem *Die Theorie* $\mathsf{BON} + (\mathsf{d_V}) + (\mathsf{Ext})$ *ist inkonsistent, d.h.*

$$\mathsf{BON} + (\mathsf{d_V}) \vdash \neg(\mathsf{Ext}).$$

Beweis Wir arbeiten informell in $\mathsf{BON} + (\mathsf{d_V})$ und betrachten die folgenden beiden Terme

$$a := (\lambda f.\mathsf{k}(\mathsf{d_V}\mathsf{1}\mathsf{0}f(\lambda y.\mathsf{0}))) \quad \text{und} \quad b := \underline{\mathrm{rec}}\, a.$$

Aufgrund des Theorems über λ-Abstraktion und des Rekursionstheorems sind a und b definiert, und es gilt

$$bx \simeq abx \simeq \mathsf{k}(\mathsf{d_V}\mathsf{1}\mathsf{0}b(\lambda y.\mathsf{0}))x \simeq \mathsf{d_V}\mathsf{1}\mathsf{0}b(\lambda y.\mathsf{0}).$$

In anderer Form geschrieben bedeutet dies

$$bx \;\simeq\; \begin{cases} \mathsf{1}, & \text{falls } b = (\lambda y.\mathsf{0}), \\ \mathsf{0}, & \text{falls } b \neq (\lambda y.\mathsf{0}). \end{cases}$$

Aus der ersten Klausel folgt $b \neq (\lambda y.\mathsf{0})$. Daraus erhalten wir $\forall x(bx = \mathsf{0})$ und somit auch $\forall x(bx = (\lambda y.\mathsf{0})x)$. Damit ist $\neg(\mathsf{Ext})$ bewiesen. □

Definition by Cases auf dem Universum ist auch nicht verträglich mit dem Totalitätsaxiom. Für den Beweis dieses Theorems sei auf den Übungsteil verwiesen.

2.3.2 Theorem $\mathsf{BON} + (\mathsf{d_V}) + (\mathsf{Tot})$ *ist inkonsistent, d.h.*

$$\mathsf{BON} + (\mathsf{d_V}) \vdash \neg(\mathsf{Tot}).$$

2.4 Induktionsprinzipien

Die Theorie BON umfasst zwar einige elementare Aussagen über natürliche Zahlen, ist jedoch mathematisch sehr schwach, da vollständige Induktion über die natürlichen Zahlen nicht zur Verfügung steht. Dieses Defizit soll nun beseitigt werden.

Teilmengen der natürlichen Zahlen wollen wir in unserem Kontext mit ihren charakteristischen Funktionen identifizieren. Entsprechend verstehen wir unter einer Teilmenge von N eine Operation a, so dass für jede natürliche Zahl b entweder $ab = \mathsf{0}$ oder $ab = \mathsf{1}$ gilt. Im ersten Fall nennen wir b ein Element von a. In Übereinstimmung damit führen wir folgende Definitionen ein:

$$\begin{aligned} Set_{\mathsf{N}}(a) &:= (\forall x \in \mathsf{N})(ax = \mathsf{0} \vee ax = \mathsf{1}), \\ a \in Set_{\mathsf{N}} &:= Set_{\mathsf{N}}(a), \\ b \,\varepsilon\, a &:= ab = \mathsf{0}. \end{aligned}$$

Nun betrachten wir zwei Formen von vollständiger Induktion, nämlich Mengeninduktion und Formelinduktion. Auf interessante Zwischenformen können wir nicht eingehen.

Mengeninduktion auf N (S-I_{N}). Für alle L_p-Terme a:

$$a \in Set_{\mathsf{N}} \;\wedge\; \mathsf{0} \,\varepsilon\, a \;\wedge\; (\forall x \in \mathsf{N})(x \,\varepsilon\, a \to x' \,\varepsilon\, a) \quad\to\quad (\forall x \in \mathsf{N})(x \,\varepsilon\, a).$$

Formelinduktion auf N (F-I_{N}). Für alle L_p-Formeln $A(x)$:

$$A(\mathsf{0}) \;\wedge\; (\forall x \in \mathsf{N})(A(x) \to A(x')) \quad\to\quad (\forall x \in \mathsf{N})A(x).$$

Mengeninduktion auf N ist selbstverständlich ein Spezialfall von Formelinduktion auf N.

Die Axiome (10) und (11) von BON sind überflüssig, falls (F-I_{N}) beziehungsweise eine hinreichend starke Form von vollständiger Induktion zur Verfügung steht. Mengeninduktion (S-I_{N}) ist allerdings für den Beweis von Theorem 2.4.1 nicht ausreichend.

Es sei BON^- die Untertheorie von BON, die auf die Axiome (10) und (11) verzichtet. Dann lässt sich folgendes Theorem zeigen.

2.4.1 Theorem *Es gibt einen geschlossenen L_p-Term $\underline{\text{prim}}$, in dem die Konstante $\mathsf{r_N}$ nicht auftritt, so dass man in $\mathsf{BON}^- + (\mathsf{F\text{-}I_N})$ beweisen kann:*

1. $(f : \mathsf{N} \to \mathsf{N}) \;\wedge\; (g : \mathsf{N}^3 \to \mathsf{N}) \quad \to \quad (\underline{\text{prim}}\, fg : \mathsf{N}^2 \to \mathsf{N}).$

2. $(f : \mathsf{N} \to \mathsf{N}) \;\wedge\; (g : \mathsf{N}^3 \to \mathsf{N}) \;\wedge\; x \in \mathsf{N} \;\wedge\; y \in \mathsf{N} \;\wedge\; h = \underline{\text{prim}}\, fg$
$\to \quad hx\mathsf{0} \simeq fx \;\wedge\; hx(y') \simeq gxy(hxy).$

Beweis Wir definieren

$$t := (\lambda hxy.\mathsf{d_N} f(\lambda z.gz(\mathsf{p_N}y)(hz(\mathsf{p_N}y)))\mathsf{0}yx \quad \text{und} \quad \underline{\text{prim}} := (\lambda fg.\underline{\text{rec}}\, t)$$

und arbeiten informell in $\mathsf{BON}^- + ((\mathsf{F\text{-}I_N}))$. Ist dann $h \simeq \underline{\text{prim}}\, fg$, so folgt

$$hxy \simeq \underline{\text{rec}}\, txy \simeq t(\underline{\text{rec}}\, t)xy \simeq thxy \simeq \mathsf{d_N} f(\lambda z.gz(\mathsf{p_N}y)(hz(\mathsf{p_N}y)))\mathsf{0}yx.$$

Daraus erhalten wir für alle $y \in \mathsf{N}$, dass

$$hx\mathsf{0} \simeq fx \quad \text{und} \quad hx(y') \simeq (\lambda z.gzy(hzy))x \simeq gxy(hxy).$$

Nun setzen wir zusätzlich $(f : \mathsf{N} \to \mathsf{N})$, $(g : \mathsf{N}^3 \to \mathsf{N})$ und $x \in \mathsf{N}$ voraus. Dann folgt mit $(\mathsf{F\text{-}I_N})$ sofort, dass

$$(\forall y \in \mathsf{N})(hxy \in \mathsf{N}).$$

Dies ergibt $(h : \mathsf{N}^2 \to \mathsf{N})$, und unsere Behauptungen sind bewiesen. Beachte dabei aber, dass $(\mathsf{S\text{-}I_N})$ nicht ausreicht, um $hxy \in \mathsf{N}$ für alle $x \in \mathsf{N}$ und $y \in \mathsf{N}$ zu zeigen. $\square$

Aus diesem Theorem ergibt sich sofort, dass jede partielle L_p-Struktur $\mathcal{M}$, die ein Modell von BON^-+$(\mathsf{F\text{-}I_N})$ ist, auch als ein Modell von BON aufgefasst werden kann. Man muss lediglich die Konstante $\mathsf{r_N}$ durch den Wert von $\underline{\text{prim}}$ in $\mathcal{M}$ interpretieren.

Ohne im Moment näher darauf einzugehen, soll hier nur erwähnt werden, dass $\mathsf{BON} + (\mathsf{S\text{-}I_N})$ von derselben beweistheoretischen Stärke ist wie die *primitiv-rekursive Arithmetik* PRA. Dagegen ist die Theorie $\mathsf{BON} + (\mathsf{F\text{-}I_N})$ beweistheoretisch äquivalent zur *Peano-Arithmetik* PA. Diese Beziehungen machen deutlich, dass es sich bei $\mathsf{BON} + (\mathsf{S\text{-}I_N})$ und $\mathsf{BON} + (\mathsf{F\text{-}I_N})$ um recht zentrale applikative Theorien handelt.

2.5 Das rekursionstheoretische Modell $\mathcal{PRF}$

Im folgenden nehmen wir an, dass $\{e\}$ für $e = 0, 1, 2, \ldots$ eine Indizierung der partiell-rekursiven Funktionen auf den natürlichen Zahlen ist, und schreiben $\bullet^{\mathcal{PRF}}$ für die partielle Funktion von $\mathbb{N} \times \mathbb{N}$ nach $\mathbb{N}$, die für alle natürlichen Zahlen e und n durch

$$e \bullet^{\mathcal{PRF}} n \;:\simeq\; \{e\}(n)$$

definiert ist. Ausserdem sei $\langle \cdot, \cdot \rangle$ eine der üblichen primitiv-rekursiven Funktionen zur Bildung von Paaren $\langle m, n \rangle$ natürlicher Zahlen. Dann existieren natürliche Zahlen $\hat{\mathsf{k}}, \hat{\mathsf{s}}, \hat{\mathsf{p}}, \hat{\mathsf{p}}_0, \hat{\mathsf{p}}_1, \hat{\mathsf{d}}_\mathsf{N}, \hat{\mathsf{s}}_\mathsf{N}$ und $\hat{\mathsf{p}}_\mathsf{N}$, so dass für alle natürlichen Zahlen e, m, n, p und q folgende Bedingungen erfüllt sind:

(1) $\{\{\hat{\mathsf{k}}\}(m)\}(n) = m$.

(2) $\{\{\hat{\mathsf{s}}\}(m)\}(n) \in \mathbb{N}$ und $\{\{\{\hat{\mathsf{s}}\}(m)\}(n)\}(p) \simeq \{\{m\}(p)\}(\{n\}(p))$.

(3) $\{\{\hat{\mathsf{p}}\}(m)\}(n) = \langle m, n \rangle$ und $\{\hat{\mathsf{p}}_0\}(m) \in \mathbb{N}$ und $\{\hat{\mathsf{p}}_1\}(m) \in \mathbb{N}$ und $\{\hat{\mathsf{p}}_0\}(\langle m, n \rangle) = m$ und $\{\hat{\mathsf{p}}_1\}(\langle m, n \rangle) = n$.

(4) $\{\{\{\{\hat{\mathsf{d}}_\mathsf{N}\}(m)\}(n)\}(p)\}(p) = m$ und $\{\{\{\{\hat{\mathsf{d}}_\mathsf{N}\}(m)\}(n)\}(p)\}(q) = n$, falls $p \neq q$.

(5) $\{\hat{\mathsf{s}}_\mathsf{N}\}(m) = m+1$ und $\{\hat{\mathsf{p}}_\mathsf{N}\}(m) = m \dot{-} 1$.

Mit $\mathcal{PRF}$ bezeichnen wir die partielle L_p-Struktur, deren Universum $|\mathcal{PRF}|$ durch die Menge $\mathbb{N}$ der natürlichen Zahlen gegeben ist und bei der das Relationssymbol N durch die Menge $\mathbb{N}$ sowie die 2-stellige Applikation $\cdot$ durch $\bullet^{\mathcal{PRF}}$ interpretiert werden; die L_p-Konstanten $\mathsf{k}, \mathsf{s}, \mathsf{p}, \ldots$ werden durch die oben eingeführten natürlichen Zahlen $\hat{\mathsf{k}}, \hat{\mathsf{s}}, \hat{\mathsf{p}}, \ldots$ interpretiert.

Mit Hilfe der Eigenschaften (1) – (5) kann man sehr leicht nachrechnen, dass $\mathcal{PRF}$ ein Modell von BON^- ist. Ausserdem ist in diesem Modell offensichtlich jede Instanz von $(\mathsf{F}\text{-}\mathsf{I}_\mathsf{N})$ erfüllt.

2.5.1 Theorem *$\mathcal{PRF}$ ist ein Modell von $\mathsf{BON}^- + (\mathsf{F}\text{-}\mathsf{I}_\mathsf{N})$ und damit auch ein Modell von $\mathsf{BON} + (\mathsf{F}\text{-}\mathsf{I}_\mathsf{N})$.*

In $\mathcal{PRF}$ gilt auch die Aussage $\forall x \mathsf{N}(x)$. Daher ist $\mathcal{PRF}$ auch ein Modell von Definition by Cases auf dem Universum, falls die Konstante d_V wie die Konstante d_N interpretiert wird.

2.6 Das Graphmodell $\mathcal{G}$

Plotkin und Scott haben ursprünglich das sogenannte Graphenmodell $\mathcal{P}(\omega)$ für den ungetypten λ-Kalkül entdeckt; eine ähnliche Konstruktion geht auch auf Engeler zurück. Das Modell $\mathcal{P}(\omega)$ wird nun zu einem Modell $\mathcal{G}$ von $\mathsf{BON}^- + (\mathsf{F}\text{-}\mathsf{I}_\mathsf{N})$ beziehungsweise $\mathsf{BON} + (\mathsf{F}\text{-}\mathsf{I}_\mathsf{N})$ erweitert. Wir geben hier nur die zentralen Definitionsschritte an und verweisen auf die Lehrbücher von Barendregt [1], Beeson [2] und Troelstra und van Dalen [20, 21] für zusätzliche Informationen.

Um $\mathcal{G}$ einzuführen, betrachten wir zuerst eine Codierung der endlichen Teilmengen von $\mathbb{N}$ als natürliche Zahlen. Ist n eine natürliche Zahl grösser als 0, so sei $Bin(n)$ die Menge $\{k_0, \ldots, k_m\}$ der eindeutig bestimmten natürlichen Zahlen mit $n = 2^{k_0} + \ldots + 2^{k_m}$ und $k_0 < \ldots < k_m$. Schliesslich sei ξ die Funktion von $\mathbb{N}$ in die Menge der endlichen Teilmengen von $\mathbb{N}$ mit $\xi(0) := \emptyset$ und $\xi(n) := Bin(n)$ für $n > 0$. Dann ist ξ offensichtlich eine Bijektion zwischen $\mathbb{N}$ und der Menge der endlichen Teilmengen von $\mathbb{N}$, so dass n als Code von $\xi(n)$ verstanden werden kann. Ausserdem ist $\xi(2^n) = \{n\}$.

Nun zur Definition von $\mathcal{G}$: Als Universum von $\mathcal{G}$ wählen wir die Potenzmenge $P(\mathbb{N})$ der Menge der natürlichen Zahlen. Die Applikation in $\mathcal{G}$ ist die Funktion $\bullet$ von $P(\mathbb{N}) \times P(\mathbb{N})$ nach $\mathbb{N}$, die für alle Teilmengen P und Q von $\mathbb{N}$ durch

$$P \bullet Q \ := \ \{n : \exists m(\xi(m) \subset Q \land \langle m, n \rangle \in P)\}$$

gegeben ist. Als Interpretation des Relationssymbols N wählen wir in $\mathcal{G}$ die Menge $\{\{n\} : n \in \mathbb{N}\}$.

$\mathcal{G}$ ist also vollständig bestimmt, wenn wir noch die Interpretationen der Konstanten angeben. Für die Konbinatoren k und s setzen wir

$$\begin{aligned}
\mathsf{k}^{\mathcal{G}} &:= \{\langle m, \langle n, p \rangle\rangle : p \in \xi(m)\}, \\
\mathsf{s}^{\mathcal{G}} &:= \{\langle m, \langle n, \langle p, q \rangle\rangle\rangle : \exists e[\exists p_0(\xi(p_0) \subset \xi(p) \land \langle p_0, \langle e, q \rangle\rangle \in \xi(m)) \land \\
&\qquad \forall e_0 \in \xi(e) \exists p_0(\xi(p_0) \subset \xi(p) \land \langle p_0, e_0 \rangle \in \xi(n))]\}.
\end{aligned}$$

Dann kann man (mit einigem Aufwand) nachrechnen, dass für alle Teilmengen M, P und Q von $\mathbb{N}$ gilt:

(1) $(\mathsf{k}^{\mathcal{G}} \bullet M) \bullet P = M$.

(2) $((\mathsf{s}^{\mathcal{G}} \bullet M) \bullet P) \bullet Q = (M \bullet Q) \bullet (P \bullet Q)$.

Um Pairing zu behandeln, interpretieren wir p als Teilmenge $\mathsf{p}^{\mathcal{G}}$ von $\mathbb{N}$, so dass

$$(\mathsf{p}^{\mathcal{G}} \bullet P) \bullet Q = \{2n : n \in P\} \cup \{2n+1 : n \in Q\}$$

für alle Teilmengen P und Q von $\mathbb{N}$ gilt. Dies wird erreicht durch

$$\mathsf{p}^{\mathcal{G}} \ := \ \{\langle 2^n, \langle 0, 2n\rangle\rangle : n \in \mathbb{N}\} \cup \{\langle 0, \langle 2^n, 2n+1\rangle\rangle : n \in \mathbb{N}\}.$$

Ausserdem sei

$$\mathsf{p}_0^{\mathcal{G}} \ := \ \{\langle 2^{2n}, n\rangle : n \in \mathbb{N}\} \ \text{ und } \ \mathsf{p}_1^{\mathcal{G}} \ := \ \{\langle 2^{2n+1}, n\rangle : n \in \mathbb{N}\}.$$

Damit gilt für alle Teilmengen P und Q von $\mathbb{N}$

(3) $\mathsf{p}_0^{\mathcal{G}} \bullet ((\mathsf{p}^{\mathcal{G}} \bullet P) \bullet Q) = P$ und $\mathsf{p}_1^{\mathcal{G}} \bullet ((\mathsf{p}^{\mathcal{G}} \bullet P) \bullet Q) = Q$.

Entsprechend der Interpretation von N setzen wir $\mathsf{0}^{\mathcal{G}} := \{0\}$ sowie

$$\mathsf{s}_{\mathsf{N}}^{\mathcal{G}} \ := \ \{\langle 2^n, n+1\rangle : n \in \mathbb{N}\} \ \text{ und } \ \mathsf{p}_{\mathsf{N}}^{\mathcal{G}} \ := \ \bullet\{\langle 2^n, n \dot{-} 1\rangle : n \in \mathbb{N}\}.$$

Daraus folgt für alle natürlichen Zahlen n, dass

(4) $\mathsf{s}_{\mathsf{N}}^{\mathcal{G}} \bullet \{n\} = \{n+1\}$ und $\mathsf{p}_{\mathsf{N}}^{\mathcal{G}} \bullet \{n+1\} = \{n\}$.

Schliesslich bleibt noch die Konstante $\mathsf{d}_{\mathsf{N}}^{\mathcal{G}}$ für Definition by Numerical Cases,

$$\mathsf{d}_{\mathsf{N}}^{\mathcal{G}} \ := \ \{\langle p, \langle q, \langle 2^m, \langle 2^n, e\rangle\rangle\rangle\rangle : (m = n \ \wedge \ e \in \xi(p)) \ \vee \ (m \neq n \ \wedge \ e \in \xi(q))\}$$

Durch Nachrechnen erhalten wir dann für alle Teilmengen P und Q von $\mathbb{N}$ und alle natürlichen Zahlen m und n:

(5) $m = n \ \rightarrow \ (((\mathsf{d}_{\mathsf{N}}^{\mathcal{G}} \bullet P) \bullet Q) \bullet \{m\}) \bullet \{n\} = P$.

(6) $m \neq n \ \rightarrow \ (((\mathsf{d}_{\mathsf{N}}^{\mathcal{G}} \bullet P) \bullet Q) \bullet \{m\}) \bullet \{n\} = Q$.

Aus den Eigenschaften (1) – (6) folgt, dass $\mathcal{G}$ ein Modell von BON^- ist. Trivialerweise erfüllt $\mathcal{G}$ ausserdem (F-$\mathsf{I_N}$) und (Tot). Aufgrund früherer Überlegungen kann $\mathcal{G}$ daher durch eine geeignete Interpretation der Konstanten $\mathsf{r_N}$ zu einem Modell von BON erweitert werden, und wir haben folgendes Theorem gezeigt.

2.6.1 Theorem *$\mathcal{G}$ ist ein Modell von* $\mathsf{BON} + (\mathsf{Tot}) + (\mathsf{F}\text{-}\mathsf{I_N})$.

2.7 Die Termmodelle $\mathcal{CNT}$ und $\mathcal{CTT}$

Bei der Definition von Termmodellen spielen Termreduktionen eine zentrale Rolle. Daher führen wir die in diesem Zusammenhang wichtigen Begriffe kurz ein.

2.7.1 Definition Für jede natürliche Zahl n definieren wir das *Numeral* $\overline{n}$ induktiv durch

$$\overline{0} := \mathsf{0} \quad \text{und} \quad \overline{n{+}1} := \overline{n}'.$$

Offensichtlich ist jedes Numeral ein L_p-Term. Die Numerale werden selbstverständlich später dafür verwendet werden, um die natürlichen Zahlen zu codieren.

2.7.2 Definition

1. *conv* ist eine 2-stellige Relation zwischen L_p-Termen. Es gilt s *conv* t genau dann, wenn es L_p-Terme a, b und c sowie voneinander verschiedene natürliche Zahlen m und n gibt, so dass einer der folgenden Fälle eintritt:

(1) $s \equiv \mathsf{k}ab$, $\quad t \equiv a$;
(2) $s \equiv \mathsf{s}abc$, $\quad t \equiv (ac)(bc)$;
(3) $s \equiv \mathsf{p}_0(a,b)$, $\quad t \equiv a$;
(4) $s \equiv \mathsf{p}_1(a,b)$, $\quad t \equiv b$;
(5) $s \equiv \mathsf{p_N}\mathsf{0}$, $\quad t \equiv \mathsf{0}$;
(6) $s \equiv \mathsf{p_N}(a')$, $\quad t \equiv a$;
(7) $s \equiv \mathsf{d_N}ab\overline{m}\,\overline{m}$, $\quad t \equiv a$;
(8) $s \equiv \mathsf{d_N}ab\overline{m}\,\overline{n}$, $\quad t \equiv b$;
(9) $s \equiv \mathsf{r_N}abc\mathsf{0}$, $\quad t \equiv a\mathsf{0}$;
(10) $s \equiv \mathsf{r_N}abc\overline{n{+}1}$, $\quad t \equiv bc\overline{n}(\mathsf{r_N}abc\overline{n})$.

Dann bezeichnen wir s als einen *Redex* und t als das *Kontraktum* von s.

2. $\rhd$ ist die 2-stellige Relation zwischen L_p-Termen, die für alle L_p-Terme a, b und c induktiv erzeugt wird durch:

   (1) $a \rhd a$;

   (2) $a \; conv \; b \implies a \rhd b$;

   (3) $a \rhd b \implies ac \rhd bc$;

   (4) $a \rhd b \implies ca \rhd cb$;

   (5) $a \rhd b$ und $b \rhd c \implies a \rhd c$.

   Gilt $a \rhd b$, so sagen wir, dass der Term a auf den Term b *reduziert*.

3. Wir schreiben $a \rhd_1 b$, wenn b aus a durch Kontraktion eines einzelnen Redex entsteht. Das bedeutet, dass $\rhd$ der reflexive und transitive Abschluss von $\rhd_1$ ist.

4. Wir scheiben $a \approx b$, wenn es eine endliche Folge von L_p-Termen $c_0, \ldots, c_n$ gibt, so dass $c_0 \equiv a$ und $c_n \equiv b$ ist und ausserdem für alle i mit $0 \leq i < n$ entweder $t_i \rhd_1 t_{i+1}$ oder $t_{i+1} \rhd_1 t_i$ gilt.

5. Eine *Reduktionsfolge* ist eine endliche oder unendliche Folge von L_p-Termen $t_0, t_1, t_2, \ldots$, so dass $t_0 \rhd_1 t_1 \rhd_1 t_2 \rhd_1 \ldots$ gilt.

6. Ein L_p-Term heisst *normal*, falls er keinen Redex enthält.

7. t heisst *Normalform* von s, falls t normal ist und $s \rhd t$ gilt.

2.7.3 Theorem (Church-Rosser-Theorem) *Sind a, b und c L_p-Terme mit $a \rhd b$ und $a \rhd c$, so existiert ein L_p-Term t mit $b \rhd t$ und $c \rhd t$.*

Wir verzichten hier auf einen Beweis dieses Theorems und verweisen auf die einschlägige Literatur. Im folgenden Theorem sind einige wichtige Folgerungen des Church-Rosser-Theorems zusammengefasst, deren Beweis wir ebenfalls übergehen.

2.7.4 Theorem

1. *Jeder L_p-Term hat höchstens eine Normalform.*

2. *Sind a und b L_p-Terme mit $a \approx b$, so gibt es einen L_p-Term c mit $a \rhd c$ und $b \rhd c$.*

3. *Sind a und b L_p-Terme mit $a \approx b$ und ist b normal, so gilt $a \rhd b$.*

4. *Sind a und b L_p-Terme mit $a \approx b$, so besitzen a und b keine Normalform oder dieselbe Normalform.*

2.7.5 Beispiel

1. Jedes Numeral $\overline{n}$ ist ein normaler L_p-Term.

2. Der L_p-Term $\omega := (\lambda x.xx)$ ist normal. Dagegen besitzt der L_p-Term $\omega\omega$ keine Normalform.

3. $\mathsf{k}\omega(\omega\omega)$ hat die Normalform ω. Es gibt aber auch unendliche Reduktionsfolgen, die mit $\mathsf{k}\omega(\omega\omega)$ beginnen.

Eine erste Idee zum Aufbau eines Termmodells von BON besteht darin, nur mit den normalen L_p-Termen zu arbeiten und darauf eine 2-stellige partielle Applikation zu definieren, indem wir

$$a * b := \text{Normalform von } ab$$

für alle normalen Terme a und b setzen. Dies ergibt jedoch Probleme im Zusammenhang mit Striktheit. Es gilt $((\mathsf{s} * \mathsf{k}) * \omega) * \omega = \omega$, so dass wir aufgrund des Axioms (2) von BON auch $(\mathsf{k} * \omega) * (\omega * \omega) = \omega$ erhalten sollten. Mit Striktheit würde daraus folgen, dass $\omega * \omega$ einen Wert besitzt. Da $\omega * \omega$ keine Normalform hat, ist dies ein Widerspruch.

Um diese Schwierigkeit zu umgehen, behandeln wir die Applikation etwas restriktiver. Wir begnügen uns nicht damit, dass eine Reduktionsfolge zu einem normalen Term führt, sondern betrachten sämtliche Reduktionsfolgen.

2.7.6 Definition

1. Eine Reduktionsfolge heisst *maximal*, falls sie unendlich ist oder mit einem normalen Term endet.
2. Ein L_p-Term t heisst *starke Normalform* eines L_p-Terms s, falls jede maximale Reduktionsfolge, die mit s beginnt, mit t endet.

Hat ein L_p-Term eine starke Normalform t, so ist t die Normalform von s. Es gibt aber L_p-Terme, die zwar eine Normalform, aber keine starke Normalform besitzen. Ein Beispiel dafür ist der Term $\mathsf{k}\omega(\omega\omega)$.

Mit *snf* bezeichnen wir nun die partielle Funktion von den L_p-Termen in die L_p-Terme, die jedem a seine starke Normalform zuordnet, falls a eine starke Normalform besitzt, und andernfalls undefiniert ist;

$$\mathit{snf}(a) \;:\simeq\; \text{starke Normalform von } a.$$

Nun führen wir zwei Termmodelle ein: Das Modell $\mathcal{CNT}$, das auf den geschlossenen Termen in Normalform basiert, und das Modell $\mathcal{CTT}$, das auf allen geschlossenen Termen aufbaut.

2.7.7 Definition Die partielle L_p-Struktur $\mathcal{CNT}$ ist durch folgende Bedingungen bestimmt:

($\mathcal{CNT}$.1) Das Universum $|\mathcal{CNT}|$ von $\mathcal{CNT}$ umfasst genau die geschlossenen L_p-Terme in Normalform.

($\mathcal{CNT}$.2) Das Relationssymbol N wird in $\mathcal{CNT}$ durch die Menge der Numerale interpretiert, d.h.

$$\mathsf{N}^{\mathcal{CNT}} := \{\overline{n} : n \in \mathbb{N}\}.$$

($\mathcal{CNT}$.3) Jede Konstante von L_p wird in $\mathcal{CNT}$ durch sich selbst interpretiert.

($\mathcal{CNT}$.4) Das 2-stellige Funktionssymbol $\cdot$ für Termanwendung wird in $\mathcal{CNT}$ durch die partielle Funktion $\bullet^{\mathcal{CNT}}$ von $|\mathcal{CNT}| \times |\mathcal{CNT}|$ nach $|\mathcal{CNT}|$ interpretiert, die für alle a und b aus $|\mathcal{CNT}|$ definiert ist durch

$$a \bullet^{\mathcal{CNT}} b \;:\simeq\; snf(ab).$$

Nun lässt sich ohne grosse Mühe nachrechnen, dass $\mathcal{CNT}$ ein Modell von BON ist. Aufgrund der Interpretation des Relationssymbols N durch die Menge der Numerale ist auch klar, dass Formelinduktion (F-I_{N}) in $\mathcal{CNT}$ gültig ist.

2.7.8 Theorem *$\mathcal{CNT}$ ist ein Modell von* $\mathsf{BON} + (\mathsf{F}\text{-}\mathsf{I}_{\mathsf{N}})$.

Es ist aber auch leicht zu sehen, dass Totalität (Tot) in $\mathcal{CNT}$ verletzt ist. Im folgenden betrachten wir nun Äquivalenzklassen von geschlossenen L_p-Termen, um ein Modell von BON + (Tot) zu definieren. Der Beweis des folgenden Lemmas ist trivial.

2.7.9 Lemma

1. *Die Relation $\approx$ zwischen L_p-Termen ist eine Äquivalenzrelation.*

2. *Sind a, b, s und t L_p-Terme mit $a \approx b$ und $s \approx t$, so gilt auch $as \approx bt$.*

Wir schreiben $[a]_\approx$ für die Äquivalenzklasse des L_p-Terms a bezüglich der Äquivalenzrelation $\approx$ und benützen diese Äquivalenzklassen als Grundobjekte des Modells $\mathcal{CTT}$.

2.7.10 Definition Die (partielle) L_p-Struktur $\mathcal{CTT}$ ist durch folgende Bedingungen bestimmt:

($\mathcal{CTT}$.1) Das Universum $|\mathcal{CTT}|$ von $\mathcal{CTT}$ umfasst genau die Äquivalenzklassen $[a]_\approx$ der geschlossenen L_p-Terme a.

($\mathcal{CTT}$.2) Das Relationssymbol N wird in $\mathcal{CTT}$ durch die Menge der Äquivalenzklassen der Numerale interpretiert, d.h.

$$\mathsf{N}^{\mathcal{CTT}} := \{[\overline{n}]_\approx : n \in \mathbb{N}\}.$$

($\mathcal{CTT}$.3) Jede Konstante von L_p wird in $\mathcal{CTT}$ durch ihre Äquivalenzklasse bezüglich $\approx$ interpretiert.

($\mathcal{CTT}$.4) Das 2-stellige Funktionssymbol $\cdot$ für Termanwendung wird in $\mathcal{CTT}$ durch die Funktion $\bullet^{\mathcal{CTT}}$ von $|\mathcal{CTT}| \times |\mathcal{CTT}|$ nach $|\mathcal{CTT}|$ interpretiert, die für alle $[a]_\approx$ und $[b]_\approx$ aus $|\mathcal{CTT}|$ definiert ist durch

$$[a]_\approx \bullet^{\mathcal{CTT}} [b]_\approx := [ab]_\approx.$$

Aufgrund des vorhergehenden Lemmas ist die Funktion $\bullet^{\mathcal{CTT}}$ wohldefiniert. Ausserdem folgt sofort, dass jeder geschlossene Term a in $\mathcal{CTT}$ den Wert $[a]_{\approx}$ besitzt.

Die Axiome von BON sowie Formelinduktion $(\mathsf{F}\text{-}\mathsf{I}_{\mathsf{N}})$ lassen sich in $\mathcal{CTT}$ leicht verifizieren. Dazu kommt, dass $\mathcal{CTT}$ trivialerweise ein Modell von (Tot) ist.

2.7.11 Theorem *$\mathcal{CTT}$ ist ein Modell von* $\mathsf{BON} + (\mathsf{Tot}) + (\mathsf{F}\text{-}\mathsf{I}_{\mathsf{N}})$.

2.8 Eine allgemeine Modellkonstruktion

In diesem Abschnitt beschreiben wir eine recht allgemeine Methode, um Modelle von $\mathsf{BON} + (\mathsf{d}_{\mathsf{V}}) + (\mathsf{F}\text{-}\mathsf{I}_{\mathsf{N}})$ zu erzeugen. Zur möglichst durchsichtigen Beschreibung dieser Konstruktion arbeiten wir mit monotonen Operatoren.

2.8.1 Definition Gegeben seien eine nichtleere Menge M und eine natürliche Zahl $n > 0$.

1. Die Abbildungen von $P(M^n)$ nach $P(M^n)$ bezeichnen wir als n-stellige *Operatoren* auf M.

2. Einen n-stelligen Operator Γ auf M bezeichnen wir als *monoton*, falls für alle Teilmengen X und Y von M^n gilt:
$$X \subset Y \quad \Longrightarrow \quad \Gamma(X) \subset \Gamma(Y).$$

3. Ist Γ ein n-stelliger Operator auf M, so heisst eine Teilmenge X von M^n *abgeschlossen* unter Γ, falls $\Gamma(X) \subset X$ gilt.

4. Ist Γ ein n-stelliger Operator auf M, so heisst eine Teilmenge X von M^n *Fixpunkt* von Γ, falls $\Gamma(X) = X$ gilt.

5. Ist Γ ein n-stelliger Operator auf M, so heisst eine Teilmenge X von M^n *kleinster Fixpunkt* von Γ, falls X ein Fixpunkt von Γ ist und $X \subset Y$ für jeden Fixpunkt Y von Γ gilt.

Monotone Operatoren besitzen die folgende wichtige Fixpunkteigenschaft. Wir übergehen aber den Beweis dieses bekannten Theorems und verweisen auf die einschlägige Literatur.

2.8.2 Theorem *Gegeben seien eine nichtleere Menge M und eine natürliche Zahl $n > 0$. Dann besitzt jeder n-stellige monotone Operator Γ auf M einen kleinsten Fixpunkt.*

Im folgenden bezeichnen wir den kleinsten Fixpunkt eines monotonen Operators Γ mit $lfp(\Gamma)$. Diesen kleinsten Fixpunkt kann man auch durch Rekursion über die Ordinalzahlen definieren. Ist Γ ein n-stelliger monotoner Operator auf der Menge M und α eine Ordinalzahl, so setzen wir

$$\Gamma^\alpha \;:=\; \Gamma(\bigcup_{\xi<\alpha} \Gamma^\xi).$$

Aufgrund der Monotonie von Γ erhalten wir sofort $\Gamma^\alpha \subset \Gamma^\beta$ für alle Ordinalzahlen α und β mit $\alpha \leq \beta$. Ausserdem gilt

$$lfp(\Gamma) = \bigcup_{\alpha \in On} \Gamma^\alpha = \bigcap\{X \subset M^n : \Gamma(X) \subset X\} = \bigcap\{X \subset M^n : \Gamma(X) = X\}.$$

Ausgangspunkt für eine sehr allgemeine Konstruktion von Modellen der Theorie $\mathsf{BON} + (\mathsf{d_V}) + (\mathsf{F\text{-}I_N})$ bilden 5-Tupel $\mathcal{K} = (M, \pi, \pi_0, \pi_1, \mathcal{F})$ mit folgenden Eigenschaften:

(1) M ist eine unendliche Menge.

(2) π ist eine Abbildung von M^2 nach M und π_0 sowie π_1 sind Abbildungen von M nach M, so dass $\pi_0(\pi(m,n)) = m$ und $\pi_1(\pi(m,n)) = n$ für alle $m, n \in M$ gilt.

(3) $\mathcal{F}$ ist eine Menge partieller Funktionen von M nach M mit $card(\mathcal{F}) \leq card(M)$.

Nun wählen wir zuerst für jedes $\iota \in \mathbb{N} \cup \mathcal{F}$ ein Element α_ι aus M, so dass dadurch die Menge $\mathbb{N} \cup \mathcal{F}$ injektiv in M abgebildet wird. Dann setzen wir

$$\begin{array}{lll}
\hat{\mathsf{k}} := \pi(\alpha_0, \alpha_0), & \hat{\mathsf{s}} := \pi(\alpha_0, \alpha_1), & \hat{\mathsf{p}} := \pi(\alpha_0, \alpha_2), \\
\hat{\mathsf{p}}_0 := \pi(\alpha_0, \alpha_3), & \hat{\mathsf{p}}_1 := \pi(\alpha_0, \alpha_4), & \hat{0} := \alpha_0, \\
\hat{\mathsf{s}}_\mathsf{N} := \pi(\alpha_0, \alpha_5), & \hat{\mathsf{p}}_\mathsf{N} := \pi(\alpha_0, \alpha_6), & \hat{\mathsf{d}}_\mathsf{N} := \pi(\alpha_0, \alpha_7), \\
\hat{\mathsf{d}}_\mathsf{V} := \pi(\alpha_0, \alpha_8), & \multicolumn{2}{l}{\text{und } \hat{\mathsf{f}} := \pi(\alpha_0, \alpha_\mathsf{f}) \text{ für jedes } \mathsf{f} \in \mathcal{F}.}
\end{array}$$

Im nächsten Schritt führen wir den 3-stelligen Operator Γ auf M ein, der dadurch definiert ist, dass $\Gamma(X)$ für jede Teilmenge X von M^3 aus allen $(e, m, n) \in M^3$ besteht, die eine der folgenden Bedingungen erfüllen:

(1) $e = \hat{\mathsf{k}} \ \wedge \ n = \pi(\alpha_1, m);$

(2) $e = \pi(\alpha_1, n);$

(3) $e = \hat{\mathsf{s}} \ \wedge \ n = \pi(\alpha_2, m);$

(4) $(\exists x \in M)[e = \pi(\alpha_2, x) \wedge n = \pi(\alpha_3, \pi(x, m))];$

(5) $(\exists x_0, x_1, y_0, y_1 \in M)[e = \pi(\alpha_3, \pi(x_0, y_0)) \wedge (x_0, m, x_1) \in X \wedge$
$(y_0, m, y_1) \in X \wedge (x_1, y_1, n) \in X];$

(6) $e = \hat{\mathsf{p}} \ \wedge \ n = \pi(\alpha_4, m);$

(7) $(\exists x \in M)[e = \pi(\alpha_4, x) \wedge n = \pi(x, m)];$

(8a) $e = \hat{\mathsf{p}}_0 \ \wedge \ (\exists x, y \in M)[m = \pi(x, y)] \wedge n = \pi_0 m;$

(8b) $e = \hat{\mathsf{p}}_0 \;\wedge\; (\forall x, y \in M)[m \neq \pi(x,y)] \wedge n = \alpha_0;$

(9a) $e = \hat{\mathsf{p}}_1 \;\wedge\; (\exists x, y \in M)[m = \pi(x,y)] \wedge n = \pi_1 m;$

(9b) $e = \hat{\mathsf{p}}_1 \;\wedge\; (\forall x, y \in M)[m \neq \pi(x,y)] \wedge n = \alpha_0;$

(10) $e = \hat{\mathsf{s}}_{\mathsf{N}} \;\wedge\; (\exists i \in \mathbb{N})[m = \alpha_i \wedge n = \alpha_{i+1}];$

(11) $e = \hat{\mathsf{p}}_{\mathsf{N}} \;\wedge\; (\exists i \in \mathbb{N})[m = \alpha_{i+1} \wedge n = \alpha_i];$

(12) $e = \hat{\mathsf{d}}_{\mathsf{N}} \;\wedge\; n = \pi(\alpha_5, m);$

(13) $(\exists x \in M)[e = \pi(\alpha_5, x) \wedge n = \pi(\alpha_6, \pi(x, m))];$

(14) $(\exists x, y \in M)[e = \pi(\alpha_6, \pi(x,y)) \wedge n = \pi(\alpha_7, \pi(\pi(x,y), m))];$

(15a) $(\exists x, y \in M)(\exists i, j \in \mathbb{N})[e = \pi(\alpha_7, \pi(\pi(x,y), \alpha_i)) \wedge m = \alpha_j \wedge$
$i = j \wedge n = x];$

(15b) $(\exists x, y \in M)(\exists i, j \in \mathbb{N})[e = \pi(\alpha_7, \pi(\pi(x,y), \alpha_i)) \wedge m = \alpha_j \wedge$
$i \neq j \wedge n = y];$

(16) $e = \hat{\mathsf{d}}_{\mathsf{V}} \;\wedge\; n = \pi(\alpha_8, m);$

(17) $(\exists x \in M)[e = \pi(\alpha_8, x) \wedge n = \pi(\alpha_9, \pi(x, m))];$

(18) $(\exists x, y \in M)[e = \pi(\alpha_9, \pi(x,y)) \wedge n = \pi(\alpha_{10}, \pi(\pi(x,y), m))];$

(19a) $(\exists x, y, z_0, z_1 \in M)[e = \pi(\alpha_{10}, \pi(\pi(x,y), z_0)) \wedge m = z_1 \wedge$
$z_0 = z_1 \wedge n = x];$

(19b) $(\exists x, y, z_0, z_1 \in M)[e = \pi(\alpha_{10}, \pi(\pi(x,y), z_0)) \wedge m = z_1 \wedge$
$z_0 \neq z_1 \wedge n = y];$

(20) $(\exists \mathsf{f} \in \mathcal{F})[e = \hat{\mathsf{f}} \wedge n \simeq \mathsf{f}(m)].$

Offensichtlich ist Γ ein monotoner Operator auf M und besitzt also einen kleinsten Fixpunkt $\mathit{lfp}(\Gamma)$. Beispielsweise durch Induktion nach α kann man ausserdem leicht zeigen, dass für alle e, m, n_0 und n_1 aus M gilt:

$$(e, m, n_0) \in \Gamma^\alpha \text{ und } (e, m, n_1) \in \Gamma^\alpha \quad \Longrightarrow \quad n_0 = n_1.$$

Daher ist $\mathit{lfp}(\Gamma) = \bigcup_{\alpha \in On} \Gamma^\alpha$ funktional im dritten Argument. Das bedeutet, dass wir durch die Definition

$$e \bullet^{\mathcal{K}} m \;\simeq\; n \quad :\Longleftrightarrow \quad (e, m, n) \in \mathit{lfp}(\Gamma)$$

für alle e, m und n aus M eine partielle Funktion $\bullet^{\mathcal{K}}$ von $M \times M$ nach M erhalten. In der Regel ist $\bullet^{\mathcal{K}}$ keine totale Funktion von $M \times M$ nach M.

Die von unserem 5-Tupel $\mathcal{K} = (M, \pi, \pi_0, \pi_1, \mathcal{F})$ erzeugte partielle L_p-Struktur $Gen(\mathcal{K})$ hat als Universum $|Gen(\mathcal{K})|$ die Menge M. Die Interpretation des Relationssymbols N ist die Menge $\{\alpha_i : i \in \mathbb{N}\}$, und die Konstanten $\mathsf{k}, \mathsf{s}, \mathsf{p}, \ldots$ werden durch die oben eingeführten Codes $\hat{\mathsf{k}}, \hat{\mathsf{s}}, \hat{\mathsf{p}}, \ldots$ interpretiert. Die Termanwendung $\cdot$ wird in $Gen(\mathcal{K})$ im Sinne der partiellen Funktion $\bullet^{\mathcal{K}}$ behandelt.

Aufgrund der Bedingungen $(1)-(20)$ in der Definition des Operators Γ kann man zeigen, dass $Gen(\mathcal{K})$ ein Modell von $\mathsf{BON}^- + (\mathsf{d_V}) + (\mathsf{F\text{-}I_N})$ ist. Durch geeignete Interpretation der Konstanten $\mathsf{r_N}$ (vgl. Theorem 2.4.1) wird also $Gen(\mathcal{K})$ auch zu einem Modell von BON.

2.8.3 Theorem *$Gen(\mathcal{K})$ ist ein Modell von* $\mathsf{BON} + (\mathsf{d_V}) + (\mathsf{F\text{-}I_N})$.

Dazu kommt, dass in $Gen(\mathcal{K})$ jede partielle Funktion f aus $\mathcal{F}$ durch das Element $\hat{\mathsf{f}}$ aus M repräsentiert wird. Für alle m und n aus M gilt nämlich $\mathsf{f}(m) \simeq n$ genau dann, wenn $\hat{\mathsf{f}} \bullet^{\mathcal{K}} m \simeq n$.

2.9 Mengentheoretische Modelle

In diesem Abschnitt machen wir von den vorhergehenden Überlegungen Gebrauch, um zu zeigen, dass jeder Abschnitt V_λ der kumulativen Hierarchie mit einer Limeszahl λ zu einem Modell von $\mathsf{BON} + (\mathsf{d_V}) + (\mathsf{F\text{-}I_N})$ gemacht werden kann. Zur Wiederholung, die kumulative Hierarchie $(V_\alpha : \alpha \in On)$ wird durch transfinite Induktion über die Ordinalzahlen definiert:

$$V_0 := \emptyset \quad \text{und} \quad V_\alpha := \bigcup_{\xi<\alpha} P(V_\xi) \;\; \text{für } \alpha > 0.$$

Nun wählen wir eine beliebige Limeszahl λ und betrachten die Menge V_λ. Ausserdem sei $Fun(V_\lambda)$ die Menge aller Funktionen im mengentheoretischen Sinn, die Elemente von V_λ sind. Schliesslich sei π die übliche Paarfunktion von $V_\lambda \times V_\lambda$ nach V_λ mit $\pi(m,n) = \{\{m\},\{m,n\}\}$, und π_0 sowie π_1 seien Funktionen von V_λ nach V_λ mit $\pi_0(\pi(m,n)) = m$ und $\pi_1(\pi(m,n)) = n$ für alle m und n aus V_λ.

Dann kann das 5-Tupel

$$\mathcal{K} = (V_\lambda, \pi, \pi_0, \pi_1, Fun(V_\lambda))$$

aufgrund der Überlegungen des vorhergehenden Abschnittes zu einem Modell $Gen(\mathcal{K})$ von $\mathsf{BON} + (\mathsf{d_V}) + (\mathsf{F\text{-}I_N})$ erweitert werden. Dabei kann man die Einbettung α von $\mathbb{N} \cup Fun(V_\lambda)$ in V_λ so wählen, dass α_i für jede natürliche Zahl i die Ordinalzahl i ist.

Kapitel 3

Explizite Mathematik

In diesem Kapitel fügen wir zu unseren applikativen Theorien eine sehr flexible Typenstruktur hinzu und erhöhen dadurch ihre Ausdrucksstärke und ihren Anwendungsbereich beträchtlich. Die so erhaltenen Systeme werden häufig als Systeme *expliziter Mathematik* bezeichnet.

Explizite Mathematik wurde von Feferman vor allem in den beiden zentralen Arbeiten [5, 6] eingeführt, in denen er auch wesentliche syntaktische und semantische Eigenschaften expliziter Mathematik behandelte. Hier verfolgen wir einen geringfügig anderen Ansatz und entwickeln explizite Mathematik basierend auf Typen und Namen; vgl. dazu [8, 14].

Unser Formalismus unterscheidet zwischen Individuen (Operationen) und Kollektionen von Individuen, die wir als Typen bezeichnen. Dazu kommt, dass jeder Typ Individuen als Namen besitzt und Typen mittels ihrer Namen angesprochen werden können.

3.1 Die Theorie EET

Wir beginnen mit der Theorie EET der elementaren expliziten Typen. Sie wird in der Sprache $\mathbb{L}_p$ zweiter Stufe formuliert, die die Sprache L_p um folgende Grundzeichen erweitert:

1. Abzählbar unendlich viele Typenvariablen $V0, V1, \ldots, Vi, \ldots (i \in \mathbb{N})$.

2. Die zweistelligen Relationssymbole $\in$ für die Elementbeziehung und $\Re$ für die Namensbeziehung.

3. Für jede natürliche Zahl e eine Individuenkonstante c_e.

Die Variablen von L_p bezeichnen wir von nun an als *Individuenvariablen.* Die $\mathbb{L}_p$-*Individuenterme* werden analog zu den L_p-Termen definiert, selbstverständlich unter Einbeziehung der Konstanten c_e für alle natürlichen Zahlen e. Die *Typenterme* von $\mathbb{L}_p$ sind die Typenvariablen.

Die *Atomformeln* von $\mathbb{L}_p$ sind alle Ausdrücke der Form $a\downarrow, (a = b), \mathsf{N}(a), (a \in X), (X = Y)$ und $\Re(a, X)$, wobei a sowie b Individuenterme und X sowie Y Typenvariablen von $\mathbb{L}_p$ sind. Davon ausgehend werden die $\mathbb{L}_p$-*Formeln* wie folgt generiert:

1. Jede Atomformel von $\mathbb{L}_p$ ist eine $\mathbb{L}_p$-Formel.

2. Ist A eine $\mathbb{L}_p$-Formel, so ist $\neg A$ eine $\mathbb{L}_p$-Formel.

3. Sind A und B $\mathbb{L}_p$-Formeln, so ist $(A \vee B)$ eine $\mathbb{L}_p$-Formel.

4. Ist A eine $\mathbb{L}_p$-Formel und sind x beziehungsweise X Individuen- beziehungsweise Typenvariablen von $\mathbb{L}_p$, so sind $\exists x A$ und $\exists X A$ $\mathbb{L}_p$-Formeln.

Wie früher lassen wir häufig Klammern fort, wenn dadurch keine Unklarheiten entstehen. $(A \wedge B)$, $(A \to B)$, $(A \leftrightarrow B)$ und $\forall x A$ werden definiert wie vorher; ausserdem sei $\forall X A := \neg\exists X \neg A$, $(\exists x{\in}V)A := \exists x(x{\in}V \wedge A)$ und $(\forall x{\in}V)A := \forall x(x{\in}V \rightarrow A)$. Ist schliesslich $\vec{a} = a_0, \ldots, a_n$ und $\vec{X} = X_0, \ldots, X_n$, so setzen wir $\Re(\vec{a}, \vec{X}) := \Re(a_0, X_0) \wedge \ldots \wedge \Re(a_n, X_n)$.

Mitteilungszeichen (auch mit Indizes)

$u, v, w, x, y, z, f, g, h$	für Individuenvariablen von $\mathbb{L}_p$;
a, b, c, r, s, t	für Individuenterme von $\mathbb{L}_p$;
U, V, W, X, Y, Z	für Typenvariablen von $\mathbb{L}_p$;
A, B, C, D, E, F	für Formeln von $\mathbb{L}_p$.

Eine $\mathbb{L}_p$-Formel heisst *stratifiziert*, falls in ihr das Relationssymbol $\Re$ nicht auftritt. Die stratifizierten $\mathbb{L}_p$-Formeln, in denen keine gebundenen Typenvariablen vorkommen, bezeichnen wir als *elementare* $\mathbb{L}_p$-Formeln.

Die Formel $\Re(a, X)$ besagt, dass der Individuenterm a den Typ X *repräsentiert*. Häufig sprechen wir in diesem Falle auch davon, dass a ein *Name* von X ist. Die folgenden Axiome über explizite Repräsentation und Extensionalität formalisieren, dass jeder Typ einen Namen besitzt und Typen extensional sind. Die Repräsentation von Typen durch ihre Namen ist dagegen intensional.

Explizite Repräsentation und Extensionalität

(E.1) $\exists x \Re(x, X)$.

(E.2) $\Re(a, X) \wedge \Re(a, Y) \;\rightarrow\; X = Y$.

(E.3) $\forall z(z \in X \leftrightarrow z \in Y) \;\rightarrow\; X = Y$.

In den folgenden Axiomen über elementare Komprehension spielen die Konstanten c_e eine ausgezeichnete Rolle. Sie dienen dazu, um die durch elementare Komprehension gebildeten Typen uniform in ihren Parametern zu repräsentieren.

Im Zusammenhang mit dem Namensgebungsprozess bei elementarer Komprehension ist es bequem, von folgenden Konventionen auszugehen:

1. Es ist eine beliebige aber feste Gödelisierung der $\mathbb{L}_p$-Formeln gegeben.

2. Ausserdem seien beliebige aber feste Aufzählungen $v0, v1, v2, \ldots$ und $V0, V1, V2, \ldots$ der Individuen- und Typenvariablen von $\mathbb{L}_p$ gegeben. Ist A eine $\mathbb{L}_p$-Formel, in der keine anderen Individuenvariablen als $v0, \ldots, vm$ und keine anderen Typenvariablen als $V0, \ldots, Vn$ frei auftreten und ist $\vec{a} = a_0, \ldots, a_m$ sowie $\vec{X} = X_0, \ldots, X_n$, so schreiben wir $A[\vec{a}, \vec{X}]$ für die $\mathbb{L}_p$-Formel, die wir aus A erhalten, indem wir simultan vi durch a_i und Vj durch X_j ersetzen $(0 \leq i \leq m, 0 \leq j \leq n)$.

Die folgenden Axiome über elementare Komprehension hängen von der gegebenen Gödelisierung und der Aufzählungen der Variablen ab. Dies bedeutet jedoch keine echte Einschränkung.

Elementare Komprehension Es sei $A[x, \vec{y}, \vec{Z}]$ eine elementare $\mathbb{L}_p$-Formel mit Gödelnummer e. Dann haben wir als Axiome:

(ECA.1) $\exists X \forall x (x \in X \leftrightarrow A[x, \vec{u}, \vec{V}])$.

(ECA.2) $\Re(\vec{v}, \vec{V}) \land \forall x (x \in X \leftrightarrow A[x, \vec{u}, \vec{V}]) \ \rightarrow \ \Re(\mathsf{c}_e(\vec{u}, \vec{v}), X)$.

Daher werden Namen für Typen, die durch elementare Komprehension gebildet werden, mit Hilfe der Konstanten c_e konstruiert. Diese Namensgebung ist uniform in den Individuen- und Typenparametern der definierenden Formel.

Die Theorie EET der expliziten elementaren Typen ist formuliert in der Sprache $\mathbb{L}_p$. Die logischen Axiome von EET umfassen die Axiome der Logik der partiellen Terme für Individuen, wobei die Striktheitsaxiome ergänzt werden durch

(i) $a \in X \ \rightarrow \ a\downarrow$,

(ii) $\Re(a, X) \ \rightarrow \ a\downarrow$

für alle $\mathbb{L}_p$-Terme a und alle Typenvariablen X. Für die Typen hat man die Axiome der klassischen Logik. Ausserdem stehen die Gleichheitsaxiome für Individuen und Typen zur Verfügung. Die nichtlogischen Axiome von EET umfassen alle Axiome von BON formuliert für die Sprache $\mathbb{L}_p$, die Axiome (E.1) – (E.3) über explizite Repräsentation und Extensionalität sowie die Axiome (ECA.1) und (ECA.2) über elementare Komprehension.

Ist $A[x, \vec{y}, \vec{Z}]$ eine elementare $\mathbb{L}_p$-Formel, so schreiben wir häufig etwas informell $\{x : A[x, \vec{u}, \vec{V}]\}$ für den durch elementare Komprehension gebildeten Typ X, der genau die Individuen x mit der Eigenschaft $A[x, \vec{u}, \vec{V}]$ enthält. In diesem Sinne lassen wir auch Ausdrücke der Form $\{x : A[x, \vec{u}, \vec{V}]\}$ als Typenterme zu; $b \in \{x : A[x, \vec{u}, \vec{V}]\}$ steht dann für $A[b, \vec{u}, \vec{V}]$.

3.1.1 Bemerkung Auf Grundlage dieser liberalisierten Schreibweise lassen sich in EET folgende Typen durch elementare Komprehension einführen:

$$\begin{aligned} \emptyset &:= \{x : x \neq x\}, \\ \mathsf{V} &:= \{x : x = x\}, \\ \{a_1, \ldots, a_n\} &:= \{x : x = a_1 \vee \ldots \vee x = a_n\}. \end{aligned}$$

Uniform in den Typentermen S und T können in EET ebenfalls folgende Typenterme durch elementare Komprehension eingeführt werden:

$$\begin{aligned} S \setminus T &:= \{x : x \in S \wedge x \notin T\}, \\ S \cup T &:= \{x : x \in S \vee x \in T\}, \\ S \cap T &:= \{x : x \in S \wedge x \in T\}, \\ S \times T &:= \{x : x = (\mathsf{p}_0 x, \mathsf{p}_1 x) \wedge \mathsf{p}_0 x \in S \wedge \mathsf{p}_1 x \in T\}, \end{aligned}$$

$$\begin{aligned} S+T &:= \{x : (x = (\mathsf{0}, \mathsf{p}_1 x) \wedge \mathsf{p}_1 x \in S) \vee (x = (\mathsf{1}, \mathsf{p}_1 x) \wedge \mathsf{p}_1 x \in T)\}, \\ (S \to T) &:= \{f : (\forall x \in S)(fx \in T)\}. \end{aligned}$$

3.2 Ontologische Überlegungen

In diesem Abschnitt wollen wir einige ontologische Eigenschaften von EET im Zusammenhang mit Komprehension, Namensgebung und Extensionalität zusammenstellen. Dadurch soll auf einige Besonderheiten des Namensprädikates hingewiesen werden. Für neuere Ergebnisse in diesem Zusammenhang sei auf [3, 13, 15] verwiesen.

3.2.1 Theorem $\mathsf{EET}+$ *volle Komprehension, d.h.*

$$\exists X \forall x (x \in X \leftrightarrow A(x))$$

für beliebige $\mathbb{L}_p$-Formeln $A(x)$, ist inkonsistent.

Beweis Wir zeigen, dass mit voller Komprehension das übliche Russel-Argument übernommen werden kann. Steht volle Komprehension zur Verfügung, so gibt es einen Typ S mit

$$a \in S \;\leftrightarrow\; \exists X (\Re(a, X) \wedge a \notin X)$$

für alle Individuenterme a. Aufgrund der Axiome über explizite Repräsentation besitzt dieser Typ einen Namen s. Dann gilt $\Re(s, S)$, und wir erhalten

$$s \in S \;\leftrightarrow\; \exists X (\Re(s, X) \wedge s \notin X) \;\leftrightarrow\; s \notin S.$$

Dies ist ein Widerspruch, so dass damit die Inkonsistenz von EET mit voller Komprehension gezeigt ist. □

3.2.2 Theorem *Die Namen eines Typs bilden in* EET *beweisbar keinen Typ, d.h.*

$$\mathsf{EET} \vdash \neg \exists Y \forall x (x \in Y \leftrightarrow \Re(x, X)).$$

Beweis Wir arbeiten informell in EET und wählen zuerst zum gegebenen Typ X einen Namen a. Ausserdem sieht man sofort, dass in Abhängigkeit von einem Individuum u und Typen V und W mit elementarer Komprehension der Typ

$$\{x : (x \notin V \wedge u \notin W) \vee (x \in V \wedge u \in W)\}$$

gebildet werden kann. Aufgrund der Uniformität der elementaren Komprehension gibt es sogar einen Individuenterm s mit

$$\Re(v, V) \wedge \Re(w, W) \to \Re(s(v, w), \{x : (x \notin V \wedge w \notin W) \vee (x \in V \wedge w \in W)\}).$$

Für $t := \lambda y.s(a, y)$ folgt daraus

$$\Re(w, W) \ \to \ \Re(tw, \{x : (x \notin X \wedge w \notin W) \vee (x \in X \wedge w \in W)\}). \tag{1}$$

Nun nehmen wir an, dass die Namen von X einen Typ bilden. Dann gibt es einen Typ Y mit

$$x \in Y \ \leftrightarrow \ \Re(x, X)$$

für alle x. Mit elementarer Komprehension können wir dann einen Typ $R := \{x : tx \notin Y\}$ einführen. Folglich gilt

$$x \notin R \ \leftrightarrow \ \Re(tx, X) \tag{2}$$

für alle x. Ausserdem sei r ein Name von R. Aus (1) folgt dann sofort

$$\Re(tr, \{x : (x \notin X \wedge r \notin R) \vee (x \in X \wedge r \in R)\}). \tag{3}$$

Nun verwenden wir (2) spezialisiert auf $x = r$ und (3), um die Äquivalenz

$$r \notin R \ \leftrightarrow \ X = \{x : (x \notin X \wedge r \notin R) \vee (x \in X \wedge r \in R)\}$$

abzuleiten. Daraus folgt, dass $r \notin R$ zu $r \in R$ äquivalent ist. Dies ist ein Widerspruch, und daher muss unsere Annahme falsch sein. $\square$

3.2.3 Folgerung Jeder Typ besitzt in EET unendlich viele Namen.

Beweis Wenn es einen Typ X mit nur endlich vielen Namen gäbe, so wäre dies ein Widerspruch zum vorhergehenden Theorem, da die Zusammenfassung von endlich vielen Individuen in EET einen Typ bildet. $\square$

3.2.4 Definition Für alle $\mathbb{L}_p$-Individuenterme a und b setzen wir

$$
\begin{aligned}
a \mathrel{\dot{\in}} b &:= \exists X(\Re(b, X) \wedge a \in X), \\
a \doteq b &:= \exists X(\Re(a, X) \wedge \Re(b, X)).
\end{aligned}
$$

3.2.5 Bemerkung In EET gilt für alle $\mathbb{L}_p$-Individuenterme a und b

$$a \doteq b \iff \exists X \Re(a, X) \ \wedge\ \exists Y \Re(b, Y) \ \wedge\ \forall x(x \mathrel{\dot{\in}} a \leftrightarrow x \mathrel{\dot{\in}} b).$$

3.2.6 Theorem *Extensionalität für Namen von Typen, d.h. die Aussage*

$$\forall x \forall y(x \doteq y \rightarrow x = y),$$

ist inkonsistent mit EET.

Beweis Wir arbeiten informell in EET und wählen einen beliebigen Typ X. Aufgrund der vorherigen Folgerung gibt es dann zwei verschiedene Namen a und b von X. Daher gilt $a \doteq b$ und $a \neq b$. $\square$

3.3 Formen der Induktion in der expliziten Mathematik

In der expliziten Mathematik unterscheidet man zwischen drei Hauptformen von vollständiger Induktion: Mengeninduktion, Typeninduktion und

Formelinduktion. Daneben gibt es noch weitere Zwischenformen, die wir hier aber nicht betrachten.

Mengeninduktion $(\mathsf{S\text{-}I_N})$.

$$a \in Set_{\mathsf{N}} \;\wedge\; \mathsf{0}\,\varepsilon\, a \;\wedge\; (\forall x \in \mathsf{N})(x\,\varepsilon\, a \to x'\,\varepsilon\, a) \quad\to\quad (\forall x \in \mathsf{N})(x\,\varepsilon\, a).$$

Typeninduktion $(\mathsf{T\text{-}I_N})$.

$$\mathsf{0} \in X \wedge (\forall x \in \mathsf{N})(x \in X \to x' \in X) \;\to\; (\forall x \in \mathsf{N})(x \in X).$$

Formelinduktion $(\mathsf{F\text{-}I_N})$. Für alle $\mathbb{L}_p$-Formeln $A(x)$:

$$A(\mathsf{0}) \wedge (\forall x \in \mathsf{N})(A(x) \to A(x')) \;\to\; (\forall x \in \mathsf{N})(A(x)).$$

Ohne auf die näheren Einzelheiten einzugehen, soll hier nur festgehalten werden, dass die Theorien $\mathsf{EET} + (\mathsf{S\text{-}I_N})$, $\mathsf{EET} + (\mathsf{T\text{-}I_N})$ und $\mathsf{EET} + (\mathsf{F\text{-}I_N})$ in engem Zusammenhang zur primitiv-rekursiven Arithmetik PRA, zur Peano-Arithmetik PA und zum System $(\Pi^0_\infty\text{-}\mathsf{CA})$ der Arithmetik zweiter Stufe mit arithmetischer Komprehension stehen.

3.3.1 Theorem

1. *Die Theorien* $\mathsf{EET} + (\mathsf{S\text{-}I_N})$, $\mathsf{BON} + (\mathsf{S\text{-}I_N})$ *und* PRA *sind beweistheoretisch äquivalent.*

2. *Die Theorien* $\mathsf{EET} + (\mathsf{T\text{-}I_N})$, $\mathsf{BON} + (\mathsf{F\text{-}I_N})$ *und* PA *sind beweistheoretisch äquivalent.*

3. *Die Theorien* $\mathsf{EET} + (\mathsf{F\text{-}I_N})$ *und* $(\Pi^0_\infty\text{-}\mathsf{CA})$ *sind beweistheoretisch äquivalent.*

Die beweistheoretische Äquivalenz von $\mathsf{EET} + (\mathsf{S\text{-}I_N})$ und $\mathsf{BON} + (\mathsf{S\text{-}I_N})$ sowie von $\mathsf{EET} + (\mathsf{T\text{-}I_N})$ und $\mathsf{BON} + (\mathsf{F\text{-}I_N})$ ergibt sich unmittelbar aus Ergebnissen von Abschnitt 3.5.

3.4 $\mathbb{L}_p$-Strukturen

3.4.1 Definition Eine $\mathbb{L}_p$-*Struktur* ist ein 5-Tupel

$$\mathcal{N} = (\mathcal{M}, \mathcal{T}, \mathcal{E}, \mathcal{R}, (m_e : e \in \mathbb{N})),$$

das folgende Bedingungen erfüllt:

1. $\mathcal{M}$ ist eine partielle L_p-Struktur.

2. $\mathcal{T}$ ist eine nichtleere Teilmenge von $P(|\mathcal{M}|)$.

3. $\mathcal{E}$ und $\mathcal{R}$ sind Teilmengen von $|\mathcal{M}| \times \mathcal{T}$.

4. $(m_e : e \in \mathbb{N})$ ist eine Familie von Elementen aus $|\mathcal{M}|$.

Eine $\mathbb{L}_p$-Struktur $\mathcal{N} = (\mathcal{M}, \mathcal{T}, \mathcal{E}, \mathcal{R}, (m_e : e \in \mathbb{N}))$ wird als *Standardstruktur* bezeichnet, falls $\mathcal{E}$ die übliche $\in$-Relation auf $|\mathcal{M}| \times \mathcal{T}$ ist. Wir schreiben dann etwas einfacher $(\mathcal{M}, \mathcal{T}, \mathcal{R}, (m_e : e \in \mathbb{N}))$ für eine solche Standardstruktur.

Ist $\mathcal{N} = (\mathcal{M}, \mathcal{T}, \mathcal{E}, \mathcal{R}, (m_e : e \in \mathbb{N}))$ eine $\mathbb{L}_p$-Struktur, so werden die Mengen $\mathcal{T}, \mathcal{E}$ sowie $\mathcal{R}$ zur Interpretation der Typen, der Elementrelation sowie der Namensrelation verwendet. Für alle natürlichen Zahlen e werden die Konstanten c_e in $\mathcal{N}$ durch m_e interpretiert.

3.4.2 Definition Ist $\mathcal{N} = (\mathcal{M}, \mathcal{T}, \mathcal{E}, \mathcal{R}, (m_e : e \in \mathbb{N}))$ eine $\mathbb{L}_p$-Struktur, so nennen wir eine Abbildung α, die jeder Individuenvariablen v von $\mathbb{L}_p$ ein Element $\alpha(v)$ aus $|\mathcal{M}|$ und jeder Typenvariablen V ein Element $\alpha(V)$ aus $\mathcal{T}$ zuordnet, eine $\mathbb{L}_p$-*Variablenbelegung* in $\mathcal{N}$.

Ist $\mathcal{N} = (\mathcal{M}, \mathcal{T}, \mathcal{E}, \mathcal{R}, (m_e : e \in \mathbb{N}))$ eine $\mathbb{L}_p$-Struktur und α eine $\mathbb{L}_p$-Variablenbelegung in $\mathcal{N}$, so wird der Wert $\mathcal{N}_\alpha(a)$ eines $\mathbb{L}_p$-Individuenterms a wie früher definiert; dabei ist nur zu beachten, dass für alle natürlichen Zahlen e die Konstanten c_e durch m_e interpretiert werden.

3.4.3 Definition Für jede $\mathbb{L}_p$-Struktur $\mathcal{N} = (\mathcal{M}, \mathcal{T}, \mathcal{E}, \mathcal{R}, (m_e : e \in \mathbb{N}))$ und jede $\mathbb{L}_p$-Variablenbelegung α in $\mathcal{N}$ wird der Wert $\mathcal{N}_\alpha(A) \in \{\mathbf{f}, \mathbf{t}\}$ einer $\mathbb{L}_p$-Formel A induktiv definiert durch:

1. Ist A eine atomare $\mathbb{L}_p$-Formel erster Stufe, so wird $\mathcal{N}_\alpha(A)$ analog zu Definition 1.2.4 bestimmt.

2. Ist A von der Form $(a \in X)$, so ist
$$\mathcal{N}_\alpha(A) \; := \; \begin{cases} \mathbf{t}, & \text{falls } (\mathcal{N}_\alpha(a), \alpha(X)) \in \mathcal{E}, \\ \mathbf{f}, & \text{sonst.} \end{cases}$$

3. Ist A von der Form $(X = Y)$, so ist
$$\mathcal{N}_\alpha(A) \; := \; \begin{cases} \mathbf{t}, & \text{falls } \alpha(X) = \alpha(Y), \\ \mathbf{f}, & \text{sonst.} \end{cases}$$

4. Ist A von der Form $\Re(a, X)$, so ist
$$\mathcal{N}_\alpha(A) \; := \; \begin{cases} \mathbf{t}, & \text{falls } (\mathcal{N}_\alpha(a), \alpha(X)) \in \mathcal{R}, \\ \mathbf{f}, & \text{sonst.} \end{cases}$$

5. Ist A von der Form $\neg B, (B \vee C)$ oder $\exists x B$, so wird $\mathcal{N}_\alpha(A)$ analog zu Definition 1.2.4 bestimmt.

6. Ist A von der Form $\exists X B$, so ist
$$\mathcal{N}_\alpha(A) \; := \; \begin{cases} \mathbf{t}, & \text{falls } \mathcal{N}_{\alpha[X=S]}(B) = \mathbf{t} \text{ für ein } S \in \mathcal{T}, \\ \mathbf{f}, & \text{sonst.} \end{cases}$$

Ist $\mathcal{N}$ eine $\mathbb{L}_p$-Struktur und A eine $\mathbb{L}_p$-Formel, so nennen wir A *gültig* in $\mathcal{N}$ und schreiben dafür $\mathcal{N} \models A$, falls $\mathcal{N}_\alpha(A) = \mathbf{t}$ für alle $\mathbb{L}_p$-Variablenbelegungen α in $\mathcal{N}$ gilt. $\mathcal{N}$ heisst *Modell* einer Menge Th von $\mathbb{L}_p$-Formeln, in Zeichen $\mathcal{N} \models \mathsf{Th}$, falls alle Formeln aus Th in $\mathcal{N}$ gültig sind.

3.5 Standardmodelle von EET

Wir beschreiben nun ein allgemeines Verfahren, das es uns ermöglicht, Modelle von BON zu Standardmodellen von EET zu erweitern.

Sei also $\mathcal{M}$ eine partielle L_p-Struktur, die ein Modell von BON ist. Zuerst ordnen wir jeder Konstanten c_e einen Wert $\mathsf{c}_e^{\mathcal{M}}$ aus $|\mathcal{M}|$ zu, indem wir für alle natürlichen Zahlen e

$$\mathsf{c}_e^{\mathcal{M}} := \text{ Wert des Terms } (\lambda x.(\mathsf{0}, \overline{e}, x)) \text{ in } \mathcal{M}$$

setzen; dabei sei $\overline{e}$ das Numeral für die natürliche Zahl e. In Ausdrücken der Form $\mathsf{c}_e^{\mathcal{M}}(\vec{m})$ sind für $\vec{m} \in |\mathcal{M}|$ die (iterierte) Paarbildung und die Anwendungsoperation im Sinne von $\mathcal{M}$ zu verstehen. Damit gilt

$$\mathsf{c}_e^{\mathcal{M}}(\vec{m}) = (\mathsf{0}, \overline{e}, (\vec{m}))$$

für alle natürlichen Zahlen e und alle $\vec{m} \in |\mathcal{M}|$ in $\mathcal{M}$. In einem nächsten Schritt definieren wir durch Induktion über die natürliche Zahl k Teilmengen $R_k^{\mathcal{M}}$ von $|\mathcal{M}|$. Simultan dazu führen wir für jedes m aus $R_k^{\mathcal{M}}$ eine Teilmenge $ext^{\mathcal{M}}(m)$ von $|\mathcal{M}|$ ein und setzen

$$\mathcal{T}_k^{\mathcal{M}} := \{ext^{\mathcal{M}}(m) : m \in R_k^{\mathcal{M}}\} \quad \text{und} \quad \mathcal{M}_k^* := (\mathcal{M}, \mathcal{T}_k^{\mathcal{M}}, \emptyset, (\mathsf{c}_e^{\mathcal{M}} : e \in \mathbb{N}))$$

$k = 0$. Für jede L_p-Formel $A[x, \vec{y}]$ mit Gödelnummer e und für alle $\vec{n} \in |\mathcal{M}|$ sei $\mathsf{c}_e^{\mathcal{M}}(\vec{n})$ aus $R_0^{\mathcal{M}}$, ausserdem gelte

$$ext^{\mathcal{M}}(\mathsf{c}_e^{\mathcal{M}}(\vec{n})) \;:=\; \{m \in |\mathcal{M}| : \mathcal{M} \models A[m, \vec{n}]\}.$$

$k > 0$. $R_k^{\mathcal{M}}$ enthalte $R_{k-1}^{\mathcal{M}}$. Ist $A[x, \vec{y}, \vec{Z}]$ eine elementare $\mathbb{L}_p$-Formel mit Gödelnummer e und sind $\vec{n} \in |\mathcal{M}|$ sowie $\vec{q} \in R_{k-1}^{\mathcal{M}}$, so sei $\mathsf{c}_e^{\mathcal{M}}(\vec{n}, \vec{q})$ ebenfalls ein Element aus $R_k^{\mathcal{M}}$; ausserdem gelte

$$ext^{\mathcal{M}}(\mathsf{c}_e^{\mathcal{M}}(\vec{n}, \vec{q})) \;:=\; \{m \in |\mathcal{M}| : \mathcal{M}_{k-1}^* \models A[m, \vec{n}, ext^{\mathcal{M}}(\vec{q})]\}.$$

Dabei sei $ext^{\mathcal{M}}(\vec{q})$ eine abkürzende Schreibweise für $ext^{\mathcal{M}}(q_1), \dots, ext^{\mathcal{M}}(q_l)$, falls $\vec{q} = q_1, \dots, q_l$ ist.

Ausgehend von diesen Definitionen führen wir nun die Mengen $\mathcal{T}^{\mathcal{M}}$ und $\mathcal{R}^{\mathcal{M}}$ ein, die zur Interpretation der Typen und der Namensrelation verwendet werden:

$$\mathcal{T}^{\mathcal{M}} := \bigcup_{k \in \mathbb{N}} \mathcal{T}_k^{\mathcal{M}} \quad \text{und} \quad \mathcal{R}^{\mathcal{M}} := \bigcup_{k \in \mathbb{N}} \{(m, ext^{\mathcal{M}}(m)) : m \in R_k^{\mathcal{M}}\}.$$

Die gewünschte $\mathbb{L}_p$-Standardstruktur ist schliesslich gegeben durch

$$\mathcal{M}^* := (\mathcal{M}, \mathcal{T}^{\mathcal{M}}, \mathcal{R}^{\mathcal{M}}, (\mathsf{c}_e^{\mathcal{M}} : e \in \mathbb{N})).$$

3.5.1 Theorem *Ist $\mathcal{M}$ ein Modell von* BON*, so ist $\mathcal{M}^*$ ein Modell von* EET*, das folgende Eigenschaften besitzt:*

1. *$\mathcal{M}^*$ ist eine konservative Erweiterung von $\mathcal{M}$ in dem Sinne, dass für alle geschlossenen L_p-Formeln A gilt:*

$$\mathcal{M}^* \models A \iff \mathcal{M} \models A.$$

2. *Ist $\mathcal{M}$ ein Modell von* (F-$\mathsf{I_N}$) *bezüglich L_p, dann ist $\mathcal{M}^*$ ein Modell von* (T-$\mathsf{I_N}$) *bezüglich $\mathbb{L}_p$.*

Beweis Durch genaues Betrachten der eben beschriebenen Konstruktion von $\mathcal{M}^*$ kann man sehr leicht nachprüfen, dass $\mathcal{M}^*$ ein Modell von EET mit den angegebenen Eigenschaften ist. □

3.5.2 Folgerung

1. EET+(S-$\mathsf{I_N}$) ist eine konservative Erweiterung von BON+(S-$\mathsf{I_N}$) bezüglich aller geschlossenen L_p-Formeln.

2. $\mathsf{EET}+(\mathsf{T}\text{-}\mathsf{I_N})$ ist eine konservative Erweiterung von $\mathsf{BON}+(\mathsf{F}\text{-}\mathsf{I_N})$ bezüglich aller geschlossenen L_p-Formeln.

Beweis Beide Behauptungen folgen unmittelbar aus dem vorhergehenden Theorem. Zum Beweis der ersten Behauptung muss man ausserdem beachten, dass $\mathcal{M}^*$ genau dann ein Modell von $(\mathsf{S}\text{-}\mathsf{I_N})$ ist, wenn $\mathcal{M}$ ein Modell von $(\mathsf{S}\text{-}\mathsf{I_N})$ ist. $\square$

3.5.3 Theorem *$\mathcal{PRF}^*$ und $\mathcal{G}^*$ sind Modelle von* $\mathsf{EET}+(\mathsf{F}\text{-}\mathsf{I_N})$.

Beweis Aufgrund von Theorem 2.5.1, Theorem 2.6.1 und Theorem 3.5.1 sind $\mathcal{PRF}^*$ und $\mathcal{G}^*$ Modelle von EET. Betrachtet man die Interpretation der natürlichen Zahlen in diesen Modellen, so sieht man sofort, dass auch $(\mathsf{F}\text{-}\mathsf{I_N})$ erfüllt ist. $\square$

3.6 Die Hierarchie endlicher Typen

In $\mathsf{EET}+(\mathsf{F}\text{-}\mathsf{I_N})$ kann ohne grossen Aufwand die übliche Hierarchie der endlichen Typen definiert werden. Ebenso kann man auch die extensionale Version dieser Typenhierarchie einführen.

Zuerst führen wir dazu Codes für die Symbole endlicher Typen (Finite Type Symbols) ein, und zwar so, dass $(\mathsf{0},\mathsf{0})$ das Symbol 0 codiert und $(\mathsf{1},(a,b))$ beziehungsweise $(\mathsf{2},(a,b))$ für $(a\times b)$ beziehungsweise $(a\rightarrow b)$ stehen. Wir betrachten nun einige Definitionen, in denen das Symbol $<$ für die übliche Kleinerrelation auf den natürlichen Zahlen verwendet wird, entsprechend steht $x\leq y$ für $(x<y\vee x=y)$. Nun setzen wir

$$A_{\mathrm{FTS}}(z, z_0, z_1, f) :=$$
$$z_0 < z \;\wedge\; z_1 < z \;\wedge\; [f(z) = (1, (f(z_0), f(z_1))) \vee f(z) = (2, (f(z_0), f(z_1)))],$$

$$B_{\mathrm{FTS}}(y, f) := (\forall z \in \mathsf{N})[z \leq y \;\rightarrow\; f(z) = (0, 0) \vee \exists z_0 \exists z_1 A_{\mathrm{FTS}}(z, z_0, z_1, f)],$$

$$C_{\mathrm{FTS}}(x, y, f) := y \in \mathsf{N} \;\wedge\; x = f(y) \;\wedge\; B_{\mathrm{FTS}}(y, f).$$

Damit ist $C_{\mathrm{FTS}}(x, y, f)$ eine elementare $\mathbb{L}_p$-Formel. Daher kann man in EET mittels elementarer Komprehension den folgenden Typ FTS einführen, der genau die Codes für die Finite Type Symbols enthält;

$$\mathrm{FTS} \;:=\; \{x : \exists y \exists f C_{\mathrm{FTS}}(x, y, f)\}.$$

Ehe wir als nächstes die Hierarchie der endlichen Typen einführen, bemerken wir, dass es wegen der Uniformität der elementaren Komprehension geschlossene $\mathbb{L}_p$-Individuenterme $\underline{\mathrm{nat}}$, $\underline{\mathrm{prod}}$ und $\underline{\mathrm{imp}}$ gibt, so dass in EET die folgenden drei Eigenschaften gelten:

(i) $$\Re(\underline{\mathrm{nat}}, \{x : \mathsf{N}(x)\}).$$

(ii) $$\Re(x, X) \wedge \Re(y, Y) \;\rightarrow\; \Re(\underline{\mathrm{prod}}(x, y), X \times Y).$$

(iii) $$\Re(x, X) \wedge \Re(y, Y) \;\rightarrow\; \Re(\underline{\mathrm{imp}}(x, y), (X \rightarrow Y)).$$

Um die folgende Definition von $\underline{\mathrm{fth}}$ etwas übersichtlicher zu gestalten, führen wir die geschlossenen $\mathbb{L}_p$-Individuenterme $\mathrm{t}_{\mathrm{prod}}$ und $\mathrm{t}_{\mathrm{imp}}$ als Abkürzungen ein. Wir schreiben

$$\begin{aligned} \mathrm{t}_{\mathrm{prod}} &:= (\lambda f u.\underline{\mathrm{prod}}(f(\mathsf{p}_0(\mathsf{p}_1 u)), f(\mathsf{p}_1(\mathsf{p}_1 u)))), \\ \mathrm{t}_{\mathrm{imp}} &:= (\lambda f u.\underline{\mathrm{imp}}(f(\mathsf{p}_0(\mathsf{p}_1 u)), f(\mathsf{p}_1(\mathsf{p}_1 u)))) \end{aligned}$$

und erhalten in EET für alle x, y und z:

(iv) $$u = (x, (y, z)) \;\rightarrow\; \mathrm{t}_{\mathrm{prod}} f u \simeq \underline{\mathrm{prod}}(fy, fz).$$

(v) $$u = (x, (y, z)) \;\rightarrow\; \mathrm{t}_{\mathrm{imp}} f u \simeq \underline{\mathrm{imp}}(fy, fz).$$

Mit Hilfe des Rekursionstheorems lässt sich nun ein geschlossener $\mathbb{L}_p$-Term $\underline{\mathrm{fth}}$ einführen, der für alle u die Rekursionsgleichung

$$\underline{\mathrm{fth}}\, u \simeq \mathsf{d_N}\underline{\mathrm{nat}}(\,\mathsf{d_N}(\mathrm{t_{prod}}\underline{\mathrm{fth}}\, u)(\mathrm{t_{imp}}\underline{\mathrm{fth}}\, u)(\mathsf{p_0}u)\mathsf{1},)(\mathsf{p_0}u)\mathsf{0}$$

erfüllt. Dann gilt in EET:

(vi) $$u = (\mathsf{0}, \mathsf{0}) \;\rightarrow\; \underline{\mathrm{fth}}\, u = \underline{\mathrm{nat}}.$$

(vii) $$u = (\mathsf{1}, (x, y)) \;\rightarrow\; \underline{\mathrm{fth}}\, u = \underline{\mathrm{prod}}(\underline{\mathrm{fth}}\, x, \underline{\mathrm{fth}}\, y).$$

(viii) $$u = (\mathsf{2}, (x, y)) \;\rightarrow\; \underline{\mathrm{fth}}\, u = \underline{\mathrm{imp}}(\underline{\mathrm{fth}}\, x, \underline{\mathrm{fth}}\, y).$$

In $\mathsf{EET} + (\mathsf{F\text{-}I_N})$ wird im nächsten Schritt mit Hilfe von Formelinduktion gezeigt, dass $\underline{\mathrm{fth}}\, u$ für jedes Finite Type Symbol u Name eines Typs ist; $\mathsf{EET} + (\mathsf{F\text{-}I_N})$ beweist also

$$(\forall u \in \mathrm{FTS})\exists X \Re(\underline{\mathrm{fth}}\, u, X).$$

Typeninduktion $(\mathsf{T\text{-}I_N})$ reicht zum Beweis dieser Behauptung nicht aus. Ist u ein Element von FTS, so bezeichnen wir von nun an den durch $\underline{\mathrm{fth}}\, u$ eindeutig bestimmten Typ mit N_u. Folglich gilt für alle u und v aus FTS:

$$\mathsf{N_0} = \mathsf{N}, \quad \mathsf{N}_{(1,(u,v))} = \mathsf{N}_u \times \mathsf{N}_v \quad \text{und} \quad \mathsf{N}_{(2,(u,v))} = (\mathsf{N}_u \rightarrow \mathsf{N}_v).$$

Das bedeutet, dass die Familie $(\mathsf{N}_u : u \in \mathrm{FTS})$ der üblichen Hierarchie der endlichen Typen entspricht.

Um die extensionale Version der Hierarchie der endlichen Typen aufzubauen, gehen wir ähnlich vor. Allerdings definieren wir nun für jedes u aus FTS einen Typ M_u und eine Gleichheitsrelation $=_u$ auf M_u, die ein Teiltyp von $\mathsf{M}_u \times \mathsf{M}_u$ ist. Dabei müssen für alle u und v aus FTS die folgenden Bedingungen (1) – (3) erfüllt sein.

(1) $\mathsf{M}_0 = \mathsf{N}$ und für alle $x, y \in \mathsf{M}_0$ gilt

$$x =_0 y \;\leftrightarrow\; x = y.$$

(2) $\mathsf{M}_{(1,(u,v))} = \mathsf{M}_u \times \mathsf{M}_v$ und für alle $x, y \in \mathsf{M}_{(1,(u,v))}$ gilt

$$x =_{(1,(u,v))} y \quad\leftrightarrow\quad \mathsf{p}_0 x =_u \mathsf{p}_0 y \;\wedge\; \mathsf{p}_1 x =_v \mathsf{p}_1 y.$$

(3) $\mathsf{M}_{(2,(u,v))} = \{f : f \in (\mathsf{M}_u \to \mathsf{M}_v) \wedge (\forall x, y \in \mathsf{M}_u)(x =_u y \to fx =_v fy)\}$ und für alle $f, g \in \mathsf{M}_{(2,(u,v))}$ gilt

$$f =_{(2,(u,v))} g \;\leftrightarrow\; (\forall x \in \mathsf{M}_u)(fx =_v gx).$$

3.7 Die Theorie SET

Ersetzen wir in EET das Schema der elementaren Komprehension durch das folgende Schema der stratifizierten Komprehension, so erhalten wir die Theorie SET der stratifizierten expliziten Typen. SET ist bedeutend stärker als EET und lässt unter anderem die Bildung (stark) imprädikativer Typen zu.

Stratifizierte Komprehension Es sei $A[x, \vec{y}, \vec{Z}]$ eine stratifizierte $\mathbb{L}_p$-Formel mit Gödelnummer e. Dann haben wir als Axiome:

(SCA.1) $$\exists X \forall x (x \in X \leftrightarrow A[x, \vec{u}, \vec{V}]).$$

(SCA.2) $$\Re(\vec{v}, \vec{V}) \wedge \forall x (x \in X \leftrightarrow A[x, \vec{u}, \vec{V}]) \;\to\; \Re(\mathsf{c}_e(\vec{u}, \vec{v}), X).$$

Die Konstanten c_e werden in SET also zur Konstruktion von Namen für Typen, die durch stratifizierte Komprehension gebildet werden, verwendet. Wie bei EET ist die Namensgebung uniform in den Individuen- und Typenparametern der definierenden Formel.

Es lässt sich leicht zeigen, dass das System PA_2 der Arithmetik zweiter Stufe mit voller Komprehension in $\mathsf{SET} + (\mathsf{T}\text{-}\mathsf{I_N})$ eingebettet werden kann. Aus einem Ergebnis von Glass [12] folgt ausserdem, dass $\mathsf{SET} + (\mathsf{F}\text{-}\mathsf{I_N})$ zu PA_2 beweistheoretisch äquivalent ist. Die Überlegungen des nächsten Abschnittes zeigen schliesslich, dass $\mathsf{SET} + (\mathsf{S}\text{-}\mathsf{I_N})$ und $\mathsf{BON} + (\mathsf{S}\text{-}\mathsf{I_N})$ dieselbe beweistheoretische Stärke besitzen.

3.7.1 Theorem

1. *Die Theorien* $\mathsf{SET} + (\mathsf{S}\text{-}\mathsf{I_N})$, $\mathsf{EET} + (\mathsf{S}\text{-}\mathsf{I_N})$, $\mathsf{BON} + (\mathsf{S}\text{-}\mathsf{I_N})$ *und* PRA *sind beweistheoretisch äquivalent.*

2. *Die Theorien* $\mathsf{SET} + (\mathsf{F}\text{-}\mathsf{I_N})$, $\mathsf{SET} + (\mathsf{T}\text{-}\mathsf{I_N})$ *und* PA_2 *sind beweistheoretisch äquivalent.*

3.8 Standardmodelle von SET

Durch eine geeignete Modifikation kann das Verfahren von Abschnitt 3.5 so erweitert werden, dass wir aus Modellen von BON Modelle von SET erhalten.

Gegeben sei wiederum eine partielle L_p-Struktur $\mathcal{M}$, die ein Modell von BON ist. Als Hilfsausdrücke führen wir zuerst für alle natürlichen Zahlen e Elemente $\mathsf{d}_e^{\mathcal{M}}$ aus $|\mathcal{M}|$ ein, die definiert sind durch

$$\mathsf{d}_e^{\mathcal{M}} \ := \ \text{Wert des Terms } (\lambda x.(\mathsf{1}, \overline{e}, x)) \text{ in } \mathcal{M}.$$

Ferner können wir annehmen, dass es eine totale Funktion φ gibt, die der Gödelnummer e einer $\mathbb{L}_p$-Formel $A[x, \vec{y}, \vec{Z}]$ die Gödelnummer $\varphi(e)$ der $\mathbb{L}_p$-Formel

$$\forall x(x \in Vj \leftrightarrow A[x, \vec{y}, \vec{Z}])$$

zuordnet, wobei j der Eindeutigkeit halber der kleinste Index sei, so dass Vj in $A[x, \vec{y}, \vec{Z}]$ nicht vorkommt. Mit Hilfe von $(\mathsf{d}_e^{\mathcal{M}} : e \in \mathbb{N})$ und φ definieren wir die Familie $(\mathsf{c}_e^{\mathcal{M}} : e \in \mathbb{N})$, indem wir für alle natürlichen Zahlen

$$\mathsf{c}_e^{\mathcal{M}} := \mathsf{d}_{\varphi(e)}^{\mathcal{M}}$$

setzen. In Ausdrücken der Form $\mathsf{c}_e^{\mathcal{M}}(\vec{m})$ und $\mathsf{d}_e^{\mathcal{M}}(\vec{m})$ sind für $\vec{m} \in |\mathcal{M}|$ die (iterierte) Paarbildung und die Anwendungsoperation im Sinne von $\mathcal{M}$ zu verstehen.

Ehe wir unser Modell von SET einführen, betrachten wir die $\mathbb{L}_p$-Struktur

$$\mathcal{M}^{(2)} := (\mathcal{M}, P(|\mathcal{M}|), \emptyset, (\mathsf{c}_e^{\mathcal{M}} : e \in \mathbb{N})).$$

Dann wählen wir für jede stratifizierte $\mathbb{L}_p$-Formel $A[\vec{y}, \vec{Z}, X]$ mit Gödelnummer e eine Skolemfunktion $\mathcal{F}_e$, so dass

$$\mathcal{M}^{(2)} \models \exists X A[\vec{m}, \vec{Q}, X] \to A[\vec{m}, \vec{Q}, \mathcal{F}_e(\vec{m}, \vec{Q})] \qquad (*)$$

für alle $\vec{m} \in |\mathcal{M}|$ und $\vec{Q} \in P(|\mathcal{M}|)$ gilt. Analog zu früher führen wir jetzt durch Induktion über die natürliche Zahl k Teilmengen $R_k^{\mathcal{M}}$ von $|\mathcal{M}|$ und simultan dazu für jedes $m \in R_k^{\mathcal{M}}$ eine Teilmenge $ext^{\mathcal{M}}(m)$ von $|\mathcal{M}|$ ein.

$k = 0$. Für jede stratifizierte $\mathbb{L}_p$-Formel $A[\vec{y}, X]$ mit Gödelnummer e und für alle $\vec{m} \in |\mathcal{M}|$ sei $\mathsf{d}_e^{\mathcal{M}}(\vec{m})$ aus $R_0^{\mathcal{M}}$; aussserdem gelte

$$ext^{\mathcal{M}}(\mathsf{d}_e^{\mathcal{M}}(\vec{m})) := \mathcal{F}_e(\vec{m}).$$

$k > 0$. $R_k^{\mathcal{M}}$ enthalte $R_{k-1}^{\mathcal{M}}$. Ist $A[\vec{y}, \vec{Z}, X]$ eine stratifizierte $\mathbb{L}_p$-Formel mit Gödelnummer e und sind $\vec{m} \in |\mathcal{M}|$ und $\vec{q} \in R_{k-1}^{\mathcal{M}}$, so sei $\mathsf{d}_e^{\mathcal{M}}(\vec{m}, \vec{q})$ ebenfalls ein Element von $R_k^{\mathcal{M}}$; ausserdem gelte

$$ext^{\mathcal{M}}(\mathsf{d}_e^{\mathcal{M}}(\vec{m}, \vec{q})) := \mathcal{F}_e(\vec{m}, ext^{\mathcal{M}}(\vec{q})).$$

Dabei sei $ext^{\mathcal{M}}(\vec{q})$ eine abkürzende Schreibweise für $ext^{\mathcal{M}}(p_1), \dots, ext^{\mathcal{M}}(q_l)$, falls $\vec{q} = q_1, \dots, q_l$ ist.

Ausgehend von diesen Definitionen führen wir nun die Interpretationen $\mathcal{T}^{\mathcal{M}}$ und $\mathcal{R}^{\mathcal{M}}$ der Typen und Namensrelation ein:

$$\begin{aligned} \mathcal{T}^{\mathcal{M}} &:= \bigcup_{k \in \mathbb{N}} \{ext^{\mathcal{M}}(m) : m \in R_k^{\mathcal{M}}\}, \\ \mathcal{R}^{\mathcal{M}} &:= \bigcup_{k \in \mathbb{N}} \{(m, ext^{\mathcal{M}}(m)) : m \in R_k^{\mathcal{M}}\}. \end{aligned}$$

Damit lässt sich unsere gewünschte $\mathbb{L}_p$-Struktur $\mathcal{M}^\sharp$ definieren als

$$\mathcal{M}^\sharp := (\mathcal{M}, \mathcal{T}^{\mathcal{M}}, \mathcal{R}^{\mathcal{M}}, (\mathsf{c}_e^{\mathcal{M}} : e \in \mathbb{N})).$$

3.8.1 Lemma *Ist $\mathcal{M}$ ein Modell von* BON, *so gilt für alle stratifizierten $\mathbb{L}_p$-Formeln $A[\vec{x}, \vec{Y}]$ sowie alle $\vec{m} \in |\mathcal{M}|$ und $\vec{Q} \in \mathcal{T}^{\mathcal{M}}$:*

$$\mathcal{M}^{(2)} \models A[\vec{m}, \vec{Q}] \iff \mathcal{M}^\sharp \models A[\vec{m}, \vec{Q}].$$

Beweis Dieses Lemma wird durch Induktion nach dem Aufbau der Formel $A[\vec{x}, \vec{Y}]$ gezeigt. Dabei ist der interessante Fall der, dass $A[\vec{x}, \vec{Y}]$ von der Form $\exists Z B[\vec{x}, \vec{Y}, Z]$ ist und $\mathcal{M}^{(2)} \models A[\vec{m}, \vec{Q}]$ gilt.

Wegen $\vec{Q} \in \mathcal{T}^{\mathcal{M}}$ gibt es dann eine natürliche Zahl k und $\vec{q} \in R_k^{\mathcal{M}}$, so dass $ext^{\mathcal{M}}(\vec{q}) = \vec{Q}$ ist. Ist e die Gödelnummer von $B[\vec{x}, \vec{Y}, Z]$, so gilt nach $(*)$

$$\mathcal{M}^{(2)} \models B[\vec{m}, \vec{Q}, \mathcal{F}_e(\vec{m}, \vec{Q})].$$

Ferner ist $\mathsf{d}_e^{\mathcal{M}}(\vec{m}, \vec{q}) \in R_{k+1}^{\mathcal{M}}$ und $\mathcal{F}_e(\vec{m}, \vec{Q}) = ext^{\mathcal{M}}(\mathsf{d}_e^{\mathcal{M}}(\vec{m}, \vec{q}))$. Daraus folgt $\mathcal{F}_e(\vec{m}, \vec{Q}) \in \mathcal{T}^{\mathcal{M}}$, so dass wir mit der Induktionsvoraussetzung auch

$$\mathcal{M}^\sharp \models B[\vec{m}, \vec{Q}, \mathcal{F}_e(\vec{m}, \vec{Q})]$$

erhalten. Dies ergibt aber sofort $\mathcal{M}^\sharp \models A[\vec{m}, \vec{Q}]$. $\square$

3.8.2 Theorem *Ist $\mathcal{M}$ ein Modell von* BON*, so ist $\mathcal{M}^\sharp$ ein Modell von* SET*. Ausserdem gilt für alle geschlossenen L_p-Formeln A:*

$$\mathcal{M}^\sharp \models A \iff \mathcal{M} \models A.$$

Beweis Aufgrund der früheren Überlegungen müssen wir nur noch das Schema der stratifizierten Komprehension betrachten. Sei also $A[x, \vec{y}, \vec{Z}]$ eine stratifizierte $\mathbb{L}_p$-Formel mit Gödelnummer e und

$$B[\vec{y}, \vec{Z}, X] := \forall x(x \in X \leftrightarrow A[x, \vec{y}, \vec{Z}])$$

die dazu passende stratifizierte $\mathbb{L}_p$-Formel mit Gödelnummer $\varphi(e)$. Seien ausserdem $\vec{m} \in |\mathcal{M}|, k \in \mathbb{N}$ und $\vec{q} \in R_k^{\mathcal{M}}$. Dann gilt

$$\mathcal{M}^{(2)} \models \exists X B[\vec{m}, ext^{\mathcal{M}}(\vec{q}), X].$$

Aufgrund von $(*)$ und den Definitionen im Zusammenhang mit dem Aufbau von $\mathcal{M}^\sharp$ erhalten wir

$$\mathcal{M}^{(2)} \models B[\vec{m}, ext^{\mathcal{M}}(\vec{q}), \mathcal{F}_{\varphi(e)}(\vec{m}, ext^{\mathcal{M}}(\vec{q}))]$$

sowie $\mathsf{d}_{\varphi(e)}^{\mathcal{M}}(\vec{m}, \vec{q}) \in F_{k+1}^{\mathcal{M}}$ und $\mathcal{F}_{\varphi(e)}(\vec{m}, ext^{\mathcal{M}}(\vec{q})) = ext^{\mathcal{M}}(\mathsf{d}_{\varphi(e)}^{\mathcal{M}}(\vec{m}, \vec{q}))$. Wegen $\mathsf{c}_e^{\mathcal{M}} = \mathsf{d}_{\varphi(e)}^{\mathcal{M}}$ und der Definition von $B[\vec{y}, \vec{Z}, X]$ ergibt sich also

$$\mathcal{M}^{(2)} \models \forall x(x \in ext^{\mathcal{M}}(\mathsf{c}_e^{\mathcal{M}}(\vec{m}, \vec{q})) \leftrightarrow A[x, \vec{m}, ext^{\mathcal{M}}(\vec{q})]).$$

Nun wenden wir das vorhergehende Lemma an und erhalten daraus, dass in $\mathcal{M}^\sharp$ für $A[x, \vec{y}, \vec{Z}]$ die Axiome (SCA.1) und (SCA.2) erfüllt sind. $\square$

3.9 Polymorphismus in expliziter Mathematik

In diesem Abschnitt möchten wir andeuten, wie polymorphe Typenstrukturen, die sich heute in der Informatik grosser Beliebtheit erfreuen, im Rahmen der expliziten Mathematik behandelt werden können. Der polymorphe

Lambdakalkül geht auf Girard [9] und Reynolds [16] zurück und lässt im Gegensatz zum üblichen getypten Lambdakalkül auch die Abstraktion über Typenvariablen zu.

Gegeben seien Basistypen $\beta_1, \beta_2, \ldots, \beta_k$ und abzählbar unendlich viele Typenvariablen $V0, V1, \ldots, Vi, \ldots$ $(i \in \mathbb{N})$. Die Kollektion der polymorphen Typen wird induktiv erzeugt durch:

1. Jeder Basistyp und jede Typenvariable ist ein polymorpher Typ.

2. Sind S und T polymorphe Typen, so ist $(S \to T)$ ein polymorpher Typ.

3. Ist T ein polymorpher Typ und ist X eine Typenvariable, so ist $(\Pi X.T)$ ein polymorpher Typ.

Diese mächtige Typenstruktur wird auch bei der Termbildung reflektiert. Der polymorphe Lambdakalkül $\mathsf{P\lambda C}$ umfasst eine gewisse Menge von Konstanten verschiedener Typen sowie abzählbar unendlich viele Variablen zu jedem polymorphen Typ T. Dabei nehmen wir zur Vereinfachung der Notation an, dass für jede Variable x^S vom Typ S auch x eine Individuenvariable von $\mathbb{L}_p$ ist. Ausserdem sollen x^S und y^T nur dann zur selben $\mathbb{L}_p$-Individuenvariablen korrespondieren, falls x und y sowie S und T identisch sind. Die Kollektion der Terme von $\mathsf{P\lambda C}$ wird induktiv erzeugt durch:

1. Jede Konstante und jede Variable vom Typ T ist ein Term vom Typ T.

2. Ist t ein Term vom Typ $(S \to T)$ und s ein Term vom Typ S, so ist $(t \circ s)$ ein Term vom Typ T.

3. Ist t ein Term vom Typ T und x^S eine Variable vom Typ S, so ist $(\lambda x^S.t)$ ein Term vom Typ $(S \to T)$.

4. Ist t ein Term vom Typ $(\Pi X.T)$ und S ein Typ, so ist $(t \bullet S)$ ein Term vom Typ $T[S/X]$.

5. Ist t ein Term vom Typ T und X eine Typenvariable, so ist $(\Lambda X.t)$ ein Term vom Typ $(\Pi X.T)$, falls X nicht im Typ einer freien Variablen von t frei vorkommt.

Ist s ein Term vom Typ S und t ein Term vom Typ T, so hat $\mathsf{P\lambda C}$ wie der übliche getypte Lambdakalkül die Reduktionsregel

$$((\lambda x^S.t) \circ s) \rhd t[s/x^S].$$

Dazu kommt aber in $\mathsf{P\lambda C}$ noch ein Reduktionsschema, das

$$((\Lambda X.t) \bullet S) \rhd t[S/X]$$

für beliebige (getypte) Terme t und Typen S fordert. Reduktionen, Normalformen und ähnliches lassen sich wie üblich definieren.

Aufgrund von Ergebnissen von Girard wissen wir, dass die Terme des polymorphen Lambdakalküls $\mathsf{P\lambda C}$ (stark) normalisiert werden können. Ausserdem hat er gezeigt, dass die zahlentheoretischen Funktionen, die in $\mathsf{P\lambda C}$ repräsentiert werden können, genau den in PA_2 beweisbar totalen rekursiven Funktionen entsprechen.

Es ist auch möglich, basierend auf $\mathsf{P\lambda C}$ sogenannte Type Assignment Calculi einzuführen. Wir verzichten jedoch darauf, dies hier näher auszuführen. Für weitere Einzelheiten über polymorphe Typensysteme sei auf die Fachliteratur verwiesen, z.B. [10, 11].

3.9.1 Beispiel

1. Es sei $\mathrm{Id} := (\Lambda Y.(\lambda x^Y.x^Y))$. Dann ist Id vom Typ $(\Pi Y.(Y \to Y))$ und repräsentiert die uniforme Identitätsfunktion. Ist S ein beliebiger polymorpher Typ, so gilt nämlich

$$(\mathrm{Id} \bullet S) \;\rhd\; (\lambda x^S.x^S),$$

und $(\lambda x^S.x^S)$ ist die Identität auf S.

2. Es sei $\mathrm{Ap} := (\Lambda X.(\Lambda Y.(\lambda y^{(X\to Y)}.(\lambda x^X.(y^{(X\to Y)} \circ x^X)))))$. Dann ist Ap vom Typ $(\Pi X.(\Pi Y.((X \to Y) \to (X \to Y))))$ und repräsentiert die allgemeine Anwendungsoperation. Sind S und T beliebige polymorphe Typen, so gilt:
   (i) $((\mathrm{Ap} \bullet S) \bullet T)$ ist vom Typ $((S \to T) \to (S \to T))$.
   (ii) $((\mathrm{Ap} \bullet S) \bullet T) \ \rhd \ (\lambda y^{(S\to T)}.(\lambda x^S.(y^{(S\to T)} \circ x^S)))$.

   Wenden wir nun $((\mathrm{Ap} \bullet S) \bullet T)$ auf einen Term t vom Typ $(S \to T)$ und das Ergebnis auf einen Term s vom Typ S an, so erhalten wir

$$(((\mathrm{Ap} \bullet S) \bullet T) \circ t) \rhd (\lambda x^S(t \circ x^S)) \text{ und } ((((\mathrm{Ap} \bullet S) \bullet T) \circ t) \circ s) \rhd (t \circ s).$$

Um $\mathsf{P\lambda C}$ in SET einzubetten, machen wir von einer *Forgetful Interpretation* in der Art von Girard und Troelstra (vgl. z.B. [19]) Gebrauch. Um dies möglichst einfach darstellen zu können, nehmen wir an, dass wir in $\mathsf{P\lambda C}$ nur einen Basistyp β für die natürlichen Zahlen haben. Ausserdem sollen in $\mathsf{P\lambda C}$ keine Termkonstanten vorkommen.

Zuerst ordnen wir jedem Typ $T(\vec{X})$, in dem höchstens die Variablen $\vec{X}$ frei vorkommen, eine stratifizierte $\mathbb{L}_p$-Formel $F_T(x, \vec{X})$ mit denselben freien Typenvariablen und einer zusätzlichen freien $\mathbb{L}_p$-Individuenvariablen x zu. Diese Zuweisung wird durch folgende Induktion nach Typenaufbau durchgeführt:

1. Ist $T(\vec{X})$ der Typ β, so ist $F_T(x) := \mathsf{N}(x)$.

2. Ist $T(\vec{X})$ eine Typenvariable X, so ist $F_T(x, X) := (x \in X)$.

3. Ist $T(\vec{X})$ von der Form $(S_0(\vec{X}) \to S_1(\vec{X}))$, so ist

$$F_T(x, \vec{X}) := \forall y(F_{S_0}(y, \vec{X}) \to F_{S_1}(xy, \vec{X})).$$

4. Ist $T(\vec{X})$ von der Form $(\Pi Y.S(\vec{X}, Y))$, so ist $F_T(x) := \forall Y F_S(x, \vec{X}, Y)$.

3.9.2 Bemerkung Ist $T(\vec{X})$ ein polymorpher Typ von $\mathsf{P\lambda C}$, so ist $F_T(x, \vec{X})$ eine stratifizierte $\mathbb{L}_p$-Formel. Folglich wird dadurch in SET ein Typ definiert.

Im nächsten Schritt übersetzen wir jeden Term t von $\mathsf{P\lambda C}$ in einen $\mathbb{L}_p$-Individuenterm t^*, indem wir bei t sämtliche Typeninformationen streichen. Diese Übersetzung wird durch folgende Induktion nach Termaufbau durchgeführt:

1. Ist t die getypte Variable x^T, so ist $t^* := x$.

2. Ist t von der Form $(s_0 \circ s_1)$, so ist $t^* := (s_0^* s_1^*)$.

3. Ist t von der Form $(\lambda x^T.s)$, so ist $t^* := (\lambda x.s^*)$.

4. Ist t von der Form $(s \bullet T)$, so ist $t^* := s^*$.

5. Ist t von der Form $(\Lambda X.s)$, so ist $t^* := s^*$.

3.9.3 Bemerkung Die Übersetzung t^* eines Terms t von $\mathsf{P\lambda C}$ enthält keine Typenvariablen.

3.9.4 Beispiel

1. Wir betrachten zuerst den Term Id aus Beispiel 3.9.1, der vom Typ $(\Pi Y.(Y \to Y))$ ist. Diesem Typ wird die Formel

$$F_{\mathrm{Id}}(x) \;=\; \forall Y \forall y(y \in Y \to xy \in Y)$$

   zugeordnet. F_{Id} bedeutet also, dass x für jeden Typ Y eine totale Funktion von Y nach Y ist. Da ausserdem $\mathrm{Id}^* = (\lambda x.x)$ gilt, erhalten wir $F_{\mathrm{Id}}(\mathrm{Id}^*)$.

2. Nun betrachten wir den Term Ap aus Beispiel 3.9.1, der vom Typ

$$T_{\mathrm{Ap}} \;=\; (\Pi X.(\Pi Y.((X \to Y) \to (X \to Y))))$$

ist. Diesem Typ wird die Formel

$$F_{\mathrm{Ap}}(x) \;=\; \forall X \forall Y \forall y[(y : X \to Y) \to (xy : X \to Y)]$$

zugeordnet. Aus $F_{\mathrm{Ap}}(x)$ folgt also, dass x eine Operation ist, die für beliebige Typen X und Y jeder totalen Funktion y von X nach Y eine totale Funktion xy von X nach Y zuordnet. Ausserdem gilt $\mathrm{Ap}^* = (\lambda y \lambda x.(yx))$. Offensichtlich gilt

$$(y : X \to Y) \;\to\; (\mathrm{Ap}^* y : X \to Y)$$

für beliebige Typen X und Y. Daher erhalten wir $F_{\mathrm{Ap}}(\mathrm{Ap}^*)$.

Das folgende Theorem besagt, dass $\mathsf{P\lambda C}$ in SET eingebettet werden kann. Die Typenzuordnung von Typen zu Termen in $\mathsf{P\lambda C}$ lässt sich in SET relativ zu den beschriebenen Übersetzungen von Typen und Termen beweisen.

3.9.5 Theorem *Es sei t ein $\mathsf{P\lambda C}$-Term vom Typ T mit den freien getypten Variablen $x_1^{S_1}, \ldots, x_n^{S_n}$ und unter Umständen freien Typenvariablen. Dann gilt*

$$\mathsf{SET} \vdash F_{S_1}(x_1) \wedge \ldots \wedge F_{S_n}(x_n) \;\to\; F_T(t^*).$$

Der Beweis dieses Theorems soll als Übungsaufgabe durchgeführt werden. Dabei ist es interessant zu sehen, an welcher Stelle von imprädikativer Komprehension Gebrauch gemacht wird.

Serie 1

Gegeben seien die folgenden Prinzipien:

(Tot) $\quad (\forall x, y)(xy\downarrow)$; $\qquad$ (Ext) $\quad (\forall x)(fx \simeq gx) \;\rightarrow\; (f = g)$.

($\mathsf{d_V}$) $\quad (u = v \rightarrow \mathsf{d_V}xyuv = x) \;\wedge\; (u \neq v \rightarrow \mathsf{d_V}xyuv = y)$.

($\mathsf{i_N}$) $\quad (\forall x)(\mathsf{i_N}x \in \mathsf{N}) \;\wedge\; (\forall x, y)(x \neq y \rightarrow \mathsf{i_N}x \neq \mathsf{i_N}y)$.

($\mathsf{c_N}$) $\quad (\forall x)(\mathsf{c_N}x = \mathsf{0} \vee \mathsf{c_N}x = \mathsf{1}) \;\wedge\; (\forall x)(\mathsf{c_N}x = \mathsf{0} \leftrightarrow x \in \mathsf{N})$.

Aufgabe 1

Finden Sie einen L_p-Term $\underline{\mathrm{rec}}_t$, so dass gilt:

$$\mathsf{BON} + (\mathsf{Tot}) \vdash \underline{\mathrm{rec}}_t f \;=\; f(\underline{\mathrm{rec}}_t f).$$

Hinweis: Modifizieren Sie den Term $\underline{\mathrm{rec}}$ aus dem Rekursionssatz für BON.

Aufgabe 2

Beweisen Sie die folgenden Aussagen:

1. $\mathsf{BON} + (\mathsf{Tot}) + (\mathsf{d_V})$ ist inkonsistent.
2. $\mathsf{BON} + (\mathsf{Tot}) + (\forall x)\mathsf{N}(x)$ ist inkonsistent.

3. $\mathsf{BON} + (\mathsf{Tot}) + (\mathsf{i_N})$ ist inkonsistent.

4. $\mathsf{BON} + (\mathsf{Tot}) + (\mathsf{c_N})$ ist inkonsistent.

Aufgabe 3

Zeigen Sie: $\mathsf{BON} \vdash \mathsf{k} \neq \mathsf{s}$.

Hinweis: Beweisen Sie in BON: $\mathsf{k} = \mathsf{s} \;\to\; (\forall x)(x = \mathsf{skk})$.

Aufgabe 4

Es seien s, t L_p-Terme, x, y verschiedene Variablen und $x \notin FV(s)$. Zeigen Sie:

$$\mathsf{BON} + (\mathsf{Ext}) \vdash s{\downarrow} \to (\lambda x.t)[s/y] = (\lambda x.t[s/y]).$$

Hinweis: Lemma 2.2.4 aus der Vorlesung.

Aufgabe 5

Finden Sie eine Sprache $\mathcal{L}$ der Logik der partiellen Terme sowie $\mathcal{L}$-Terme a und b, so dass

$$\mathsf{LPT} \not\vdash a{\downarrow} \wedge b{\downarrow} \;\to\; a[b/x]{\downarrow}.$$

Hinweis: Theorem 1.2.5 aus der Vorlesung.

Abgabe: 6. Mai 1996

Serie 2

Aufgabe 1

Zeigen Sie, dass es natürliche Zahlen $\hat{\mathsf{k}}$ und $\hat{\mathsf{s}}$ gibt, so dass für alle natürlichen Zahlen m, n und p folgende Bedingungen erfüllt sind:

(1) $\{\{\hat{\mathsf{k}}\}(m)\}(n) = m$.

(2) $\{\{\hat{\mathsf{s}}\}(m)\}(n) \in \mathbb{N}$ und $\{\{\{\hat{\mathsf{s}}\}(m)\}(n)\}(p) \simeq \{\{m\}(p)\}(\{n\}(p))$.

Aufgabe 2

Erweitern Sie das normale Termmodell $\mathcal{CNT}$ so, dass darin *full definition by cases* ($\mathsf{d_V}$) erfüllt ist.

Aufgabe 3

Im folgenden betrachten wir das Join-Axiom (J) in expliziter Mathematik. Dazu schreiben wir $Z = \Sigma(U, f)$ für die Aussage

$$\forall x(x \in Z \leftrightarrow x = (\mathsf{p_0}x, \mathsf{p_1}x) \wedge \mathsf{p_0}x \in U \wedge \exists Y(\Re(f(\mathsf{p_0}x), Y) \wedge \mathsf{p_1}x \in Y)),$$

d.h. Z ist die disjunkte Vereinigung über alle $x \in U$ derjenigen Typen, die durch fx benannt werden. Nun hat das Join-Axiom die Form

$$(\mathsf{J}) \quad \Re(u, U) \wedge (\forall x \in U)\exists Y \Re(fx, Y) \rightarrow \exists Z(\Re(\mathsf{j}(u, f), Z) \wedge Z = \Sigma(U, f)).$$

Hier ist j eine neue Konstante unserer Sprache. Erweitern Sie die Standardmodell-Konstruktion für EET, so dass das Join-Axiom (J) mitbehandelt wird.

Aufgabe 4

Ein Typ X heisst *berechenbar*, falls er eine charakteristische Funktion auf V besitzt:

$$\begin{aligned} f \in P(\mathsf{V}) &:= \forall x(fx = \mathsf{0} \vee fx = \mathsf{1}), \\ \mathit{Comp}(X) &:= (\exists f \in P(\mathsf{V}))\forall x(x \in X \leftrightarrow fx = \mathsf{0}). \end{aligned}$$

Zeigen Sie: $\mathsf{EET} \vdash \exists X \neg \mathit{Comp}(X)$.

Abgabe: 17. Juni oder 24. Juni (mit Adresse).