Vergleich unstetiger Funktionen: „Principle of Omniscience" und Vollständigkeit in der C-Hierarchie

Dissertation

zur Erlangung des naturwissenschaftlichen Doktorgrades der Fakultät für Mathematik und Informatik der Fernuniversität Hagen

Vorgelegt von: Uwe Mylatz aus Lüneburg
Eingereicht im Mai 2006
Hauptgutachter Klaus Weihrauch
Zweitgutachter Rutger Verbeek

Inhaltsverzeichnis

1 Einleitung

In dieser Dissertation sollen Themen aus der Berechenbarkeitstheorie betrachtet werden. Wir möchten unstetige Funktionen oder Mengen von unstetigen Funktionen miteinander vergleichen, einige von ihnen bekannt aus dem Gebiet der Analysis. Die Betrachtungen sind vor allem motiviert durch das Buch „Computability" von Klaus Weihrauch und die Diplomarbeit von Thorsten von Stein mit dem Titel „Vergleich nicht konstruktiv lösbarer Probleme in der Analysis". Der erste Teil basiert auf dem Bericht „The TTE-Interpretation of Three Hierarchies of Omniscience Principles" von Klaus Weihrauch.

Die Funktionen, die wir betrachten, sind Typ-II-Funktionen. Das bedeutet, ihr Definitionsbereich hat die Mächtigkeit des Kontinuums. Die Argumente dieser Funktionen lassen sich deshalb nicht mehr durch eine Nummerierung mit natürlichen Zahlen oder einer Notation mit Worten endlicher Länge benennen. Die Argumente können nur noch durch Darstellungen unendlicher Länge repräsentiert werden. Es handelt sich z.B. um Folgen von natürlichen Zahlen, Mengen von natürlichen Zahlen, reelle Zahlen usw. Aber die Mächtigkeit des Definitionsbereichs ist nicht größer als die Kardinalität des Kontinuums, bzw. des Intervalls $[0, 1]$ der reellen Zahlen, welches der Mächtigkeit der Potenzmenge oder der Menge aller Folgen der natürlichen Zahlen entspricht. Eine Theorie der Typ-II-Berechenbarkeit beschäftigt sich nicht mit Funktionen auf noch größeren Definitionsbereichen. So werden wir nicht beliebige Funktionen auf den reellen Zahlen als Argumente zulassen, denn die Menge aller Funktionen dieser Art ist größer als das Kontinuum. Als Argumente werden wir nur Teilmengen der reellen Funktionen mit der Mächtigkeit des Kontinuums betrachten, z.B. die Menge der stetigen Funktionen auf einem kompakten Intervall.

Die Theorie der Typ-II-Berechenbarkeit weist Ähnlichkeiten mit der Berechenbarkeitstheorie von Typ-I-Funktionen auf. Der Hauptunterschied zeigt sich beim Konzept der Stetigkeit bzw. Unstetigkeit bei Typ-II-Funktionen. Eine Funktion heißt stetig, wenn für die Berechnung eines endlichen Teils des Resultates nur ein endliches Teil des Arguments benutzt werden muss. Der Ausdruck Stetigkeit ist adäquat, weil er mit der topologischen Stetigkeit übereinstimmt, wenn eine geeignete Topologie gewählt wird. Ich werde darauf kurz im ersten Kapitel eingehen.

In der Theorie der Typ-I-Berechenbarkeit ist ein wichtiges und faszinierendes Ergebnis, dass es Funktionen gibt, die nicht berechenbar sind, Funktionen, die nicht von einem Computer mit einem endlichen Programm berechnet werden können. Solche Funktionen sind zu kompliziert, um durch eine endliche Zeichenkette (Programm) repräsentiert zu werden. In Beweisen wird sehr oft die Idee der Selbstanwendbarkeit als ein Werkzeug benutzt. Einige Funktionen können so interpretiert werden, dass sie Eigenschaften von sich selbst berechnen. Mit diesem Trick ist es oft möglich zu beweisen, dass eine Funktion nicht berechenbar ist.

Eine ähnliche Rolle wie die nicht berechenbaren Funktionen in der Theorie der Berechenbarkeit spielen die unstetigen Funktionen in der Typ-II-Theorie. Unstetigkeit einer Funktion bedeutet, dass die Funktion nicht einmal durch einen Computer mit einem unendlichen Programm berechnet werden kann. Viele bekannte reelle Funktionen, die wir z.B. aus der Schule kennen, sind unstetig und damit nicht berechenbar. Es gibt in der Typ-II-Theorie neben den berechenbaren und den unstetigen Funktionen auch stetige, aber nicht berechenbare

Funktionen. Das sind Funktionen, die sich nicht mit endlich langen Programmen, aber doch noch mit unendlich langen berechnen lassen. Z.B. ist eine konstante Funktion mit einem nicht berechenbaren reellen Funktionswert stetig, aber nicht berechenbar. Die meisten Funktionen, die in der Analysis vorkommen, sind aber entweder berechenbar oder unstetig. Funktionen der dritten Gruppe werden eher künstlich konstruiert.

Ich finde, dass die Existenz von nicht berechenbaren Funktionen bemerkenswert ist. Die Möglichkeiten von Computern und von exakten Methoden und Algorithmen sind eng begrenzt. Der Mensch ist nicht frei von diesen Schranken in seinem Denken, aber er hat die Möglichkeit und den Wunsch, feste Methoden und Verfahren zu tranzendieren. So entspringt die Faszination der Mathematik teilweise der Tatsache, dass es keinen Algorithmus gibt, der für jedes Problem zu einer Lösung führt. Besonders die ungelösten Probleme der Mathematik ziehen Interessierte am meisten an. So sind auch viele Existenz- oder Widerspruchsbeweise in dieser Arbeit nicht konstruktiv.

Ferner zeigt die Untersuchung von unstetigen und damit nicht berechenbaren reellen oder komplexen Funktionen, dass sogar sehr einfache Funktionen, wie z.B. Treppenfunktionen, nicht berechenbar sein können, obwohl sie im Mathematikunterricht als unproblematisch dargestellt werden. Auch stetige Funktionen wie die reelle Addition erfordern einiges an Überlegung, da sie nur bezüglich einer geeigneten Darstellung stetig sind, z.B. nicht in der üblicherweise verwendeten Dezimaldarstellung.[1]

Diese Arbeit hat zwei Hauptteile. Im ersten Teil werden wir vor allem Funktionen behandeln, die n Folgen $p_1, \ldots, p_n$ oder unendlich viele Folgen $p_1, p_2, \ldots$ als Argumente und denselben Funktionswert für verschiedene Eingaben $p_1, p_2, \ldots$ und $q_1, q_2, \ldots$ haben, wenn dieselben Folgen Nullfolgen sind:

$$(\forall i \in \{1, \ldots, n\})(p_i = 0^\omega \Leftrightarrow q_i = 0^\omega) \Longrightarrow f\langle p_1, \ldots, p_n\rangle = f\langle q_1, \ldots, q_n\rangle$$

$$\text{bzw. } (\forall i \geq 1)(p_i = 0^\omega \Leftrightarrow q_i = 0^\omega) \Longrightarrow f\langle p_1, p_2, \ldots\rangle = f\langle q_1, q_2, \ldots\rangle$$

Wir werden auch mehrwertige Funktionen dieses Typs betrachten. Das ist motiviert durch die Funktionenmengen $LLPO_n$ in [Wei92a]. Diese Mengen von Funktionen bilden eine Hierarchie, in welcher die Elemente mit zunehmendem n immer einfacher werden. Für $n \geq 2$ sind sie einfacher als das principle of omniscience Ω von Errett Bishop und Douglas Bridges.
Für den Vergleich von Funktionen und Funktionenmengen muss eine geeignete Reduzierbarkeitsrelation definiert werden.
Es war mein Ziel, eine einfache Regel für die Entscheidung zu finden, welche Funktionen oder Funktionenmengen reduzierbar aufeinander oder äquivalent zueinander sind. Zum Beispiel war ich daran interessiert, die Anzahl der Äquivalenzklassen von totalen und partiellen Funktionen der angegebenen Art zu berechnen. Für diesen Zweck wurden Algorithmen entwickelt. Mit diesen Algorithmen ist es im Prinzip möglich zu berechnen, ob eine Funktion auf eine andere reduzierbar ist, und welche Äquivalenzklassen es gibt. Für kleine Anzahlen von Argumenten ($n \leq 2$) wurde dies auch für totale und partielle Funktionen erreicht. Aber die Algorithmen benötigen sehr viel Rechenzeit. So ist es nicht

[1] Dieser Teil der Einleitung ist ähnlich zu der Einleitung meiner Diplomarbeit, da das Hauptthema in beiden Arbeiten dasselbe ist.

möglich, diese Informationen für größere Probleme mit dem Computer zu berechnen.

Der zweite Teil behandelt die C-Hierarchie. Die Funktion C ist schwieriger als Ω. Ω gibt die Information, ob eine Folge 0^ω ist oder nicht. Das ist äquivalent zu dem Problem, ob in einer Folge die Zahl 0 vorkommt oder nicht. Die Funktion C gibt diese Information für alle $i \in \mathbb{N}$:

$$C(p)(i) := \begin{cases} 0 & \text{falls } (\exists n)p(n) = i+1 \\ 1 & \text{sonst} \end{cases}$$

Durch wiederholte Anwendung von C erhalten wir noch schwierigere Funktionen. Eine Funktion f heißt C^n-stetig, wenn sie berechnet werden kann durch $A \circ C \circ B_1 \circ \ldots \circ C \circ B_n$ mit stetigen Funktionen $A, B_1, \ldots, B_n$.
Die C^n-stetigen Funktionen bilden eine echte Hierarchie von zunehmend schwieriger werdenden Funktionen. In [St89] und [Myl92] wurden einige C^n-stetige Funktionen betrachtet. Zwei offene Probleme wurden dort noch nicht behandelt:

- Welche Funktionen sind vollständig für die C^n-Klassen? Gibt es vollständige Funktionen? Welche bekannten Funktionen in speziellen Klassen sind nicht vollständig?
- Ist es möglich, Funktionen in einer C^n-Klasse durch logische Ausdrücke zu charakterisieren?

Der zweite Teil wird versuchen, auf diese Fragen eine Antwort zu geben.

Gerne möchte ich an dieser Stelle noch meinen Dank an Prof. Klaus Weihrauch aussprechen, der mich zum Ende meines Studiums an der Fernuniversität zu einer Diplomarbeit in der Theoretischen Informatik ermuntert hat und mich bei der Dissertation beraten hat. Großer Dank gebührt auch meiner Frau Sigrid, die soviel Geduld und Verständnis aufgebracht hat, und ebenfalls meinen Kollegen Martin Schreiber und Martin Warnke vom Rechen- und Medienzentrum der Universität Lüneburg, die mich oft unterstützt haben, auch wenn diese Dissertation nicht so ganz in mein dortiges Aufgabengebiet gehört.

2 Symbole

Hier sollen einige oft benutzte Symbole definiert werden.

$\mathbb{N}$ sei definiert durch $\mathbb{N} := \{0, 1, 2, \ldots\}$. n-Tupel aus der Menge der natürlichen Zahlen $\mathbb{N}$ werden geschrieben als $(a_0, a_1, \ldots, a_{n-1}) \in \mathbb{N}^n$, konstante n-Tupel als a^n.[2] Folgen aus $\mathbb{N}$ werden notiert durch $(a_0, a_1, \ldots) \in \mathbb{N}^\omega$, konstante Folgen durch a^ω. Hier symbolisiert $\mathbb{N}^\omega$ oder $\mathbb{B}$ die Menge aller Folgen über $\mathbb{N}$ und $\mathbb{N}^*$ die Menge aller endlichen Tupel über $\mathbb{N}$. Gemischte Notationen sind auch möglich. So bedeutet $(1, 2, \ldots, 6)2^3$ das gleiche wie $(1, 2, 3, 4, 5, 6, 2, 2, 2)$ und $(0, 1, 2)0^3 3^\omega$ ist die Folge $(0, 1, 2, 0, 0, 0, 3, 3, 3, \ldots)$. Für die Präfixrelation von Worten oder Folgen wird $\sqsubseteq$ oder $\sqsubset$ benutzt.

Die ersten n Elemente eines Tupels oder einer Folge p werden durch $[[p]]_n := (p(0), p(1), \ldots, p(n-1))$ bezeichnet. M_p ist die Menge, welche durch die Folge p repräsentiert wird: $M_p := \{k | (\exists i) p(i) = k + 1\}$. $M_p = \emptyset \iff p = 0^\omega$.

Cantors Funktion π, die auch als $\langle , \rangle$ notiert wird, wird auf Folgen erweitert[3]. Sie wird definiert durch: $\langle p, q \rangle (2n) := p(n)$ und $\langle p, q \rangle (2n + 1) := q(n)$ für alle Folgen $p, q \in \mathbb{B}$, $\langle p \rangle := p$ und $\langle p_1, p_2, \ldots, p_{k+1} \rangle := \langle \langle p_1, p_2, \ldots, p_k \rangle, p_{k+1} \rangle$ für alle Folgen $p, p_1, p_2, \ldots, p_{k+1} \in \mathbb{B}$. $\langle i, p \rangle (0) := i$ und $\langle i, p \rangle (n{+}1) := p(n)$ für alle $i \in \mathbb{N}$ und alle Folgen $p \in \mathbb{B}$. $\langle p_0, p_1, p_2, \ldots \rangle \langle i, j \rangle := p_i(j)$ für alle $p_0, p_1, p_2, \ldots \in \mathbb{B}$. Die i-te Folge in $\langle p_0, p_1, \ldots \rangle$ wird bezeichnet durch $\langle p \rangle_i$. Dies soll die Verwechslung mit p_i als i-te Folge in einer Aufzählung vermeiden.

Partielle Funktionen f von M nach M' werden durch $f :\subseteq M \longrightarrow M'$ bezeichnet, totale Funktionen durch $f : M \longrightarrow M'$, wenn betont werden soll, dass sie auf der ganzen Menge M definiert sind.

Einige einfache Funktionen werden manchmal in den Beweisen benutzt.
$NEG : \mathbb{N} \to \mathbb{N}$ wird definiert durch

$$NEG(n) := \begin{cases} 0 & \text{falls } n = 1 \\ 1 & \text{sonst} \end{cases}$$

für alle $n \in \mathbb{N}$. $INV : \mathbb{B} \to \mathbb{B}$ wird definiert durch

$$INV(p)(n) := \begin{cases} 0 & \text{falls } p(n) = 1 \\ 1 & \text{sonst} \end{cases}$$

für alle $p \in \mathbb{B}$ und $n \in \mathbb{N}$. $SIGN : \mathbb{N} \to \mathbb{N}$ wird definiert durch

$$SIGN(n) := \begin{cases} 0 & \text{falls } n = 0 \\ 1 & \text{sonst} \end{cases}$$

für alle $n \in \mathbb{N}$. $INC : \mathbb{B} \to \mathbb{B}$ wird definiert durch

$$INC(p)(n) := p(n) + 1$$

[2] Die Zählung beginnt manchmal mit 0 und manchmal mit 1, ist also nicht immer einheitlich.

[3] $\langle x, y \rangle = \pi(x, y)$ für $x, y \in \mathbb{N}$, $\langle x_1, \ldots, x_n \rangle = \pi^n(x_1, \ldots, x_n)$ für $x_1, \ldots, x_n \in \mathbb{N}$ und $\pi_i^{(n)} \langle x_1, \ldots, x_n \rangle := x_i$ für $x_1, \ldots, x_n \in \mathbb{N}$ und $1 \leq i \leq n$.

für alle $p \in \mathbb{B}$ und $n \in \mathbb{N}$.
$\overline{0} : [0,1] \to \mathbb{R}$ ist die konstante Nullfunktion auf dem Intervall $[0,1]$: $\overline{0}(x) := 0$ für alle $x \in [0,1]$.

$\nu_Q : \mathbb{N} \to \mathbb{Q}$ ist eine Standardnummerierung der rationalen Zahlen:

$$\nu_Q\langle i,j,k\rangle := \frac{i-j}{k+1}$$

Eine Repräsentation $\rho :\subseteq \mathbb{B} \to \mathbb{R}$ der reellen Zahlen ist gegeben durch

$$Def(\rho) = \{p \in \mathbb{B} | (\forall n > m) |\nu_Q(p(m)) - \nu_Q(p(n))| < 2^{-m}\} \text{ und}$$

$$\rho(p) := \lim_{n\to\infty} \nu_Q(p(n)) \text{ für alle } p \in Def(\rho)$$

Diese Repräsentation benutzt eine schnelle Approximation durch rationale Zahlen. Eine schwächere Repräsentation benutzt eine Approximation ohne Information über ihre Geschwindigkeit. Sei $\rho_n :\subseteq \mathbb{B} \to \mathbb{R}$ definiert durch

$$\rho_n(p) := \begin{cases} x & \text{falls } \lim_{n\to\infty} \nu_Q(p(n)) = x \text{ existiert} \\ div & \text{sonst} \end{cases}$$

Wir werden auch eine Repräsentation für stetige Funktionen auf dem Intervall $[0,1]$ verwenden. Die Menge der Polygone mit rationalen Stützpunkten ist dicht in der Menge der stetigen Funktionen auf $[0,1]$. Sei α eine natürliche Nummerierung dieser Polygone. Dann kann eine stetige Funktion durch eine Approximation mit Polygonen repräsentiert werden .
Sei $C[0,1]$ die Menge der stetigen Funktionen auf $[0,1]$.
Definiere $\delta_\alpha :\subseteq \mathbb{B} \to C[0,1]$ durch

$$\delta_\alpha(p) := \begin{cases} f & \text{falls } \lim_{n\to\infty} \alpha(p(n)) = f \\ & \text{und } (\forall n > m)|\alpha(p(m)) - \alpha(p(n))| < 2^{-m} \\ div & \text{sonst} \end{cases}$$

Ein Baum ist hier ein endlicher, zusammenhängender, gerichteter Graph $G = (V,E)$, bei dem jeder Knoten höchstens eine eingehende Kante hat. Die Wurzel ist der einzige Knoten ohne eingehende Kante. Der Vater eines Knotens $v_1 \in V$ ist der Knoten $v_2 \in V$ mit $(v_2, v_1) \in E$. Ein Sohn eines Knotens $v_1 \in V$ ist ein Knoten $v_2 \in V$ mit $(v_1, v_2) \in E$. Ein Nachfolger eines Knotens $v \in V$ ist ein Knoten $w \in V$, so dass es Knoten $v_1, \ldots, v_n \in V$ gibt mit $(v_i, v_{i+1}) \in E$ für alle $1 \le i < n$ und $v = v_1$ und $w = v_n$.

3 Principle of Omniscience

3.1 Hierarchien und Reduzierbarkeitsrelationen

3.1.1 Definition von $LLPO_n$, $MLPO_n$ und LPO_n

In [Wei92a] hat Klaus Weihrauch drei Hierarchien von unstetigen Funktionen eingeführt. In dieser Arbeit möchte ich unter anderem diese Hierarchien behandeln.

Dies sind zunächst die wichtigsten Definitionen und Resultate aus [Wei92a]. Sie werden teilweise in anderer Form formuliert. Zum Beispiel werden in [Wei92a] Funktionen von $(\Sigma^\omega)^n$ nach Σ^* verwendet. Hier betrachten wir Funktionen von $\mathbb{B}^n$ nach $\mathbb{N}$. Aber die Definitionen und Sätze sind ähnlich.

Definition 1 ($\leq$ für Funktionen)**.** *Seien $f :\subseteq \mathbb{B} \to X$ und $g :\subseteq Y \to Z$ Funktionen mit $X, Y, Z \in \{\mathbb{B}, \mathbb{N}\}$.*
$f \leq g :\Longleftrightarrow$ es gibt stetige Funktionen $A :\subseteq \mathbb{B} \to X$ und $B :\subseteq \mathbb{B} \to Y$ mit

$$f(p) = A\langle p, g \circ B(p)\rangle \text{ für alle } p \in Def(f).$$

$f \equiv g :\Longleftrightarrow f \leq g$ und $g \leq f$. $f < g :\Longleftrightarrow f \leq g$ und $f \not\equiv g$.

Definition 2 ($\leq$ für Funktionenmengen)**.** *Seien $S \subseteq \{f :\subseteq \mathbb{B} \to X\}$ und $T \subseteq \{f :\subseteq Y \to Z\}$ mit $X, Y, Z \in \{\mathbb{B}, \mathbb{N}\}$.*
$S \leq T :\Longleftrightarrow$ es gibt stetige Funktionen $A :\subseteq \mathbb{B} \to X$ und $B :\subseteq \mathbb{B} \to Y$ mit

$$p \mapsto A\langle p, g \circ B(p)\rangle \in S \text{ für alle } g \in T.$$

$S \equiv T :\Longleftrightarrow S \leq T$ und $T \leq S$. $S < T :\Longleftrightarrow S \leq T$ und $S \not\equiv T$.

$\leq$ ist die Reduzierbarkeitsrelation, die wir benutzen, um Funktionen miteinander zu vergleichen. Wenn $f \leq g$, dann ist f nicht schwieriger wie g. Zum Beispiel folgt, dass f stetig ist, falls g stetig ist, und dass g unstetig ist, falls f unstetig ist. Später werden wir auch andere Reduzierbarkeitsrelationen verwenden und werden die hier definierte Relation zur besseren Unterscheidung meistens $\leq_2$ nennen.

Für Mengen gilt ähnliches. Es gelte $S \leq T$. Falls es eine stetige Funktion $g \in T$ gibt, dann ist eine Funktion $f \in S$ auch stetig. Falls alle $f \in S$ unstetig sind, dann sind auch alle $g \in T$ unstetig.

Definition 3 (Principle of Omniscience Ω)**.** *Sei $\Omega : \mathbb{B} \to \mathbb{N}$ definiert durch*

$$\Omega(p) := \begin{cases} 0 & \text{falls } p = 0^\omega \\ 1 & \text{sonst} \end{cases}$$

für alle $p \in \mathbb{B}$. Ein anderer Name für Ω ist LPO (Limited Principle of Omniscience).

Dies ist das Principle of Omniscience, das von Bishop and Bridges in [BiBr85] verwendet wurde. Für sie zeigt das Vorkommen dieses Prinzips in einem Beweis oder einer Definition, dass sie nicht konstruktiv vorgehen, sondern Objekte benutzen, die man nicht berechnen kann.

Definition 4 (LPO_n)**.** *Für $n \in \mathbb{N}$ definiere $f_n :\subseteq \mathbb{B} \to \mathbb{N}$ durch*

$$f_n(p) := \begin{cases} i & \textit{falls } i = \#_1(p) \leq n \\ div & \textit{sonst} \end{cases}$$

für alle $p \in \mathbb{B}$. Definiere $LPO_n := \{f_n\}$.

$\#_1(p)$ ist definiert als die Anzahl der Einsen in p.

Definition 5 (g_n)**.** *Für $n \in \mathbb{N}$ definiere $g_n : \mathbb{B} \to \mathbb{N}$ durch*

$$g_n\langle p_1, \ldots, p_n\rangle := m, \textit{ falls } m = card\{i | p_i \neq 0^\omega\}$$

für alle $p_1, \ldots, p_n \in \mathbb{B}$.

Satz 1 ($g_n \equiv f_n$)**.**

1. *$g_1 \equiv \Omega$.*
2. *$g_n \equiv f_n$ für alle $n \geq 1$.*

Die Beweise werden ausgelassen. Sie stehen in [Wei92a] und müssten hier leicht abgeändert werden, da wir andere Definitions- und Wertebereiche verwenden.

Definition 6 (Partition einer Funktion)**.** *Eine Partition einer Funktion $f : X \to Y$ ist eine Menge $\{f|_A | A \in \pi\}$, wobei π eine Partition von X ist. Eine Funktion $f : X \to Y$ heißt k-stetig, gdw. f eine Partition von höchstens k stetigen Funktionen hat.*

Satz 2 (LPO_n-Hierarchie)**.**

1. *$LPO_0 < LPO_1 < LPO_2 < \ldots$*
2. *LPO_n ist $(n+1)$-stetig, aber nicht n-stetig.*
3. *f ist nicht n-stetig, gdw. $f_n \leq f$. Also ist f_n die einfachste Funktion, die nicht n-stetig ist.*

Definition 7 ($LLPO_n$)**.** *Für $n \in \mathbb{N}$, $n \geq 2$ sei $LLPO_n$ die Menge aller Funktionen f mit*

$$Def(f) = \{\langle p_1, p_2, \ldots, p_n\rangle \mid \textit{ es gibt höchstens ein } k \textit{ mit } p_k \neq 0^\omega\}$$
$$\textit{und } f\langle p_1, p_2, \ldots, p_n\rangle = k \textit{ für ein } k \textit{ mit } p_k = 0^\omega.$$

$LLPO$ ist eine Abkürzung für Lesser Limited Principle of Omniscience.

Satz 3 ($LLPO_n$-Hierarchie)**.**

1. *$LLPO_2 < LPO_1$*
2. *$LLPO_{n+1} < LLPO_n$ für alle $n \geq 2$.*

Definition 8 ($MLPO_n$)**.** *Für $n \in \mathbb{N}$, $n \geq 2$ sei $MLPO_n$ die Menge aller Funktionen f mit*

$$Def(f) = \{\langle p_1, p_2, \ldots, p_n\rangle \mid \textit{ es gibt eine Zahl } k \textit{ mit } p_k = 0^\omega\}$$
$$\textit{und } f\langle p_1, p_2, \ldots, p_n\rangle = k \textit{ für ein } k \textit{ mit } p_k = 0^\omega.$$

Satz 4 ($MLPO_n$-Hierarchie)**.**

1. $MLPO_n < MLPO_{n+1}$ *für* $n \geq 2$.
2. $MLPO_{n+1} < LPO_n$ *für* $n \geq 1$.

Die folgende Abbildung zeigt einige der Beziehungen. Sie ist aus [Wei92a] übernommen.

$$
\begin{tikzcd}
 & LPO_n \\
MLPO_{n+1} \arrow[ur] & LPO_{n-1} \arrow[u] \\
MLPO_n \arrow[u] \arrow[ur] & \vdots \arrow[u] \\
\vdots \arrow[u] & LPO_2 \\
MLPO_3 \arrow[ur] & LPO_1 = \Omega \arrow[u] \\
MLPO_2 = LLPO_2 \arrow[u] \arrow[ur] & \\
LLPO_3 \arrow[u] & \\
\vdots \arrow[u] & \\
LLPO_n &
\end{tikzcd}
$$

Abbildung 1: Reduzierbarkeitsrelationen bei Weihrauch

3.1.2 Andere Definitionen von Reduzierbarkeit

Die obige Definition der Reduzierbarkeit ist nur eine mögliche unter vielen anderen. In dieser Arbeit werden meistens die obigen Definitionen verwendet. Zur Unterscheidung wird die angegebene Relation durch „$\leq_2$" bezeichnet, und zwar für die Reduzierbarkeit von Funktionen und Mengen von Funktionen. Zwei schwächere Relationen werden noch eingeführt, „$\leq_0$" und „$\leq_1$".

Definition 9 ($\leq_0$ für Funktionen)**.** *Seien $f :\subseteq \mathbb{B} \to X$ und $g :\subseteq Y \to X$ mit $X, Y \in \{\mathbb{B}, \mathbb{N}\}$.*
$f \leq_0 g :\Longleftrightarrow$ es gibt eine stetige Funktion $B :\subseteq \mathbb{B} \to Y$ mit

$$f(p) = g \circ B(p) \text{ für alle } p \in Def(f).$$

$f \equiv_0 g :\Longleftrightarrow f \leq_0 g$ und $g \leq_0 f$. $f <_0 g :\Longleftrightarrow f \leq_0 g$ und $f \not\equiv_0 g$.

Definition 10 ($\leq_0$ für Funktionenmengen)**.** *Seien $S \subseteq \{f :\subseteq \mathbb{B} \to X\}$ und $T \subseteq \{f :\subseteq Y \to X\}$ mit $X, Y \in \{\mathbb{B}, \mathbb{N}\}$.*
$S \leq_0 T :\Longleftrightarrow$ es gibt eine stetige Funktion $B :\subseteq \mathbb{B} \to Y$ mit

$$p \mapsto g \circ B(p) \in S \text{ für alle } g \in T.$$

$S \equiv_0 T :\Longleftrightarrow S \leq_0 T$ und $T \leq_0 S$. $S <_0 T :\Longleftrightarrow S \leq_0 T$ und $S \not\equiv_0 T$.

Definition 11 ($\leq_1$ für Funktionen)**.** *Seien $f :\subseteq \mathbb{B} \to X$ und $g :\subseteq Y \to Z$ mit $X, Y, Z \in \{\mathbb{B}, \mathbb{N}\}$.*
$f \leq_1 g :\Longleftrightarrow$ es gibt stetige Funktionen $A :\subseteq Z \to X$ und $B :\subseteq \mathbb{B} \to Y$ mit

$$f(p) = A \circ g \circ B(p) \text{ für alle } p \in Def(f).$$

$f \equiv_1 g :\Longleftrightarrow f \leq_1 g$ und $g \leq_1 f$. $f <_1 g :\Longleftrightarrow f \leq_1 g$ und $f \not\equiv_1 g$.

Definition 12 ($\leq_1$ für Funktionenmengen)**.** *Seien $S \subseteq \{f :\subseteq \mathbb{B} \to X\}$ und $T \subseteq \{f :\subseteq Y \to Z\}$ mit $X, Y, Z \in \{\mathbb{B}, \mathbb{N}\}$.*
$S \leq_1 T :\Longleftrightarrow$ es gibt stetige Funktionen $A :\subseteq Z \to X$ und $B :\subseteq \mathbb{B} \to Y$ mit

$$p \mapsto A \circ g \circ B(p) \in S \text{ für alle } g \in T.$$

$S \equiv_1 T :\Longleftrightarrow S \leq_1 T$ und $T \leq_1 S$. $S <_1 T :\Longleftrightarrow S \leq_1 T$ und $S \not\equiv_1 T$.

3.2 Spezielle Probleme

3.2.1 LPO_n

Die Funktionen LPO_n bilden eine Hierarchie in Bezug auf die Reduzierbarkeitsrelation $\leq_2$.

In [Myl92] habe ich schon das Principle of Omniscience und Verallgemeinerungen davon untersucht. Die Idee war, das Principle of Omniscience mehr als einmal zu benutzen. Eine Funktion, die zu ihrer Berechnung dieses Prinzip n-mal braucht, müsste einfacher sein als eine Funktion, die es $n+1$-mal benötigt.

In [Myl92] wurden Ω^n-stetige Funktionen definiert durch

Definition 13 (Ω^n-stetige Funktionen).
Eine Funktion $\Gamma :\subseteq \mathbb{B} \longrightarrow \mathbb{N}$ (bzw. $\Gamma :\subseteq \mathbb{B} \longrightarrow \mathbb{B}$) ist Ω^0-stetig, gdw. sie stetig ist, und Ω^{n+1}-stetig, gdw. es eine stetige Funktion $A :\subseteq \mathbb{B} \longrightarrow \mathbb{N}$ (bzw. $A :\subseteq \mathbb{B} \longrightarrow \mathbb{B}$) und eine Ω^n-stetige Funktion $\Sigma :\subseteq \mathbb{B} \longrightarrow \mathbb{B}$ gibt mit $\Gamma(p) = A\langle p, \Omega \circ \Sigma(p)\rangle$ für alle $p \in Def(\Gamma)$.

Diese Definition hat den Nachteil, dass das meiste der Information, die bei der Anwendung von Ω gewonnen wurde, verloren geht. Eventuell kann mehr Information genutzt werden, wenn die Funktion A alle Anwendungen von Ω auf einmal benutzt.

Definition 14 (Ω_n-stetige Funktionen).
Für $n \geq 0$ ist eine Funktion $\Gamma :\subseteq \mathbb{B} \longrightarrow \mathbb{N}$ (bzw. $\Gamma :\subseteq \mathbb{B} \longrightarrow \mathbb{B}$) Ω_n-stetig, gdw. es stetige Funktionen $A :\subseteq \mathbb{B} \longrightarrow \mathbb{N}$ (bzw. $A :\subseteq \mathbb{B} \longrightarrow \mathbb{B}$) und stetige Funktionen $A_1, \ldots, A_n :\subseteq \mathbb{B} \longrightarrow \mathbb{B}$ gibt mit

$$\Gamma(p) = A\langle p, \Omega \circ A_1(p), \ldots, \Omega \circ A_n(p)\rangle \text{ für alle } p \in Def(\Gamma).$$

Die Funktionenklassen Ω_n und Ω^n sind nicht miteinander vergleichbar. Es gibt Funktionen f mit $f \in \Omega_n$ und $f \notin \Omega^n$ und Funktionen f mit $f \in \Omega^n$ und $f \notin \Omega_n$:

Satz 5 (Ω^n-stetig $\not\Rightarrow$ Ω_n-stetig).
Es gibt eine Funktion $f : \mathbb{B} \to \{0,1\}$, die Ω^2-stetig und nicht Ω_2-stetig ist.

Beweis. Sei f definiert durch

$$f(p) := \begin{cases} 1 & \text{falls } (\exists k)p(k) = 3 \text{ und } (\exists k)p(k) = 5 \\ 1 & \text{falls } \neg(\exists k)p(k) = 3 \text{ und } (\exists k)p(k) = 7 \\ 0 & \text{sonst} \end{cases}$$

<u>f ist Ω^2-stetig:</u>
Definiere $Y : \mathbb{B} \longrightarrow \mathbb{B}$ durch

$$Y(p)(n) := \begin{cases} 1 & \text{falls } p(n) = 3 \\ 0 & \text{sonst} \end{cases}$$

und $X : \mathbb{B} \longrightarrow \mathbb{N}$ durch

$$X\langle p, x\rangle(n) := \begin{cases} 1 & \text{falls } x = 1 \text{ und } p(n) = 5 \\ 1 & \text{falls } x = 0 \text{ und } p(n) = 7 \\ 0 & \text{sonst} \end{cases}$$

für alle $p \in \mathbb{B}, x \in \mathbb{N}$ und $A : \mathbb{B} \longrightarrow \mathbb{N}$ durch $A\langle p, x\rangle := x$.
A, X und Y sind stetig.
Falls $(\exists k)p(k) = 3$ und $(\exists k)p(k) = 5$, dann gilt
$A\langle p, \Omega \circ X\langle p, \Omega \circ Y(p)\rangle\rangle = A\langle p, \Omega \circ X\langle p, 1\rangle\rangle = A\langle p, 1\rangle = 1 = f(p)$.
Falls $\neg(\exists k)p(k) = 3$ und $(\exists k)p(k) = 7$, dann gilt
$A\langle p, \Omega \circ X\langle p, \Omega \circ Y(p)\rangle\rangle = A\langle p, \Omega \circ X\langle p, 0\rangle\rangle = A\langle p, 1\rangle = 1 = f(p)$.
Falls $(\exists k)p(k) = 3$ und $(\neg\exists k)p(k) = 5$, dann gilt
$A\langle p, \Omega \circ X\langle p, \Omega \circ Y(p)\rangle\rangle = A\langle p, \Omega \circ X\langle p, 1\rangle\rangle = A\langle p, 0\rangle = 0 = f(p)$.
Falls $\neg(\exists k)p(k) = 3$ und $(\neg\exists k)p(k) = 7$, dann gilt
$A\langle p, \Omega \circ X\langle p, \Omega \circ Y(p)\rangle\rangle = A\langle p, \Omega \circ X\langle p, 0\rangle\rangle = A\langle p, 0\rangle = 0 = f(p)$.
Also gilt $f(p) = A\langle p, \Omega \circ X\langle p, \Omega \circ Y(p)\rangle\rangle$ für alle $p \in \mathbb{B}$.

<u>f ist nicht Ω_2-stetig:</u>
Angenommen, $f(p) = A\langle p, \Omega \circ B_1(p), \Omega \circ B_2(p)\rangle$ mit stetigen Funktionen A, B_1 und B_2.
Definiere $p_0 := 0^\omega$. Sei $x_0 := \Omega \circ B_1(p_0)$ und $y_0 := \Omega \circ B_2(p_0)$.
Dann gilt $f(p_0) = A\langle p_0, x_0, y_0\rangle = 0$.
Weil A stetig ist, gibt es eine Zahl k' mit $A\langle p, x_0, y_0\rangle = 0$ für alle p mit $0^{k'} \sqsubseteq p$.
Und weil B_1 und B_2 stetig sind, gibt es eine Zahl k'' mit der Eigenschaft

$$\Omega \circ B_1(p_0) = 1 \Longrightarrow \Omega \circ B_1(p) = 1 \text{ für alle } p \text{ mit } 0^{k''} \sqsubseteq p$$

und

$$\Omega \circ B_2(p_0) = 1 \Longrightarrow \Omega \circ B_2(p) = 1 \text{ für alle } p \text{ mit } 0^{k''} \sqsubseteq p$$

Sei $k := \max\{k', k''\}$.
Sei $p_1 := 0^k 7 0^\omega$. Dann gilt $f(p_1) = 1$ und $A\langle p_1, x_0, y_0\rangle = 0$.
Also gilt $x_1 := \Omega \circ B_1(p_1) > x_0$ oder $y_1 := \Omega \circ B_2(p_1) > y_0$.
Dann gilt $f(p_1) = A\langle p_1, \Omega \circ B_1(p_1), \Omega \circ B_2(p_1)\rangle = A\langle p_1, x_1, y_1\rangle = 1$.
Mit dem analogen Argument wie oben gibt es eine Zahl l mit

$$A\langle p, x_1, y_1\rangle = 1 \text{ für alle } p \text{ mit } 0^k 7 0^l \sqsubseteq p$$

und

$$\Omega \circ B_1(p_1) = 1 \Longrightarrow \Omega \circ B_1(p) = 1 \text{ für alle } p \text{ mit } 0^k 7 0^l \sqsubseteq p$$

und

$$\Omega \circ B_2(p_1) = 1 \Longrightarrow \Omega \circ B_2(p) = 1 \text{ für alle } p \text{ mit } 0^k 7 0^l \sqsubseteq p$$

Sei $p_2 := 0^k 7 0^l 3 0^\omega$.
Dann gilt $x_2 := \Omega \circ B_1(p_2) > x_1$ oder $y_2 := \Omega \circ B_2(p_2) > y_1$
und $f(p_2) = A\langle p_2, \Omega \circ B_1(p_2), \Omega \circ B_2(p_2)\rangle = A\langle p_2, x_2, y_2\rangle = 0$.
Mit demselben Argument muss es eine Zahl m geben mit

$$A\langle p, x_2, y_2\rangle = 0 \text{ für alle } p \text{ mit } 0^k 7 0^l 3 0^m \sqsubseteq p$$

und

$$\Omega \circ B_1(p_2) = 1 \Longrightarrow \Omega \circ B_1(p) = 1 \text{ für alle } p \text{ mit } 0^k 7 0^l 3 0^m \sqsubseteq p$$

und

$$\Omega \circ B_2(p_2) = 1 \Longrightarrow \Omega \circ B_2(p) = 1 \text{ für alle } p \text{ mit } 0^k 7 0^l 3 0^m \sqsubseteq p$$

Sei $p_3 := 0^k 7 0^l 3 0^m 5 0^\omega$.
Dann folgt $x_3 := \Omega \circ B_1(p_2) > x_2$ oder $y_3 := \Omega \circ B_2(p_2) > y_2$

und $f(p_3) = A\langle p_3, \Omega \circ B_1(p_3), \Omega \circ B_2(p_3)\rangle = A\langle p_3, x_3, y_3\rangle = 1$.
Dann muss es Zahlen $x_0 \leq x_1 \leq x_2 \leq x_3 \in \{0,1\}$
und $y_0 \leq y_1 \leq y_2 \leq y_3 \in \{0,1\}$ geben mit
$(x_1 > x_0 \vee y_1 > y_0)$ und $(x_2 > x_1 \vee y_2 > y_1)$ und $(x_3 > x_2 \vee y_3 > y_2)$.
Das ist nicht möglich. Also ist f nicht Ω_2-stetig. □

Satz 6 (Ω_n-stetig $\not\Rightarrow$ Ω^n-stetig)**.**
Es gibt eine Funktion $f : \mathbb{B} \to \{0,1,2\}$, die Ω_2-stetig und nicht Ω^2-stetig ist.

Beweis. Definiere $f : \mathbb{B} \longrightarrow \{0,1,2\}$ durch

$$f(p) := \begin{cases} 1 & \text{falls } (\exists k) p(k) = 1 \\ 2 & \text{falls } (\exists k) p(k) = 2 \text{ und } (\forall k) p(k) \neq 1 \\ 0 & \text{sonst} \end{cases}$$

f ist Ω_2-stetig:
Definiere $B_1 : \mathbb{B} \longrightarrow \mathbb{B}$ durch

$$B_1(p)(n) := \begin{cases} 1 & \text{falls } p(n) = 1 \\ 0 & \text{sonst} \end{cases}$$

Definiere $B_2 : \mathbb{B} \longrightarrow \mathbb{B}$ durch

$$B_2(p)(n) := \begin{cases} 1 & \text{falls } p(n) = 2 \\ 0 & \text{sonst} \end{cases}$$

Definiere $A : \mathbb{B} \longrightarrow \mathbb{B}$ durch

$$A\langle p, x, y\rangle := \begin{cases} 1 & \text{falls } x = 1 \\ 2 & \text{falls } x = 0 \text{ und } y = 1 \\ 0 & \text{sonst} \end{cases}$$

für alle $p \in \mathbb{B}, x, y \in \mathbb{N}$.
A, B_1 und B_2 sind stetig und $f(p) = A\langle p, \Omega \circ B_1(p), \Omega \circ B_2(p)\rangle$.

f ist nicht Ω^2-stetig:
Aber es gibt keine stetigen Funktionen A, B_1 und B_2 mit
$f(p) = A\langle p, \Omega \circ B_1\langle p, \Omega \circ B_2(p)\rangle\rangle$.
Andernfalls gäbe es eine Zahl $z \in \{0,1\}$, so dass $A\langle p, z\rangle$ einen Funktionswert a für unendlich viele Folgen und ebenfalls einen davon unterschiedenen Wert b für unendlich viele Folgen hätte.
Sei $p' := 0^\omega$, $p_k := 0^k 2 0^\omega$ und $p_{k,j} := 0^k 2 0^j 1 0^\omega$.
Man nehme an, dass $\Omega \circ B_1\langle p', \Omega \circ B_2(p')\rangle = z$ und es unendlich viele k gibt mit $\Omega \circ B_1\langle p_k, \Omega \circ B_2(p_k)\rangle = z$.
Da A stetig ist und $f(p') = A\langle p', z\rangle = 0$, gibt es eine Zahl l mit $A\langle p, z\rangle = 0$ für alle p mit $0^l \sqsubseteq p$.
Dann gibt es eine Zahl $k > l$ mit $\Omega \circ B_1\langle p_k, \Omega \circ B_2(p_k)\rangle = z$ und $A\langle p_k, z\rangle = 0$.
Aber $f(p_k) = 2$.
Dasselbe würde gelten, wenn $\Omega \circ B_1\langle p', \Omega \circ B_2(p')\rangle = z$ und es unendlich viele k, j gäbe mit $\Omega \circ B_1\langle p_{k,j}, \Omega \circ B_2(p_{k,j})\rangle = z$.
Also gibt es eine Zahl $z \in \{0,1\}$ mit $\Omega \circ B_1\langle p_k, \Omega \circ B_2(p_k)\rangle = z$ und $\Omega \circ B_1\langle p_{k,j}, \Omega \circ$

$B_2(p_{k,j})\rangle = z$ für alle bis auf endlich viele k und j.
Sei $\Omega \circ B_1\langle p_k, \Omega \circ B_2(p_k)\rangle = z$.
Dann gibt es wegen der Stetigkeit von A eine Zahl l mit $A\langle p, z\rangle = 2$ für alle p mit $0^k 2 0^l \sqsubseteq p$.
Aber dann gibt es Zahlen $j > l$ mit $\Omega \circ B_1\langle p_{k,j}, \Omega \circ B_2(p_{k,j})\rangle = z$ und $A\langle p_{k,j}, z\rangle = 2$. Aber $f(p_{k,j}) = 1$.
Das ist ein Widerspruch. Also ist f nicht Ω^2-stetig. □

Definition 15 (C_n). *Für alle $n \in \mathbb{N}$ definiere $C_n : \mathbb{B} \longrightarrow \mathbb{B}$ durch*

$$C_n(p)(i) := \begin{cases} 0 & \textit{falls } i < n \textit{ und } (\exists k)p(k) = i+1 \\ 1 & \textit{falls } i < n \textit{ und } (\forall k)p(k) \neq i+1 \\ 0 & \textit{sonst} \end{cases}$$

Diese Definition von C_n wurde in [Myl92] gemacht und sollte nicht mit der Definition in [Bra03] verwechselt werden. In [Myl92] wurde der folgende Satz bewiesen:

Satz 7 (Vergleich von C_n und Ω_n).

$$f \leq_2 C_{n+1} \Longleftrightarrow f \textit{ ist } \Omega_n\textit{-stetig}$$

für alle $n \in \mathbb{N}$ und $f :\subseteq \mathbb{B} \to \mathbb{B}$ oder $f :\subseteq \mathbb{B} \to \mathbb{N}$.

Das bedeutet für alle f und $n \in \mathbb{N}$, f ist C_{n+1}-stetig, gdw. f Ω_n-stetig ist. Dabei heißt eine Funktion f C_n-stetig, gdw. es stetige Funktionen A und B gibt mit $f(p) = A\langle p, C_n \circ B(p)\rangle$ für alle $p \in Def(f)$.

Satz 8 (Vergleich von C_n und f_n).

$$C_n \equiv f_n \textit{ für alle } n \in \mathbb{N}.$$

Mit f_n sind die Funktionen aus Definition 4 gemeint.

Beweis.
$\underline{C_n \leq_2 f_n}$:
Definiere $B : \mathbb{B} \longrightarrow \mathbb{B}$ durch

$$B(p)(0) := \begin{cases} 1 & \text{falls } 1 \leq p(0) \leq n+1 \\ 0 & \text{sonst} \end{cases}$$

und

$$B(p)(k+1) := \begin{cases} 1 & \text{falls } 1 \leq p(k+1) \leq n+1 \\ & \text{und } p(k+1) \notin \{p(0), \ldots, p(k)\} \\ 0 & \text{sonst} \end{cases}$$

Definiere $A :\subseteq \mathbb{B} \longrightarrow \mathbb{B}$ durch

$$A\langle p, i\rangle(k) := \begin{cases} 0 & \text{falls } k < n \text{ und } (\exists l)(\exists x \leq l)(p(x) = k+1 \\ & \text{und } |\{p(0), \ldots, p(l)\} \cap \{1, \ldots, n\}| = i) \\ 1 & \text{falls } k < n \text{ und } (\exists l)(\forall x \leq l)(p(x) \neq k+1 \\ & \text{und } |\{p(0), \ldots, p(l)\} \cap \{1, \ldots, n\}| = i) \\ 0 & \text{falls } k \geq n \\ div & \text{sonst} \end{cases}$$

für alle $p \in \mathbb{B}, i \in \mathbb{N}$.
Dann gilt $C_n(p) = A\langle p, f_n \circ B(p)\rangle$ für alle $p \in \mathbb{B}$.
Die Idee hinter dem Beweis: Die Funktion B erzeugt eine 1 für alle unterschiedlichen Zahlen zwischen 1 und $n+1$ in p. Die Funktion f_n zählt die Vorkommen der Einsen und die Funktion A berechnet, welche Zahlen in p vorkommen.
$\underline{f_n \leq_2 C_n}$:
Definiere $B :\subseteq \mathbb{B} \longrightarrow \mathbb{B}$ durch

$$B(p)(0) := \begin{cases} 1 & \text{falls } p(0) = 1 \\ 0 & \text{sonst} \end{cases}$$

$$B(p)(k+1) := \begin{cases} z+1 & \text{falls } p(k+1) = 1 \text{ und } \sharp_1(\{p(0), \ldots, p(k+1)\}) \leq n \\ & \text{und } \max(\{B(p)(0), \ldots, B(p)(k)\} = z) \\ div & \text{falls } p(k+1) = 1 \text{ und } \sharp_1(\{p(0), \ldots, p(k)\}) > n \\ 0 & \text{sonst} \end{cases}$$

Definiere $A :\subseteq \mathbb{B} \longrightarrow \mathbb{N}$ durch

$$A\langle p, q\rangle := \max\{i \mid q(i) = 0 \text{ und } i < n\}$$

für alle $p, q \in \mathbb{B}$.
Dann gilt $f_n(p) = A\langle p, C_n \circ B(p)\rangle$ für alle $p \in \mathbb{B}$ und $n \in \mathbb{N}$. $\square$

3.2.2 $LLPO_\infty$

Definition 16 ($LLPO_\infty$)**.**
Sei $LLPO_\infty$ die Menge aller Funktionen f mit

$$\begin{aligned}&Def(f) = \{\langle p_1, p_2, \dots\rangle \mid \textit{ es gibt höchstens ein } k \textit{ mit } p_k \neq 0^\omega\}\\ &\textit{und } f(\langle p_1, p_2, \dots\rangle) = k \textit{ für ein } k \textit{ mit } p_k = 0^\omega.\end{aligned}$$

Satz 9 ($LLPO_\infty \leq_2 LLPO_n$)**.**

$$LLPO_\infty \leq_2 LLPO_n \textit{ für alle } n \geq 2.$$

Beweis. Sei $B_n : \mathbb{B} \longrightarrow \mathbb{B}$ definiert durch

$$B_n\langle p_1, p_2, \dots\rangle := \langle p_1, p_2, \dots, p_n\rangle$$

für $p_1, p_2, \dots \in \mathbb{B}$ und $A : \mathbb{B} \longrightarrow \mathbb{N}$ durch

$$A\langle p, i\rangle := i$$

für $p \in \mathbb{B}, i \in \mathbb{N}$.
Sei $g \in LLPO_n$.
Dann gilt $f \in LLPO_\infty$ für f, definiert durch $f\langle p_1, p_2, \dots\rangle :=$
$A\langle\langle p_1, p_2, \dots\rangle, g \circ B_n\langle p_1, p_2, \dots\rangle\rangle = g \circ B_n\langle p_1, p_2, \dots\rangle = g\langle p_1, p_2, \dots, p_n\rangle$.
Also gilt $LLPO_\infty \leq_2 LLPO_n$ für alle $n \geq 2$. □

Satz 10 ($LLPO_n \not\leq_2 LLPO_\infty$)**.**

$$LLPO_n \not\leq_2 LLPO_\infty \textit{ für alle } n \geq 2.$$

Der Beweis dieses Satzes ähnelt dem Beweis in [Wei92a] auf Seite 16-19.

Beweis. Nehme an, dass $LLPO_n \leq_2 LLPO_\infty$ für ein $n \geq 2$. Dann gibt es stetige Funktionen A und B, so dass für alle $f \in LLPO_\infty$

$$q \longmapsto A\langle q, f \circ B(q)\rangle \in LLPO_n \text{ für alle } q \in \mathbb{B}^n.$$

Behauptung: $B\langle 0^\omega, \dots, 0^\omega\rangle = \langle 0^\omega, \dots\rangle$.
Beweis der Behauptung: Für $i, k \in \mathbb{N}$ definiere

$$\begin{aligned}p_{k,i} &:= \langle p_1, p_2, \dots\rangle \text{ mit } p_j = 0^\omega \text{ für } j \neq k \text{ und } p_k = 0^i10^\omega\\ q_{k,i} &:= \langle p_1, p_2, \dots, p_n\rangle \text{ mit } p_j = 0^\omega \text{ für } j \neq k \text{ und } p_k = 0^i10^\omega.\end{aligned}$$

Annahme: $B\langle 0^\omega, \dots, 0^\omega\rangle = p_{k,i}$ für $i, k \in \mathbb{N}$.
Da B stetig ist, gibt es ein j mit

$$B\langle q_1, \dots, q_n\rangle \subseteq \langle p_1, \dots, p_{k-1}, 0^i1\mathbb{B}, p_{k+1}, \dots\rangle$$

für alle $q_1, \dots, q_n \in 0^j\mathbb{B}$ und $p_i \in \mathbb{B}$ für $1 \leq i < k$ und $k < i$.
Also gilt $B(q_{m,a}) = p_{k,i}$ für alle $1 \leq m \leq n$ und $a \geq j$.
Sei $f \in LLPO_\infty$ und sei b definiert durch

$$b := A\langle\langle 0^\omega, \dots, 0^\omega\rangle, f \circ B\langle 0^\omega, \dots, 0^\omega\rangle\rangle.$$

Für alle $a > j$ gilt nach Annahme für A, B und f

$$A\langle q_{b,a}, f \circ B(q_{b,a})\rangle \neq b.$$

Also gilt

$$A\langle\langle 0^\omega, \dots, 0^\omega\rangle, f(p_{k,i})\rangle = b \text{ (nach Def. von } b\text{) und}$$
$$A\langle q_{b,a}, f(p_{k,i})\rangle \neq b \text{ für alle } a \geq j.$$

A ist stetig und $q_{b,a} \to \langle 0^\omega, \dots, 0^\omega\rangle$ für $a \to \infty$.
Das ist ein Widerspruch. Damit ist die Behauptung bewiesen.

Definiere nun

$$M_k^\infty := \{\langle p_1, p_2, \dots\rangle \mid p_j = 0^\omega \text{ für } j \neq k \text{ und } p_k \in 0^*10^\omega\} \text{ und}$$
$$M_k^n := \{\langle p_1, p_2, \dots, p_n\rangle \mid p_j = 0^\omega \text{ für } j \neq k \text{ und } p_k \in 0^*10^\omega\}.$$

<u>Fall 1</u>:
Für jedes j mit $1 \leq j \leq n$ gibt es ein $k(j) \in \mathbb{N}$ mit $B(p) \in M_{k(j)}^\infty \cup \{\langle 0^\omega, \dots\rangle\}$ für unendlich viele $p \in M_j^n$.
Sei N_j diese unendliche Teilmenge von M_j^n.
Dann gibt es ein $k \in \mathbb{N}\backslash\{k(1), \dots, k(n)\}$.
Es gibt ein $f \in LLPO_\infty$ mit

$$f\langle 0^\omega, \dots\rangle = k \text{ und}$$
$$f(p) = k \text{ für alle } p \in M_{k(1)}^\infty \cup \dots \cup M_{k(n)}^\infty.$$

Dann gilt nach Annahme

$$f \circ B(p) = k$$

für alle $p \in N_1 \cup \dots \cup N_n \cup \{\langle 0^\omega, \dots, 0^\omega\rangle\}$.
Nach Annahme gibt es ein $g \in LLPO_n$ mit

$$g(p) = A\langle p, B \circ f(p)\rangle = A\langle p, k\rangle$$

für alle $p \in N := N_1 \cup \dots \cup N_n \cup \{\langle 0^\omega, \dots, 0^\omega\rangle\}$.
Dann ist g stetig auf N.
Sei $g\langle 0^\omega, \dots, 0^\omega\rangle = m$. Da N_m unendlich ist, gibt es Zahlen $i(0) < i(1) < \dots$, so dass $q_{m,i(j)} \in N_m$ für $j = 0, 1, 2, \dots$
Dann gilt $g(q_{m,i(j)}) \neq m$ für alle j und $q_{m,i(j)} \to \langle 0^\omega, \dots, 0^\omega\rangle$ für $j \to \infty$ und $g\langle 0^\omega, \dots, 0^\omega\rangle = m$. Also ist g nicht stetig auf $N_m \cup \{\langle 0^\omega, \dots, 0^\omega\rangle\}$. Das ist ein Widerspruch.

<u>Fall 2</u>:
Es gibt Zahlen j, so dass keine Zahl $k(j) \in \mathbb{N}$ existiert mit $B(p) \in M_{k(j)}^\infty \cup \{\langle 0^\omega, \dots\rangle\}$ für unendlich viele $p \in M_j^n$.

Sei die Menge all dieser Zahlen mit U bezeichnet und sei V definiert durch $V := \{1, \ldots, n\} \backslash U$. V mag leer sein.
Dann gibt es für jedes $j \in V$ ein $k(j) = \{x_j\} \subseteq \mathbb{N}$ mit $B(p) \in M_{x_j}^{\infty} \cup \{\langle 0^\omega, \ldots \rangle\}$ für unendlich viele $p \in M_j^n$.
Sei N_j diese unendliche Teilmenge von M_j^n.
Definiere k durch

$$k := \begin{cases} \max\{x \mid x \in k(y), y \in V\} + 1 & \text{falls } V \neq \emptyset \\ 0 & \text{sonst} \end{cases}$$

Andernfalls (falls $j \notin V$) definiere $k(j)$ durch $k(j) := \mathbb{N} \backslash \{1, \ldots, k\}$ für alle $j \in U$.
Sei N_j die unendliche Teilmenge von allen $p \in M_j^n$, so dass $B(p) \in M_x^{\infty}$ für ein $x \in k(j)$.
Es gibt ein $f \in LLPO_\infty$ mit

$$\begin{aligned} &f\langle 0^\omega, \ldots \rangle = k \text{ und} \\ &f(p) = k \text{ für alle } p \in \bigcup_{\substack{x \in k(i) \\ 1 \leq i \leq n}} M_x^{\infty}. \end{aligned}$$

Nach Annahme gilt

$$f \circ B(p) = k$$

für alle $p \in N_1 \cup \ldots \cup N_n \cup \{\langle 0^\omega, \ldots, 0^\omega \rangle\}$.
Nach Annahme gibt es ein $g \in LLPO_n$ mit

$$g(p) = A\langle p, B \circ f(p) \rangle = A\langle p, k \rangle$$

für alle $p \in N := N_1 \cup \ldots \cup N_n \cup \{\langle 0^\omega, \ldots, 0^\omega \rangle\}$.
Dann ist g stetig auf N.
Sei $g\langle 0^\omega, \ldots, 0^\omega \rangle = m$. Falls $m \in V$, dann gibt es Zahlen $h(0) < h(1) < \ldots$, so dass $q_{m,h(j)} \in N_m$ für $j = 0, 1, 2, \ldots$
Dann gilt $g(q_{m,h(j)}) \neq m$ für alle j und $q_{m,h(j)} \to \langle 0^\omega, \ldots, 0^\omega \rangle$ für $j \to \infty$ und $g\langle 0^\omega, \ldots, 0^\omega \rangle = m$.
Also ist g nicht stetig auf $N_m \cup \{\langle 0^\omega, \ldots, 0^\omega \rangle\}$. Das ist ein Widerspruch.
Falls $m \in U$, dann gilt für jede Folge $p(0) < p(1) < \ldots \in N_m$
$p(i) \to \langle 0^\omega, \ldots, 0^\omega \rangle$ für $i \to \infty$.
Dann gilt $g(p(i)) \neq m$ für alle i und $p(i) \to \langle 0^\omega, \ldots, 0^\omega \rangle$ für $i \to \infty$ und $g\langle 0^\omega, \ldots, 0^\omega \rangle = m$.
Also ist g nicht stetig auf $N_m \cup \{\langle 0^\omega, \ldots, 0^\omega \rangle\}$. Das ist auch ein Widerspruch.

Also ist $LLPO_n \leq LLPO_\infty$ falsch für alle $n \geq 2$. □

Es gibt einen anderen Beweis, der wesentlich einfacher ist und den Satz 3 verwendet, der von Klaus Weihrauch in [Wei92a] bewiesen wurde.

Beweis. Nehme an, es gibt eine Zahl n mit $LLPO_n \leq LLPO_\infty$.
Dann gilt $LLPO_\infty \leq LLPO_{n+1} \leq LLPO_n \leq LLPO_\infty$.
Dies impliziert $LLPO_{n+1} \equiv LLPO_n$.
Aber das ist ein Widerspruch zu Satz 3. □

Natürlich ist $LLPO_\infty$ nicht stetig:

Satz 11 ($LLPO_\infty$ ist nicht stetig)**.**
Es gibt keine stetige Funktion f mit

$$f\langle p_1, p_2, \ldots\rangle = k$$

für eine Zahl k mit $p_k = 0^\omega$, falls es höchstens ein i gibt mit $p_i \neq 0^\omega$.

Beweis. Nehme an, es gibt eine stetige Funktion f mit der im Satz angegebenen Eigenschaft. Sei $p_i := 0^\omega$ für alle $i \geq 1$. Dann gilt $f\langle p_1, p_2, \ldots\rangle = k$ für ein $k > 0$.
Da f stetig ist, gibt es eine Zahl $z \in \mathbb{N}$ mit $f(p) = k$ für alle p mit $0^z \sqsubseteq p$.
Definiere q_i durch $q_i := 0^\omega$ für alle $i \neq k$ und $q_k := 0^z 10^\omega$.
Dann gilt $f\langle q_1, q_2, \ldots\rangle = k$, aber $q_k \neq 0^\omega$. Dies ist ein Widerspruch. □

Ω ist die einfachste unstetige Funktion. Es ergibt sich die Frage, ob $LLPO_\infty$ die einfachste Funktionenmenge ist, die nicht stetig ist.

Satz 12 (Ω einfachste unstetige Funktion)**.**
Wenn $f \leq_2 \Omega$ für $f :\subseteq \mathbb{B} \longrightarrow \mathbb{N}$, dann ist f stetig oder $f \equiv_2 \Omega$.

Beweis. Wenn $f :\subseteq \mathbb{B} \longrightarrow \mathbb{N}$ unstetig ist, dann gibt es ein $p \in \mathbb{N}$ und ein $k \in \mathbb{N}$ mit $f(p) = k$ und in jeder Umgebung von p eine Folge q mit $f(q) \neq k$.
Dann gibt es zu jedem $i \in \mathbb{N}$ ein $q_i \in \mathbb{B}$ mit $[[p]]_i \sqsubseteq q_i$ und $f(q_i) \neq k$.
Definiere $B : \mathbb{B} \longrightarrow \mathbb{B}$ durch

$$B(r)(i) := \begin{cases} p(i) & \text{falls } (\forall j \leq i) r(j) = 0 \\ q_j(i) & \text{falls } j = \min\{s \leq i | r(s) \neq 0\} \end{cases}$$

und $A : \mathbb{B} \longrightarrow \mathbb{N}$ durch

$$A\langle q, x\rangle := \begin{cases} 0 & \text{falls } x = k \\ 1 & \text{sonst} \end{cases}$$

für $q \in \mathbb{B}, x \in \mathbb{N}$ und A und B sind stetig.
Dann gilt $\Omega(p) = A\langle p, f \circ B(p)\rangle$ für alle $p \in \mathbb{B}$, also $\Omega \leq_2 f$. Aus $f \leq_2 \Omega$ folgt dann $\Omega \equiv_2 f$. □

Satz 13 ($LLPO_\infty$ nicht einfachste Menge von unstetigen Funktionen).
Es gibt eine Menge P von Funktionen $f :\subseteq \mathbb{B} \longrightarrow \mathbb{N}$ mit der Eigenschaft

- $P \leq_2 LLPO_\infty$
- $LLPO_\infty \not\leq_2 P$
- *es gibt keine stetige Funktion $g \in P$*

Beweis. Sei P definiert durch $P := LLPO_\infty \cup \{f_k\}$ mit
$Def(f_k) = \{\langle p_1, p_2, \ldots\rangle |$es gibt höchstens ein j mit $p_j \neq 0^\omega\}$ und

$$f_k\langle p_1, p_2, \ldots\rangle := \begin{cases} j & \text{falls } p_j \neq 0^\omega \\ k & \text{falls } (\forall j > 0) p_j = 0^\omega \end{cases}$$

Es gilt $P \leq_2 LLPO_\infty$, da $LLPO_\infty \subseteq P$ und $Def(P) = Def(LLPO_\infty)$. P enthält auch keine stetige Funktion.
Aber $LLPO_\infty \leq_2 P$ ist nicht richtig. Denn seien A und B stetige Funktionen mit $p \mapsto A(p, g \circ B(p)) \in LLPO_\infty$ für alle $g \in P$. Aus $g \circ B(p) = k$ ist nicht zu ersehen, ob $p_k = 0^\omega$ oder $p_k \neq 0^\omega$:
Sei $p := \langle p_1, p_2, \ldots\rangle := \langle 0^\omega, 0^\omega, \ldots\rangle$.
Sei $f(p) = A\langle p, g \circ B(p)\rangle$ für $g \in P$ und $g \circ B(p) = k$. Dann gibt es ein $j \in \mathbb{N}$ und ein $i \in \mathbb{N}$ mit $f(q) = A\langle q, k\rangle = j$ für alle q mit $0^i \sqsubseteq q$. Definiere q' durch $q'_j := 0^i 10^\omega$ und $q'_n := 0^\omega$ für $n \neq j$.
Es gibt eine Funktion $g \in P$ mit $g \circ B(q') = k$. Dann folgt $A\langle q', g \circ B(q')\rangle = A\langle q', k\rangle = j$, aber $f(q') \neq j$ für alle $f \in LLPO_\infty$. □

Auch wenn man $LLPO_\infty$ als mehrwertige Funktion auffasst, handelt es sich nicht um die einfachste unstetige Funktion. Dafür definieren wir:

Definition 17 (mehrwertige unstetige Funktion).
Eine mehrwertige Funktion f heißt stetig, wenn es eine stetige einwertige Funktion g gibt mit $f(p) \neq \emptyset \Longrightarrow g(p) \in f(p)$. Andernfalls heißt f unstetig.

Definition 18 ($\leq_2$ für mehrwertige Funktionen).
Für zwei mehrwertige Funktionen f und g gilt $f \leq_2 g$, gdw. es stetige Funktionen A und B gibt mit $A\langle p, k\rangle \in f(p)$ für alle $p \in \mathbb{B}$ und $k \in g \circ B(p)$.

Satz 14 ($LLPO_\infty$ nicht einfachste mehrwertige unstetige Funktion).
Es gibt eine mehrwertige unstetige Funktion $P :\subseteq \mathbb{B} \longrightarrow \mathbb{N}$ mit $P \leq_2 LLPO_\infty$ und $LLPO_\infty \not\leq_2 P$.

Beweis. Definiere die mehrwertige Funktion P durch

$$Def(P) := \{\langle p_1, p_2, \ldots\rangle \in \mathbb{B} | \text{es gibt höchstens ein } k \text{ mit } p_k \neq 0^\omega\}$$

und

$$\begin{aligned} k \in P(\langle p_1, p_2, \ldots\rangle) :\Longleftrightarrow & (k \geq 3 \text{ und } p_{k-2} = 0^\omega) \\ & \text{oder } (k = 1 \text{ und } (\exists i) p_i = 10^\omega) \\ & \text{oder } (k = 1 \text{ und } (\forall i) p_i = 0^\omega) \\ & \text{oder } (k = 2 \text{ und } (\exists i)(\exists j > 0) p_i = 0^j 10^\omega) \\ & \text{oder } (k = 2 \text{ und } (\forall i) p_i = 0^\omega) \end{aligned}$$

$P(p)$ hat also beliebige Werte, falls $p_k = 0^\omega$ für alle $k > 0$. Falls $p_k = 0^j 10^\omega$ für ein k, ist das Ergebnis $n + 2$ für ein $n \neq k$ oder 1, falls die 1 in einer Folge an erster Stelle kommt, bzw. 2, falls die 1 in einer Folge nicht an erster Stelle kommt.
P ist nicht stetig. Der Beweis ist einfach.
$\underline{P \leq_2 LLPO_\infty}$: Definiere B durch $B(p) := p$ und A durch $A\langle p, k\rangle := k+2$. Dann gilt $A\langle p, k\rangle \in P(p)$ für alle $k \in LLPO_\infty \circ B(p)$.
$\underline{LLPO_\infty \not\leq_2 P}$: Nehme an, dass $LLPO_\infty \leq_2 P$ gilt. Es gibt dann stetige Funktionen A und B mit $A\langle p, k\rangle \in LLPO_\infty(p)$ für alle $k \in P \circ B(p)$.
Sei q die Folge mit $q_n = 0^\omega$ für alle n. Es ist klar, dass $B(q) = q$ gelten muss. Dann gilt $1 \in P \circ B(q)$ und ebenso $2 \in P \circ B(q)$. Es gelte $A\langle q, 1\rangle = k$ und $A\langle q, 2\rangle = l$.
Wegen der Stetigkeit von A gibt es ein $j' > 0$ mit $A\langle p, 2\rangle = l$ für alle p mit $0^{j'} \sqsubseteq p$.
Wegen der Stetigkeit von B gibt es ein $j^* > 0$ mit $0 \sqsubseteq B(p)(l)$ für alle p mit $0^{j^*} \sqsubseteq p$.
Definiere $j := \max\{j', j^*\}$.
Definiere q' durch $q'_l := 0^j 10^\omega$ und $q'_n := 0^\omega$ für $n \neq l$.
Dann gilt $0 \sqsubseteq B(q')(l)$. Deshalb folgt $2 \in P \circ B(q')$. Daraus folgt $A\langle q', 2\rangle = l$.
Aber $l \notin LLPO_\infty(q')$. □

Hier ist ein Beispiel für eine mehrwertige unstetige Funktion, die äquivalent zu $LLPO_\infty$ ist:

Beispiel 1. Sei die mehrwertige Funktion $P1$ definiert durch

$$Def(P1) := \{\langle p_1, p_2, \ldots\rangle | \text{ es gibt höchstens zwei Zahlen } k \text{ mit } p_k \neq 0^\omega\}$$

und $P1\langle p_1, p_2, \ldots\rangle := \{\langle k_1, k_2\rangle | \; p_{k_1} = 0^\omega \text{ oder } p_{k_2} = 0^\omega\}$.
Es gilt $P1 \equiv_2 LLPO_\infty$.
$\underline{P1 \leq_2 LLPO_\infty}$:
Definiere B durch

$$\langle B(p)\rangle_{\langle i,j\rangle}(n) := \begin{cases} 1 & \text{falls } (\exists x \leq n)\langle p\rangle_i(x) = 1 \\ & \text{und } (\exists x \leq n)\langle p\rangle_j(x) = 1 \\ & \text{und } (\langle p\rangle_i(n) = 1 \text{ oder } \langle p\rangle_j(n) = 1) \\ & \text{und } i \leq j \\ 0 & \text{sonst} \end{cases}$$

Definiere A durch $A\langle p,a\rangle := a$ für alle $p \in \mathbb{B}$ und $a \in \mathbb{N}$.
Dann gilt $A\langle p,\langle k_1,k_2\rangle\rangle \in P1(p)$ für alle $\langle k_1,k_2\rangle \in g \circ B(p)$.
$\underline{LLPO_\infty \leq_2 P1}$:
Definiere B durch

$$\langle B(p)\rangle_i(n) := \begin{cases} 1 & \text{falls } (\exists a,b) p_{\langle a,b\rangle}(m) = 1 \\ & \text{und } (i = a \text{ oder } i = b) \\ & \text{und } \langle a,b\rangle \leq n \text{ und } m \leq n \\ & \text{und } \langle B(p)\rangle_i(j) = 0 \text{ für alle } 0 < j < n \\ 0 & \text{sonst} \end{cases}$$

Definiere A durch $A\langle p,\langle i,j\rangle\rangle := \langle i,j\rangle$ für alle $p \in \mathbb{B}$ und $i,j \in \mathbb{N}$.
Dann gilt $A\langle p,k\rangle \in LLPO_\infty(p)$ für alle $p \in Def(LLPO_\infty)$ und $k \in P1 \circ B(p)$.
Denn für $p = \langle 0^\omega, 0^\omega, \ldots\rangle$ gilt $\langle i,j\rangle \in LLPO_\infty(p)$ für alle $\langle i,j\rangle \in \mathbb{N}$.
Sei $p \in Def(LLPO_\infty)$ eine Folge mit $p_{\langle i,j\rangle} \neq 0^\omega$. Dann gilt $B(p)_i \neq 0^\omega$ und $B(p)_j \neq 0^\omega$. Hieraus folgt $\langle i,j\rangle \notin P1 \circ B(p)$. Daraus folgt $A\langle p,k\rangle \neq \langle i,j\rangle$ für alle $k \in P1 \circ B(p)$. Und daraus folgt $A\langle p,k\rangle \in LLPO_\infty(p)$ für alle $k \in P1 \circ B(p)$.

Es gibt auch Funktionenmengen P, die mit $LLPO_\infty$ nicht vergleichbar sind.

Beispiel 2. Sei $P2 := \{f_1, f_2\}$ eine Menge mit

$$Def(f_1) = Def(f_2) = \{\langle p_1, p_2, \ldots\rangle |\text{ es gibt höchstens eine Zahl } k \text{ mit } p_k \neq 0^\omega\}$$

und

$$f_1\langle p_1,p_2,\ldots\rangle := \begin{cases} 0 & \text{falls } p_1 = 0^\omega \\ 1 & \text{sonst} \end{cases}$$

und

$$f_2\langle p_1,p_2,\ldots\rangle := \begin{cases} 1 & \text{falls } p_1 = 0^\omega \\ 0 & \text{sonst} \end{cases}$$

Dann gilt $P2 \not\leq_2 LLPO_\infty$ und $LLPO_\infty \not\leq_2 P2$.

$\underline{P2 \not\leq_2 LLPO_\infty}$: Annahme: $P2 \leq_2 LLPO_\infty$. Dann gibt es stetige Funktionen A und B mit $p \mapsto A\langle p, g\circ B(p)\rangle = f_1$ oder $p \mapsto A\langle p, g\circ B(p)\rangle = f_2$ für $g \in LLPO_\infty$.

Definiere P_k durch $P_k := \{\langle p_1,p_2,\ldots\rangle \in \mathbb{B} | p_k \neq 0^\omega \text{ und } p_j = 0^\omega \text{ für } j \neq k\}$.

Sei $q := 0^\omega$. Dann gilt $B(q) = 0^\omega$. Andernfalls würde es eine Zahl k und eine Zahl N geben mit $B(p) \in P_k$ für alle $p \in \mathbb{B}$ mit $0^N \sqsubseteq p$.
Definiere g durch $g(p) := k+1$, falls $p \in P_k$, k sonst.
Dann gilt $A\langle q, g\circ B(q)\rangle = A\langle q, k+1\rangle = a$ mit $a = 0$ oder $a = 1$.
Da A stetig ist, gibt es eine Zahl N' mit $A\langle p, k+1\rangle = a$ für alle p mit $0^{N'} \sqsubseteq p$.
Definiere $N^* := \max\{N, N'\}$.
Definiere q' durch $\langle q'\rangle_1 := 0^{N^*}10^\omega$ und $\langle q'\rangle_j := 0^\omega$ für alle $j \neq 1$. Das impliziert $a = A\langle q, g\circ B(q)\rangle = A\langle q, k+1\rangle = A\langle q', k+1\rangle = A\langle q', g\circ B(q')\rangle = 1 - a$.
Das ist ein Widerspruch.

Sei $g \in LLPO_\infty$ definiert durch $g(p) := 1$ für $\langle p\rangle_1 = 0^\omega$ und $g(p) := 2$ sonst.
Annahme: $f_1(p) = A\langle p, g\circ B(p)\rangle$ für alle $p \in \mathbb{B}$.

Dann gilt $f_1(q) = 0 = A\langle q, g \circ B(q)\rangle = A\langle q, 1\rangle$. Dann gibt es eine Zahl N mit $f_1(p) = 0 = A\langle p, 1\rangle$ für alle p mit $0^N \sqsubseteq p$.
Definiere q' durch $\langle q'\rangle_1 := 0^N 10^\omega$ und $\langle q'\rangle_j := 0^\omega$ für $j \neq 1$.
$f_1(q') = 1 = A\langle q', g \circ B(q')\rangle$ impliziert dann $g \circ B(q') = 2$.
Also gilt $B(q') \in P_1$. Dann gilt $A\langle q', g \circ B(q')\rangle = A\langle q', 2\rangle = 1$.
Definiere nun $g' \in LLPO_\infty$ durch $g'(p) := 2$ für $\langle p\rangle_2 = 0^\omega$ und $g'(p) := 1$ sonst.
Dann ist $A\langle p, g' \circ B(p)\rangle = f_1(p)$ nicht wahr für alle $p \in \mathbb{B}$, und zwar wegen $B(q) = 0^\omega$. Das impliziert $f_1(q) = 1$. Das ist ein Widerspruch.
Also gilt $f_2(p) = A\langle p, g' \circ B(p)\rangle$ für alle $p \in \mathbb{B}$.
Dann folgt $f_2(q) = 1 = A\langle q, g' \circ B(q)\rangle = A\langle q, 2\rangle$. Dann gilt $f_2(q') = 0 = A\langle q', g' \circ B(q')\rangle = A\langle q', 1\rangle$. Also muss B die Folge q' im ersten Fall auf eine Folge in P_1 und im zweiten Fall auf eine Folge in P_2 abbilden. Das ist ein Widerspruch.
Genauer: Es gibt eine Zahl $N \in \mathbb{N}$ mit $B(p) \in P_1$ für alle $p \neq 0^\omega$ und $0^N \sqsubseteq p$.
Auch im zweiten Fall gibt es eine Zahl $N' \in \mathbb{N}$ mit $B(p) \in P_2$ für alle $p \neq 0^\omega$ und $0^{N'} \sqsubseteq p$.
Definiere $N^* := \max\{N, N'\}$. Dann gilt $B(p) \in P_1 \cap P_2$ für alle $p \neq 0^\omega$ und $0^{N^*} \sqsubseteq p$.
Definiere q^* durch $\langle q^*\rangle_1 := 0^{N^*} 10^\omega$ und $\langle q^*\rangle_j := 0^\omega$ für alle $j \neq 1$. Dann gilt $B(q^*) \in P_1 \cap P_2 \notin Def(LLPO_\infty)$. Daraus folgt $A\langle q^*, g \circ B(q^*)\rangle = A\langle q^*, div\rangle = div$. Das ist ein Widerspruch.

$LLPO_\infty \not\leq_2 P2$:
Annahme: $LLPO_\infty \leq_2 P2$. Dann existieren stetige Funktionen A und B mit $p \mapsto A\langle p, f_1 \circ B(p)\rangle \in LLPO_\infty$ und $p \mapsto A\langle p, f_2 \circ B(p)\rangle \in LLPO_\infty$.
Sei $q := 0^\omega$. Sei $A\langle q, f_1 \circ B(q)\rangle = k$ und $f_1 \circ B(q) = 1$ o.B.d.A.[4]
Dann gibt es ein $j \in \mathbb{N}$ mit $A\langle p, 1\rangle = k$ für alle p mit $0^j \sqsubseteq p$.
Definiere q' durch $\langle q'\rangle_k := 0^j 10^\omega$ und $\langle q'\rangle_i := 0^\omega$ für alle $i \neq k$.
Dann gilt $A\langle q', 1\rangle = k$. Also folgt $f_1 \circ B(q') = 0$.
Aber dann gilt $f_2 \circ B(q) = 0$ und $f_2 \circ B(q') = 1$ und damit $A\langle q, 0\rangle = k$ und $A\langle q', 1\rangle \neq k$. Das ist ein Widerspruch.

Hier ist ein anders Problem, das schwieriger als $LLPO_\infty$ ist.

Beispiel 3. Sei $LLPO'_\infty$ die Menge aller Funktionen f mit der Eigenschaft

$$Def(f) = \{\langle p_1, p_2, \ldots\rangle |\text{ es gibt endlich viele Zahlen } k \text{ mit } p_k \neq 0^\omega\}$$

und $f\langle p_1, p_2, \ldots\rangle = \langle n, k_1, \ldots, k_n\rangle$, wobei eine Zahl m existiert mit $1 \leq m \leq n$ und $p_{k_m} = 0^\omega$.

Es kann also endlich viele Folgen geben, die keine Nullfolgen sind, und die Funktionen in $LLPO'_\infty$ können endlich viele Antworten geben, von denen eine korrekt sein muss.

Dann gilt $LLPO'_\infty \not\leq_2 LLPO_\infty$ und $LLPO_\infty \leq_2 LLPO'_\infty$.

$LLPO'_\infty \not\leq_2 LLPO_\infty$: Nehme an, dass $LLPO'_\infty \leq_2 LLPO_\infty$. Dann existieren stetige Funktionen A und B mit $p \mapsto A\langle p, g \circ B(p)\rangle \in LLPO'_\infty$ für alle $g \in LLPO_\infty$.

[4] Die Annahme $f_1 \circ B(q) = 0$ führt ebenfalls zum Widerspruch.

Definiere p durch $\langle p\rangle_i := 0^\omega$ für alle $i \geq 1$. Dann gilt $B(p) = 0^\omega$. Andernfalls würde es eine Zahl k geben mit $\langle B(p)\rangle_k \neq 0^\omega$. Nehme an, dass $\langle n, x_1, \ldots, x_n\rangle = A\langle p, g \circ B(p)\rangle$ für ein spezielles $g \in LLPO_\infty$. Wegen der Stetigkeit von A und B würde es eine Zahl y geben mit $\langle n, x_1, \ldots, x_n\rangle = A\langle q, g \circ B(q)\rangle$ für alle q mit $0^y \sqsubseteq q$. Definiere $r \in \mathbb{B}$ durch $\langle r\rangle_{x_i} := 0^y 10^\omega$ für alle $1 \leq i \leq n$ und $\langle r\rangle_j := 0^\omega$ für alle $j \neq x_i$. Dann folgt $\langle n, x_1, \ldots, x_n\rangle = A\langle r, g \circ B(r)\rangle$, aber $\langle r\rangle_{x_i} \neq 0^\omega$ für alle $1 \leq i \leq n$. Also gilt $B(0^\omega) = 0^\omega$.

Nun definiere p durch $\langle p\rangle_1 := 10^\omega$ und $\langle p\rangle_i := 0^\omega$ für alle $i > 1$.
$B(p) = 0^\omega$ würde auch zu einem Widerspruch führen. Also gilt $B(p) = q$, wobei $\langle q\rangle_i \neq 0^\omega$ für ein $i \in \mathbb{N}$ und $\langle q\rangle_j = 0^\omega$ für alle $j \neq i$.
Sei $\langle n, x_1, \ldots, x_n\rangle = A\langle p, g \circ B(p)\rangle$. Wegen der Stetigkeit von A und B würde es eine Zahl y geben mit $\langle n, x_1, \ldots, x_n\rangle = A\langle q, g \circ B(q)\rangle$ für alle q mit $\langle q\rangle_1 = 10^\omega$ und $0^y \sqsubseteq \langle q\rangle_j$ für alle $j > 1$.
Nun definiere $q \in \mathbb{B}$ durch $\langle q\rangle_1 := 10^\omega$ und $\langle q\rangle_{x_i} := 0^y 10^\omega$ für alle $1 \leq i \leq n$ und $\langle q\rangle_j := 0^\omega$ für alle $j \neq x_i$.
Dann folgt $\langle n, x_1, \ldots, x_n\rangle = A\langle q, g \circ B(q)\rangle$, aber $\langle q\rangle_{x_i} \neq 0^\omega$ für alle $1 \leq i \leq n$.
Das ist ein Widerspruch.

<u>$LLPO_\infty \leq_2 LLPO'_\infty$</u>: Definiere B durch

$$B(p)\langle i, j\rangle := \begin{cases} 1 & \text{falls } p\langle i, j\rangle = 1 \\ 1 & \text{falls } (\exists l < j) p\langle k, l\rangle = 1 \text{ und } i + j = k + l \\ 0 & \text{sonst} \end{cases}$$

Dann ist B stetig und $B(0^\omega) = 0^\omega$ und $\langle B\langle p_0, p_1, \ldots\rangle\rangle_j \neq 0^\omega$ für $j \leq k$ und $\langle B\langle p_0, p_1, \ldots\rangle\rangle_j = 0^\omega$ für $j > k$, falls $p_k \neq 0^\omega$ und $p_i = 0^\omega$ für $i \neq k$.
Falls $p_k(l) = 1$, dann gilt $B(p)_i(j) = 1$ mit $j = k + l - i$. Für $i < k$ bedeutet das $j = k + l - i > l$. Für $i > k$ bedeutet das $j = k + l - i < l$. Also gilt $B(p)_i(j) = 0$ für alle $i > k$ und $j \in \mathbb{N}$.
Definiere A durch $A\langle p, \langle n, a_1, \ldots, a_n\rangle\rangle := \max\{a_1, \ldots, a_n\}$.
Dann gilt $p \mapsto A\langle p, g \circ B(p)\rangle \in LLPO_\infty$ für alle $g \in LLPO'_\infty$.

3.2.3 $LLPO_{n,m}$

Definition 19 ($LLPO_{n,m}$).
Für alle $n \geq m \geq 0$ sei $LLPO_{n,m}$ die Menge aller Funktionen f mit

$$Def(f) = \{\langle p_1, p_2, \ldots, p_n\rangle \mid \text{es gibt höchstens } m \text{ Zahlen } j \text{ mit } p_j \neq 0^\omega \text{ und mindestens ein } k \text{ mit } p_k = 0^\omega\}$$

und $f\langle p_1, p_2, \ldots, p_n\rangle = k$ für ein k mit $p_k = 0^\omega$.

Die Funktionenmenge $MLPO_n$ ist ein Spezialfall von $LLPO_{n,m}$:

$$MLPO_n = LLPO_{n,n-1} \text{ für alle } n \geq 2.$$

Die Funktionenmenge $LLPO_n$ ist ebenfalls ein Spezialfall von $LLPO_{n,m}$:

$$LLPO_n = LLPO_{n,1} \text{ für alle } n \geq 2.$$

Für alle $n \in \mathbb{N}$ ist $LLPO_{n,0}$ stetig. Denn für $n > 0$ ist die konstante Funktion f mit $f\langle p_1, p_2, \ldots, p_n\rangle := 1$ in $LLPO_{n,0}$. Für $n = 0$ ist die nirgendwo definierte Funktion f mit $f() := div$ in $LLPO_{n,0}$.

Für alle $n > 0$ gilt $LLPO_{n,n} \equiv_2 LLPO_{n,n-1}$. Somit gilt $LLPO_{1,1} \equiv_2 LLPO_{1,0}$. Also ist auch $LLPO_{1,1}$ stetig. Für alle $n > m > 0$ und damit auch für alle $n = m > 1$ gilt dagegen: $LLPO_{n,m}$ ist unstetig. Denn aus dem nächsten Satz folgt für alle $n > m > 0$ $LLPO_{n,1} \leq_2 LLPO_{n,m}$.

Satz 15 ($LLPO_{n,m} \leq_2 LLPO_{l,k}$).
Sei $n > m > 0$ und $l > k > 0$.
Dann gilt $LLPO_{n,m} \leq_2 LLPO_{l,k}$
$\iff (m \leq k \text{ und } k \geq m(l \text{ div } n) + \min(l \bmod n, m))$.

Hierbei sind *div* und *mod* definiert durch

$$z = a \ div \ b :\iff z = \max\{i \in \mathbb{N} | i \leq \frac{a}{b}\}$$

und

$$z = a \ mod \ b :\iff z = a - (a \ div \ b) \cdot b$$

Für $LLPO_{n,m} \leq_2 LLPO_{l,k}$ bedeutet die Bedingung $m \leq k$, dass die Zahl der Eingabefolgen ungleich 0^ω von $LLPO_{n,m}$ kleiner oder gleich der Zahl der Eingabefolgen ungleich 0^ω von $LLPO_{l,k}$ sein muss.

Die Bedingung $l > m$ muss auch erfüllt sein. Aber sie wird ausgelassen, da sie schon direkt aus $l > k$ und der ersten Bedingung $k \geq m$ folgt.

Die Idee hinter der Formel $k \geq m(l \ div \ n) + \min(l \ mod \ n, m))$ ist die folgende:

Wenn wir das Problem $LLPO_{n,m}$ auf $LLPO_{l,k}$ mit einem unbekannten k reduzieren möchten, können wir die n Folgen des ersten Problems kopieren und sie als Argumente für $LLPO_{l,k}$ verwenden. Aber wenn $l > n$, müssen Folgen auf mehr als eine Eingabefolge von $LLPO_{l,k}$ abgebildet werden. Wenn wir die Eingabefolgen möglichst gleich auf die Eingabefolgen von $LLPO_{l,k}$ verteilen wollen (und das ist die beste Strategie), dann haben wir jede Folge auf l *div* n Folgen abzubilden. Aber falls l nicht durch n teilbar ist, müssen l *mod* n Folgen jeweils auf $(l \ div \ n) + 1$ Folgen abgebildet werden.
Es kann m von n Folgen geben, die keine Nullfolgen sind. Also produziert die Abbildung $m(l \ div \ n)$ Folgen, die eventuell keine Nullfolgen sind. Aber l *mod* n Folgen werden auf eine zusätzliche Folge abgebildet. Also können l *mod* n weitere Folgen auf eine Folge ungleich 0^ω abgebildet werden. Das ist allerdings nur möglich, falls $m \geq (l \ mod \ n)$. Wenn $m < (l \ mod \ n)$, können nur m weitere Folgen auf eine Folge ungleich 0^ω abgebildet werden. So erzeugt die Abbildung im schlechtesten Fall höchstens $m(l \ div \ n) + \min(l \ mod \ n, m))$ Folgen ungleich 0^ω.

Beispiel 4. $LLPO_{4,3} \leq_2 LLPO_{10,8}$.
Setze $n = 4$, $m = 3$ und $l = 10$ und $k = 8$.
Wir können leicht $m \leq k$ verifizieren.
Wir berechnen den minimalen Wert für k:
$k = 3(10 \text{ div } 4) + \min(10 \text{ mod } 4, 3)$
$= 3 \cdot 2 + \min(2, 3) = 6 + 2 = 8.$

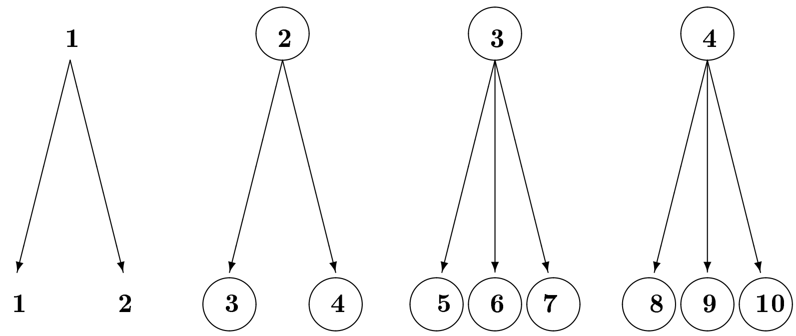

Abbildung 2: $LLPO_{4,3} \leq LLPO_{10,8}$ (Beste Strategie)

In diesem Beispiel und mit dieser Abbildung enthalten im schlechtesten Fall die zweite, dritte und vierte Folge eine 1. Sie werden auf die dritte bis zehnte Eingabefolge von $LLPO_{10,k}$ abgebildet. Damit sind 8 von den 10 Eingabefolgen von $LLPO_{10,k}$ keine Nullfolgen.

Wenn die Abbildung keine Gleichverteilung versucht, erhalten wir eventuell mehr Folgen, die keine Nullfolgen sind:

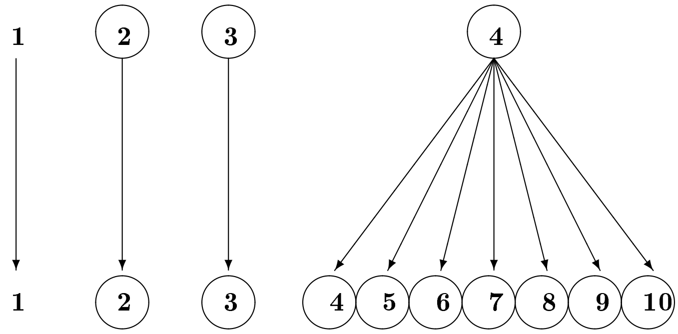

Abbildung 3: $LLPO_{4,3} \leq LLPO_{10,9}$ (Schlechte Strategie)

Beweis von Satz 15. (der Richtung „$\Longleftarrow$")
Sei $k := m(l \text{ div } n) + \min(l \text{ mod } n, m)$.
Falls $l \geq n$, dann folgt $k \geq m$ und die erste Bedingung ist auch erfüllt.
Falls $l < n$, dann folgt $l \; mod \; n = l$. Aus $l > m$ folgt
$k = m(l \text{ div } n) + \min(l \text{ mod } n, m) = \min(l, m) = m$.
Dann ist auch die erste Bedingung $k \geq m$ erfüllt.
Definiere $a := (l \text{ div } n) + 1$, $b := (l \text{ div } n)$ und $x := l \text{ mod } n$.
Definiere die stetige Funktion $B : \mathbb{B} \longrightarrow \mathbb{B}$ durch

$$B\langle p_1, \ldots, p_n \rangle := \langle p_1^a \ldots p_x^a p_{x+1}^b \ldots p_n^b \rangle.$$

Dann gilt $x \cdot a + (n - x) \cdot b$
$= (l \text{ mod } n)((l \text{ div } n) + 1) + (n - (l \text{ mod } n))(l \text{ div } n)$
$= (l \text{ mod } n)(l \text{ div } n) + (l \text{ mod } n) + n(l \text{ div } n) - (l \text{ mod } n)(l \text{ div } n)$
$= (l \text{ mod } n) + n(l \text{ div } n) = l$.
Also ist $B\langle p_1, \ldots, p_n \rangle$ ein l-Tupel.
Falls es in $\langle p_1, \ldots, p_n \rangle$ nicht mehr als m Folgen gibt, die keine Nullfolgen sind, dann gibt es in $B\langle p_1, \ldots, p_n \rangle$ nicht mehr als k Folgen ungleich 0^ω.
Denn falls $m \leq x$, folgt $k = ma = m(l \text{ div } n + 1) = m(l \text{ div } n) + m = m(l \text{ div } n) + \min(l \text{ mod } n, m)$.
Falls $m > x$, folgt $k = xa + b(m - x) = x(l \text{ div } n + 1) + (l \text{ div } n)(m - x) = x(l \text{ div } n) + x + (l \text{ div } n)m - (l \text{ div } n)x = (l \text{ mod } n) + (l \text{ div } n)m = m(l \text{ div } n) + \min(l \text{ mod } n, m)$.
Definiere die stetige Funktion A durch

$$A\langle p, j \rangle := \begin{cases} i & \text{falls } j \leq x \cdot a \text{ und } i \text{ ist die eindeutige Zahl} \\ & \text{mit } i(a-1) + 1 \leq j \leq i \cdot a \\ i & \text{falls } j > x \cdot a \text{ und } i \text{ ist die eindeutige Zahl} \\ & \text{mit } x + i(b-1) + 1 \leq j \leq x + i \cdot b \\ div & \text{sonst} \end{cases}$$

Dann gilt $B\langle p_1, \ldots, p_n \rangle \in Def(LLPO_{l,k})$ für alle $\langle p_1, \ldots, p_n \rangle \in Def(LLPO_{n,m})$.
Falls $g(B\langle p_1, \ldots, p_n \rangle) = d'$ für $g \in LLPO_{l,k}$, dann gilt $A\langle \langle p_1, \ldots, p_n \rangle, d' \rangle = d$ mit $p_d = 0^\omega$.
Also folgt

$$\langle p_1, \ldots, p_n \rangle \longmapsto A\langle \langle p_1, \ldots, p_n \rangle, g \circ B\langle p_1, \ldots, p_n \rangle \rangle \in LLPO_{n,m}$$

für alle $g \in LLPO_{l,k}$. $\square$

Beweis von Satz 15. (der Richtung „$\Longrightarrow$”)
Wir müssen das Folgende zeigen:
Wenn nicht ($m \leq k$ und $k \geq m(l \text{ div } n) + \min(l \text{ mod } n, m)$),
dann ist $LLPO_{n,m} \leq_2 LLPO_{l,k}$ falsch.
Das ist äquivalent zu:
Wenn $m > k$ oder $k < m(l \text{ div } n) + \min(l \text{ mod } n, m)$, dann
$LLPO_{n,m} \not\leq_2 LLPO_{l,k}$.
Das werden wir in den beiden folgenden Lemmas beweisen. □

Lemma 1. *Aus $k < m$ folgt $LLPO_{n,m} \not\leq_2 LLPO_{l,k}$.*

Beweis. Annahme: $k < m$ und $LLPO_{n,m} \leq_2 LLPO_{l,k}$ und

$$\langle p_1, \dots, p_n \rangle \longmapsto A\langle\langle p_1, \dots, p_n \rangle, g \circ B\langle p_1, \dots, p_n \rangle\rangle \in LLPO_{n,m}$$

für jedes $g \in LLPO_{l,k}$.
Definiere $v_i := 0^\omega$ für $1 \leq i \leq n$ für diesen Beweis.
Definiere $q_0 := \langle v_1, \dots, v_n \rangle$.
Sei $cp(B(q_0)) = (a_1, \dots, a_l) =: b_0$.[5] Dann gilt

$$0 \leq \sharp_1(b_0) \leq k.$$

Definiere $p_{j,i} := \langle v_1 \dots, v_{j-1}, 0^i 10^\omega, v_{j+1}, \dots, v_n \rangle$.
Wenn $m > 0$, muss es eine Zahl j geben mit $cp(B(p_{j,i})) =: b_1$ und

$$\sharp_1(b_0) < \sharp_1(b_1) \leq k$$

für unendlich viele i (das impliziert $\sharp_1(b_0) < k$).
Andernfalls gilt $A\langle p_{j,i}, g \circ B(p_{j,i})\rangle = A\langle p_{j,i}, g(b_0)\rangle$ für alle j und $g \in LLPO_{l,k}$ und alle bis auf endlich viele $i \in \mathbb{N}$.[6]
Sei $A\langle q_0, g(b_0)\rangle =: x$. Dann existiert wegen der Stetigkeit von A eine Zahl y mit $A\langle\langle w_1, \dots, w_n\rangle, g(b_0)\rangle = x$ für alle $w_1, \dots, w_n$ mit $w_r \in 0^y\mathbb{N}^\omega$ für $1 \leq r \leq n$.
Dann gilt $A\langle p_{x,y}, g \circ B(p_{x,y})\rangle = A\langle p_{x,y}, g(b_0)\rangle = x$.
Das ist ein Widerspruch.

Also gibt es eine Zahl j mit $cp(B(p_{j,i})) =: b_1$ und $\sharp_1(b_0) < \sharp_1(b_1) \leq k$ für unendlich viele i. Definiere $q_1 := p_{j,i}$ für j und i mit $cp(B(p_{j,i})) =: b_1$ und $\sharp_1(b_0) < \sharp_1(b_1) \leq k$.

Seien $q_0, \dots, q_r \in \mathbb{B}^n$ Folgen mit $\sharp_1(q_t) = t$ für alle $0 \leq t \leq r$ und

$$0 \leq \sharp_1(B\langle q_0\rangle) < \sharp_1(B\langle q_1\rangle) < \ldots < \sharp_1(B\langle q_r\rangle) \leq k.$$

Sei $E :\subseteq \mathbb{B} \times \mathbb{B} \times \mathbb{N} \to \mathbb{B}$ definiert durch

$$E(\langle p_1 \dots, p_n\rangle, p', t) := \langle p_1, \dots, p_{t-1}, p', p_{t+1}, \dots, p_n\rangle.$$

Falls $\sharp_1(B\langle q_r\rangle) < k$, dann gibt es eine Zahl j mit $q_{r+1} := E(q_r, 0^i 10^\omega, j)$ und $\sharp_1(q_{r+1}) = r + 1$ und

$$\sharp_1(B\langle q_r\rangle) < \sharp_1(B\langle q_{r+1}\rangle) \leq k$$

[5] $cp(p_1, \dots, p_n) := (a_1, \dots, a_n)$ mit $a_i = 0$ für $p_i = 0^\omega$ und $a_i = 1$ für $p_i \neq 0^\omega$. Für Details siehe Definition 28.
[6] Mit $g(a_1, \dots, a_n)$ meinen wir $g\langle p_1, \dots, p_n\rangle$, falls $cp(p_i) = a_i$ für $1 \leq i \leq n$.

für unendlich viele $i \in \mathbb{N}$. Das Argument ist dasselbe wie für q_1.

Dies impliziert, dass es Folgen $q_0, \ldots, q_t$ gibt mit $\sharp_1(q_r) = r$ für alle $0 \leq r \leq t$ und

$$0 \leq \sharp_1(B\langle q_0\rangle) < \ldots < \sharp_1(B\langle q_t\rangle) = k$$

und $\sharp_1(q_t) \leq \sharp_1(B\langle q_t\rangle) = k$.
Also gilt $m > k = \sharp_1(B\langle q_t\rangle)$. Da $m > k$, gibt es für jedes j mit $q_t(j) = 0^\omega$ eine Zahl N, so dass $q := E(q_t, 0^i 10^\omega, j)$ und $\sharp_1(q) = t+1$ und $B\langle q_t\rangle = B\langle q\rangle$ für alle $i > N$.
Sei $A\langle\langle q_t\rangle, g \circ B\langle q_t\rangle\rangle = z$. Dann gilt $A\langle E(q_t, 0^i 10^\omega, z), g \circ B(E(q_t, 0^i 10^\omega, z))\rangle = A\langle E(q_t, 0^i 10^\omega, z), g \circ B\langle q_t\rangle\rangle = z$.
Das ist ein Widerspruch. Also gilt: $LLPO_{n,m} \leq_2 LLPO_{l,k}$ impliziert $k \geq m$. $\square$

Lemma 2. *Aus* $LLPO_{n,m} \leq_2 LLPO_{l,k}$ *folgt* $k \geq m(l \text{ div } n) + \min(l \bmod n, m)$.

Beweis. Seien A und B stetige Funktionen mit

$$q \longmapsto A\langle q, g \circ B\langle q\rangle\rangle \in LLPO_{n,m}$$

für alle $q \in Def(LLPO_{n,m})$ und $g \in LLPO_{l,k}$.
Definiere $p_0 := \langle 0^\omega, \ldots, 0^\omega\rangle$.
Definiere $q_0 := cp(B(p_0))$ und $M_0 := \{j | q_0(j) = 1\}$.
Entweder gilt $q_0 = (0, \ldots, 0)$ (und $M_0 = \emptyset$) oder es gibt eine Zahl i_0 mit $q_0(i) = 1 \Rightarrow cp(B\langle 0^{i_0}\langle q\rangle_1, \ldots, 0^{i_0}\langle q\rangle_n\rangle)(i) = 1$ für alle $q \in \mathbb{B}$ und $i \in \{1, \ldots, l\}$.
Sei $k_0 := |M_0|$. Für den Fall $q_0 = (0, \ldots, 0)$ setze $i_0 := 1$.
Definiere $p_{j,x} := \langle\langle p_0\rangle_1, \ldots, \langle p_0\rangle_{j-1}, (0^x 10^\omega), \langle p_0\rangle_{j+1}, \ldots, \langle p_0\rangle_n\rangle$.
Für jedes $j \in \{1, \ldots, n\}$ und $x > j_0$ sei $N_{j,x}$ definiert durch

$$N_{j,x} = \{y | cp(B(p_{j,x}))(y) = 1\}$$

Aus der Stetigkeit von B folgt $M_0 \subseteq N_{j,x}$ für alle $1 \leq j \leq n$ und $x > j_0$.
Definiere $N'_{j,x} := N_{j,x} \backslash M_0$ für alle $j \in \{1, \ldots, n\}$ und $x > j_0$.
Es muss ein $j \in \{1, \ldots, n\}$ und ein $x > j_0$ geben mit[7]

$$|N'_{j,x}| \geq \left\lceil \frac{l - k_0}{n} \right\rceil$$

Wenn das nicht der Fall wäre, dann gäbe es eine Zahl $s \in \{1, \ldots, l\}$ mit $s \notin \bigcup_{j=1}^n N_{j,x}$ für unendlich viele $x \in \mathbb{N}$.
Dann gibt es eine Funktion $g \in LLPO_{l,k}$, und für alle $j \in \{1, \ldots, n\}$ gibt es unendlich viele $x \in \mathbb{N}$ mit $g \circ B(p_{j,x}) = s$.
Also gilt $z := A\langle 0^\omega, s\rangle = A\langle p_{j,x}, g \circ B(p_{j,x})\rangle$ für unendlich viele x und alle $j \in \{1, \ldots, n\}$.
Aber dann folgt $z = A\langle p_{z,x}, g \circ B(p_{z,x})\rangle$ für unendlich viele x.
Das ist ein Widerspruch.

Seien $i \in \{1, \ldots, n\}$ und $x > j_0$ Zahlen, so dass $|N'_{i,x}|$ maximal ist. Definiere $M_1 := N'_{i,x}$. Dann gilt

$$|M_1| \geq \left\lceil \frac{l - k_0}{n} \right\rceil$$

[7] $\lceil\ \rceil$ und $\lfloor\ \rfloor$ sind wie üblich definiert durch $\lceil x \rceil := \min\{n \in \mathbb{N} | n \geq x\}$ und $\lfloor x \rfloor := \max\{n \in \mathbb{N} | n \leq x\}$ für $x \in \mathbb{Q}$ oder $x \in \mathbb{R}$.

Durch Induktion lassen sich Mengen $M_0, \ldots, M_m$ definieren mit

$$k_i = |M_i| \text{ für } i = 0, \ldots, m,$$

$$(\forall i \in \{0, \ldots, m\})(\exists p_i)\sharp_1 cp(p_i) = i \text{ und}$$

$$cp(B(p_i))(j) = 1 \Leftrightarrow j \in \bigcup_{k \in \{0,\ldots,i\}} M_k,$$

$$M_i \cap M_j = \emptyset \text{ für alle } i, j \in \{0, \ldots, n\}, j \neq i,$$

$$k_i \geq \left\lceil \frac{l - \sum_{j=0}^{i-1} k_j}{n - i + 1} \right\rceil \text{ für alle } i \in \{1, \ldots, m\}$$

Denn seien Mengen $M_0, \ldots, M_r$ mit $r < m$ und den angegebenen Eigenschaften gegeben.
Dann gibt es ein p_r mit $\sharp_1(cp(p_r)) = r$ und $cp(B(p_r))(j) = 1 \Leftrightarrow j \in \bigcup_{k \in \{0,\ldots,r\}} M_k$.
Definiere $q_r := cp(B(p_r))$ und $M := \{k \in \{1, \ldots, n\} | cp(p_r)(k) \neq 1\}$.
Sei $k_r := |M_r|$.
Sei i_r eine Zahl mit $(\forall j \in \{1, \ldots, m\}) cp(B(p_r))(j) = 1 \Rightarrow cp(B(p)(j) = 1$ für alle p mit $(\forall k \in \{1, \ldots, n\})[[\langle p_r \rangle_k]]_{j_r} \sqsubseteq \langle p \rangle_k$.
Definiere $p_{j,x} := \langle \langle p_r \rangle_1, \ldots, \langle p_r \rangle_{j-1}, (0^x 10^\omega), \langle p_r \rangle_{j+1}, \ldots, \langle p_r \rangle_n \rangle$ für alle $j \in M$.
Für jedes $j \in M$ und $x > i_r$ sei $N_{j,x}$ definiert durch

$$N_{j,x} = \{y | cp(B(p_{j,x}))(y) = 1\}$$

Aus der Stetigkeit von B folgt $\bigcup_{k=0,\ldots,r} M_k \subseteq N_{j,x}$ für alle $j \in M$ und $x > i_r$.
Definiere $N'_{j,x} := N_{j,x} \setminus \bigcup_{k=0,\ldots,r} M_k$ für alle $j \in M$ und $x > i_r$.
Es muss ein $j \in M$ und ein $x > i_r$ geben mit

$$|N'_{j,x}| \geq \left\lceil \frac{l - \sum_{i=0}^{r} k_i}{n - r} \right\rceil$$

Wenn das nicht der Fall wäre, dann gäbe es eine Zahl $s \in \{1, \ldots, l\}$ mit $s \notin \bigcup_{j=0}^{r} M_j \cup \bigcup_{j \in M} N'_{j,x}$ für unendlich viele $x \in \mathbb{N}$.
Dann gibt es eine Funktion $g \in LLPO_{l,k}$, und für alle $j \in M$ gibt es unendlich viele $x \in \mathbb{N}$ mit $g \circ B(p_{j,x}) = s$.
Also gilt $z := A\langle p_r, s \rangle = A\langle p_{j,x}, g \circ B(p_{j,x}) \rangle = A\langle p_{j,x}, s \rangle$ für unendlich viele x und alle $j \in M$, wobei $z \in M$ gelten muss.
Aber dann folgt $z = A\langle p_{z,x}, g \circ B(p_{z,x}) \rangle$ für unendlich viele x.
Das ist ein Widerspruch.

Seien $i \in M$ und $x > i_r$ Zahlen, so dass $|N'_{i,x}|$ maximal ist. Definiere $M_{r+1} := N'_{i,x}$. Dann gilt

$$|M_{r+1}| \geq \left\lceil \frac{l - \sum_{j=0}^{r} k_j}{n - r} \right\rceil$$

Für die maximale Anzahl k von Folgen ungleich 0^ω in $B(p)$ gilt dann

$$k \geq \sum_{i=0}^{m} k_i$$

Es soll berechnet werden, wie groß k mindestens werden muss.

Behauptung: Für alle $x, y, l, n \in \mathbb{N}, n > 1$ gilt

$$x \geq y \Rightarrow x + \left\lceil \frac{l-x}{n} \right\rceil \geq y + \left\lceil \frac{l-y}{n} \right\rceil$$

Beweis:
Für $x = y$ ist die Behauptung klar. Sei $x = y + z$ mit $z \in \mathbb{N}$, $z > 0$. Dann gilt

$$x + \left\lceil \frac{l-x}{n} \right\rceil = y + z + \left\lceil \frac{l-y-z}{n} \right\rceil = y + \left\lceil \frac{l-y-z}{n} + z \right\rceil$$

$$= y + \left\lceil \frac{l-y-z+z\cdot n}{n} \right\rceil = y + \left\lceil \frac{l-y+z\cdot(n-1)}{n} \right\rceil \geq y + \left\lceil \frac{l-y}{n} \right\rceil$$

Somit wird die Summe $\sum_{i=0}^{m} k_i$ minimal, wenn alle Summanden minimal werden. Somit kann man $k_0 = 0$ setzen.

Setze $a := l$ div n, $x := l$ mod n und $y := n - x$.
Dann gilt $l = (a+1)x + ay$.

Behauptung: Wenn die Zahlen k_i für $1 \leq i \leq m$ minimal werden sollen, gilt

$$k_i = a + 1 \text{ für } 1 \leq i \leq x \text{ und } k_i = a \text{ für } x < i \leq m$$

Beweis:
Falls $1 \leq x$ gilt

$$k_1 = \left\lceil \frac{l}{n} \right\rceil = \left\lceil \frac{(a+1)x+ay}{n} \right\rceil = \left\lceil \frac{ax+x+ay}{n} \right\rceil = \left\lceil \frac{an+x}{n} \right\rceil = a+1.$$

Falls $x = 0$ gilt

$$k_1 = \left\lceil \frac{l}{n} \right\rceil = \left\lceil \frac{an}{n} \right\rceil = a.$$

Für $1 < i \leq x$ gilt mit Induktion

$$k_i = \left\lceil \frac{l-k_1-\ldots-k_{i-1}}{n-i+1} \right\rceil = \left\lceil \frac{(a+1)x+ay-(a+1)(i-1)}{n-i+1} \right\rceil$$

$$= \left\lceil \frac{ax+x+ay-ai-i+a+1}{n-i+1} \right\rceil = \left\lceil \frac{a(n-i+1)+x-i+1}{n-i+1} \right\rceil = a+1.$$

Für $x < i \leq m$ gilt mit Induktion

$$k_i = \left\lceil \frac{l-k_1-\ldots-k_{i-1}}{n-i+1} \right\rceil = \left\lceil \frac{(a+1)x+ay-(a+1)x-a(i-1-x)}{n-i+1} \right\rceil$$

$$= \left\lceil \frac{ax+x+ay-ax-x-ai+a+ax}{n-i+1} \right\rceil = \left\lceil \frac{a(n-i+1)}{n-i+1} \right\rceil = a.$$

Hieraus folgt, wenn die Werte k_i für $0 \leq i \leq m$ minimal sein sollen:

$$k \geq \sum_{i=0}^{m} k_i = \sum_{i=1}^{m} k_i = \begin{cases} x(a+1) + (m-x)a & \text{falls } m > x \\ m(a+1) & \text{falls } m \leq x \end{cases}$$

Für $m > x$ folgt dann $k \geq x(a+1) + (m-x)a$
$= xa + x + ma - xa = ma + x$
$= m(l \text{ div } n) + \min(m, l \bmod n)$.

Für $m \leq x$ folgt $k \geq m(a+1)$
$= ma + m$
$= m(l \text{ div } n) + \min(m, l \bmod n)$.

In jedem Fall gilt also

$$k \geq= m \cdot (l \text{ div } n) + \min(m, (l \bmod n))$$

Somit folgt:
$LLPO_{n,m} \leq_2 LLPO_{l,k}$ impliziert $k \geq m \cdot (l \text{ div } n) + \min(m, (l \bmod n))$. □

Beispiel 5. Für alle $l \geq 2$ gilt $LLPO_{n,1} \leq_2 LLPO_{l,1}$, gdw. $n \geq l$.
Beweis von $\Longleftarrow$:
Sei $m = k = 1$. Nach Satz 15 haben wir zu verifizieren, dass $m \leq k$ und $k \geq m(l \text{ div } n) + \min(l \bmod n, 1)$.
Die erste Bedingung ist natürlich erfüllt. Wenn $n = l$, erhalten wir
$m(l \text{ div } n) + \min(l \bmod n, 1)$
$= 1 \cdot 1 + \min(0, 1) = 1 = k$.
Wenn $n > l$, erhalten wir $m(l \text{ div } n) + \min(l \bmod n, 1)$
$= 1 \cdot 0 + 1 = 1 = k$.
Somit gilt $LLPO_{n,1} \leq_2 LLPO_{l,1}$.
Beweis von $\Longrightarrow$:
Sei $m = k = 1$. Nach Satz 15 wissen wir, dass $m \leq k$ und $k \geq m(l \text{ div } n) + \min(l \bmod n, 1)$.
Aus der zweiten Bedingung folgt $1 \geq 1(l \text{ div } n) + \min(l \bmod n, 1)$.
Das bedeutet, dass nur einer der Terme $(l \text{ div } n)$ und $(l \bmod n)$ größer als 0 sein kann. Also folgt $(l \text{ div } n) = 0$ oder $l \bmod n = 0$. Also gilt $n > l$ oder $n = l$.

Dies ist ein sehr einfacher Beweis des Satzes

$$LLPO_{n+1} <_2 LLPO_n \text{ für alle } n \geq 2$$

in Satz 3 und in [Wei92a].

Die Behauptung

$$MLPO_n <_2 MLPO_{n+1} \text{ für alle } n \geq 2$$

kann genauso leicht bewiesen werden:

Beispiel 6. $LLPO_{n,n-1} \leq_2 LLPO_{l,l-1}$, gdw. $n \leq l$.
Beweis von $\Longleftarrow$:

Sei $m = n - 1$ und $k = l - 1$. Nach Satz 15 haben wir zu verifizieren, dass $m \leq k$ und $k \geq m(l \text{ div } n) + \min(l \bmod n, 1)$.
Falls $l \geq n$, ist natürlich die erste Bedingung erfüllt.
Sei $l = x \cdot n + y$ mit $x \geq 1$ und $0 \leq y < n$.
Dann gilt $m(l \text{ div } n) + \min(l \bmod n, m)$
$= (n-1)((x \cdot n + y) \text{ div } n) + \min((x \cdot n + y) \bmod n, n-1)$
$= (n-1)x + \min(y, n-1) = (n-1)x + y = x \cdot n - x + y$
$\leq x \cdot n + y - 1 = l - 1 = k$.
Beweis von $\Longrightarrow$:
Sei $l = x \cdot n + y$ mit $x \geq 0$ und $0 \leq y < n$.
Sei $k \geq m(l \text{ div } n) + \min(l \bmod n, 1)$.
Dann gilt $k \geq m(l \text{ div } n) + \min(l \bmod n, 1)$
$\Longrightarrow x \cdot n + y - 1 \geq (n-1)((x \cdot n + y) \text{ div } n) + \min((x \cdot n + y) \bmod n, n-1)$
$\Longrightarrow x \cdot n + y - 1 \geq (n-1)(x) + \min(y, n-1)$
$\Longrightarrow x \cdot n + y - 1 \geq x \cdot n - x + y$
$\Longrightarrow -1 \geq -x$
$\Longrightarrow x \geq 1$.
Also gilt $l = x \cdot n + y \geq n$.

Beispiel 7. $LLPO_{n,1} \leq_2 LLPO_{n+1,2}$ für alle $n > 1$.
Sei $m = 1$, $k = 2$ und $l = n + 1$.
Wir können $m \leq k$ leicht verifizieren.
Ferner können wir das erlaubte Minimum für k berechnen:
Falls $n > 1$, dann gilt $k = m(l \text{ div } n) + \min(l \bmod n, m)$
$= 1((n+1) \text{ div } n) + \min((n+1) \bmod n, 1)$
$= 1 + \min(1, 1) = 2$.

Beispiel 8. $LLPO_{n,1} \leq_2 LLPO_{n \cdot k', k'}$.
Sei $m = 1$ und $l = n \cdot k'$.
Wir können $m \leq k'$ leicht verifizieren.
Ferner können wir das erlaubte Minimum für k berechnen:
$k = m((n \cdot k') \text{ div } n) + \min((n \cdot k') \bmod n, m)$
$= 1(k') + \min(0, 1) = k'$.

Beispiel 9. $LLPO_{n+1,m} \leq_2 LLPO_{n,m+1}$, für $n + 1 > m \geq 0$ und $n > m + 1 \geq 0$ (wobei $n + 1 > m$ schon aus $n > m + 1$ folgt).
Wir haben $m \leq m + 1$ zu verifizieren. Das ist trivial.
Ferner haben wir zu verifizieren, dass
$m + 1 \geq m(n \text{ div } (n+1)) + \min(n \bmod (n+1), m)$:
$m(n \text{ div } (n+1)) + \min(n \bmod (n+1), m)$
$= m \cdot 0 + \min(n, m) = m \leq m + 1$.
Also ist die zweite Bedingung auch erfüllt.

3.2.4 $LLPO_{\infty,m}$

Definition 20 ($LLPO_{\infty,m}$).
Sei $LLPO_{\infty,m}$ die Menge aller Funktionen f mit

$$Def(f) = \{\langle p_1, p_2, \ldots\rangle \mid \text{ es gibt höchstens } m \text{ Zahlen } k \text{ mit } p_k \neq 0^\omega\}$$
$$\text{und } f(\langle p_1, p_2, \ldots\rangle) = k \text{ für ein } k \text{ mit } p_k = 0^\omega.$$

Satz 16 ($LLPO_{\infty,m} \leq LLPO_{l,k}$).
$LLPO_{\infty,m} \leq LLPO_{l,k} \Longleftrightarrow m \leq k$.

Beweis. (von „$\Longleftarrow$")
Definiere $A :\subseteq \mathbb{B} \longrightarrow \mathbb{N}$ durch $A\langle\langle p_1, p_2, \ldots\rangle, w\rangle := w$ und B durch $B\langle p_1, p_2, \ldots\rangle := \langle p_1, p_2, \ldots, p_l\rangle$. Definiere g durch

$$\begin{aligned} g\langle p_1, p_2, \ldots\rangle &:= A\langle\langle p_1, p_2, \ldots\rangle, f \circ B\langle p_1, p_2, \ldots\rangle\rangle \\ &= f \circ B\langle p_1, p_2, \ldots\rangle \\ &= f\langle p_1, p_2, \ldots, p_l\rangle \\ &= k \text{ für ein } k < l \text{ mit } p_k = 0^\omega. \end{aligned}$$

Da $f \in LLPO_{l,k}$, es eine Folge 0^ω in $p_1, p_2, \ldots, p_l$ gibt, und es $m \leq k$ Folgen ungleich 0^ω gibt, findet f die Position einer Folge 0^ω.
Also gilt $g \in LLPO_{\infty,m}$. □

Beweis. (von „$\Longrightarrow$")
Annahme: $k < m$ und $LLPO_{\infty,m} \leq_2 LLPO_{l,k}$.
Definiere $q_0 := \langle 0^\omega, 0^\omega, \ldots\rangle$.
Sei $cp(B(q_0)) = (a_1, \ldots, a_l) =: b_0$. Dann gilt

$$0 \leq \sharp_1(b_0) \leq k.$$

Definiere $v_i := 0^\omega$ für $i \geq 0$ für diesen Beweis.
Definiere $p_{j,i} := \langle v_1, \ldots, v_{j-1}, 0^i 10^\omega, v_{j+1}, \ldots\rangle$.
Es muss eine Zahl j existieren mit $cp(B(p_{j,i})) =: b_1$ und

$$\sharp_1(b_0) < \sharp_1(b_1) \leq k$$

für unendlich viele i.
Andernfalls gilt $A\langle p_{j,i}, g \circ B(p_{j,i})\rangle = A\langle p_{j,i}, g(b_0)\rangle$ für alle j und $g \in LLPO_{l,k}$ und alle bis auf endlich viele $i \in \mathbb{N}$.
Sei $A\langle q_0, g(b_0)\rangle =: x$. Dann gibt es wegen der Stetigkeit von A eine Zahl y mit

$$A\langle\langle w_1, w_2, \ldots\rangle, g(b_0)\rangle = x$$

für alle $w_1, w_2, \ldots$ mit $w_r \in 0^y\mathbb{N}^\omega$ für $1 \leq r$.
Dann gilt $A\langle p_{x,y}, g \circ B(p_{x,y})\rangle = A\langle p_{x,y}, g(b_0)\rangle = x$.
Das ist ein Widerspruch.

Also gibt es eine Zahl j mit $cp(B(p_{j,i})) =: b_1$ und

$$\sharp_1(b_0) < \sharp_1(b_1) \leq k$$

für unendlich viele i. Definiere $q_1 := p_{j,i}$ für j und i mit $cp(B(p_{j,i})) =: b_1$ und $\sharp_1(b_0) < \sharp_1(b_1) \leq k$.

Seien $q_0, \ldots, q_r \in \mathbb{B}^\omega$ Folgen mit $\sharp_1(q_t) = t$ für alle $0 \leq t \leq r$ und

$$0 \leq \sharp_1(B\langle q_0\rangle) < \sharp_1(B\langle q_1\rangle) < \ldots < \sharp_1(B\langle q_r\rangle) \leq k.$$

Sei $E :\subseteq \mathbb{B} \times \mathbb{B} \times \mathbb{N} \to \mathbb{B}$ definiert durch

$$E(\langle p_1, \ldots\rangle, p', t) := \langle p_1, \ldots, p_{t-1}, p', p_{t+1}, \ldots\rangle.$$

E ersetzt die t-te Folge in $\langle p_1, \ldots\rangle$ durch p'.
Falls $\sharp_1(B\langle q_r\rangle) < k$, dann gibt es eine Zahl j mit $q_{r+1} := E(q_r, 0^i10^\omega, j)$ und $\sharp_1(q_{r+1}) = r + 1$ und $\sharp_1(B\langle q_r\rangle) < \sharp_1(B\langle q_{r+1}\rangle) \leq k$ für unendlich viele $i \in \mathbb{N}$. Das Argument ist dasselbe wie für q_1.

Das impliziert, dass es Folgen $q_0, \ldots, q_t$ gibt mit $\sharp_1(q_r) = r$ für alle $0 \leq r \leq t$ und

$$0 \leq \sharp_1(B\langle q_0\rangle) < \ldots < \sharp_1(B\langle q_t\rangle) = k$$

und $\sharp_1(q_t) \leq \sharp_1(B\langle q_t\rangle) = k$.
Also gilt $m > k = \sharp_1(B\langle q_t\rangle)$. Da $m > k$, gibt es für jedes j mit $q_t(j) = 0^\omega$ eine Zahl N, so dass $q := E(q_t, 0^i10^\omega, j)$ und $\sharp_1(q) = t + 1$ und $B\langle q_t\rangle = B(q)$ für alle $i > N$.
Sei $A\langle q_t, g \circ B(q_t)\rangle = z$. Dann gilt $A\langle E(q_t, 0^i10^\omega, z), g \circ B(E(q_t, 0^i10^\omega, z))\rangle = A\langle E(q_t, 0^i10^\omega, z), g \circ B(q_t)\rangle = z$.
Das ist ein Widerspruch. Also gilt: $LLPO_{\infty,m} \leq_2 LLPO_{l,k}$ impliziert $k \geq m$. ☐

Es ist möglich, diesen Satz als eine Variation des Satzes 15 zu interpretieren. Wenn wir $n := \infty$ in Satz 15 definieren, dann folgt $LLPO_{\infty,m} \leq_2 LLPO_{l,k}$ gdw. $m \leq k$ und $k \geq m(l \text{ div } \infty) + \min(l \bmod \infty, m)$.
Die Bedingung $m \leq k$ ist dieselbe in beiden Sätzen.
Wenn wir $l \text{ div } \infty := 0$ und $l \bmod \infty := l$ setzen, erhalten wir
$m(l \text{ div } \infty) + \min(l \bmod \infty, m) = m \cdot 0 + \min(l, m) = m \leq k$.
Somit ist mit der ersten Bedingung auch die zweite erfüllt.

Satz 17 ($LLPO_{\infty,m} \leq LLPO_{\infty,k}$)**.**
$LLPO_{\infty,m} \leq LLPO_{\infty,k} \iff m \leq k$.

Beweis. (von „$\Longleftarrow$")
Definiere A durch $A\langle p, w\rangle := w$ und B durch $B\langle p_1, p_2, \ldots\rangle := \langle p_1, p_2, \ldots\rangle$. Definiere g durch

$$\begin{aligned} g\langle p_1, p_2, \ldots\rangle &:= A\langle\langle p_1, p_2, \ldots\rangle, f \circ B\langle p_1, p_2, \ldots\rangle\rangle \\ &= f \circ B\langle p_1, p_2, \ldots\rangle \\ &= f\langle p_1, p_2, \ldots\rangle \\ &= k \text{ für ein } k \in \mathbb{N} \text{ mit } p_k = 0^\omega. \end{aligned}$$

Da $f \in LLPO_{\infty,k}$ und $k \geq m$, ist f anwendbar auf $\langle p_1, p_2, \ldots\rangle$, und f findet die Position einer Folge 0^ω. Somit gilt $g \in LLPO_{\infty,m}$. ☐

Beweis. (von „$\Longrightarrow$”)
Der Beweis ist ähnlich wie die entsprechende Beweisrichtung von Satz 16. $\square$

Dieser Satz kann ebenfalls als eine Variation des Satzes 15 betrachtet werden. Wenn wir $n := l := \infty$ in Satz 15 setzen, dann gilt $LLPO_{\infty,m} \leq_2 LLPO_{\infty,k}$ gdw. $m \leq k$ und $k \geq m(\infty \text{ div } \infty) + \min(\infty \text{ mod } \infty, m)$.
Die Bedingung $m \leq k$ ist in beiden Sätzen dieselbe.
Wenn wir ∞ div $\infty := 1$ und ∞ mod $\infty := 0$ definieren, erhalten wir
$m(\infty \text{ div } \infty) + \min(\infty \text{ mod } \infty, m) = m \cdot 1 + \min(0, m) = m \leq k$.
Somit ist mit der ersten Bedingung auch die zweite erfüllt.

Ohne Beweis sei noch der folgende Satz angefügt:

Satz 18 ($LLPO_{n,m} \leq LLPO_{\infty,k}$)**.**
$LLPO_{n,m} \leq LLPO_{\infty,k} \Longleftrightarrow (m = 0$ *oder* $n = m = 1)$.

Dieser Satz kann für unstetige $LLPO_{n,m}$ ebenfalls als eine Variation des Satzes 15 betrachtet werden. Wenn wir $l := \infty$ in Satz 15 setzen, dann gilt für $m > 0$
$m(\infty \text{ div } n) + \min(\infty \text{ mod } n, m) = m \cdot \infty + \min(0, m) = \infty > k$.

3.2.5 $LLPO_{\infty,-m}$

Definition 21 ($LLPO_{\infty,-m}$).
Sei $LLPO_{\infty,-m}$ die Menge aller Funktionen f mit

$$Def(f) = \{\langle p_1, p_2, \ldots\rangle \mid \text{ es gibt höchstens } m \text{ Zahlen } k \text{ mit } p_k = 0^\omega\}$$
$$\text{und } f(\langle p_1, p_2, \ldots\rangle) := k \text{ für ein } k \text{ mit } p_k = 0^\omega.$$

Satz 19 ($LLPO_{l,k} \leq LLPO_{\infty,-m}$).

$$LLPO_{l,k} \leq LLPO_{\infty,-m}, \text{ gdw. } m \geq 1.$$

Beweis. (von „$\Longleftarrow$")
Definiere $h : \mathbb{N} \setminus \{0\} \times \{1, \ldots, l\} \longrightarrow \mathbb{N} \setminus \{0\}$ durch $h(i, j) := (i - 1) \cdot l + j$.
Definiere $B :\subseteq \mathbb{B} \longrightarrow \mathbb{B}$ durch $B\langle p_1, \ldots, p_l\rangle := \langle q_1, q_2, \ldots\rangle$ mit

$$q_{h(i,j)}(x) := \begin{cases} 1 & \text{falls } i = 1 = x \text{ und } p_j(1) = 1 \\ 1 & \text{falls } i = 1 = x \text{ und} \\ & j \neq \min\{s \in \{1, \ldots, l\} | p_s(1) = 0\} \\ 1 & \text{falls } i = x > 1 \text{ und } \min\{s \in \{1, \ldots, l\} | [[p_s]]_x = 0^x\} \\ & = \min\{t \in \{1, \ldots, l\} | [[p_t]]_{x-1} = 0^{x-1}\} \\ 1 & \text{falls } i = x > 1 \text{ und} \\ & j \neq \min\{s \in \{1, \ldots, l\} | [[p_s]]_i = 0^i\} \\ 1 & \text{falls } i < x \text{ und } \min\{s \in \{1, \ldots, l\} | [[p_s]]_x = 0^x\} \\ & \neq \min\{t \in \{1, \ldots, l\} | [[p_t]]_{x-1} = 0^{x-1}\} \text{ und} \\ & q_{h(i,j)}(1) = \ldots = q_{h(i,j)}(x - 1) = 0 \\ 0 & \text{sonst} \end{cases}$$

Die Funktion B ist stetig und berechenbar. Im ersten Berechnungsschritt können die Werte $q_{h(1,j)}(1)$ für $1 \leq j \leq l$ berechnet werden.
Im n-ten Berechnungsschritt können die Werte $q_{h(i,j)}(n)$ für $1 \leq i \leq n$ und $1 \leq j \leq l$ und die Werte $q_{h(n,j)}(x)$ für $1 \leq x \leq n$ und $1 \leq j \leq l$ berechnet werden.

Definiere $A :\subseteq \mathbb{B} \longrightarrow \mathbb{N}$ durch $A\langle p, y\rangle := ((y - 1) \bmod l) + 1$.
A ist ebenfalls stetig.

Dann gilt das Folgende:

$(\forall 1 \leq n \leq l) p_n \neq 0^\omega \Longrightarrow (\forall 1 \leq n') B\langle p_1, \ldots, p_l\rangle(n') \neq 0^\omega$

$a = \min\{n | p_n = 0^\omega\} \Longrightarrow$ es gibt nur eine Folge
q_z in $B\langle p_1, \ldots, p_l\rangle$ mit $q_z = 0^\omega$ und $((z - 1) \bmod l) + 1 = a$

Es ist möglich, diese Behauptungen durch Induktion zu beweisen.

Sei $a = \min\{n | p_n = 0^\omega\}$. Sei z die Zahl mit q_z in $B\langle p_1, \ldots, p_l\rangle$ mit $q_z = 0^\omega$ und $((z - 1) \bmod l) + 1 = a$.
Sei ferner $g \in LLPO_{\infty,-m}$.

Dann gilt $A\langle\langle p_1, \ldots, p_l\rangle, g \circ B\langle p_1, \ldots, p_l\rangle\rangle$
$= A\langle\langle p_1, \ldots, p_l\rangle, z\rangle = a$.
Also gilt $LLPO_{l,k} \leq_2 LLPO_{\infty,-m}$. □

In diesem Beweis wird eine der Nullfolgen auf eine Nullfolge abgebildet. Die Funktion B versucht, die erste Nullfolge auf eine Nullfolge im Bild abzubilden, von deren Position wir die Position der Nullfolge der Eingabe berechnen können. Wenn die Berechnung herausfindet, dass eine andere Folge die erste Nullfolge ist, wird die Ausgabe abgeändert. An die bisherige Nullfolge wird eine 1 angehängt und in eine andere Ausgabefolge an geeigneter Stelle werden nur Nullen geschrieben.

Beispiel 10. Sei $p_1 := 00010^\omega$, $p_2 := 0^\omega$ und $p_3 := 10^\omega$.
Die Funktion B berechnet die folgenden Folgen:

h(i,j)		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	1	**0**	**1**	**1**	0	0	0	0	0	0	0	0	0	0	0	0
	2	0	0	0	**1**	**1**	**1**	0	0	0	0	0	0	0	0	0
	3	0	0	0	0	0	0	**1**	**1**	**1**	0	0	0	0	0	0
x	4	**1**	**0**	**0**	0	0	0	0	0	0	**1**	**0**	**1**	0	0	0
	5	0	0	0	0	0	0	0	0	0	0	0	0	**1**	**1**	**1**
	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

In den ersten Berechnungsschritten wird p_1 als Nullfolge angesehen. Also werden in die Folge q_1 nur Nullen geschrieben. In die Folgen q_2 und q_3 wird an der ersten Stelle eine 1 geschrieben.
In den folgenden Schritten wird in die Folgen q_4, q_5 und q_6 eine 1 an der zweiten Stelle geschrieben und in die Folgen q_7, q_8 und q_9 eine 1 an der dritten Stelle. Dann findet die Berechnung, dass p_1 nicht gleich 0^ω ist, sondern eventuell $p_2 = 0^\omega$. Deshalb wird die 1 in q_{10} und q_{12} geschrieben und die 0 in q_{11} an dem vierten Platz.
Die falsche Vermutung zuvor wird dadurch korrigiert, dass eine 1 in q_1 an der vierten Stelle geschrieben wird.

Beweis. (von „$\Longrightarrow$")
Falls $m = 0$, dann ist $LLPO_{\infty,-m}$ nicht definiert.
Also ist $LLPO_{l,k} \leq_2 LLPO_{\infty,-m}$ nicht möglich. □

Satz 20 ($LLPO_{\infty,-m} \not\leq_2 LLPO_{l,k}$)**.**

$$LLPO_{\infty,-m} \leq_2 LLPO_{l,k} \text{ ist falsch für alle } m,\ l \text{ und } k.$$

Beweis. Wenn $LLPO_{\infty,-m} \leq LLPO_{l,k}$, dann gibt es stetige Funktionen A und B mit

$$A\langle\langle p_1, \ldots\rangle, g \circ B\langle p_1, \ldots\rangle\rangle \in LLPO_{\infty,-m}$$

für alle $g \in LLPO_{l,k}$.
Für $g \in LLPO_{l,k}$ gibt es nicht mehr als l unterschiedliche mögliche Werte. Sei

$$j(i) := A\langle\langle 0^\omega, 0^\omega, \ldots\rangle, i\rangle$$

für alle $1 \leq i \leq l$.
Da A stetig ist, gibt es eine Zahl x mit

$$A\langle\langle w_1, w_2, \ldots\rangle, i\rangle = j(i)$$

für alle $1 \leq i \leq l$ und $w_1, w_2, \ldots \in 0^x\mathbb{B}$.
Sei $b := \max\{j(i) | 1 \leq i \leq l\} + 1$.
Sei

$$q_j := \begin{cases} 0^\omega & \text{falls } j = b \\ 0^b 1 0^\omega & \text{sonst} \end{cases}$$

Dann gilt $A\langle\langle q_1, q_2, \ldots\rangle, i\rangle < b$ für alle i, aber $f \in LLPO_{\infty,-m}$ impliziert $f\langle q_1, q_2, \ldots\rangle = b$.
Das ist ein Widerspruch. Also gilt $LLPO_{\infty,-m} \not\leq_2 LLPO_{l,k}$. □

Satz 21 ($LLPO_{\infty,-m} \leq_2 LLPO_{\infty,-k}$)**.**

$$LLPO_{\infty,-m} \leq_2 LLPO_{\infty,-k} \Longleftrightarrow m = 0 \textit{ oder } k \geq 1.$$

Beweis. (von „$\Longleftarrow$")
Falls $m = 0$, dann gilt $f = div$ für alle $f \in LLPO_{\infty,-m}$.
Somit folgt $LLPO_{\infty,-m} \leq_2 LLPO_{\infty,-k}$.
Sei $m > 0$ und $k > 0$. Dann kann dieselbe Technik wie in dem Beweis von Satz 19 angewendet werden.
Definiere $B :\subseteq \mathbb{B} \longrightarrow \mathbb{B}$ durch $B\langle p_0, \ldots\rangle := \langle q_0, q_1, \ldots\rangle$ mit

$$q_{\langle i,j\rangle}(x) := \begin{cases} 1 & \text{falls } \langle i,j\rangle = x \text{ und es gibt ein } y \leq x \text{ mit } p_j(y) = 1 \\ 1 & \text{falls } \langle i,j\rangle = x \text{ und es gibt eine Zahl } \langle a,b\rangle < \langle i,j\rangle \\ & \text{mit } b \leq j \text{ und } p_b(y) = 0 \text{ für alle } y \leq x \\ 1 & \text{falls } \langle i,j\rangle < x \text{ und } q_{\langle i,j\rangle}(y) = 0 \text{ für alle } y < x \\ & \text{und } p_j(x) = 1 \\ 0 & \text{sonst} \end{cases}$$

Die Funktion B ist stetig und berechenbar. Im ersten Berechnungsschritt wird der Wert $q_{\langle 0,0\rangle}(0)$ berechnet.
Im n-ten Schritt werden die Werte $q_i(n)$ für $0 \leq i \leq n$ und die Werte $q_n(x)$ für $0 \leq x \leq n$ berechnet.

Definiere $A :\subseteq \mathbb{B} \longrightarrow \mathbb{N}$ durch $A\langle p, \langle x, y\rangle\rangle := y$.
A ist ebenfalls stetig.

Dann gilt das Folgende:

$(\forall n \geq 0) p_n \neq 0^\omega \Longrightarrow (\forall n' \geq 0) B\langle p_0, p_1 \ldots\rangle(n') \neq 0^\omega$

$a = \min\{n | p_n = 0^\omega\} \Longrightarrow$ es gibt nur eine Folge
q_z in $B\langle p_0, p_1, \ldots\rangle$ mit $q_z = 0^\omega$ und $\pi_2(z) = a$.

Es ist möglich, diese Behauptungen durch Induktion zu beweisen.

Sei $a = \min\{n | p_n = 0^\omega\}$. Sei z die Zahl mit q_z in $B\langle p_1, \ldots, p_l\rangle$ mit $q_z = 0^\omega$ und $a = \pi_2(z)$.
Sei ferner $g \in LLPO_{\infty,-k}$.
Dann gilt $A\langle\langle p_1, p_2, \ldots\rangle, g \circ B\langle p_1, p_2, \ldots\rangle\rangle$
$= A\langle\langle p_1, p_2, \ldots\rangle, z\rangle = a$.
Also gilt $LLPO_{\infty,-m} \leq_2 LLPO_{\infty,-k}$. □

Beweis. (von „$\Longrightarrow$”)
Annahme: $m > 0$ und $k = 0$. Dann ist $LLPO_{\infty,-k}$ nicht definiert.
Also ist $LLPO_{\infty,-m} \leq_2 LLPO_{\infty,-k}$ nicht möglich. □

Satz 22 ($LLPO_{\infty,m} \leq_2 LLPO_{\infty,-k}$)**.**

$$LLPO_{\infty,m} \leq_2 LLPO_{\infty,-k}, \text{ gdw. } m = 0 \text{ oder } k \geq 1.$$

Beweis. Der Beweis ist ähnlich wie der Beweis von Satz 19. □

3.2.6 $LLPO_{\infty,\infty}$

Definition 22 ($LLPO_{\infty,\infty}$).
Sei $LLPO_{\infty,\infty}$ die Menge aller Funktionen f mit

$$Def(f) = \{\langle p_1, p_2, \ldots\rangle \mid \text{ es gibt unendlich viele Zahlen } k \text{ und } l \text{ mit } p_k = 0^\omega \text{ und } p_l \neq 0^\omega\}$$

und $f(\langle p_1, p_2, \ldots\rangle) := k$ für ein k mit $p_k = 0^\omega$.

Satz 23 ($LLPO_{\infty,\infty} \leq LLPO_{\infty,-k}$).

$$LLPO_{\infty,\infty} \leq LLPO_{\infty,-k} \text{ gdw. } k \geq 1.$$

Beweis. Der Beweis ist ähnlich wie der Beweis von Satz 19. □

Satz 24 ($LLPO_{\infty,m} \leq LLPO_{\infty,\infty}$).

$$LLPO_{\infty,m} \leq LLPO_{\infty,\infty} \text{ für alle } m.$$

Beweis. Wieder kann dieselbe Technik wie im Beweis von Satz 19 angewendet werden.
Aber diesmal müssen wir mit der Funktion B unendlich viele Folgen gleich 0^ω und unendlich viele Folgen ungleich 0^ω erzeugen.
Eine Idee, dies zu erreichen, ist es, eine Nullfolge und dann eine Folge ungleich 0^ω unendlich oft für eine Nullfolge an der niedrigsten Position zu erzeugen.

Definiere $B :\subseteq \mathbb{B}^\omega \longrightarrow \mathbb{B}^\omega$ durch $B\langle p_0, \ldots\rangle := \langle q_0, q_1, \ldots\rangle$ mit

$$q_{\langle i,j\rangle}(x) := \begin{cases} 1 & \text{falls } \langle i,j\rangle = x \text{ und es gibt ein } y \leq x \text{ mit } p_j(y) = 1 \\ 1 & \text{falls } \langle i,j\rangle = x \text{ und es gibt eine Zahl } \langle a,b\rangle < \langle i,j\rangle \\ & \text{mit } b \leq j \text{ und } p_b(y) = 0 \text{ für alle } y \leq x \\ & \text{und } i \text{ ist ungerade} \\ 1 & \text{falls } \langle i,j\rangle < x \text{ und } q_{\langle i,j\rangle}(y) = 0 \text{ für alle } y < x \\ & \text{und } p_j(x) = 1 \\ 0 & \text{sonst} \end{cases}$$

Die Funktion B ist stetig und berechenbar. Im ersten Berechnungsschritt wird der Wert $q_{\langle 0,0\rangle}(0)$ berechnet.
Im n-ten Schritt werden die Werte $q_i(n)$ für $0 \leq i \leq n$ und die Werte $q_n(x)$ für $0 \leq x \leq n$ berechnet.

Definiere $A :\subseteq \mathbb{B} \longrightarrow \mathbb{N}$ durch $A\langle p, \langle x, y\rangle\rangle := y$ für $p \in \mathbb{B}, x, y \in \mathbb{N}$.
A ist ebenfalls stetig.

Dann gilt das Folgende:

$(\forall 0 \leq n) p_n \neq 0^\omega \Longrightarrow (\forall 0 \leq n') B\langle p_0, p_1 \ldots\rangle(n') \neq 0^\omega$

$a = \min\{n | p_n = 0^\omega\} \Longrightarrow$ es gibt unendlich viele Folgen
q_z in $B\langle p_0, p_1, \ldots\rangle$ mit $q_z = 0^\omega$ und $\pi_2(z) = a$ und unendlich viele Folgen q_z in

$B\langle p_0, p_1, \ldots\rangle$ mit $q_z \neq 0^\omega$ und $\pi_2(z) = a$.

Sei $a = \min\{n | p_n = 0^\omega\}$. Sei z eine Zahl mit q_z in $B\langle p_0, p_1, \ldots\rangle$ mit $q_z = 0^\omega$ und $a = \pi_2(z)$.
Sei ferner $g \in LLPO_{\infty,\infty}$.
Dann gilt $A\langle\langle p_0, p_1, \ldots\rangle, g \circ B\langle p_0, p_1, \ldots\rangle\rangle$
$= A\langle\langle p_0, p_1, \ldots\rangle, z\rangle = a$.
Also gilt $LLPO_{\infty,-m} \leq_2 LLPO_{\infty,\infty}$. □

Satz 25 ($LLPO_{\infty,\infty} \not\leq LLPO_{l,k}$)**.**
$LLPO_{\infty,\infty} \leq LLPO_{l,k}$ ist falsch für alle l und k.

Beweis. Angenommen $LLPO_{\infty,\infty} \leq LLPO_{l,k}$. Nach Satz 24 gilt $LLPO_{\infty,m} \leq LLPO_{\infty,\infty}$ für alle $m \geq 1$. Sei $m > k$. Die Transitivität von $\leq_2$ impliziert $LLPO_{\infty,m} \leq LLPO_{l,k}$.
Aber das ist nicht wahr nach Satz 18.
Also ist $LLPO_{\infty,\infty} \leq LLPO_{l,k}$ nicht wahr. □

Satz 26 ($\Omega <_2 LLPO_{\infty,\infty}$)**.**
$\Omega <_2 LLPO_{\infty,\infty}$.

Beweis.
<u>$\Omega \leq_2 LLPO_{\infty,\infty}$:</u>
Definiere $B :\subseteq \mathbb{B} \longrightarrow \mathbb{B}$ durch

$$B(p)\langle j, i\rangle := \begin{cases} 1 & \text{falls } j = 2\cdot i + 1 \text{ und } [[p]]_i = 0^i \\ 1 & \text{falls } j = 2\cdot i \text{ und } [[p]]_i \neq 0^i \\ 1 & \text{falls } j < 2\cdot i \text{ und } [[p]]_i \neq 0^i \\ & \text{und } [[p]]_{i-1} = 0^i \\ 0 & \text{sonst} \end{cases}$$

Die Funktion B ist stetig und berechenbar. Im ersten Berechnungsschritt werden die Werte $B(p)\langle 0, 0\rangle$ und $B(p)\langle 0, 1\rangle$ berechnet.
Im n-ten Berechnungsschritt werden die Werte $B(p)\langle 2n, i\rangle$ und $B(p)\langle 2n+1, i\rangle$ für $0 \leq i \leq n$ und die Werte $B(p)\langle 2i, n\rangle$ und $B(p)\langle 2i+1, n\rangle$ für $0 \leq i < n$ berechnet.

Definiere $A :\subseteq \mathbb{B} \longrightarrow \mathbb{N}$ durch

$$A\langle p, x\rangle := \begin{cases} 0 & \text{falls } x \text{ gerade} \\ 1 & \text{falls } x \text{ ungerade} \end{cases}$$

für $p \in \mathbb{B}, x \in \mathbb{N}$. A ist ebenfalls stetig.

Dann gilt das Folgende:

$$\begin{aligned} p = 0^\omega \Longrightarrow & \\ & (\forall n \geq 0) B(p)(2n) = 0^\omega \\ & \text{und } (\forall n \geq 0) B(p)(2n+1) = 0^n 1 0^\omega \end{aligned}$$

$$
\begin{aligned}
p = 0^i10^\omega \implies & \\
& (\forall n \geq i) B(p)(2n) = 0^n10^\omega \\
& \text{und } (\forall n \geq i) B(p)(2n+1) = 0^\omega \\
& \text{und } (\forall n < i) B(p)(2n) \neq 0^\omega \\
& \text{und } (\forall n < i) B(p)(2n+1) \neq 0^\omega
\end{aligned}
$$

Es ist möglich, diese Behauptungen durch Induktion zu beweisen.

Falls $p = 0^\omega$ und $g \in LLPO_{\infty,\infty}$, dann ist $g \circ B(p)$ gerade und $A(p, g \circ B(p)) = 0$.
Falls $p \neq 0^\omega$ und $g \in LLPO_{\infty,\infty}$, dann ist $g \circ B(p)$ ungerade und $A(p, g \circ B(p)) = 1$.
Also gilt $\Omega \leq_2 LLPO_{\infty,\infty}$.
$\underline{LLPO_{\infty,\infty} \not\leq_2 \Omega}$:
Ähnlich wie im ersten Teil des Beweises kann man $MLPO_n \leq_2 LLPO_{\infty,\infty}$ für alle $n \geq 2$ nachweisen. Aus $LLPO_{\infty,\infty} \leq_2 \Omega$ würde dann $MLPO_n \leq_2 \Omega$ folgen, was aber an früherer Stelle widerlegt wurde. □

Zusammenfassend gelten folgende Beziehungen:

Satz 27.

- $LLPO_{\infty,m} \leq LLPO_{\infty,\infty}$ *für alle* m
- $LLPO_{\infty,\infty} \leq LLPO_{l,k}$ *ist falsch für alle* l *und* k
- $LLPO_{n,m} \leq LLPO_{\infty,\infty}$ *für alle* $n, m \in \mathbb{N}$
- $LLPO_{\infty,\infty} \leq LLPO_{\infty,k}$ *ist falsch für alle* $k \in \mathbb{N}$
- $LLPO_{\infty,\infty} \leq LLPO_{\infty,\infty}$

Die noch fehlenden Beweise sind einfach. Es fällt auf, dass auch hier die Formeln von Satz 15 anwendbar sind:

- $m \leq \infty$ und $m(\infty \text{ div } \infty) + min(\infty \bmod \infty, m) = m + 0 \leq \infty$.
- $\infty \leq k$ ist für kein $k \in \mathbb{N}$ erfüllt.
- $m \leq \infty$ und $m(\infty \text{ div } n) + min(\infty \bmod n, m) = m \cdot \infty + 0 \leq \infty$.
- $\infty \leq k$ ist für kein $k \in \mathbb{N}$ erfüllt.
- $\infty \leq \infty$ und $\infty(\infty \text{ div } \infty) + min(\infty \bmod \infty, \infty) = \infty + 0 \leq \infty$.

Somit kann man verallgemeinern:

Definition 23 ($LLPO_{n,m}$ verallgemeinert)**.**
Für alle $\infty \geq n \geq m \geq 0$ *sei* $LLPO_{n,m}$ *die Menge aller Funktionen* f *mit*

$$\begin{aligned} Def(f) &= \{\langle p_1, p_2, \ldots, p_n\rangle \mid \textit{es gibt höchstens } m \textit{ Zahlen } j \textit{ mit } p_j \neq 0^\omega \\ &\qquad \textit{und ein } k \textit{ mit } p_k = 0^\omega\} \\ \textit{bzw. } Def(f) &= \{\langle p_1, p_2, \ldots\rangle \mid \textit{es gibt höchstens } m \textit{ Zahlen } j \textit{ mit } p_j \neq 0^\omega \\ &\qquad \textit{und unendlich viele Zahlen } k \textit{ mit } p_k = 0^\omega\} \end{aligned}$$

und $f\langle p_1, p_2, \ldots, p_n\rangle = k$ *bzw.* $f\langle p_1, p_2, \ldots\rangle = k$ *für ein* k *mit* $p_k = 0^\omega$.

Satz 28 ($LLPO_{n,m} \leq_2 LLPO_{l,k}$ verallgeinert)**.**
Sei $\infty \geq n \geq m \geq 0$ *und* $\infty \geq l \geq k \geq 0$.
Dann gilt $LLPO_{n,m} \leq_2 LLPO_{l,k}$
$\iff m = 0$ *oder* $m = n = 1$ *oder* ($m \leq k$ *und* $k \geq m(l \operatorname{div} n) + \min(l \bmod n, m)$).

3.2.7 $LLPO^\infty$

Definition 24 ($LLPO^\infty$).
Sei $LLPO^\infty$ die Menge aller Funktionen f mit

$$\begin{aligned}&Def(f) = \{\langle i, p_1, p_2, \ldots, p_i\rangle \mid \text{ es gibt höchstens eine Zahl } k\\ &\text{mit } p_k \neq 0^\omega \text{ und } i \geq 2 \text{ und } k \leq i\}\\ &\text{und } f(\langle i, p_1, p_2, \ldots, p_i\rangle) := k \text{ für ein } k \text{ mit } p_k = 0^\omega \text{ und } k \leq i.\end{aligned}$$

Satz 29 ($LLPO_{n,1} \leq LLPO^\infty$).
$LLPO_{n,1} \leq LLPO^\infty$ für alle n.

Beweis. Sei B definiert durch $B\langle p_1, p_2, \ldots, p_n\rangle := \langle n, \langle p_1, p_2, \ldots, p_n\rangle\rangle$ und A durch $A\langle p, q\rangle := q$.
Sei $f \in LLPO^\infty$. Dann gilt

$$\begin{aligned}&A\langle p_1, p_2, \ldots, p_n, f \circ B\langle p_1, p_2, \ldots, p_n\rangle\rangle\\ &= f \circ B\langle p_1, p_2, \ldots, p_n\rangle\\ &= f\langle n, \langle p_1, p_2, \ldots, p_n\rangle\rangle\end{aligned}$$

für alle $\langle p_1, p_2, \ldots, p_n\rangle \in \mathbb{B}$. Also gilt $LLPO_{n,1} \leq LLPO^\infty$ für alle $n > 0$. □

Satz 30 ($LLPO^\infty \not\leq LLPO_{n,1}$).
$LLPO^\infty \leq LLPO_{n,1}$ ist falsch für alle $n > 2$ und wahr für $n = 2$.

Beweis. Definiere B durch $B\langle i, \langle p_1, \ldots, p_i\rangle\rangle := \langle p_1, p_2\rangle$
und A durch $A\langle q, x\rangle := x$. Sei $g \in LLPO_{2,1}$.
Dann gilt $A\langle\langle i, \langle p_1, \ldots, p_i\rangle\rangle, g \circ B\langle i, \langle p_1, \ldots, p_i\rangle\rangle\rangle$
$= g \circ B\langle i, \langle p_1, \ldots, p_i\rangle\rangle = g(\langle p_1, p_2\rangle)$
für alle $\langle i, \langle p_1, \ldots, p_i\rangle\rangle \in \mathbb{B}$. Also gilt $LLPO^\infty \leq LLPO_{2,1}$.

Annahme: $LLPO^\infty \leq_2 LLPO_{n,1}$ für $n > 2$.
Nach Satz 29 gilt $LLPO_{2,1} \leq_2 LLPO^\infty$.
Dann folgt $LLPO_{2,1} \leq_2 LLPO_{n,1}$ wegen der Transitivität von $\leq_2$.
Aber für $n > 2$ ist das nicht richtig nach Satz 3. □

Satz 31 ($LLPO_\infty \leq LLPO^\infty$).
$LLPO_\infty \leq LLPO^\infty$.

Beweis. Sei $B :\subseteq \mathbb{B} \longrightarrow \mathbb{B}$ definiert durch $B\langle p_1, p_2, \ldots\rangle := \langle 2, \langle p_1, p_2\rangle\rangle$
und $A :\subseteq \mathbb{B} \longrightarrow \mathbb{N}$ durch $A\langle p, x\rangle := x$.
Sei $f \in LLPO^\infty$. Dann gilt

$$\begin{aligned}&A\langle\langle p_1, p_2, \ldots\rangle, f \circ B\langle p_1, p_2, \ldots\rangle\rangle\\ &= f \circ B\langle p_1, p_2, \ldots\rangle\\ &= f\langle 2, \langle p_1, p_2\rangle\rangle\end{aligned}$$

für alle $\langle p_1, p_2, \ldots\rangle \in \mathbb{B}$. Also gilt $LLPO_\infty \leq LLPO^\infty$. □

Satz 32 ($LLPO^\infty \not\leq LLPO_\infty$).
$LLPO^\infty \leq LLPO_\infty$ ist falsch.

Beweis. Angenommen $LLPO^{\infty} \leq LLPO_{\infty}$.
Nach Satz 29 gilt $LLPO_{n,1} \leq LLPO^{\infty}$.
Dann gilt $LLPO_{n,1} \leq_2 LLPO_{\infty}$ wegen der Transitivität von $\leq_2$.
Aber das ist falsch nach Satz 10.
Also gilt $LLPO^{\infty} \not\leq_2 LLPO_{\infty}$. $\square$

3.3 $\leq_2$-Reduzierbarkeit auf PO_n

Wir möchten uns mit speziellen Problemen beschäftigen, um das *LLPO*-Problem zu verallgemeinern:

Definition 25. *Sei S_1 definiert durch $S_1 := \{0^\omega\} \cup 0^*10^\omega$.*

Definition 26. *Sei PO_n die Menge aller Funktionen $f : S_1^n \longrightarrow \mathbb{N}$ mit der Eigenschaft*

$$\begin{aligned}&(p_i = 0^\omega \Longleftrightarrow q_i = 0^\omega \text{ für alle } 1 \leq i \leq n)\\ &\Longrightarrow f(p_1, \ldots, p_n) = f(q_1, \ldots, q_n).\end{aligned}$$

Also hängt der Wert einer Funktion $f \in PO_n$ nur davon ab, welche Argumente Nullfolgen sind und nicht davon, welche Position eine 1 in einer Folge hat.

Für das Folgende definieren wir eine Relation „$\preceq$".

Definition 27. *Auf $\{0,1\} \times \{0,1\}$ definiere $\preceq$ durch*

$$i \preceq j :\Longleftrightarrow i \leq j.$$

Auf $\{0,1\}^n \times \{0,1\}^n$ definiere $\preceq$ durch

$$(i_1, \ldots, i_n) \preceq (j_1, \ldots, j_n) :\Longleftrightarrow i_k \leq j_k \text{ für alle } 1 \leq k \leq n.$$

Auf $\mathcal{P}(\{0,1\}^n) \times \mathcal{P}(\{0,1\}^n)$ definiere $\preceq$ durch

$$M \preceq N :\Longleftrightarrow \text{ für jedes } x \in M \text{ gibt es ein } y \in N \text{ mit } x \preceq y.$$

$\prec$ wird definiert durch

$$x \prec y :\Longleftrightarrow x \preceq y \text{ und } x \neq y$$

3.3.1 Beschreibung von totalen Funktionen durch charakteristische Muster

Zunächst konzentrieren wir uns nur auf totale Funktionen $f \in PO_n$.

Die Funktionswerte dieser Funktionen hängen nur davon ab, welche Eingabefolgen Nullfolgen sind und welche nicht. Wenn in einer der Eingabefolgen ein Wert ungleich 0 auftaucht, dann ist es gleichgültig, an welcher Stelle.

Um es einfach zu machen, betrachten wir erstmal nur totale Funktionen mit einem Funktionswert (Funktionen im üblichen Sinn). Diese Funktionen haben exakt einen Wert für jede Eingabe. Partielle Funktionen werden später diskutiert und danach Funktionen mit mehr als einem Wert (mehrwertige Funktionen).

Totale Funktionen $f \in PO_n$ können einfach beschrieben werden, in dem man alle Funktionswerte für die verschiedenen Eingaben aufzählt. Da es nur darauf ankommt, ob die Eingabefolgen Nullfolgen sind oder nicht, können wir den Typ der Eingabe durch ein Bitmuster beschreiben. Das Muster 001 zum Beispiel soll bedeuten, dass die ersten beiden Folgen Nullfolgen sind und die letzte nicht. Die Funktion cp soll dieses Bitmuster liefern, z.B. beschreibt $cp(p_1, p_2, p_3) = 001$ das obige Beispiel.

Definition 28. *Die Funktion $cp : S_1^n \longrightarrow \{0,1\}^n$ ist definiert durch*

$$cp(p_1, \ldots, p_n)(i) := \begin{cases} 0 & \textit{falls } p_i = 0^\omega \\ 1 & \textit{sonst} \end{cases}$$

.

Wenn wir die verschiedenen Bitmuster als Binärzahlen betrachten, sie sortieren und die Funktionswerte einer Funktion $f \in PO_n$ in dieser Reihenfolge aufzählen, können wir f exakt beschreiben. So soll $cp(f) = (0, 1, 4, 0)$ das gleiche bedeuten wie die längere Beschreibung $f(p_1, p_2) = 0$, falls $cp(p_1, p_2) = 00$, $f(p_1, p_2) = 1$, falls $cp(p_1, p_2) = 01$, $f(p_1, p_2) = 4$, falls $cp(p_1, p_2) = 10$ und $f(p_1, p_2) = 0$, falls $cp(p_1, p_2) = 11$.

Definition 29. *Die Funktion $cp : PO_n \longrightarrow \mathbb{N}^{2^n}$ ist definiert durch*

$$cp(f)(i) := j \textit{ gdw. } f(p_1, \ldots, p_n) = j \textit{ für } bin(cp(p_1, \ldots, p_n)) = i,$$

wobei bin definiert ist durch $bin(w) := \sum_{i=1}^{n} w(i) \cdot 2^{n-i}$ für $w \in \{0,1\}^*$ und $n = Länge(w)$.

Um über die Reduzierbarkeit von f auf g zu entscheiden, ist es nicht von Interesse, welche speziellen Werte die Funktion hat. Aus einer endlichen Menge von Werten der Funktion g können andere Werte berechnet werden. Z.B. ist die konstante Funktion mit dem einzigen Wert 0 stetig genauso wie die konstante Funktion mit dem Funktionswert 1. Oder die Funktion f mit $cp(f) = (0, 0, 1, 1)$ ist äquivalent zu g mit $cp(g) = (3, 3, 1, 1)$.

Es ist nur von Interesse, ob es gleiche Funktionswerte für unterschiedliche Eingabemuster gibt und welche von ihnen gleich sind.

Dafür wird der Begriff des Funktionentyps definiert.

Definition 30 (Funktionentyp)**.** *Auf den Funktionen in PO_n wird durch $\equiv$, definiert durch $f :\equiv g$ gdw.*

$$f(p_1, \ldots, p_n) = f(q_1, \ldots, q_n) \Leftrightarrow g(p_1, \ldots, p_n) = g(q_1, \ldots, q_n)$$

für alle $p_i \in \mathbb{B}$ und $q_i \in \mathbb{B}$, $1 \leq i \leq n$, eine Äquivalenzrelation definiert.
Eine Äquivalenzklasse nennen wir einen Funtionentyp.
Wir sagen, dass zwei Funktionen $f \in PO_n$ und $g \in PO_n$ vom selben Typ sind, gdw. f und g zur selben Klasse gehören.

Wir bezeichnen eine Klasse meistens durch Angabe eines Vertreters, z.B. steht das Muster 0113 für den Funktionentyp, der auch die Funktionen mit dem Muster 2339, 0991 usw. enthält.

Es gibt nur endlich viele Typen von Funktionen bei einer gegebenen Stelligkeit.

Beispiel 11. Es gibt nur einen Typ einer totalen null-stelligen Funktion:

Nr	Typ
1	0

Beispiel 12. Es gibt zwei Typen einer einstelligen totalen Funktion:

Nr	Typ
1	01
2	00

Beispiel 13. Es gibt 15 Typen von zweistelligen totalen Funktionen:

Nr	Typ
1	0000
2	0003
3	0020
4	0100
5	0111
6	0022
7	0101
8	0110
9	0023
10	0103
11	0120
12	0113
13	0121
14	0122
15	0123

3.3.2 Reduzierbarkeit von totalen Funktionen

Wir werden jetzt die Reduzierbarkeit auf totalen Funktionen untersuchen.

Für die Funktionen ohne Argument gibt es nur einen Funktionentyp:

Beispiel 14. Reduzierbarkeit unter nullstelligen Funktionen:

Klasse	Typ
1	0

Dieser Funktionstyp bildet die einzige Äquivalenzklasse, die Klasse der konstanten Funktionen.

Beispiel 15. Für einstellige Funktionen gibt es zwei Äquivalenzklassen.

Für $cp(f) = (0,0)$ und $cp(g) = (0,1)$ gilt $f \leq_2 g$ und $g \not\leq_2 f$, da f stetig ist und g nicht. Eine Tabelle der Funktionstypen mit absteigendem Grade der Unstetigkeit sieht folgendermaßen aus:

Klasse	Typ
1	01
2	00

Für mehrstellige Funktionen mit zwei Argumenten gibt es 4 Äquivalenzklassen von Funktionen, wieder mit absteigendem Grade der Unstetigkeit:

Beispiel 16. Reduzierbarkeit unter 2-stelligen Funktionen:

Klasse	Typen					
1	0123	0113	0122	0023	0103	0121
2	0100	0110	0020	0120		
3	0101	0022	0111	0003		
4	0000					

Beispiel 17. Reduzierbarkeit unter 0-, 1- und 2-stelligen Funktionen:

Klasse	Typen					
1	0123	0113	0122	0023	0103	0121
2	0100	0110	0020	0120		
3	0101	0022	0111	0003	01	
4	0000	00	0			

Im nächsten Abschnitt wird ein Algorithmus präsentiert, um eine Antwort auf die Frage zu berechnen, ob eine Funktion reduzierbar auf eine andere ist oder nicht.

Die Äquivalenzklassen in den obigen Beispielen sind mit diesem Algorithmus berechnet worden.

3.3.3 Entscheidungsalgorithmus für die Reduzierbarkeit von totalen Funktionen

Es erhebt sich die Frage, ob es möglich ist, die Reduzierbarkeit einer Funktion $f \in PO_n$ auf eine andere Funktion $g \in PO_m$ zu entscheiden.
Die Antwort ist positiv. Hier ist ein Algorithmus:

Zunächst baue einen Baum für die $\preceq$-Relation der Bitmuster der Eingabe von f auf.
In diesem Baum kann ein Muster y mehr als einmal auftauchen, da es mehr als ein Muster x geben kann, welches Präfix von y ist. Z.B. gilt $01 \preceq 11$ und $10 \preceq 11$. So wird 11 als Sohn des Knotens 01 und ebenso als Sohn von Knoten 10 vorkommen.

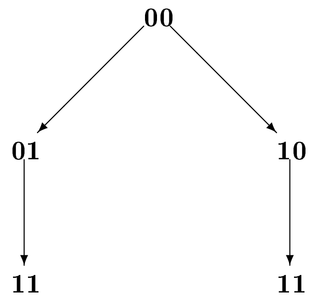

Abbildung 4: Baum der Eingabemuster

Dann ordne allen Knoten die Bitmuster der Eingabe von g zu, z.B.

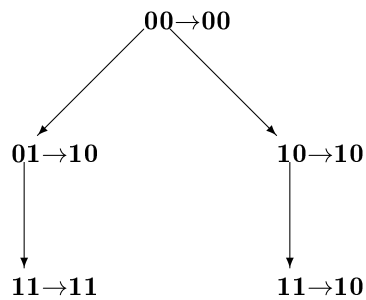

Abbildung 5: Labeling des Baums

Diese Zuordnung (Labeling) hat die Einschränkung zu respektieren, dass jedes Label eines Knotens zu allen Labeln der Söhne und Nachfolger in der $\preceq$-Relation stehen muss.

Dann ordnen wir jedem Bitmuster der Argumente von f die Menge von allen Labeln von Knoten für dieses Muster im Baum zu. Wir nennen diese Mengen *Labelsets*.

cp(p)	labelset(p)
00	{00}
01	{10}
10	{10}
11	{11,10}

Dann müssen wir die folgende Regel für alle Muster von Folgen p und q testen.
Da alle Folgen mit demselben charakteristischen Muster die selben Funktionswerte haben, schreiben wir $f(i) = q$ statt $f(p) = q$ für alle p mit $cp(p) = i$ oder $cp(f(i)) = j$ statt $cp(q) = j$ für alle q mit $f(p) = q$ und $cp(p) = i$.
Die Menge der Funktionswerte $g(labelset(i))$ soll $labelset_g(i)$ genannt werden.

Regel 1. Wenn $i \preceq j$ und $f(i) \neq f(j)$, dann gilt $labelset_g(i) \cap labelset_g(j) = \emptyset$.

Wenn diese Regel nicht erfüllt ist für ein Paar i, j, dann muss eine andere Zuordnung mit Bitmustern getestet werden. Wenn die Regel erfüllt ist, dann ist $f \leq_2 g$ wahr. Wenn alle Zuordnungen ohne Erfolg getestet werden, dann ist $f \leq_2 g$ falsch.

Beispiel 18. Sei $cp(f) = (0,1,2,3)$ und $cp(g) = (0,3,3,4)$. Dann ergibt das Labeling der Abbildungen oben das folgende Resultat:

i	$f(i)$	$labelset(i)$	$labelset_g(i)$
00	0	{00}	{0}
01	1	{10}	{3}
10	2	{10}	{3}
11	3	{11,10}	{3,4}

Dann gilt $10 \preceq 11$ und $f(10) = 2 \neq f(11) = 3$. Aber wegen $g(\{10\}) = \{3\}$ und $g(\{10, 11\}) = \{3, 4\}$ erhalten wir $g(\{10\}) \cap g(\{10, 11\}) = \{3\} \neq \emptyset$. Damit ist die Regel nicht erfüllt. Aber mit einer anderen Zuordnung ist es möglich, die Regel zu erfüllen:

i	$f(i)$	$labelset(i)$	$labelset_g(i)$
00	0	{00}	{0}
01	1	{01}	{3}
10	2	{10}	{3}
11	3	{11}	{4}

Vergleich von allen Paaren i, j mit $i \prec j$:

i	$f(i)$	j	$f(j)$	$labelset(i)$	$labelset_g(i)$	$labelset(j)$	$labelset_g(j)$
00	0	01	3	{00}	{0}	{01}	{3}
00	0	10	3	{00}	{0}	{10}	{3}
00	0	11	4	{00}	{0}	{11}	{4}
01	1	11	4	{01}	{3}	{11}	{4}
10	2	11	4	{10}	{3}	{11}	{4}

In diesem Fall sind $f(i)$ und $f(j)$ in allen Fällen ungleich. Aber $labelset_g(i) \cap labelset_g(j) = \emptyset$ in jedem Fall. Also gilt $f \leq_2 g$.

Hier ist der Algorithmus zum Testen der Reduzierbarkeit zwischen totalen Funktionen in PO_n und PO_m:

Algorithmus 1.

```
program compare
Input: n := number of arguments of f
         m := number of arguments of g
         ♯ the function values of f
         for i := 0 to 2^n − 1 begin
              f(i) := cp(f)(i)
         end
         ♯ the function values of g
         for i := 0 to 2^m − 1 begin
              g(i) := cp(g)(i)
         end
Output: true, if f ≤_2 g
            false, if f ≰_2 g
begin
   f_n := 2^n − 1
   g_n := 2^m − 1

   ♯ create the tree t
   counter:=0
   create_tree (t,0^n)

   ♯ labeling of the tree t
   red:=false
   label_tree (t)

   ♯ output of the result
   output red
end


procedure create_tree (t,content)
♯ creates a node t with content and sons, in which
♯ each 0 in the content will be replaced by a 1
begin
   create a new node and let t point to it
   counter:=counter+1
   t.argument:=content
   t.counter:=counter
   for all i with content(i) = 0 do
        ♯ replace the 0 at place i with 1
        content2 := content or 0^(i−1)10^(len(content)−i)
        create_tree (s,content2)
        t.sohn_i :=s
   od
```

```
end

procedure label_tree (s)
♯ creates recursively a label for every node in the tree
begin
  for i := 0 to g_n do
     if father(s) does not exist or father(s).label ⪯ i then
       s.label:=i
       if s.counter=counter then
         if test(t)=true then red:=true fi
       fi
       for i:=1 to number of sons of s do
          label_tree (s.son_i)
       od
     fi
  od
end

function test(t)
begin
  ♯ initialization of the sets of labels
  for i:=0 to f_n do
     labelset_g(i) := ∅
  od
  ♯ computation of the sets of labels
  collect(t)
  ♯ testing the rule
  test:=true
  for j:=1 to f_n do
     for i := 0 to j − 1 do
        if rule(i,j)=false then test:=false
     od
  od
  return test
end

procedure collect(s)
♯ computation of the sets of labels
begin
  labelset_g(s.argument) := labelset_g(s.argument) ∪ g(s.label)
  for i:=1 to number of sons of s do
     collect(s.son_i)
  od
end

function rule(i,j)
♯ testing the rule
♯ if i ⪯ j and f(i) ≠ f(j)
♯ then labelset_g(i) and labelset_g(j) have to be disjoint
begin
  rule:=true
```

```
  if i ≺ j and f(i) ≠ f(j) then
    if labelset_g(i) ∩ labelset_g(j) ≠ ∅ then
      rule:=false
    fi
  fi
  return rule
end
```

Das Programm berechnet zuerst f_n und g_n. Das sind die Anzahlen aller möglichen charakteristischen Muster für die Eingaben der n-stelligen Funktion f und der m-stelligen Funktion g, um 1 dekrementiert.

Dann wird der Baum für f rekursiv aufgebaut. Jeder Knoten enthält danach ein passendes charakteristisches Muster für Eingaben von f.

Dann wird die Variable *red* mit $false$ initialisiert. Darauf wird die Prozedur *label_tree*(t) aufgerufen. Dabei ist t beim Aufruf eine Referenz auf die Wurzel des Baums. Die Prozedur probiert für den aktuellen Knoten alle charakteristischen Muster für Eingaben der Funktion g aus, wobei die $\preceq$-Relation zum Vaterknoten überprüft wird.

Wenn alle Knoten mit einem *label* versehen sind, wird die Funktion *test* aufgerufen. Diese erzeugt zu jedem Knoten i die Menge $labelset_g(i)$, das ist die Menge aller *label* zu dem charakteristischen Muster i im Baum. Das erledigt die Prozedur *collect*. Dann wird eine Variable *test* auf *true* gesetzt.

Dann findet der eigentliche Test zwischen allen charakteristischen Mustern statt, wobei die Funktion *rule* aufgerufen wird. In dieser wird die Regel implementiert. Falls in einem Fall die Regel nicht erfüllt wird, wird die Variable *test* auf $false$ gesetzt.

Wenn *test* den Wert $false$ zurückgibt, dann probiert die Prozedur *label_tree* rekursiv das nächste Labeling. Wenn der Wert *true* zurückgegeben wird, dann wird *red* auf *true* gesetzt. Sind alle möglichen Labelings ohne Erfolg probiert worden, behält *red* den Wert $false$.

Dieser Inhalt von *red* wird schließlich im Hauptprogramm ausgegeben.

3.3.4 Beweis der Korrektheit des Algorithmus für totale Funktionen

Wir wollen jetzt die Korrektheit von Algorithmus 1 beweisen. $f \leq_2 g$ bedeutet, dass es stetige Funktionen A und B gibt mit

$$f(p_1, \ldots, p_n) = A(p_1, \ldots, p_n, g \circ B(p_1, \ldots, p_n)$$

für alle $p_1, \ldots, p_n \in \mathbb{B}$.
Zunächst merken wir an, dass das Labeling des Baums die Aufgabe hat, die stetige Funktion B zu berechnen. Aber es gibt dabei einige Restriktionen. Wir haben zu zeigen, dass diese Restriktionen beachtet werden können.

Zunächst definieren wir die Funktion arr, die nicht nur die Information liefert, welche Folgen Nullfolgen sind (wie die Funktion cp), sondern auch in welchen Folgen ein Wert ungleich 0 zuerst auftritt. In einem späteren Abschnitt wird ein Beispiel gegeben, in welchem gezeigt wird, dass nicht nur das charakteristische Muster cp der Eingabefolgen, sondern die Information über die Reihenfolge, in der die Einsen in den Folgen erscheinen, eine Rolle spielt. Diese Information liefert die Funktion arr.

Definition 31. *Definiere arr durch*

$$arr(p_1, \ldots, p_n)(i) := \begin{cases} 0 & \textit{falls } p_i = 0^\omega \\ k & \textit{falls es } k-1 \textit{ Folgen } p_j \textit{ gibt mit} \\ & p_j \prec p_i \textit{ oder } (p_j = p_i \textit{ und } j < i) \end{cases}$$

für alle $p_1, \ldots, p_n \in \{0,1\}^\omega$.

Beispiel 19. $arr(00010^\omega, 0^\omega, 010^\omega) = (2, 0, 1)$.
$arr(10^\omega, 10^\omega, 0^\omega) = (1, 2, 0)$.

Wir erweitern die Definition auf endliche Folgen:

Definition 32. *Definiere arr durch*

$$arr(p_1, \ldots, p_n)(i) := \begin{cases} 0 & \textit{falls } p_i = 0^\omega \textit{ oder } p_i = 0^m \textit{ für ein } m \in \mathbb{N} \\ k & \textit{falls es } k-1 \textit{ Folgen } p_j \textit{ gibt mit} \\ & p_j \prec p_i \textit{ oder } (p_j = p_i \textit{ und } j < i). \end{cases}$$

für alle $p_1, \ldots, p_n \in \{0,1\}^\omega \cup \{0,1\}^*$.

Beispiel 20. $arr(00010, 0^\omega, 010) = (2, 0, 1)$.
$arr(10^\omega, 100, 0) = (1, 2, 0)$.

Für arr können wir eine $\preceq$-Relation definieren:

Definition 33. $arr(p_1, \ldots, p_n) \preceq arr(q_1, \ldots, q_n)$
gdw. $arr(p_1, \ldots, p_n) = c_1 \ldots c_n$ *und* $arr(q_1, \ldots, q_n) = d_1 \ldots d_n$
und $c_i \leq d_i$ *für alle* $1 \leq i \leq n$.
$arr(p_1, \ldots, p_n) \prec arr(q_1, \ldots, q_n)$
$:\Longleftrightarrow arr(p_1, \ldots, p_n) \preceq arr(q_1, \ldots, q_n)$ *und* $arr(p_1, \ldots, p_n) \neq arr(q_1, \ldots, q_n)$

Wir beweisen zwei Lemmas:

Lemma 3. *Wenn $f \leq_2 g$, dann findet der Algorithmus ein Labeling des Baumes, das die Regel 1 erfüllt.*

Lemma 4. *Wenn der Algorithmus ein Labeling des Baumes erzeugt, welches die Regel 1 erfüllt, dann gilt $f \leq_2 g$.*

Beweis von Lemma 3.
Wir haben zu zeigen: Wenn $f \leq_2 g$, dann findet der Algorithmus ein Labeling des Baumes, das die Regel erfüllt.
Falls $f \leq_2 g$, dann gibt es stetige Funktionen A und B mit $f(p_1, \dots, p_n) = A(p_1, \dots, p_n, g \circ B(p_1, \dots, p_n))$.
Wir müssen zeigen, dass der Algorithmus mindestens ein Labeling erzeugt, welches die Regel erfüllt.

Wir zeigen, dass es stetige Funktionen A' und B' gibt mit den folgenden Eigenschaften:

1. $B'(p_1, \dots, p_n) = B'(q_1, \dots, q_n)$ für alle $p_1, \dots, p_n$ und $q_1, \dots, q_n$ mit $arr(p_1, \dots, p_n) = arr(q_1, \dots, q_n)$ (B' hängt nur von arr ab)

2. $A(p_1, \dots, p_n, g \circ B(p_1, \dots, p_n)) = A'(p_1, \dots, p_n, g \circ B'(p_1, \dots, p_n))$ für alle $p_1, \dots, p_n$ (A' und B' verhalten sich zusammen wie A und B)

3. Für alle $p_1, \dots, p_n$ gibt es $p_1^*, \dots, p_n^*$ mit $arr(p_1^*, \dots, p_n^*) = arr(p_1, \dots, p_n)$ und $cp(B(p_1^*, \dots, p_n^*)) = cp(B'(p_1^*, \dots, p_n^*))$ (es gibt eine Eingabe, für die sich B' verhält wie B)

Diese Funktionen werden rekursiv definiert.

Induktionsanfang:
Sei $(p_1, \dots, p_n) = (0^\omega, \dots, 0^\omega)$. Daraus folgt $arr(p_1, \dots, p_n) = cp(p_1, \dots, p_n) = (0, \dots, 0)$.
Sei $f(p_1, \dots, p_n) = A(p_1, \dots, p_n, g \circ B(p_1, \dots, p_n)) = k$.
Sei $g \circ B(p_1, \dots, p_n) = k'$.
Dann gilt $f(p_1, \dots, p_n) = A(p_1, \dots, p_n, k') = k$.
Sei $cp(B(p_1, \dots, p_n)) = (c_1, \dots, c_n) =: K(0, \dots, 0)$.
Definiere B' durch $B'(p_1, \dots, p_n) := (c_1 0^\omega, \dots, c_n 0^\omega)$.
B' ist stetig auf $\{(0^\omega, \dots, 0^\omega)\}$.
Definiere A' durch $A'(q_1, \dots, q_n, k') := k$ für alle $q_1, \dots, q_n \in \mathbb{B}$.
A' ist stetig auf $\{(q_1, \dots, q_n, k') | q_1, \dots, q_n \in \mathbb{B}\}$.
Dann gilt $f(p_1, \dots, p_n) = A(p_1, \dots, p_n, g \circ B(p_1, \dots, p_n))$
$= A(p_1, \dots, p_n, k') = A'(p_1, \dots, p_n, k') = A'(p_1, \dots, p_n, g \circ B'(p_1, \dots, p_n))$.
Somit ist die zweite Eigenschaft erfüllt.
Die erste Eigenschaft ist trivialerweise erfüllt, und die dritte auch.

Induktionsschluss:
Seien nun A' und B' definiert für $p_1, \dots, p_n$ mit $arr(p_1, \dots, p_n) = (c_1, \dots, c_n)$ und $max(c_1, \dots, c_n) = j - 1 < n$. Das bedeutet, dass es $j-1$ Folgen in $p_1, \dots, p_n$ gibt, die ungleich 0^ω sind.
Nach Induktionsannahme gibt es $p_1^*, \dots, p_n^*$ mit $arr(p_1^*, \dots, p_n^*) = (c_1, \dots, c_n)$ und $cp(B(p_1^*, \dots, p_n^*)) = cp(B'(p_1^*, \dots, p_n^*))$.
Seien $P_1, \dots, P_n$ Tupel mit $(P_1 0^\omega, \dots, P_n 0^\omega) = (p_1^*, \dots, p_n^*)$.
Für $1 \leq i \leq n$ sei $(e_1, \dots, e_n)$ ein Muster mit $c_i = 0$ und $0 < e_i = j$ und $c_r = e_r$ für alle $1 \leq r \leq n$ und $r \neq i$.

Das bedeutet, in $p_1, \ldots, p_n$ taucht eine neue 1 in Folge p_i auf, und die anderen Folgen bleiben unverändert.
Es gibt ein Muster K mit $cp(B(q_1, \ldots, q_n)) = K =: K(e_1, \ldots, e_n)$ für unendlich viele $z \in \mathbb{N}$ mit $(P_1 0^z, \ldots, P_n 0^z) \sqsubseteq (q_1, \ldots, q_n)$ und $arr(q_1, \ldots, q_n) = (e_1, \ldots, e_n)$, da es nur endlich viele unterschiedliche Werte für $cp(B(q_1, \ldots, q_n))$ gibt.

Sei $y := K - cp(B(p_1^*, \ldots, p_n^*))$.[8]

Nach Umbenennung von B' in B^* erweitere die Definition von B^* durch

$B'(q_1, \ldots, q_n)(i)(j)$

$$:= \begin{cases} B^*(q_1, \ldots, q_n)(i)(j) & \text{falls } arr([[q_1]]_i, \ldots, [[q_n]]_i) = (c_1, \ldots, c_n) \\ y(j) & \text{falls } arr([[q_1]]_i, \ldots, [[q_n]]_i) = (e_1, \ldots, e_n) \\ & \text{und } arr([[q_1]]_{i-1}, \ldots, [[q_n]]_{i-1}) \\ & = (c_1, \ldots, c_n) \\ 0 & \text{sonst.} \end{cases}$$

Dann ist B' stetig auf $\{(q_1, \ldots, q_n) | arr(q_1, \ldots, q_n) \preceq (e_1, \ldots, e_n)\}$ und $cp(B'(q_1, \ldots, q_n)) = K$ für $arr(q_1, \ldots, q_n) = (e_1, \ldots, e_n)$.
Sei $K' := g \circ B'(q_1, \ldots, q_n)$ für $arr(q_1, \ldots, q_n) = K$.
Dann gilt $f(q_1, \ldots, q_n) = A(q_1, \ldots, q_n, g \circ B(q_1, \ldots, q_n))$
$= A(q_1, \ldots, q_n, g \circ B'(q_1, \ldots, q_n))$
$= A(q_1, \ldots, q_n, K')$ für $arr(q_1, \ldots, q_n) = (e_1, \ldots, e_n)$.

Erweitere die Definition von A' durch
$A'(q_1, \ldots, q_n, K') := f(q_1, \ldots, q_n)$ für alle $q_1, \ldots, q_n$ mit $arr(q_1, \ldots, q_n) = (e_1, \ldots, e_n)$.
Dann ist A' stetig.

Dann gilt $cp(B'(q_1, \ldots, q_n)) = cp(B'(r_1, \ldots, r_n))$ für alle $q_1, \ldots, q_n$ und $r_1, \ldots, r_n$ mit $arr(q_1, \ldots, q_n) = arr(r_1, \ldots, r_n) = (e_1, \ldots, e_n)$.
Somit ist die erste Eigenschaft erfüllt.

Ferner gilt $A'(q_1, \ldots, q_n, g \circ B'(q_1, \ldots, q_n))$
$= A'(q_1, \ldots, q_n, K')$
$= f(q_1, \ldots, q_n)$
$= A(q_1, \ldots, q_n, g \circ B(q_1, \ldots, q_n))$
für $arr(q_1, \ldots, q_n) = (e_1, \ldots, e_n)$.
Somit ist die zweite Eigenschaft erfüllt.

Und nach Konstruktion gibt es $q_1^*, \ldots, q_n^*$ mit $arr(q_1^*, \ldots, q_n^*) = (e_1, \ldots, e_n)$ und $cp(B(q_1^*, \ldots, q_n^*)) = cp(B'(q_1^*, \ldots, q_n^*))$.
Somit ist die dritte Eigenschaft erfüllt.

[8] Die Subtraktion ist komponentenweise definiert: $(a_1, \ldots, a_n) - (b_1, \ldots, b_n) = (a_1 - b_1, \ldots, a_n - b_n)$

Also gibt es eine stetige Funktion B' mit dem selben Muster für alle Folgen mit demselben Wert für arr. Die Funktionswerte unter der Funktion g bei Argumenten mit diesem Muster werden im Labeling des Baums benutzt.

Die Stetigkeit von A' erfordert, dass $f(p_1, \ldots, p_n) = f(q_1, \ldots, q_n)$ oder $g \circ B'(p_1, \ldots, p_n) \neq g \circ B'(q_1, \ldots, q_n)$, falls $arr(p_1, \ldots, p_n) \preceq arr(q_1, \ldots, q_n)$.

Dann gilt $labelset_g(cp(p_1, \ldots, p_n)) \cap labelset_g(cp(q_1, \ldots, q_n)) = \emptyset$.

Somit wird die Regel des Algorithmus erfüllt. □

Die stetige Funktion B' in diesem Beweis kann mit dem folgenden Algorithmus berechnet werden:

Algorithmus:

input: sequences $p_1, \ldots, p_n$
output: sequences $B'(p_1, \ldots, p_n)$

$k := 0$
$x(0) := (p_1(0), \ldots, p_n(0))$
pattern:=arr(x(0))
result:=K(pattern)
output result
Loop
    $k := k + 1$
    $x(k) := (p_1(k), \ldots, p_n(k))$
    pattern:=$arr([[x(1)]]_k, \ldots, [[x(n)]]_k)$
    new:=K(pattern)
    result:=new-result
    output result
    result:=new
end loop

Beweis von Lemma 4.
Wenn der Algorithmus ein Labeling des Baumes erzeugt, das die Regel einhält, dann ist es möglich, stetige Funktionen B und A zu konstruieren mit $f(p_1, \ldots, p_n) = A(p_1, \ldots, p_n, g \circ B(p_1, \ldots, p_n))$.
Der folgende Algorithmus berechnet die stetige Funktion B.

Algorithmus:

$k := 0$
$x(0) := (p_1(0), \ldots, p_n(0))$
pattern:=x(0)
$t :=$ the first node reached by depth-first-search in the tree
    with t.argument=pattern
result:=t.label
output result
Loop
    $k := k + 1$

```
    x(k) := (p_1(k), ..., p_n(k))
    pattern:=max(pattern, x(k))
    t := the first node reached by depth-first-search in the subtree of t
         with t.argument=pattern
    new:=t.label
    result:=new-result
    output result
    result:=new
end loop
```

Die Funktion max im Algorithmus ist komponentenweise definiert:
$max((a_1, \ldots, a_n), (b_1, \ldots, b_n)) := (max(a_1, b_1), \ldots, max(a_n, b_n))$.

Es ist ebenso möglich, die stetige Funktion A zu konstruieren. Sie hängt von dem Labeling ab, das die Regel erfüllt.

Definiere A durch

$$A(p_1, \ldots, p_n, k) := \begin{cases} f(p_1, \ldots, p_n) & \text{falls ein Knoten t durch Tiefensuche} \\ & \text{gefunden wird mit} \\ & t.argument = cp(p_1, \ldots, p_n) \\ & \text{und } t.label = (e_1, \ldots, e_n) \\ & \text{und } g(q_1, \ldots, q_n) = k \\ & \text{für } cp(q_1, \ldots, q_n) = (e_1, \ldots, e_n) \\ div & \text{sonst.} \end{cases}$$

Wenn $arr(p_1, \ldots, p_n) \preceq arr(q_1, \ldots, q_n)$ und t_1 ist der Knoten für $(p_1, \ldots, p_n)$ und t_2 ist der Knoten für $(q_1, \ldots, q_n)$, dann gilt nach der Regel für das Labeling $f(p_1, \ldots, p_n) = f(q_1, \ldots, q_n)$ oder $g(t_1.label) \neq g(t_2.label)$.
Somit ist A stetig.

Weiter gilt nach Definition von A und B

$$f(p_1, \ldots, p_n) = A(p_1, \ldots, p_n, g \circ B(p_1, \ldots, p_n))$$

für alle $p_1, \ldots, p_n \in \mathbb{B}$. □

Aus Lemma 3 und 4 folgt unmittelbar[9]:

[9] Eine kleine Auslassung könnte irritieren: in den Beweisen der Lemmas nehmen wir an, dass der Wert 1 in einer Folge nach der anderen erscheint, aber mehr als eine 1 kann in verschiedenen Folgen an gleicher Stelle vorkommen. Die Funktion arr interpretiert in diesem Falle die 1 in der ersten Folge als das erste Vorkommen und die 1 in der zweiten Folge als das nächste. So wird das gleichzeitige Vorkommen wie ein Vorkommen nacheinander behandelt. Aber dieses Vorgehen ist zulässig. Da die Funktionen in PO_n liegen, hängt der Funktionswert nur vom charakteristischen Muster ab. Wenn die Zahl 1 in zwei Folgen auf einmal auftaucht, gibt es unendlich viele Eingabefolgen mit dem selben Muster, aber die 1 erscheint nacheinander.

Satz 33 (Algorithmus für die Reduzierbarkeit von totalen Funktionen).
Sei $f \in PO_n$ *and* $g \in PO_m$.
$f \leq_2 g$, *gdw. der Algorithmus ein Labeling des Baumes findet, das die Regel 1 erfüllt:*
Falls $i \preceq j$ *und* $f(i) \neq f(j)$, *dann gilt* $labelset_g(i) \cap labelset_g(j) = \emptyset$.

3.3.5 Anzahl der Typen von n-stelligen totalen Funktionen

Es wäre interessant, die Zahl der Äquivalenzklassen von Funktionen in PO_n zu berechnen. Hier werden wir nur die Anzahl von Funktionstypen berechnen.
Es ist klar, dass die Anzahl von Funktionstypen eine obere Grenze für die Anzahl von Äquivalenzklassen von totalen Funktionen in PO_n bildet. Alle Funktionen vom gleichen Typ sind reduzierbar aufeinander bezüglich $\leq_2$. Die Anzahl von Äquivalenzklassen wird wesentlich kleiner sein.

Aber wenn wir die Äquivalenzklassen bezüglich $\leq_2$ ermitteln wollen, können wir testen, welche Funktionstypen äquivalent sind. Da ihre Zahl endlich ist, handelt es sich um ein endliches Problem[10].

Es gibt 1 Typ von totalen null-stelligen Funktionen und es gibt 2 Typen von totalen 1-stelligen Funktionen, 15 Typen von totalen 2-stelligen Funktionen, 4140 Typen von totalen 3-stelligen Funktionen usw.

Wie viele Typen gibt es im allgemeinen? Es ist möglich, diese Zahl zu berechnen:

Definition 34. *Definiere $y_{n,i}$ für $n, i \in \mathbb{N}$ durch*

$$\begin{aligned}
y_{n,0} &:= 0 \\
y_{n,1} &:= 1 \text{ für } n > 1 \\
y_{n,n} &:= 1 \\
y_{n,i} &:= 0 \text{ für alle } i > n \\
y_{n,i} &:= i \cdot y_{n-1,i} + y_{n-1,i-1} \text{ für alle } n > i > 1.
\end{aligned}$$

Definition 35. *Definiere z_n durch $z_0 := 1$ und*

$$z_n := \sum_{i=1}^{n} y_{n,i}$$

für $n > 0$.

Das Resultat:

n	z_n
0	1
1	1
2	2
3	5
4	15
5	52
6	203
7	877
8	4140
9	21147
10	115975
...	...

Dann gilt das Folgende:

[10] Die Rechenzeit wächst aber vermutlich mit zunehmender Stelligkeit n exponentiell.

Satz 34 (Anzahl von Funktionstypen)**.**
Die Anzahl von Funktionstypen von totalen Funktionen aus PO_n ist z_{2^n}.

Beweis. Zunächst stellen wir fest, dass $y_{n,i}$ die Anzahl von Typen von n-Tupeln mit i verschiedenen Zahlen ist. Für $y_{n,1}$ und $y_{n,n}$ ist das klar. Für $i > n$ kann es kein Tupel geben. Wir beweisen die Behauptung für die letzte Gleichung durch Induktion über n:
$n = 1$: $y_{1,1} = 1$ und es gibt nur einen Typ eines 1-Tupels mit einem Wert.
$n \to n + 1$: $y_{n+1,i} = i \cdot y_{n,i} + y_{n,i-1}$.
Falls es x $(n + 1)$-Tupel mit i verschiedenen Zahlen gibt, dann können die ersten n Elemente i verschiedene Werte oder nur $i - 1$ verschiedene Werte haben. Im ersten Fall kann der letzte Wert irgendeine der i verschiedenen Zahlen sein, im zweiten Fall muss die letzte Zahl eine von den anderen verschiedene Zahl sein. Die Induktionsannahme sagt, dass es $y_{n,i}$ n-Tupel mit i verschiedenen Werten gibt und $y_{n,i-1}$ n-tupels mit $i - 1$ verschiedenen Werten. Somit gibt es $i \cdot y_{n,i} + y_{n,i-1}$ $(n + 1)$-Tupel mit i verschiedenen Werten.

Nun müssen wir nur bedenken, dass eine Funktion aus PO_n mit n Argumenten 2^n verschiedene Eingabemuster besitzt und somit jede Funktion in PO_n 2^n verschiedene Funktionswerte haben kann. Da es $1, 2, \ldots, 2^n$ verschiedene Werte geben kann, ist die Anzahl aller Funktionstypen $y_{2^n,1} + y_{2^n,2} + \ldots + y_{2^n,2^n} = z_{2^n}$. □

Hier ist ein Algorithmus, mit dem man alle Funktionentypen mit n Argumenten erzeugen kann:
Der Aufruf wird gemacht mit

```
create (2^n-1,0,"").
```

Algorithmus 2.

```
procedure create(k,i,v)
  w:=v
  for j=0 to i
    if j=i or (j appears in v)
    then
      w:=v & j
      if i=k
      then
        print w
      else
        create (k,i+1,w)
      end
    end
  next j
end
```

3.3.6 Reduzierbarkeit zwischen partiellen Funktionen

Nun soll die Reduzierbarkeit auf partiellen Funktionen untersucht werden.

Auch in diesem Fall können die Funktionen einfach beschrieben werden. Es muss nur ein weiterer möglicher „Funktionswert" d für *div* hinzugefügt werden. Wir werden ihn manchmal als „–" schreiben, um ihn in Tabellen besser von richtigen Funktionswerten unterscheiden zu können.

Für partielle Funktionen können wir zwischen normaler und strenger Reduzierbarkeit (siehe Def. 44-46) unterscheiden. Zunächst soll die normale Reduzierbarkeitsrelation „$\leq_2$" auf partiellen Funktionen betrachtet werden.

Beispiel 21. Es gibt zwei Typen von partiellen null-stelligen Funktionen. Sie bilden zwei unterschiedliche Äquivalenzklassen.

Klasse	Typ
1	0
2	-

Beispiel 22. Es gibt 5 Typen von partiellen einstelligen Funktionen. Sie bilden drei Äquivalenzklassen bezüglich $\leq_2$.

Klasse	Typ		
1	01		
2	0-	00	-1
3	--		

Beispiel 23. Es gibt 52 Typen von partiellen 2-stelligen Funktionen. Sie bilden 5 Äquivalenzklassen bezüglich $\leq_2$.

Klasse	Typ							
1	0123	0113	0122	0023	0103	0121	01-3	0-23	
2	0100	0110	0020	0120	0-20	01-0			
3	01--	011-	002-	010-	012-	0-2-			
	0101	0022	0111	0003	-122	--23	-113	-121	-123
	0--3	0-22	00-3	0-03	01-1	-1-3			
4	-12-	-1--	--2-	-11-	0---	000-			
	00-0	0-00	0--0	0-0-	00--				
	0000	---3	--22	-111	-1-1				
5	----								

Die Klassen sind absteigend nach dem Grad der Unstetigkeit geordnet. Die Reduzierbarkeitsklassen aller bisherigen Typen:

Klasse	Typ								
1	0123	0113	0122	0023	0103	0121	01-3	0-23	
2	0100	0110	0020	0120	0-20	01-0			
3	01--	011-	002-	010-	012-	0-2-			
	0101	0022	0111	0003	-122	--23	-113	-121	-123
	0--3	0-22	00-3	0-03	01-1	-1-3	01		
4	-12-	-1--	--2-	-11-	0---	000-			
	00-0	0-00	0--0	0-0-	00--				
	0000	---3	--22	-111	-1-1				
	0-	00	-1	0					
5	----	--	-						

Die Klassen sind wieder absteigend nach ihrer Unstetigkeit geordnet.

Der Algorithmus zum Testen der Reduzierbarkeit auf partiellen Funktionen ist derselbe wie für totale Funktionen, es muss nur die Regel geändert werden:

Regel 2 ($\leq_2$ für partielle Funktionen)**.**
(Falls $f(i) \neq div$, dann $div \notin labelset_g(i)$)
und (falls $f(j) \neq div$, dann $div \notin labelset_g(j)$)
und (falls ($i \preceq j$ und $f(i) \neq div$ und $f(j) \neq div$ und $f(i) \neq f(j)$),
dann $labelset_g(i) \cap labelset_g(j) = \emptyset$)

Die Regel im vorigen Algorithmus muss dann durch die folgende Funktion ersetzt werden:

Algorithmus 3. (für normale Reduzierbarkeit auf partiellen Funktionen)

function rule(i,j)
begin
  rule:=true
  if $f(i) \neq div$ and $div \in labelset_g(i)$ then
    rule:=false
  fi
  if $f(j) \neq div$ and $div \in glabelset(j)$ then
    rule:=false
  fi
  if $i \preceq j$ and $f(i) \neq div$ and $f(j) \neq div$ and $f(i) \neq f(j)$ then
    if $labelset_g(i) \cap labelset_g(j) \neq \emptyset$ then
      rule:=false
    fi
  fi
end

3.3.7 Beweis der Korrektheit des Algorithmus für partielle Funktionen

Wir haben die Korrektheit von Algorithmus 3 zu beweisen. Wieder bemerken wir, dass das Labeling des Baums den Zweck hat, die stetige Funktion B zu berechnen. Aber es gibt dabei Einschränkungen. Wir haben zu zeigen, dass diese Einschränkungen beachtet werden können.

Wir beweisen zwei Lemmas:

Lemma 5. *Wenn $f \leq_2 g$, dann findet der Algorithmus ein Labeling des Baums, das die Regel 2 erfüllt.*

Lemma 6. *Wenn der Algorithmus ein Labeling des Baumes erzeugt, das die Regel 2 erfüllt, dann gilt $f \leq_2 g$.*

Beweis von Lemma 5.
Wir haben zu zeigen: Wenn $f \leq_2 g$, dann findet der Algorithmus ein Labeling des Baums, das die Regel 2 erfüllt.
Wenn $f \leq_2 g$, dann gibt es stetige Funktions A und B mit $f(p_1, \ldots, p_n) = A(p_1, \ldots, p_n, g \circ B(p_1, \ldots, p_n))$ oder $f(p_1, \ldots, p_n) = div$.
Wir müssen zeigen, dass der Algorithmus ein Labeling erzeugt, das die Regel erfüllt.

Wir zeigen, dass es stetige Funktionen A' und B' gibt mit den folgenden Eigenschaften:

1. $B'(p_1, \ldots, p_n) = B'(q_1, \ldots, q_n)$ für alle $p_1, \ldots, p_n$ and $q_1, \ldots, q_n$ mit $arr(p_1, \ldots, p_n) = arr(q_1, \ldots, q_n)$ (B' hängt nur von arr ab)

2. $A(p_1, \ldots, p_n, g \circ B(p_1, \ldots, p_n)) = A'(p_1, \ldots, p_n, g \circ B'(p_1, \ldots, p_n))$ für alle $p_1, \ldots, p_n$ mit $f(p_1, \ldots, p_n) \neq div$ (A' und B' zusammen verhalten sich wie A und B)

3. Für alle $p_1, \ldots, p_n$ mit $f(p_1, \ldots, p_n) \neq div$ gibt es $p_1^*, \ldots, p_n^*$ mit $arr(p_1^*, \ldots, p_n^*) = arr(p_1, \ldots, p_n)$ und $cp(B(p_1^*, \ldots, p_n^*)) = cp(B'(p_1^*, \ldots, p_n^*))$ (es gibt eine Eingabe, für welche sich B' wie B verhält)

Diese Funktionen werden rekursiv definiert.

Induktionsanfang:
Sei $(p_1, \ldots, p_n) = (0^\omega, \ldots, 0^\omega)$. Also gilt $arr(p_1, \ldots, p_n) = cp(p_1, \ldots, p_n) = (0, \ldots, 0)$.
Fall 1:
Sei $f(p_1, \ldots, p_n) = A(p_1, \ldots, p_n, g \circ B(p_1, \ldots, p_n) = k$.
Sei $g \circ B(p_1, \ldots, p_n)) = k'$.
Dann folgt $f(p_1, \ldots, p_n) = A(p_1, \ldots, p_n, k') = k$.
Sei $cp(B(p_1, \ldots, p_n)) = (c_1, \ldots, c_n) =: K(0, \ldots, 0)$.
Definiere B' durch

$$B'(q_1, \ldots, q_n) := \begin{cases} (c_1 0^\omega, \ldots, c_n 0^\omega) & \text{falls } (q_1, \ldots, q_n) = (0, \ldots, 0) \\ div & \text{sonst} \end{cases}$$

B' ist stetig auf $\mathbb{B}^n$.
Definiere A' durch

$$A'(q_1, \ldots, q_n, l) := \begin{cases} k & \text{falls } l = k' \\ div & \text{sonst} \end{cases}$$

für alle $q_1, \ldots, q_n \in \mathbb{B}$ und $l \in \mathbb{N}$.
A' ist stetig auf $\{(q_1, \ldots, q_n, l) | q_1, \ldots, q_n \in \mathbb{B}, l \in \mathbb{N}\}$.
Dann gilt $f(p_1, \ldots, p_n) = A(p_1, \ldots, p_n, g \circ B(p_1, \ldots, p_n))$
$= A(p_1, \ldots, p_n, k') = A'(p_1, \ldots, p_n, k') = A'(p_1, \ldots, p_n, g \circ B'(p_1, \ldots, p_n))$.
Somit ist die zweite Eigenschaft erfüllt.
Die erste Eigenschaft ist trivialerweise erfüllt und die dritte auch.
Fall 2:
Sei $f(p_1, \ldots, p_n) = div$.
Definiere $K(0, \ldots, 0) := (0, \ldots, 0)$.
Definiere B' durch $B'(q_1, \ldots, q_n) := (0^\omega, \ldots, 0^\omega)$ für alle $q_1, \ldots, q_n \in \mathbb{B}$.
B' ist stetig auf $\mathbb{B}^n$.
Definiere A' durch $A'(q_1, \ldots, q_n, k) := div$ für alle $q_1, \ldots, q_n \in \mathbb{B}$ und $k \in \mathbb{N}$.
A' ist stetig auf $\{(q_1, \ldots, q_n, k) | q_1, \ldots, q_n \in \mathbb{B}\}$.
Dann gilt $f(p_1, \ldots, p_n) = div = A'(p_1, \ldots, p_n, g \circ B'(p_1, \ldots, p_n))$.
Somit ist die zweite Eigenschaft erfüllt.
Aber sie ist trivialerweise sowieso erfüllt. Da $f(p_1, \ldots, p_n) = div$, ist das Resultat von $A'(p_1, \ldots, p_n, g \circ B'(p_1, \ldots, p_n))$ unwichtig.
Die erste Eigenschaft ist trivialerweise erfüllt, und die dritte auch.

Induktionsschluss:
Nun seien A' und B' definiert für $p_1, \ldots, p_n$ mit $arr(p_1, \ldots, p_n) = (c_1, \ldots, c_n)$ mit $max(c_1, \ldots, c_n) = j - 1 < n$. Das bedeutet, dass es $j - 1$ Folgen in $p_1, \ldots, p_n$ ungleich 0^n gibt.
Nach Induktionsannahme gibt es $p_1^*, \ldots, p_n^*$ mit $arr(p_1^*, \ldots, p_n^*) = (c_1, \ldots, c_n)$ und $cp(B(p_1^*, \ldots, p_n^*)) = cp(B'(p_1^*, \ldots, p_n^*))$ oder $g \circ B(p_1^*, \ldots, p_n^*) = div$.
Seien $P_1, \ldots, P_n$ Tupel mit $(P_1 0^\omega, \ldots, P_n 0^\omega) = (p_1^*, \ldots, p_n^*)$.
Für $1 \le i \le n$ sei $(e_1, \ldots, e_n)$ ein Muster mit $c_i = 0$ und $0 < e_i = j$ und $c_r = e_r$ für alle $1 \le r \le n$ und $r \neq i$.
Das bedeutet, in $p_1, \ldots, p_n$ erscheint eine neue 1 in Folge p_i, und die anderen Folgen bleiben unverändert.
Es gibt wieder zwei Fälle:
Fall 1:
Sei $f(q_1, \ldots, q_n) = A(q_1, \ldots, q_n, g \circ B(q_1, \ldots, q_n)) = k$ für alle $q_1, \ldots, q_n$ mit $arr(q_1, \ldots, q_n) = (e_1, \ldots, e_n)$.
Dann gilt $g \circ B(q_1, \ldots, q_n) \neq div$ für alle $q_1, \ldots, q_n$ mit $arr(q_1, \ldots, q_n) = (e_1, \ldots, e_n)$.
Es gibt ein Muster K mit $cp(B(q_1, \ldots, q_n)) = K =: K(e_1, \ldots, e_n)$ für unendlich viele $z \in \mathbb{N}$ mit $(P_1 0^z, \ldots, P_n 0^z) \sqsubseteq (q_1, \ldots, q_n)$ und $arr(q_1, \ldots, q_n) = (e_1, \ldots, e_n)$, weil es nur endlich viele unterschiedliche Werte für $cp(B(q_1, \ldots, q_n))$ gibt.

Sei $y := K - cp(B(p_1^*, \ldots, p_n^*))$.[11]

[11] Die Subtraktion ist komponentenweise definiert: $(a_1, \ldots, a_n) - (b_1, \ldots, b_n) = (a_1 - b_1, \ldots, a_n - b_n)$

Nach Umbenennung von B' in B^* erweitern wir die Definition von B^* durch
$B'(q_1, \ldots, q_n)(i)(j)$

$$:= \begin{cases} B^*(q_1, \ldots, q_n)(i)(j) & \text{falls } arr([[q_1]]_i, \ldots, [[q_n]]_i) = (c_1, \ldots, c_n) \\ y(j) & \text{falls } arr([[q_1]]_i, \ldots, [[q_n]]_i) = (e_1, \ldots, e_n) \\ & \text{und } arr([[q_1]]_{i-1}, \ldots, [[q_n]]_{i-1}) \\ & = (c_1, \ldots, c_n) \\ 0 & \text{sonst.} \end{cases}$$

Dann ist B' stetig auf $\mathbb{B}^n$
und $cp(B'(q_1, \ldots, q_n)) = K$ für $arr(q_1, \ldots, q_n) = (e_1, \ldots, e_n)$.
Sei $K' := g \circ B'(q_1, \ldots, q_n)$ für $arr(q_1, \ldots, q_n) = K$.
Dann gilt $f(q_1, \ldots, q_n) = A(q_1, \ldots, q_n, g \circ B(q_1, \ldots, q_n))$
$= A(q_1, \ldots, q_n, g \circ B'(q_1, \ldots, q_n))$
$= A(q_1, \ldots, q_n, K')$ für $arr(q_1, \ldots, q_n) = K$.

Erweitere die Definition von A' durch
$A'(q_1, \ldots, q_n, K') := k$ für alle $q_1, \ldots, q_n$ mit $arr(q_1, \ldots, q_n) = (e_1, \ldots, e_n)$.

Dann gilt $cp(B'(q_1, \ldots, q_n)) = cp(B'(r_1, \ldots, r_n))$ für alle $q_1, \ldots, q_n$ und $r_1, \ldots, r_n$ mit $arr(q_1, \ldots, q_n) = arr(r_1, \ldots, r_n) = (e_1, \ldots, e_n)$.
Somit ist die erste Eigenschaft erfüllt.

Weiter gilt $A'(q_1, \ldots, q_n, g \circ B'(q_1, \ldots, q_n))$
$= A'(q_1, \ldots, q_n, K')$
$= f(q_1, \ldots, q_n)$
$= A(q_1, \ldots, q_n, g \circ B(q_1, \ldots, q_n))$
für $arr(q_1, \ldots, q_n) = (e_1, \ldots, e_n)$.
Somit ist die zweite Eigenschaft erfüllt.

Und nach Konstruktion gibt es $q_1^*, \ldots, q_n^*$ mit $arr(q_1^*, \ldots, q_n^*) = (e_1, \ldots, e_n)$ und $cp(B(q_1^*, \ldots, q_n^*)) = cp(B'(q_1^*, \ldots, q_n^*))$.
Somit ist auch die dritte Eigenschaft erfüllt.

Also gibt es eine stetige Funktion B' mit demselben Muster für alle Folgen mit demselben Wert für arr. Die Funktionswerte von g für Argumente mit diesen Mustern werden in dem Labeling des Baumes benutzt.

Die Stetigkeit von A' erfordert, dass $f(p_1, \ldots, p_n) = f(q_1, \ldots, q_n)$ oder
$g \circ B'(p_1, \ldots, p_n) \neq g \circ B'(q_1, \ldots, q_n)$, falls $arr(p_1, \ldots, p_n) \preceq arr(q_1, \ldots, q_n)$.

Dann gilt $labelset_g(cp(p_1, \ldots, p_n)) \cap labelset_g(cp(q_1, \ldots, q_n)) = \emptyset$.
Aber diese Bedingung muss nur erfüllt werden, wenn $f(p_1, \ldots, p_n) \neq div$.
Ferner folgt aus $f(q_1, \ldots, q_n) \neq div$, dass $labelset_g(cp(q_1, \ldots, q_n)) \neq \emptyset$.

Somit ist die Regel des Algorithmus erfüllt.

<u>Fall 2</u>:
Sei $f(q_1, \ldots, q_n) = div$ für alle $q_1, \ldots, q_n$ mit $arr(q_1, \ldots, q_n) = (e_1, \ldots, e_n)$.

Definiere $K(e_1, \ldots, e_n) := d$.

Nach Umbenennung von B' in B^* erweitere die Definition von B^* durch

$B'(q_1, \ldots, q_n)(i)(j)$

$$:= \begin{cases} B^*(q_1, \ldots, q_n)(i) & \text{falls } arr([[q_1]]_i, \ldots, [[q_n]]_i) = (c_1, \ldots, c_n) \\ 0 & \text{sonst.} \end{cases}$$

Dann ist B' stetig auf $\{(q_1, \ldots, q_n) | arr(q_1, \ldots, q_n) \sqsubseteq (e_1, \ldots, e_n)\}$ und $cp(B'(q_1, \ldots, q_n)) = cp(B'(p_1, \ldots, p_n))$ für $arr(q_1, \ldots, q_n) = (e_1, \ldots, e_n)$ und $arr(p_1, \ldots, p_n) = (c_1, \ldots, c_n)$.

Dann gilt $cp(B'(q_1, \ldots, q_n)) = cp(B'(r_1, \ldots, r_n))$ für alle $q_1, \ldots, q_n$ und $r_1, \ldots, r_n$ mit $arr(q_1, \ldots, q_n) = arr(r_1, \ldots, r_n) = (e_1, \ldots, e_n)$.
Somit ist die erste Eigenschaft erfüllt.

Ferner gilt $f(q_1, \ldots, q_n) = div$
für $arr(q_1, \ldots, q_n) = (e_1, \ldots, e_n)$.
Somit ist die zweite Eigenschaft erfüllt und die dritte auch.

Somit gibt es eine stetige Funktion B' mit demselben Muster für alle Folgen mit dem gleichen Wert für arr. Die Funktionswerte von g für Argumente mit diesen Mustern werden im Labeling des Baumes benutzt.
Wegen $f(q_1, \ldots, q_n) = div$ ist die Regel des Algorithmus trivialerweise erfüllt.

Die stetige Funktion B' kann mit dem folgenden Algorithmus berechnet werden:

Algorithmus:

```
input sequences p_1, ..., p_n
k := 0
x(0) := (p_1(0), ..., p_n(0))
pattern:=arr(x(0))
if K(pattern)=d
  then result:=(0, ..., 0)
  else result:=K(pattern)
endif
output result
Loop
     k := k + 1
     x(k) := (p_1(k), ..., p_n(k))
     pattern:=arr([[x(1)]]_k, ..., [[x(n)]]_k)
     if K(pattern)=d
        then output (0, ..., 0)
        else new:=K(pattern)
             output new-result
```

```
            result:=new
      endif
end loop
```

□

Beweis von Lemma 6.
Wenn der Algorithmus ein Labeling des Baumes erzeugt, das die Regel erfüllt, dann ist es möglich, stetige Funktionen B und A mit $f(p_1,\ldots,p_n) = A(p_1,\ldots,p_n, g \circ B(p_1,\ldots,p_n))$ zu konstruieren.
Der folgende Algorithmus berechnet die stetige Funktion B. Er hängt von einem Labeling ab, das die Regel erfüllt.

Algorithmus:

```
k := 0
x(0) := (p_1(0),...,p_n(0))
pattern:=x(0)
t := the first node reached by depth-first-search in the tree
     with t.argument=pattern
result:=t.label
output result
Loop
      k := k + 1
      x(k) := (p_1(k),...,p_n(k))
      pattern:=max(pattern, x(k))
      t := the first node reached by depth-first-search in the subtree of t
            with t.argument=pattern
      new:=t.label
      result:=new-result
      output result
      result:=new
end loop
```

Hier ist max wieder komponentenweise definiert:
$max((a_1,\ldots,a_n),(b_1,\ldots,b_n)) := (max(a_1,b_1),\ldots,max(a_n,b_n))$.

Es ist auch möglich, die stetige Funktion A zu berechnen. Sie hängt wieder von einem Labeling ab, das die Regel erfüllt.

Definiere A durch $A(p_1,\ldots,p_n,k)$

$$:= \begin{cases} f(p_1,\ldots,p_n) & \text{falls ein Knoten t durch Tiefensuche gefunden wird} \\ & \text{mit } cp(p_1,\ldots,p_n) \sqsubseteq t.argument \text{ und} \\ & t.label = (e_1,\ldots,e_n) \text{ und} \\ & f(p_1,\ldots,p_n) \neq div \text{ und} \\ & g(q_1,\ldots,q_n) = k \text{ für } cp(q_1,\ldots,q_n) = (e_1,\ldots,e_n) \\ div & \text{sonst.} \end{cases}$$

Falls $arr(p_1, \ldots, p_n) \preceq arr(q_1, \ldots, q_n)$ und t_1 ist der Knoten für $(p_1, \ldots, p_n)$ und t_2 ist der Knoten für $(q_1, \ldots, q_n)$, dann gilt nach der Regel für das Labeling $f(p_1, \ldots, p_n) = f(q_1, \ldots, q_n)$ oder $g(t_1.label) \neq g(t_2.label)$ oder $f(p_1, \ldots, p_n) = div$ oder $f(q_1, \ldots, q_n) = div$.
Somit ist A stetig.

Weiter gilt nach Definition von A und B

$$f(p_1, \ldots, p_n) = A(p_1, \ldots, p_n, g \circ B(p_1, \ldots, p_n)) \text{ oder } f(p_1, \ldots, p_n) = div$$

für alle $p_1, \ldots, p_n \in \mathbb{B}$. □

Aus Lemma 5 und 6 folgt unmittelbar:

Satz 35 (Algorithmus für die Reduzierbarkeit von partiellen Funktions)**.**
Seien $f \in PO_n$ und $g \in PO_m$ partielle Funktionen.
$f \leq_2 g$, gdw. Algorithmus 3 findet ein Labeling des Baumes, das Regel 2 erfüllt:

(Falls $f(i) \neq div$, dann $div \notin labelset_g(i)$)
und (falls $f(j) \neq div$, dann $div \notin labelset_g(j)$)
und (falls($i \preceq j$ und $f(i) \neq div$ und $f(j) \neq div$ und $f(i) \neq f(j)$),
dann gilt $labelset_g(i) \cap labelset_g(j) = \emptyset$)

3.3.8 Anzahl von Typen von n-stelligen partiellen Funktionen

Es gibt 2 Typen von partiellen null-stelligen Funktionen. Es gibt 5 Typen von partiellen einstelligen Funktionen und 52 Typen von partiellen 2-stelligen Funktionen.

Wie viele Typen gibt es im allgemeinen? Das kann rekursiv berechnet werden:

Definition 36. *Definiere $d_{n,i}$ durch*

$$\begin{aligned}
d_{n,0} &:= 0 \\
d_{n,1} &:= 1 \\
d_{n,n} &:= n \\
d_{n,i} &:= 0 \text{ für alle } i > n \\
d_{n,i} &:= i \cdot d_{n-1,i} + d_{n-1,i-1} + y_{n-1,i-1} \text{ für alle } n > i > 1.
\end{aligned}$$

Die Zahl $y_{n,i}$ wurde in Definition 34 erklärt.

Definition 37. *Definiere zp_n durch $zp_0 := 1$ und*

$$zp_n := \sum_{i=1}^{n} (y_{n,i} + d_{n,i})$$

für $n > 0$.

Das Resultat:

n	zp_n
0	1
1	2
2	5
3	15
4	52
5	203
6	877
7	4140
8	21147
9	115975
10	678570
...	...

Es gilt das Folgende:

Satz 36. *Die Anzahl von Funktionstypen von n-stelligen partiellen Funktionen ist zp_{2^n}.*

Beweis. Zunächst stellen wir fest, dass $d_{n,i}$ die Anzahl der n-Tupel mit i verschiedenen Werten ist, wobei d vorkommt. Für $d_{n,1}$ und $d_{n,n}$ ist das klar[12]. Für $i > n$ kann es kein Tupel geben. Wir beweisen die Behauptung für die letzte

[12] Das einzige n-Tupel mit einem Wert ist $(d, \ldots, d)$. Und es gibt n n-Tupel mit n unterschiedlichen Werten: $(d, 1, \ldots, n-1), (0, d, 2, \ldots, n-1), \ldots, (0, 1, \ldots, n-2, d)$.

Gleichung durch Induktion über n:
$n = 1$: $d_{1,1} = 1$ und es gibt nur ein 1-Tupel mit einem Wert d.
$n \to n + 1$: $d_{n+1,i} = i \cdot d_{n,i} + d_{n,i-1} + y_{n,i-1}$.
Wenn es x $(n+1)$-Tupel gibt mit i verschiedenen Werten, dann können die ersten n Elemente i verschiedene Werte oder nur $i - 1$ unterschiedliche Werte haben. Im ersten Fall kann der letzte Wert einer der i verschiedenen Zahlen sein, im zweiten Fall kann die letzte Zahl n oder d sein, aber n nur, falls die ersten Werte d nicht schon enthalten. Die Induktionsannahme sagt, dass es $d_{n,i}$ n-Tupel gibt mit i verschiedenen Werten, einschließlich d, und $d_{n,i-1}$ n-Tupel mit $i - 1$ verschiedenen Werten, einschließlich d. Somit gibt es $i \cdot d_{n,i} + d_{n,i-1} + y_{n,i-1}$ $(n+1)$-Tupel mit i verschiedenen Werten, wobei d mindestens einmal vorkommt.

Nun müssen wir nur noch beachten, dass eine Funktion mit n Argumenten 2^n verschiedene Muster von Eingaben hat, und somit jede Funktion in PO_n 2^n verschiedene Funktionswerte haben kann. Da es $1, 2, \ldots, 2^n$ verschiedene Werte geben kann, ist die Anzahl aller Funktionstypen

$$y_{2^n,1} + d_{2^n,1} + y_{2^n,2} + d_{2^n,2} + \ldots + y_{2^n,2^n} + d_{2^n,2^n} = z_{2^n}.$$

□

Wenn wir die Funktionen z und zp vergleichen, fällt ein einfacher Zusammenhang auf:

Satz 37. *$z_{n+1} = zp_n$ für alle $n \in \mathbb{N}$.*

Beweis.

$$\begin{aligned}
z_{n+1} &= \sum_{i=1}^{n+1} y_{n+1,i} \\
&= \sum_{i=1}^{n+1} (i \cdot y_{n,i} + y_{n,i-1}) \\
&= \sum_{i=1}^{n+1} (i \cdot y_{n,i}) + \sum_{i=1}^{n+1} (y_{n,i-1}) \\
&= \sum_{i=1}^{n} (i \cdot y_{n,i}) + (n+1) \cdot y_{n,n+1} + \sum_{i=0}^{n} (y_{n,i}) \\
&= \sum_{i=1}^{n} (i \cdot y_{n,i}) + (n+1) \cdot 0 + \sum_{i=1}^{n} (y_{n,i}) + y_{n,0} \\
&= \sum_{i=1}^{n} (i \cdot y_{n,i}) + \sum_{i=1}^{n} (y_{n,i}) + 0 \\
&= \sum_{i=1}^{n} (i \cdot y_{n,i}) + \sum_{i=1}^{n} (y_{n,i}) \text{ und} \\
zp_n &= \sum_{i=1}^{n} (d_{n,i} + y_{n,i}) \\
&= \sum_{i=1}^{n} (d_{n,i}) + \sum_{i=1}^{n} (y_{n,i}).
\end{aligned}$$

Somit gilt $z_{n+1} = zp_n$, gdw.

$$\sum_{i=1}^{n}(i \cdot y_{n,i}) = \sum_{i=1}^{n}(d_{n,i}).$$

Durch Induktion über n können wir eine stärkere Gleichung beweisen:
$i \cdot y_{n,i} = d_{n,i}$ für alle $n, i \in \mathbb{N}$
$n = 1$: $y_{1,i} = 0 = d_{1,i}$ für $i > 1$ und $y_{1,1} = 1 = d_{1,1}$.
Somit gilt $1 \cdot y_{1,i} = d_{1,i}$ für alle $i \in \mathbb{N}$.
$n \to n + 1$:

$$\begin{aligned} i \cdot y_{n+1,i} &= i^2 \cdot y_{n,i} + i \cdot y_{n,i-1} \\ &= i^2 \cdot y_{n,i} + (i-1) \cdot y_{n,i-1} + y_{n,i-1} \\ &= i \cdot d_{n,i} + d_{n,i-1} + y_{n,i-1} \text{ nach Induktionsannahme} \\ &= d_{n+1,i}. \end{aligned}$$

□

Das impliziert folgendes Resultat:

Satz 38. *Die Anzahl von Funktionstypen von n-stelligen partiellen Funktionen ist z_{2^n+1}.*

Beweis. Der Satz folgt direkt aus den Sätzen 37 und 34. □

Es gibt einen Algorithmus, um alle Funktionstypen von partiellen n-stelligen Funktionen zu generieren:
Der Aufruf erfolgt durch

```
create_d (2^n-1,0,"").
```

Algorithmus 4.

```
procedure create_d(k,i,v)
  w:=v
  for j=0 to i
    if j=i or (j appears in v)
    then
      w:=v & j
      if i=k
        then print w
      else
        create_d (k,i+1,w)
      end
    end
  next j
  w:=v & "d"
  if i=k
    then print w
  else
    create_d (k,i+1,w)
  end
end
```

3.3.9 Weitere Regeln zum Beweis von Nicht-Reduzierbarkeit

Die Berechnung der Reduzierbarkeitsrelation erfordert viel Rechenzeit. Es gibt einige einfache Regeln, um Nicht-Reduzierbarkeit zu beweisen, die in einigen Fällen anwendbar sind:

Satz 39. *Seien $f \in PO_n$ und $g \in PO_m$ totale Funktionen.*
Falls es für f i verschiedene Werte auf einem Pfad im Baum gibt und für g weniger als i Werte auf allen Pfaden, dann ist $f \leq_2 g$ nicht wahr.

Beweis. Falls $f \leq_2 g$, dann gibt es ein Labeling des Baumes für f, das die Regel 1 erfüllt. Falls f k verschiedene Werte auf einem Pfad hat und g weniger als k verschiedene Werte auf allen Pfaden des Baums, dann müssen auf diesem Pfad zwei Knoten mit verschiedenen Werten für f mit dem selben Label versehen werden. Wenn die Knoten die Muster i und j haben, dann gilt $i \preceq j$ oder $j \preceq i$ und $f(i) \neq f(j)$ und $labelset_g(i) \cap labelset_g(j) \neq \emptyset$. Das ist ein Widerspruch zur Regel. □

Beispiel 24. Sei $cp(f) = (0, 1, 1, 2)$ und $cp(g) = (0, 0, 0, 1)$. Dann kann f nicht reduzierbar auf g sein entsprechend der Regel.

Satz 40. *Seien $f \in PO_n$ und $g \in PO_m$ totale Funktionen.*
Falls es für f i Wechsel von Werten auf einem Pfad des Baums gibt und für g weniger als i Wechsel auf allen Pfaden, dann ist $f \leq_2 g$ nicht wahr.

Beweis. Falls $f \leq_2 g$, dann gibt es ein Labeling des Baums für f, das die Regel 1 erfüllt. Falls f k Wechsel von Werten auf einem Pfad hat und g weniger als k Wechsel auf allen Pfaden des Baumes, dann müssen auf diesem Pfad zwei Knoten mit unterschiedlichen Werten für f mit dem selben Label versehen werden. Wenn die Knoten die Muster i und j haben, dann gilt $i \preceq j$ oder $j \preceq i$ und $f(i) \neq f(j)$ und $labelset_g(i) \cap labelset_g(j) \neq \emptyset$. Das ist ein Widerspruch zur Regel. □

Beispiel 25. Sei $cp(f) = (0, 1, 1, 0)$ und $cp(g) = (0, 2, 2, 2)$. Dann kann f nicht reduzierbar auf g sein nach der Regel, da es einen Pfad $(0, 1, 0)$ für f gibt und alle Pfade für g das Muster $(0, 2, 2)$ haben.

Satz 41. *Seien $f \in PO_n$ und $g \in PO_m$ partielle Funktionen.*
Falls es für f i verschiedene Werte auf einem Pfad im Baum gibt und für g weniger als i Werte auf allen Pfaden, dann ist $f \leq g$ nicht wahr.

Beweis. Der Beweis ist ähnlich zum vorigen. □

Beispiel 26. Seien $cp(f) = (0, 1, 1, 2)$ und $cp(g) = (d, 0, 0, 1)$. Dann kann f nicht reduzierbar auf g sein nach der Regel.

Satz 42. *Seien $f \in PO_n$ und $g \in PO_m$ partielle Funktionen.*
Falls es für f i Wechsel von Werten auf einem Pfad im Baum gibt und für g weniger als i Wechsel auf allen Pfaden, dann ist $f \leq g$ nicht wahr.

Beweis. Der Beweis ist ähnlich wie der vorige. □

Beispiel 27. Seien $cp(f) = (0, 1, 1, 0)$ und $cp(g) = (0, 2, 2, d)$. Dann kann f nicht reduzierbar auf g sein wegen der Regel, da es einen Pfad $(0, 1, 0)$ gibt für f und alle Pfade für g das Muster $(0, 2, d)$ haben.

3.3.10 Reduzierbarkeit von mehrwertigen Funktionen

Zum Schluss wollen wir die Resultate auf mehrwertige Funktionen anwenden. Zum Beispiel kann $LLPO_2$ durch ein charakteristisches Muster beschrieben werden: $LLPO_2 = (\{1,2\}, 1, 2, d)$

Das bedeutet $LLPO_2(p_1,p_2) \in \{1,2\}$ für $cp(p_1,p_2) = (0,0)$, $LLPO_2(p_1,p_2) = 1$ für $cp(p_1,p_2) = (0,1)$, $LLPO_2(p_1,p_2) = 2$ für $cp(p_1,p_2) = (1,0)$ und $LLPO_2(p_1,p_2) = div$ für $cp(p_1,p_2) = (1,1)$.

In dieser Weise können alle mehrwertigen Funktionen beschrieben werden. Zum Beispiel ist die Funktion $f = (\{d,1\}, 1, 1, \{d,1\}))$ eine Funktion, die den Wert 1 berechnet, aber für $cp(p) = 00$ oder $cp(p) = 11$ eventuell kein Resultat berechnet.

Die Funktionsmenge $LLPO_3$ hat die Beschreibung:
$LLPO_3 = (\{1,2,3\}, \{1,2\}, \{1,3\}, d, \{2,3\}, d, d, d)$

In der Definition in [Wei92a] wurde $LLPO_n$ definiert als eine Menge von Funktionen. Somit kann $LLPO_2$ als $LLPO_2 = \{(1,1,2,d), (2,1,2,d)\}$ geschrieben werden. $LLPO_2$ ist die Menge von zwei Funktionen f_1 und f_2 mit $cp(f_1) = (1,1,2,d)$ und $cp(f_2) = (2,1,2,d)$. Aber $LLPO_n$ kann auch als mehrwertige Funktion für alle $n \in \mathbb{N}$ beschrieben werden. Wir benutzen diese Definition, weil sie die Formulierung der nächsten Algorithmen und der Beweise einfacher macht.

Für die Berechnung der Reduzierbarkeitsrelation von partiellen mehrwertigen Funktionen können wir dieselben Algorithmen wie für die Reduzierbarkeit auf einwertigen partiellen Funktionen verwenden. Aber da die Funktionswerte Mengen sind, haben wir die Regel zu ändern.

Für die Regel benötigen wir eine neue Definition:

Definition 38. *Definiere die Funktion fm durch*

$$fm(f)(i,x) := \bigcap_{\substack{i \preceq j,\ div \notin f(j) \\ x \in labelset_g(j)}} f(j)$$

für alle Knoten i des Baums.

Der leere Durchschnitt und die leere Vereinigung werden wie üblich definiert:

$$\bigcap_{i\in\emptyset} M_i := \mathbb{N} \text{ und } \bigcup_{i\in\emptyset} M_i := \emptyset$$

Die Regel kann wie folgt formuliert werden. Sie vergleicht nicht zwei Knoten i und j miteinander, sie führt den Test auf jedem Knoten i einzeln aus.

Regel 3 ($\leq_2$ für mehrwertige Funktionen)**.**
Falls $div \notin f(i)$, dann $(div \notin labelset_g(i)$ und $(\forall x \in labelset_g(i))fm(f)(i,x) \neq \emptyset)$

Hier ist der Algorithmus zum Testen der Reduzierbarkeit zwischen mehrwertigen Funktionen in PO_n und PO_m:

Algorithmus 5.

```
program compare
Input: n := number of arguments of f
       m := number of arguments of g
       ♯ the function values of f
       for i := 0 to 2^n − 1 begin
           f(i) := cp(f)(i)
       end
       ♯ the function values of g
       for i := 0 to 2^m − 1 begin
           g(i) := cp(g)(i)
       end
Output: true, if f ≤_2 g
        false, if f ≰_2 g
begin
   f_n := 2^n − 1
   g_n := 2^m − 1

   ♯ create the tree t
   counter:=0
   create_tree (t,0^n)

   ♯ labeling of the tree t
   red:=false
   label_tree (t)

   ♯ output the result
   output red
end


procedure create_tree (t,content)
♯ creates a node t with content and sons, in which
♯ each 0 in the content will be replaced by a 1
begin
   create a new node and let t point to it
   counter:=counter+1
   t.argument:=content
   t.counter:=counter
```

```
  for all i with content(i) = 0 do
      ♯ replace the 0 at place i with 1
      content2 := content or 0^{i-1}10^{len(content)-i}
      create_tree (s,content2)
      t.sohn_i :=s
  od
end

procedure label_tree (s)
♯ creates recursively a label for every node in the tree
begin
  for i := 0 to g_n do
      if father(s) does not exist or father(s).label ⪯ i then
        s.label:=i
        if s.counter=counter then
          if test(t)=true then red:=true
        fi
        for i:=1 to number of sons of s do
            label_tree (s.son_i)
        od
      fi
  od
end

function test(t)
♯ initialization of the sets of labels
begin
  for i:=0 to f_n do
      labelset_g(i) := ∅
  od
  ♯ computation of the sets of labels
  collect(t)
  ♯ computation of fm and testing the rule
  test:=true
  for i:=0 to f_n do
      if div ∉ f(i) then
        for every x ∈ labelset_g(i) do
            fm(i,x)=f(i)
            for j:=0 to f_n do
                if i ⪯ j and x ∈ labelset_g(j) and div ∉ f(j) then
                  fm(i,x) = fm(i,x) ∩ f(j)
                fi
            od
            if rule(i,x)=false then test:=false
        od
      fi
  od
  return test
end

procedure collect(s)
```

```
♯ computation of the set of labels
begin
  labelset_g(s.argument) := labelset_g(s.argument) ∪ g(s.label)
  for i:=1 to number of sons of s do
     collect(s.son_i)
  od
end

function rule(i,x)
♯ testing the rule
♯ if div ∉ f(i) then div ∉ labelset_g(i)
♯ and fm(i,x) ≠ ∅
begin
  rule:=true
  if div ∉ f(i) and div ∈ labelset_g(i) then
    rule:=false
  fi
  if fm(i,x) = ∅ then
    rule:=false
  fi
end
```

In den Algorithmen zum Testen der Reduzierbarkeit auf einwertigen Funktionen war der Hauptteil

$$f(i) \neq f(j) \Rightarrow labelset_g(i) \cap labelset_g(j) = \emptyset$$

Mit diesem Teil der Regel sollte garantiert werden, dass unterschiedliche Werte von f für Argumente $i \preceq j$ mit Hilfe der Funktion g berechnet werden können.

Wenn wir diese Regel auf mehrwertige Funktionen übertragen, würde es naheliegen, so zu formulieren:

$$f(i) \cap f(j) = \emptyset \Rightarrow labelset_g(i) \cap labelset_g(j) = \emptyset$$

Das würde bedeuten, wenn es keinen gemeinsamen Wert in $f(i)$ und $f(j)$ gibt, dann wird die Funktion g für die Berechnung eines Funktionswertes benötigt. Aber das ist nur möglich, wenn es keine gemeinsamen Elemente in $labelset_g(i)$ und $labelset_g(j)$ gibt.

Aber diese Abänderung reicht nicht aus. Der Hauptteil muss folgendermaßen formuliert werden:

$$fm(i,x) \neq \emptyset \text{ für alle } i \text{ und } x \in labelset_g(i)$$

Das liegt daran, dass die zuvor vorgeschlagene Regel auf zwei verschiedenen Pfaden des Baums erfüllt werden kann, aber die Resultate nicht miteinander vereinbar sind:

Beispiel 28. Sei die Funktion f definiert durch $cp(f) := (\{1,2\},1,2,\{1,2\})$. Wenn wir sie mit der Funktion g vergleichen, die definiert ist durch $cp(g) :=$

$(1, 1, 1, 1)$, bekommt der Baum als Label eine 1 in jedem Knoten:
$labelset_g(00) = labelset_g(01) = labelset_g(10) = labelset_g(11) = \{1\}$. Und es gilt $f(00) \cap f(01) \cap f(11) = \{1\}$ und $f(00) \cap f(10) \cap f(11) = \{2\}$.
Somit ist die zuerst vorgeschlagene Regel erfüllt. Aber $f(01)$ kann nicht den gleichen Wert wie $f(10)$ bekommen. Somit haben wir zu entscheiden, welches charakteristische Muster die Eingabe hat. Aber das ist nicht möglich. Die Funktion g gibt keine Information, die für die Entscheidung hilfreich sein kann.
Der abgewandelte Algorithmus würde dagegen $fm(f)(00, 1) = f(00) \cap f(01) \cap f(10) \cap f(11) = \{1, 2\} \cap \{1\} \cap \{2\} \cap \{1, 2\} = \emptyset$ berechnen.

Beispiel 29. $LLPO_3 \leq_2 LLPO_2$:
Definiere $f := LLPO_3$ und $g := LLPO_2$.
f hat das charakteristische Muster $(\{1, 2, 3\}, \{1, 2\}, \{1, 3\}, d, \{2, 3\}, d, d, d)$ und der Baum sieht folgendermaßen aus:

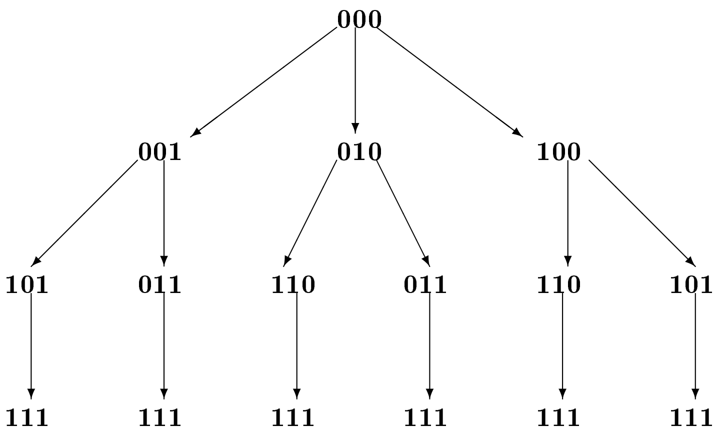

Abbildung 6: Baum für $LLPO_3$

Ein mögliches Labeling wird induziert durch $B(p_1, p_2, p_3) := (p_1, p_2)$.

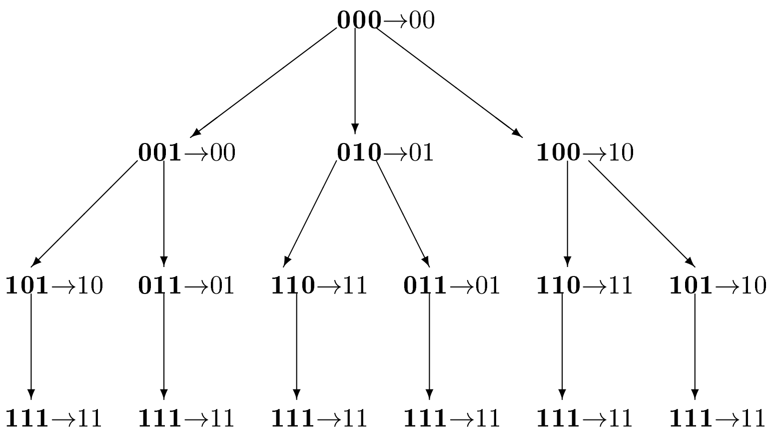

Abbildung 7: Baum für $LLPO_3$ mit Labeling

cp	labelset	f	$labelset_g$	fm(f)(cp,1)	fm(f)(cp,2)
000	{00}	{1,2,3}	{1,2}	{1}	{2}
001	{00}	{1,2}	{1,2}	{1,2}	{1,2}
010	{01}	{1,3}	{1}	{1,3}	$\mathbb{N}$
100	{10}	{2,3}	{2}	$\mathbb{N}$	{2,3}
101	{10}	d	{2}	$\mathbb{N}$	$\mathbb{N}$
011	{01}	d	{1}	$\mathbb{N}$	$\mathbb{N}$
110	{11}	d	{d}	$\mathbb{N}$	$\mathbb{N}$
111	{11}	d	{d}	$\mathbb{N}$	$\mathbb{N}$

Aus dieser Tabelle folgt $div \notin f(i) \Rightarrow div \notin labelset_g(i)$ für alle $0 \leq i \leq 7$.
Ferner gilt $fm(f)(i,1) \neq \emptyset$ und $fm(f)(i,2) \neq \emptyset$ für alle $0 \leq i \leq 7$.
Somit ist die Regel erfüllt und es folgt $LLPO_3 \leq_2 LLPO_2$.

Beispiel 30. $LLPO_2 \not\leq_2 LLPO_3$:
Definiere $f := LLPO_2$ und $g := LLPO_3$.

Der Baum sieht folgendermaßen aus:

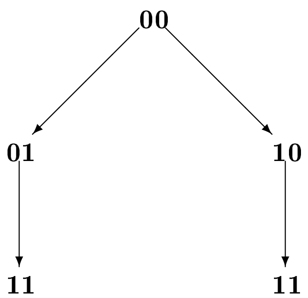

Abbildung 8: Baum für $LLPO_2$

Da die Knoten von 00, 01 und 10 Funktionswerte ungleich div haben, müssen sie als Label 000, 001, 010 oder 100 bekommen. Und es gilt $g(000) = \{1, 2, 3\}$, $g(001) = \{1, 2\}$, $g(010) = \{1, 3\}$ und $g(100) = \{2, 3\}$.
Da das Labeling die $\preceq$-Relation berücksichtigen muss, können wir den Knoten 00 nur mit 000 labeln, oder alle Knoten müssten das gleiche Label bekommen.
Dann gilt aber $labelset_g(00) = labelset_g(01) = labelset_g(10) = \{x, y\}$ mit $x, y \in \{1, 2, 3\}$ und $x \neq y$.
Dann folgt $fm(f)(00, x) = \{1, 2\} \cap \{1\} \cap \{2\} = \emptyset$. Das widerspricht der Regel.
Somit muss 00 das Label 000 bekommen. Dann muss 01 ein anderes Label bekommen, o.B.d.A. 001. Dann gilt $labelset_g(00) = \{1, 2, 3\}$ und $labelset_g(01) \subseteq \{1, 2\}$.
Weier gilt $fm(f)(00, 1) \subseteq \{1, 2\} \cap \{1\} = \{1\}$ und $fm(f)(00, 2) \subseteq \{1, 2\} \cap \{1\} = \{1\}$.
Es ist gleichgültig, welches Label 10 bekommt, es gilt
$labelset_g(10) \cap labelset_g(01) \cap labelset_g(00) \neq \emptyset$.
Somit gibt es eine Zahl $x \in labelset_g(10) \cap labelset_g(01) \cap labelset_g(00)$.
Dann gilt $fm(f)(00, x) = \{1, 2\} \cap \{1\} \cap \{2\} = \emptyset$.
Das widerspricht der Regel. Somit gilt $LLPO_2 \not\leq_2 LLPO_3$.

3.3.11 Beweis der Korrektheit des Algorithmus für mehrwertige Funktionen

We haben die Korrektheit von Algorithmus 5 zu beweisen. Wieder hat das Labeling des Baums den Sinn, die stetige Funktion B zu berechnen. Aber es gibt wieder Einschränkungen, die beachtet werden müssen.

Wir beweisen zwei Lemmas:

Lemma 7. *Wenn $f \leq_2 g$, dann findet der Algorithmus ein Labeling des Baums, das die Regel 3 erfüllt.*

Lemma 8. *Wenn der Algorithmus ein Labeling des Baums generiert, das die Regel 3 erfüllt, dann gilt $f \leq_2 g$.*

Beweis von Lemma 7.
Wie im Beweis von Lemma 3 und 5 können wir beweisen, dass es stetige und totale Funktionen $B' : \mathbb{B}^n \longrightarrow \mathbb{B}^m$ und $A' : \mathbb{B}^n \times \mathbb{N} \longrightarrow \mathbb{N}$ gibt mit den Eigenschaften:

1. $B'(p_1, \dots, p_n) = B'(q_1, \dots, q_n)$ für alle $p_1, \dots, p_n$ and $q_1, \dots, q_n$ mit $arr(p_1, \dots, p_n) = arr(q_1, \dots, q_n)$ (B' hängt nur von arr ab)
2. $A(p_1, \dots, p_n, g \circ B(p_1, \dots, p_n)) = A'(p_1, \dots, p_n, g \circ B'(p_1, \dots, p_n))$ für alle $p_1, \dots, p_n$ mit $div \notin f(p_1, \dots, p_n)$ (A' und B' zusammen verhalten sich wie A und B)
3. Für alle $p_1, \dots, p_n$ mit $div \notin f(p_1, \dots, p_n)$ gibt es $p_1^*, \dots, p_n^*$ mit $arr(p_1^*, \dots, p_n^*) = arr(p_1, \dots, p_n)$ und $cp(B(p_1^*, \dots, p_n^*)) = cp(B'(p_1^*, \dots, p_n^*))$ (es gibt eine Eingabe, für die sich B' wie B verhält)

Die Funktion B' kann für das Labeling des Baums benutzt werden.
Und die Definition von A' impliziert einige Restriktionen.
Falls $div \notin f(p_1, \dots, p_n)$, dann folgt $div \notin A(p_1, \dots, p_n, g \circ B(p_1, \dots, p_n))$.
Dann gilt $div \notin A'(p_1, \dots, p_n, g \circ B'(p_1, \dots, p_n))$ wegen der Eigenschaft 2.
Dann gilt $div \notin B'(p_1, \dots, p_n)$.
Das ist ein Teil der Regel.

Sei i ein Knoten im Baum mit $div \notin f(i)$. Sei $x \in labelset_g(i)$.
Sei $M := \{j | i \preceq j, x \in labelset_g(j), div \notin f(j)\}$. Dann gilt

$$\bigcap_{j \in M} f(j) \neq \emptyset$$

Beweis: Sei $g \circ B'(p_1, \dots, p_n) = x$ und $A'(p_1, \dots, p_n, x) = a \in f(i)$ für alle $p_1, \dots, p_n$ mit $arr(p_1, \dots, p_n) = i$.
Da A' stetig ist, gilt $A'(p_1, \dots, p_n, x) = a$ für alle $p_1, \dots, p_n$ mit $j = arr(p_1, \dots, p_n)$ und $i \sqsubseteq j$.
Aus $A'(p_1, \dots, p_n, x) = A(p_1, \dots, p_n, x)$ für alle $p_1, \dots, p_n$ und $x \in g \circ B(p_1, \dots, p_n)$ folgt $a \in f(j)$ für alle j mit $i \preceq j$ und $x \in g \circ B'(p_1, \dots, p_n)$ mit $arr(p_1, \dots, p_n) = j$.
Das impliziert

$$a \in \bigcap_{j \in M} f(j)$$

Somit ist der zweite Teil der Regel erfüllt. □

Beweis von Lemma 8.
Die Argumentation folgt den Beweisen von Lemma 4 und 6. Wenn der Algorithmus ein Labeling des Baums erzeugt, das die Regel erfüllt, dann ist es möglich, stetige Funktionen B und A zu konstruieren mit

$$f(p_1, \ldots, p_n) \in A(p_1, \ldots, p_n, g \circ B(p_1, \ldots, p_n)).$$

Der Algorithmus zum Berechnen der stetigen Funktion B ist derselbe wie in dem Beweis von Lemma 6. Er basiert auf einem Labeling, das die Regel erfüllt.

Es ist ebenso möglich, die stetige Funktion A zu konstruieren. Sie basiert ebenso auf dem Labeling, das die Regel erfüllt.

Die Definition von A im Beweis von Lemma 6 war

$$A(p_1, \ldots, p_n, k) := \begin{cases} f(p_1, \ldots, p_n) & \text{falls ein Knoten t durch} \\ & \text{Tiefensuche gefunden wird} \\ & \text{mit } cp(p_1, \ldots, p_n) \sqsubseteq t.argument \\ & \text{und } t.label = (e_1, \ldots, e_n) \text{ und} \\ & f(p_1, \ldots, p_n) \neq div \text{ und} \\ & g(q_1, \ldots, q_n) = k \\ & \text{für } cp(q_1, \ldots, q_n) = (e_1, \ldots, e_n) \\ div & \text{sonst} \end{cases}$$

Diese Definition ist nicht nützlich, da f viele verschiedene Werte haben kann. Die Definition wird deshalb durch folgende ersetzt:

$$A(i, k) := \begin{cases} \min(fm(f)(i, k)) & \text{falls } fm(f)(i, k) \neq \emptyset \\ div & \text{sonst.} \end{cases}$$

Dann gilt $A(p_1, \ldots, p_n, x) \in f(p_1, \ldots, p_n)$ für alle $x \in g \circ B(p_1, \ldots, p_n)$.
Somit gilt $f \leq_2 g$. □

3.3.12 Bemerkungen

Notwendigkeit für die Verwendung der Funktion arr

Im Baum haben verschiedene Knoten dasselbe charakteristische Muster, aber können mit verschiedenen Labeln versehen werden. Somit wurde die Funktion arr statt cp verwendet.

Das folgende Beispiel zeigt, dass das notwendig ist.

Beispiel 31. Sei
$cp(f) = (\{1,2,3\}, \{1,2\}, \{1,3\}, \{1,2,3\}, \{2,3\}, \{1,2,3\}, \{1,2,3\}, d)$
und $cp(g) = (\{1,2\}, 1, 2, d)$.
Dann gilt $f \leq_2 g$. Ein mögliches Labeling ist:

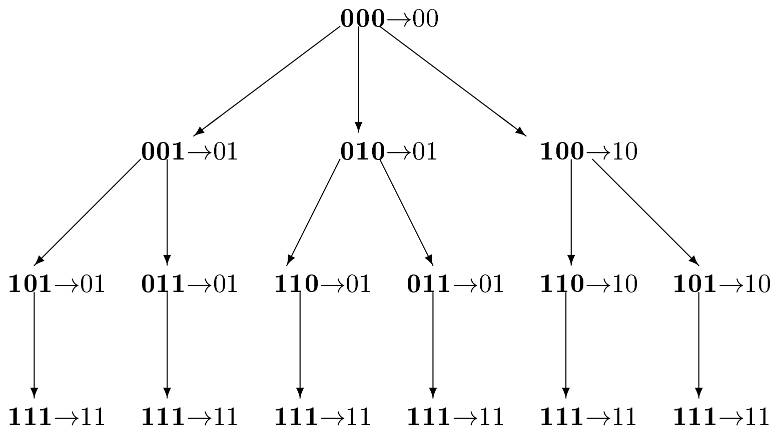

Abbildung 9: Baum für f

cp	labelset	f	$labelset_g$
000	{00}	{1,2,3}	{1,2}
001	{01}	{1,2}	{1}
010	{01}	{1,3}	{1}
100	{10}	{2,3}	{2}
101	{01,10}	{1,2,3}	{1,2}
011	{01}	{1,2,3}	{1}
110	{01,10}	{1,2,3}	{1,2}
111	{11}	{d}	{d}

Definiere A durch

$$A(p_1, p_2, p_3, k) := \begin{cases} 1 & \text{falls } k = 1 \\ 2 & \text{falls } k = 2 \\ div & \text{sonst} \end{cases}$$

und sei B die Funktion, die durch das Labeling induziert wird.
Dann gilt $f(p_1, p_2, p_3) \in A(p_1, p_2, p_3, g \circ B(p_1, p_2, p_3))$ für alle $p_1, p_2, p_3 \in \mathbb{B}$.

In diesem Beispiel wird der Knoten 110 mit den Labeln 01 und 10 versehen.

Und es ist leicht zu zeigen, dass es kein zulässiges Labeling gibt, das allen Knoten mit dem gleichen charakteristischen Muster das gleiche Label gibt.

Vergleich mit der Beschreibung von Hertling

Peter Hertling bescheibt unstetige Funktionen in [Hert96] durch Grade der Unstetigkeit.

Beispiel 32. Sei $cp(f) = (0, 1, 1, 2)$. Dann gilt
$A_0^{(1)}(f) = PO_2$,
$A_1^{(1)}(f) = \{(0^\omega, 0^\omega)\} \cup \{(0^n 10^\omega, 0^\omega)\} \cup \{(0^\omega, 0^m 10^\omega)\}$,
$A_2^{(1)}(f) = \{(0^\omega, 0^\omega)\}$,
$A_3^{(1)}(f) = \emptyset$ und
$A_0^{(2)}(f) = PO_2$,
$A_1^{(2)}(f) = \{(0^\omega, 0^\omega)\} \cup \{(0^n 10^\omega, 0^\omega)\} \cup \{(0^\omega, 0^m 10^\omega)\}$,
$A_2^{(2)}(f) = \{(0^\omega, 0^\omega)\}$,
$A_3^{(2)}(f) = \emptyset$.
Somit gilt $Lev^{(1)}(f) = Lev^{(2)}(f) = 3$.
$Lev^{(1)}(f) = Lev^{(2)}(f) = 2$ für $cp(f) = (0, 1, 1, 1)$.
$Lev^{(1)}(f) = Lev^{(2)}(f) = 1$ für $cp(f) = (0, 0, 0, 0)$.
Aber $Lev^{(1)}(f) = Lev^{(2)}(f) = 3$ für $cp(f) = (0, 1, 1, 0)$.
Somit gibt es mehr Äquivalenzklassen als Level. Der Level ist nicht geeignet, um die Äquivalenzklassen zu beschreiben.

Peter Hertling beschreibt unstetige Funktionen durch Wälder. Es ist möglich, jede Funktion in PO durch einen Wald zu beschreiben.

Beispiel 33. Sei $cp(f) = (0, 1, 1, 2)$. Dann hat der Wald für f die Form
$B(f) = \{(0), (1), (2), (0, (1, 2)), (1, (2)), (0, (1, 2, (1, (2))))\}$.
In unserer Charakterisierung wird dagegen f durch den Baum $(0, (1, (2)), (1, (2)))$ beschrieben.

Ein anderer Unterschied liegt darin, dass in der Beschreibung durch Peter Hertling die direkt verbundenen Knoten der Bäume verschiedene Werte haben. In unserem Ansatz können sie gleich sein, da die Knoten charakteristische Muster und keine Funktionswerte an Unstetigkeitspunkten repräsentieren. Ferner können in den Wäldern, wie sie von Hertling beschrieben werden, unendlich viele Bäume auftreten, falls es unendlich viele Funktionswerte gibt.

3.3.13 Weitere Sätze für mehrwertige Funktionen

Definition 39. *Seien $f \in PO_n$ und $f' \in PO_n$ mehrwertige Funktionen. Definiere $f \cup f'$ durch $(f \cup f')(p) := f(p) \cup f'(p)$ für alle $p \in \mathbb{B}^n$.*

Definition 40. *Seien $f \in PO_n$ und $f' \in PO_n$ mehrwertige Funktionen. Definiere $f \cap f'$ durch $(f \cap f')(p) := f(p) \cap f'(p)$ für alle $p \in \mathbb{B}^n$.*

Satz 43. *Seien $f \in PO_n$, $f' \in PO_n$ und $g \in PO_m$ mehrwertige Funktionen. Dann gilt $f \leq_2 g \Rightarrow f \cup f' \leq_2 g$.*

Beweis. Sei $f \leq_2 g$. Nach Regel 3 gilt $div \notin f(i) \Rightarrow div \notin labelset_g(i)$.
Das ist äquivalent zu $div \in labelset_g(i) \Rightarrow div \in f(i)$.
Aber $div \in f(i)$ impliziert $div \in f(i) \cup f'(i)$.
Somit gilt $div \in labelset_g(i) \Rightarrow div \in f(i) \cup f'(i)$.
Somit folgt, $div \notin f(i) \cup f'(i)$ impliziert $div \notin labelset_g(i)$.
Somit ist der erste Teil der Regel erfüllt für $f(i) \cup f'(i)$.

Wir haben den zweiten Fall zu verifizieren.
$f \leq_2 g$ impliziert $fm(f)(i,x) \neq \emptyset$ für alle i und $x \in labelset_g(i)$.
$fm(f)(i,x) \subseteq fm(f \cup f')(i,x)$ impliziert $fm(f \cup f')(i,x) \neq \emptyset$.
Somit ist auch der zweite Teil der Regel erfüllt.
Somit folgt $f \cup f' \leq_2 g$. □

Satz 44. *Seien $f \in PO_n$, $g \in PO_m$ und $g' \in PO_m$ mehrwertige Funktionen. Dann gilt $f \leq_2 g \cup g' \Rightarrow f \leq_2 g$.*

Beweis. Es gelte $f \leq_2 g \cup g'$. Nach Regel 3 gilt $div \notin f(i) \Rightarrow div \notin labelset_{g \cup g'}(i)$.
Das impliziert $div \notin labelset_g(i)$.
Somit ist der erste Teil der Regel erfüllt für $f(i)$ und $g(i)$.

Wir müssen noch den zweiten Teil verifizieren.
$f \leq_2 g \cup g'$ impliziert $fm(f)(i,x) \neq \emptyset$ für alle i und $x \in labelset_{g \cup g'}(i)$.
Dann gilt $fm(f)(i,x) \neq \emptyset$ für alle i und $x \in labelset_g(i)$.
Damit ist auch der zweite Teil der Regel erfüllt.
Somit gilt $f \leq_2 g$. □

Satz 45. *Seien $f \in PO_n$, $f' \in PO_n$ und $g \in PO_m$ mehrwertige Funktionen. Dann gilt $f \cap f' \leq_2 g \Rightarrow f \leq_2 g$.*

Satz 46. *Seien $f \in PO_n$, $g \in PO_m$ und $g' \in PO_m$ mehrwertige Funktionen. Dann gilt $f \leq_2 g \Rightarrow f \leq_2 g \cap g'$.*

Beweis. Die Beweise sind ähnlich. □

Satz 47. *Seien $f \in PO_n$, $g \in PO_m$ mehrwertige Funktionen. Falls es eine Zahl k gibt mit $k \in g(i)$ für alle i und $div \notin f(i)$ für alle i, dann impliziert $f \leq_2 g$*

$$(\exists m) m \in f(i) \text{ für alle } i$$

Beweis. Sei k eine Zahl mit $k \in g(j)$ für alle j. Es gelte $div \notin f(j)$ für alle j.
$f \leq_2 g$ impliziert $fm(f)(i,k) \neq \emptyset$ für alle i.

Dann gilt $fm(f)(0,k) \neq \emptyset$.
Das impliziert
$$\bigcap_j f(j) \neq \emptyset$$
Somit gibt es eine Zahl m mit $m \in f(i)$ für alle i. □

Satz 48. *Seien $f \in PO_n$, $g \in PO_m$ mehrwertige Funktionen. Es gebe ein Labeling, das die Regel erfüllt. Dann gilt*

$$f(i) \cap f(j) = \emptyset \Longrightarrow labelset_g(i) \cap labelset_g(j) = \emptyset$$

für alle Knoten i und j des Baums mit $i \preceq j$ und $div \notin f(i)$ und $div \notin f(j)$.

Beweis. Der Beweis folgt unmittelbar aus der allgemeinen Regel. □

Satz 49. *Seien $f \in PO_n$, $g \in PO_m$ mehrwertige Funktionen. Es gebe ein Labeling, das die Regel erfüllt.*
Sei M die Menge von Knoten auf einem Pfad des Baums mit $div \notin f(i)$ für alle $i \in M$.
Dann gilt
$$\bigcap_{i \in M} f(i) = \emptyset \Longrightarrow \bigcap_{i \in M} labelset_g(i) = \emptyset$$
für alle Knoten i und j des Baums mit $i \preceq j$.

Beweis. Der Beweis folgt wieder aus der Regel 3. □

3.4 Anwendung auf spezielle Reduzierbarkeitsprobleme

3.4.1 Anwendung auf LLPO

Mit den Resultaten des letzten Abschnitts können wir einige Sätze beweisen. Zunächst zwei Lemmas:

Lemma 9. *Sei M eine endliche Menge mit $|M| = n+1$ und M_i, $1 \leq i \leq n$, seien n Mengen mit $M_i \subseteq M$ und $| M_i | = n$ oder $| M_i | = n+1$ für alle $i \in \{1, \ldots, n\}$. Dann gilt*

$$\bigcap_{i=1}^{n} M_i \neq \emptyset$$

Beweis. Für jedes $i \in \{1, \ldots, n\}$ enthält M_i alle Elemente von M, bis auf eins. Somit fehlt in jeder Menge M_i höchstens ein Element von M. Da M $n+1$ Elemente hat, gibt es ein Element $x \in M$, das in allen Mengen M_i enthalten ist. Somit gilt

$$x \in \bigcap_{i=1}^{n} M_i$$

Also ist der Durchschnitt der Mengen M_i nicht leer. □

Lemma 10. *Sei $M := \{1, \ldots, n\}$ und $M_i := M \backslash \{i\}$ für $1 \leq i \leq n$. Dann gilt*

$$M \cap \bigcap_{i=1}^{n} M_i = \emptyset$$

Beweis. $i \notin M_i$ für alle $i \in \{1, \ldots, n\}$. Somit gilt

$$i \notin M \cap \bigcap_{i=1}^{n} M_i$$

Wegen

$$M \cap \bigcap_{i=1}^{n} M_i \subseteq \{1, \ldots, n\}$$

folgt

$$M \cap \bigcap_{i=1}^{n} M_i = \emptyset$$

□

Der nächste Satz wurde von Weihrauch in [Wei92a] bewiesen. Mit dem Algorithmus und Regel 3 wird der Beweis wesentlich kürzer:

Satz 50. *$LLPO_n \not\leq_2 LLPO_{n+1}$ für jedes $n \in \mathbb{N}$.*

Beweis. Definiere $f := LLPO_n$ und $g := LLPO_{n+1}$.
Definiere $k_{m,i} := 0^{i-1}10^{m-i}$ für $1 \leq i \leq m$.
Der Baum für f muss mit den charakteristischen Mustern für die Eingaben der Funktion g als Label versehen werden. Der Knoten 0^n hat n Söhne.
$f(0^n) = \{1, \ldots, n\}$ und $f(k_{n,i}) = \{1, \ldots, n\} \backslash \{i\}$ für $1 \leq i \leq n$.
Nach Lemma 10 ist der Durchschnitt dieser Mengen leer.

Falls die Wurzel nicht mit 0^{n+1} als Label versehen wird, dann müssen alle Label gleich sein. Dann gibt es ein $x \in \{1, \ldots, n\}$ im Durchschnitt aller Label. Dann ist die Regel nicht erfüllt, da $fm(f)(0^n, x) = \emptyset$.
Falls 0^n als Label 0^{n+1} bekommt, dann bekommen die n Söhne als Label auch 0^{n+1} oder $k_{n+1,i}$ für $i \in \{1, \ldots, n+1\}$.
Dann gilt $labelset_g(0^{n+1}) = \{1, \ldots, n+1\}$ und die Mengen $labelset_g(k_{n+1,i})$ sind n Teilmengen von $\{1, \ldots, n+1\}$ mit Kardinalität $n+1$ oder n.
Nach Lemma 9 ist der Durchschnitt von diesen Mengen nicht leer. Also gibt es eine Zahl x in diesem Durchschnitt. Somit gilt $fm(f)(0^n, x) = \emptyset$. Nach Regel 3 gilt $LLPO_n \not\leq_2 LLPO_{n+1}$. □

Definition 41. *Für $n \geq 2$ sei $LLPO_{n\vee}$ definiert durch*

$$Def(LLPO_{n\vee}) = \{(p_1, \ldots, p_n) \mid \text{ es gibt ein } k \text{ mit } p_k = 0^\omega\}$$
$$\text{und } LLPO_{n\vee}(p_1, \ldots, p_n) = \{\langle i, j\rangle | i < j \text{ und } (p_i = 0^\omega \text{ oder } p_j = 0^\omega)\}$$

Definition 42. *Für $n \geq 2$ sei $LLPO_{n\wedge}$ definiert durch*

$$Def(LLPO_{n\vee}) = \{(p_1, \ldots, p_n) \mid \text{ es gibt Zahlen } k \neq l \text{ mit } p_k = p_l = 0^\omega\}$$
$$\text{und } LLPO_{n\vee}(p_1, \ldots, p_n) = \{\langle i, j\rangle | i < j \text{ und } p_i = 0^\omega \text{ und } p_j = 0^\omega\}$$

Satz 51. *$LLPO_{n\vee} \leq_2 LLPO_{n\wedge}$ für alle $n \in \mathbb{N}$, $n \geq 2$.*

Beweis. Definiere $f := LLPO_{n\vee}$ und $g := LLPO_{n\wedge}$.
Es gibt stetige Funktionen A und B mit

$$A(p_1, \ldots, p_n, g \circ B(p_1, \ldots, p_n)) \subseteq f(p_1, \ldots, p_n)$$

für alle $(p_1, \ldots, p_n) \in Def(f)$.
Die stetige Funktion B kann mit dem folgenden Algorithmus berechnet werden. Der Algorithmus berechnet im Prinzip die Identität auf $\mathbb{B}^n$. Aber falls es nur eine Folge $p_i = 0^\omega$ gibt, erzeugt er zwei Nullfolgen in der Ausgabe. Er setzt die erste Folge q_1 auf 0^ω. Falls die Folge p_1 die einzige Nullfolge zu sein scheint, setzt er die zweite Folge q_2 auf 0^ω.
Das garantiert, dass die Ausgabe in $Def(g)$ liegt, falls es nur eine Nullfolge in $p_1, \ldots, p_n$ gibt.
$A :\subseteq \mathbb{B}^n \times \mathbb{N} \longrightarrow \mathbb{N}$ ist definiert durch $A(p_1, \ldots, p_n, x) := x$ für alle $p_1, \ldots, p_n \in \mathbb{B}$ und $x \in \mathbb{N}$. □

Algorithmus 6.

$x(0) := (p_1(0), \ldots, p_n(0))$
$s := x$
$y := \sharp_1(s)$
$k := 0$
if $y = n$ then
    $\sharp$ there is no null-sequence in $(p_1, \ldots, p_n)$
    $q(0) := (1, \ldots, 1)$
else if $y < n - 1$ then
    $\sharp$ there are possibly more than two null-sequences in $(p_1, \ldots, p_n)$
    $q(0) := x(0)$
else if $y = n - 1$ then

```
    ♯ there is at most one null-sequence in (p_1, ..., p_n)
    if s(1) = 1 then
        if q(0) := (0, s(2), ..., s(n))
    else
        q(0) := (s(1), 0, s(3), ..., s(n))
    endif
endif
result:=q(0)
output q(0)
loop
    k := k + 1
    x(k) := (p_1(k), ..., p_n(k))
    s := max(s, x(k))
    y := ♯_1(s)
    if y = n then
        ♯ there is no null-sequence in (p_1, ..., p_n)
        q(k) := (1, ..., 1) − result
    else if y < n − 1 then
        ♯ there are possibly more than two null-sequences in (p_1, ..., p_n)
        q(k) := x(k) − result
    else if y = n − 1 then
        ♯ there is at most one null-sequence in (p_1, ..., p_n)
        if s(1) = 1 then
            q(k) := (0, s(2), ..., s(n)) − result
        else
            q(k) := (s(1), 0, s(3), ..., s(n)) − q(k − 1)
        endif
    endif
    result:=max(result,q(k))
    output q(k)
end loop
```

Satz 52. *$LLPO_{2\wedge} \leq_2 LLPO_{2\vee}$.*

Beweis. Definiere $f := LLPO_{2\wedge}$ und $g := LLPO_{2\vee}$.
Definiere B durch $B(p_1, p_2) := (p_1, p_2)$ und $A(p_1, p_2, x) := x$.
Dann gilt $A(0^\omega, 0^\omega, g \circ B(0^\omega, 0^\omega)) = A(0^\omega, 0^\omega, g(0^\omega, 0^\omega)) = A(0^\omega, 0^\omega, \langle 1, 2\rangle) = \langle 1, 2\rangle$
$= f(0^\omega, 0^\omega)$ und $f(p_1, p_2) = div$ für $(p_1, p_2) \neq (0^\omega, 0^\omega)$. □

Satz 53. *$LLPO_{n\wedge} \not\leq_2 LLPO_{n\vee}$ für alle $n \in \mathbb{N}$, $n \geq 3$.*

Beweis. Definiere $f := LLPO_{n\wedge}$ und $g := LLPO_{n\vee}$ für $n \geq 3$.
Nehme $f \leq_2 g$ an. Dann gibt es stetige Funktionen A und B mit

$$A(p_1, \ldots, p_n, g \circ B(p_1, \ldots, p_n)) \subseteq f(p_1, \ldots, p_n)$$

für alle $(p_1, \ldots, p_n) \in Def(f)$.
Dann gilt $f(0, 0, 1, 1^{n-3}) = \{\langle 1, 2\rangle\}$, $f(1, 0, 0, 1^{n-3}) = \{\langle 2, 3\rangle\}$,
$f(0, 1, 0, 1^{n-3}) = \{\langle 1, 3\rangle\}$ und

$f(0,0,0,1^{n-3}) = \{\langle 1,2\rangle, \langle 1,3\rangle, \langle 2,3\rangle\}$.
Die möglichen Resultate für $g \circ B(p_1,\ldots,p_n)$ sind

$$M_i := \{\langle 1,i\rangle,\ldots,\langle i-1,i\rangle,\langle i,i+1\rangle,\ldots,\langle i,n\rangle\}$$

und Vereinigungsmengen der M_i.
Sei $a := labelset_g(0,0,1,1,\ldots,1)$,
$b := labelset_g(1,0,0,1,1,\ldots,1)$ und
$c := labelset_g(0,0,0,1,1,\ldots,1)$.
Es gibt eine Zahl i mit $M_i \subseteq a$ und eine Zahl j mit $M_j \subseteq b$.
Falls $i = j$, dann gilt $M_i = M_j \subseteq c$.
Falls $i < j$, dann gilt $\langle i,j\rangle \in M_i \cap M_j \cap c \subseteq a \cap b \cap c$.
Falls $j < i$, dann gilt $\langle j,i\rangle \in M_i \cap M_j \cap c \subseteq a \cap b \cap c$.
In den ersten beiden Fällen gilt
$fm(f)((0,0,0,1,1,\ldots,1),\langle i,j\rangle)$
$\subseteq \{\langle 1,2\rangle,\langle 1,3\rangle,\langle 2,3\rangle\} \cap \{\langle 1,2\rangle\} \cap \{\langle 2,3\rangle\} = \emptyset$.
Im letzten Fall gilt
$fm(f)((0,0,0,1,1,\ldots,1),\langle j,i\rangle)$
$\subseteq \{\langle 1,2\rangle,\langle 1,3\rangle,\langle 2,3\rangle\} \cap \{\langle 1,2\rangle\} \cap \{\langle 2,3\rangle\} = \emptyset$.

Somit folgt $LLPO_{n\wedge} \not\leq_2 LLPO_{n\vee}$ nach der Regel 3. $\square$

3.4.2 *LLPO*-stetige reelle Funktionen

Definition 43. *Für alle n seien* MAX_n *bzw.* MIN_n *die Menge aller Funktionen* $f :\subseteq \mathbb{B}^n \longrightarrow \mathbb{N}$ *mit*

$$\begin{aligned} &Def(f) = Def(\rho^n) \\ &und\ f(\langle p_1, p_2, \ldots, p_n\rangle) = k \end{aligned}$$

für ein k mit $\rho(p_k) = \max(\rho(p_1), \rho(p_2), \ldots, \rho(p_n))$
bzw. $\rho(p_k) = \min(\rho(p_1), \rho(p_2), \ldots, \rho(p_n))$.

Satz 54. $MAX_n \leq LLPO_{n,n-1}$

Beweis. Definiere A durch $A(p_1, \ldots, p_n, w) := w$ und B durch
$B(p_1, \ldots, p_n) := (q_1, \ldots, q_n)$ mit

$$q_i(j) := \begin{cases} 0 & \text{falls } \nu_Q(p_i(j)) \geq \nu_Q(p_k(j)) - 2^{-j} \text{ für alle } 1 \leq k \leq n,\ k \neq i \\ & \text{oder } q_i(l) = 1 \text{ für ein } l < j \\ 1 & \text{sonst} \end{cases}$$

Dann gilt für $f \in LLPO_{n,n-1}$ $A(p_1, \ldots, p_n, f \circ B(p_1, \ldots, p_n))$
$= f \circ B(p_1, \ldots, p_n)$
$= f(q_1, \ldots, q_n) = k$
für ein k mit $\rho(p_k) = \max(\rho(p_1), \rho(p_2), \ldots, \rho(p_n))$. □

Satz 55. $LLPO_{n,n-1} \leq MAX_n$

Beweis. Definiere A durch $A(p_1, \ldots, p_n, w) := w$ und B durch
$B(p_1, \ldots, p_n) := (q_1, \ldots, q_n)$ mit $q_i(0) := \langle 0, 1, 0\rangle$ ($\nu_Q\langle 0, 1, 0\rangle = -1$) und

$$q_i(j) := \begin{cases} \langle 0, 1, 2^j - 1\rangle & \text{falls } p_i(k) = 0 \text{ für alle } k \leq j \\ q_i(j-1) & \text{sonst} \end{cases}$$

für $j > 0$.
Dann gilt für $f \in MAX_n$ $A(p_1, \ldots, p_n, f \circ B(p_1, \ldots, p_n, w))$
$= f \circ B(p_1, \ldots, p_n)$
$= f(q_1, \ldots, q_n)$
$= k$ für ein k mit $p_k = 0^\omega$. □

Resultat:

Satz 56. $MAX_n \equiv LLPO_{n,n-1}$

In ähnlicher Weise erhalten wir:

Satz 57. $MIN_n \equiv LLPO_{n,n-1}$

Schlussfolgerung:

Satz 58. $MIN_m \leq MIN_n$, *gdw.* $m \leq n$.

Beweis. Dies folgt aus $MIN_m \equiv LLPO_{m,m-1}$ und $MIN_n \equiv LLPO_{n,n-1}$ und $LLPO_{m,m-1} \leq LLPO_{n,n-1} \Longleftrightarrow m \leq n$. □

Weiter folgt:

Satz 59. $MAX_m \leq MAX_n$, *gdw.* $m \leq n$. $MAX_m \leq MIN_n$, *gdw.* $m \leq n$. $MIN_m \leq MAX_n$, *gdw.* $m \leq n$.

3.5 Andere Reduzierbarkeitsrelationen

Die Reduzierbarkeitsrelation $\leq_2$ wurde definiert durch
$f \leq_2 g$, gdw. es stetige Funktionen A und B gibt mit

$$f(p_1, \ldots, p_n) = A(p_1, \ldots, p_n, g \circ B(p_1, \ldots, p_n))$$

$$\text{oder } A(p_1, \ldots, p_n, g \circ B(p_1, \ldots, p_n)) \subseteq f(p_1, \ldots, p_n)$$

für alle $p_1, \ldots, p_n \in Def(f)$.

Es gibt schwächere Typen von Reduzierbarkeitsrelationen $\leq_0$ und $\leq_1$, für die man die Reduzierbarkeitseigenschaft und die Klassen auch berechnen kann. Diese Definitionen wurden in Abschnitt 3.1.2 gegeben.

In allen diesen Definitionen kann der Funktionswert $f(p_1, \ldots, p_n)$ berechnet werden mit Hilfe des Wertes von $g \circ B(p_1, \ldots, p_n)$ für alle $p_1, \ldots, p_n \in Def(f)$. Für alle $p_1, \ldots, p_n \notin Def(f)$ kann irgendein Wert oder *div* berechnet werden.

Auch eine strengere Version von Reduzierbarkeit kann definiert werden:

Definition 44 (Reduzierbarkeit $\leq_0^s$)**.**
Die Relation $\leq_0^s$ *ist definiert durch*

$$\begin{aligned} f \leq_0^s g, \textit{ gdw. es gibt eine stetige Funktion } B \textit{ mit} \\ & f(p_1, \ldots, p_n) = g \circ B(p_1, \ldots, p_n) \\ & \textit{für alle } p_1, \ldots, p_n \in Def(f) \textit{ und} \\ & g \circ B(p_1, \ldots, p_n) = div \\ & \textit{für alle } p_1, \ldots, p_n \notin Def(f). \end{aligned}$$

Definition 45 (Reduzierbarkeit $\leq_1^s$)**.**
Die Relation $\leq_1^s$ *ist definiert durch*

$$\begin{aligned} f \leq_1^s g, \textit{ gdw. es gibt stetige Funktionen } A \textit{ und } B \textit{ mit} \\ & f(p_1, \ldots, p_n) = A \circ g \circ B(p_1, \ldots, p_n) \\ & \textit{für alle } p_1, \ldots, p_n \in Def(f) \textit{ und} \\ & A \circ g \circ B(p_1, \ldots, p_n) = div \\ & \textit{für alle } p_1, \ldots, p_n \notin Def(f). \end{aligned}$$

Definition 46 (Reduzierbarkeit $\leq_2^s$)**.**
Die Relation $\leq_2^s$ *ist definiert durch*

$$\begin{aligned} f \leq_2^s g, \textit{ gdw. es gibt stetige Funktionen } A \textit{ und } B \textit{ mit} \\ & f(p_1, \ldots, p_n) = A(p_1, \ldots, p_n, g \circ B(p_1, \ldots, p_n) \\ & \textit{für alle } p_1, \ldots, p_n \in Def(f) \textit{ und} \\ & A(p_1, \ldots, p_n, g \circ B(p_1, \ldots, p_n)) = div \\ & \textit{für alle } p_1, \ldots, p_n \notin Def(f). \end{aligned}$$

3.5.1 Reduzierbarkeit von Funktionen für $\leq_0$ und $\leq_1$

Für $\leq_0$ gibt es unendlich viele Äquivalenzklassen, da Resultate für g nicht umgerechnet werden können. Z.B. sei $cp(f) = (0,1,2,3)$ und $cp(g) = (3,4,5,6)$. Beide Funktionen sind nicht reduzierbar aufeinander. Aber der Algorithmus für $\leq_2$ kann im Prinzip benutzt werden. Z.B. ist f reduzierbar auf g für $cp(f) = (0,1,2,3)$ und $cp(g) = (0,2,1,3)$.

Für $\leq_1$ gibt es nur endlich viele Äquivalenzklassen, aber mehr als für $\leq_2$. Z.B. ist für $cp(f) = (0,1,2,3)$ und $cp(g) = (0,1,1,3)$ f nicht reduzierbar auf g, da 01 und 10 den selben Funktionswert haben und A keinen Zugriff auf (p_1, p_2) hat, um zu testen, welches Eingabemuster vorliegt.

Die Algorithmen zum Testen der Reduzierbarkeit für $\leq_0$ sind dieselben wie die Algorithmen für $\leq_2$, aber es gibt nicht die Notwendigkeit, zwei Knoten im Baum zu vergleichen. In den Regeln der Algorithmen werden nur die einzelnen Knoten des Baums isoliert getestet.

Regel 4 ($\leq_0$ für totale Funktionen)**.**
$labelset_g(i) = \{f(i)\}$ für alle Knoten i des Baums.

Es gibt auch für $\leq_0$ Äquivalenzklassen. Zum Beispiel sind für $cp(f) = (0,1,1,1)$ und $cp(g) = (0,0,0,1)$ f und g äquivalent.
Es gibt eine stetige Funktion B mit $cp(B(p_1,p_2)) = 00$ für $cp(p_1,p_2) = 00$ und $cp(B(p_1,p_2)) = 11$ für $cp(p_1,p_2) \neq 00$, dann gilt $f(p) = g \circ B(p)$.
Und es gibt eine stetige Funktion B mit $cp(B(p_1,p_2)) = 00$ für $cp(p_1,p_2) \neq 11$ und $cp(B(p_1,p_2)) = 11$ für $cp(p_1,p_2) = 11$, dann gilt $g(p) = f \circ B(p)$.

Regel 5 ($\leq_0$ für partielle Funktionen)**.**
Wenn $f(i) \neq div$, dann $labelset_g(i) = \{f(i)\}$.

Regel 6 ($\leq_0$ für mehrwertige Funktionen)**.**
Wenn $div \notin f(i)$, dann $labelset_g(i) \subseteq f(i)$.

Die Algorithmen zum Testen der Reduzierbarkeit für $\leq_1$ sind den Algorithmen für $\leq_2$ ähnlicher. Hier müssen alle Paare von Knoten getestet werden. Die Tests werden nicht auf Knoten i und j mit $i \preceq j$ beschränkt. Das ist der Hauptunterschied.

Regel 7 ($\leq_1$ für totale Funktionen)**.**
Wenn $f(i) \neq f(j)$, dann $labelset_g(i) \cap labelset_g(j) = \emptyset$.

Regel 8 ($\leq_1$ für partielle Funktionen)**.**
(Falls $f(i) \neq div$, dann $div \notin labelset_g(i)$)
und (falls $f(j) \neq div$, dann $div \notin labelset_g(j)$)
und (falls ($f(i) \neq div$ und $f(j) \neq div$ und $f(i) \neq f(j)$),
dann $labelset_g(i) \cap labelset_g(j) = \emptyset$)

Regel 9 ($\leq_1$ für mehrwertige Funktionen)**.**
$(\forall i) div \notin f(i) \Longrightarrow div \notin labelset_g(i)$ und

$$(\forall x) \bigcap_{x \in labelset_g(j)} f(j) \neq \emptyset$$

Die Korrektheit der Regel 6 und 9 wird in den nächsten Abschnitten bewiesen.

Beispiel 34 ($\leq_1$)**.**
Für $\leq_1$ gibt es 6 Äquivalenzklassen von 2-stelligen totalen Funktionen:

Klasse	Typen				
1	0123				
2	0120				
3	0023	0103	0113	0121	0122
4	0020	0100	0110		
5	0003	0022	0101	0111	
6	0000				

Die Reduzierbarkeitsrelation zwischen den Funktionstypen sieht folgendermaßen aus:

$$\begin{tikzcd}
& \mathbf{1} & \\
\mathbf{2} \arrow[ur] & & \mathbf{3} \arrow[ul] \\
& \mathbf{4} \arrow[ul] \arrow[ur] & \\
& \mathbf{5} \arrow[u] & \\
& \mathbf{6} \arrow[u] &
\end{tikzcd}$$

Abbildung 10: Reduzierbarkeit bezüglich $\leq_1$

Für $\leq_0$ ist die Anzahl von Äquivalenzklassen unendlich. Aber wenn wir nur Funktionen mit denselben Funktionswerten vergleichen, gibt es endlich viele Klassen für n-stellige Functionen in PO_n. Eine obere Grenze für die Anzahl von Äquivalenzklassen ist wieder die Anzahl der Funktionstypen. Aber für $\leq_0$ muss der Funktionstyp anders definiert werden. Zum Beispiel gilt $f \not\leq_0 g$ und $g \not\leq_0 f$ für $cp(f) = (1,0,0,0)$ und $cp(g) = (0,1,1,1)$. Aber beide Funktionen sind vom selben Typ nach Definition 30.

3.5.2 Beweis der Korrektheit des Algorithmus für $\leq_0$-Reduzierbarkeit auf mehrwertigen Funktionen

Wir haben die Korrektheit des Algorithmus mit Regel 6 zu beweisen. Wieder hat das Labeling des Baums den Zweck, die stetige Funktion B zu berechnen.

Wir beweisen zwei Lemmas:

Lemma 11. *Wenn $f \leq_0 g$, dann findet der Algorithmus ein Labeling des Baums, das die Regel 6 erfüllt.*

Lemma 12. *Wenn der Algorithmus ein Labeling des Baums findet, das die Regel 6 erfüllt, dann gilt $f \leq_0 g$.*

Beweis von Lemma 11.
Sei $g \circ B(p_1, \ldots, p_n) \subseteq f(p_1, \ldots, p_n)$ für alle $p_1, \ldots, p_n$ mit $div \notin f(p_1, \ldots, p_n)$. Wie in dem Beweis von Lemma 3 und 5 können wir beweisen, dass es eine stetige und totale Funktion $B' : \mathbb{B}^n \longrightarrow \mathbb{B}^m$ gibt mit den Eigenschaften:

1. $B'(p_1, \ldots, p_n) = B'(q_1, \ldots, q_n)$ für alle $p_1, \ldots, p_n$ und $q_1, \ldots, q_n$ mit $arr(p_1, \ldots, p_n) = arr(q_1, \ldots, q_n)$ (B' hängt nur von arr ab)
2. $g \circ B(p_1, \ldots, p_n) = g \circ B'(p_1, \ldots, p_n)$
für alle $p_1, \ldots, p_n$ mit $div \notin f(p_1, \ldots, p_n)$ ($g \circ B'$ verhält sich wie $g \circ B$)
3. Für alle $p_1, \ldots, p_n$ mit $div \notin f(p_1, \ldots, p_n)$ gibt es $p_1^*, \ldots, p_n^*$ mit
$arr(p_1^*, \ldots, p_n^*) = arr(p_1, \ldots, p_n)$ und
$cp(B(p_1^*, \ldots, p_n^*)) = cp(B'(p_1^*, \ldots, p_n^*))$ (es gibt eine Eingabe, für die sich B' wie B verhält)

Die Funktion B' kann für das Labeling des Baums benutzt werden.
Falls $div \notin f(p_1, \ldots, p_n)$, dann gilt $g \circ B'(p_1, \ldots, p_n) \subseteq f(p_1, \ldots, p_n)$ wegen Eigenschaft 2.
Das impliziert $labelset_g(i) \subseteq f(i)$ für jeden Knoten. □

Beweis von Lemma 12.
Die Argumentation folgt den Beweisen von Lemma 4 und 6. Wenn der Algorithmus ein Labeling des Baums erzeugt, das die Regel erfüllt, dann ist es möglich, eine stetige Funktion B zu konstruieren mit

$$g \circ B(p_1, \ldots, p_n)) \subseteq f(p_1, \ldots, p_n).$$

Der Algorithmus für die Berechnung der stetigen Funktion B ist derselbe wie im Beweis von Lemma 6. Er basiert auf einem Labeling, das die Regel erfüllt. □

Lemma 11 und lemma 12 zusammen ergeben den Beweis für die Korrektheit der Regel 6.

3.5.3 Beweis der Korrektheit des Algorithmus für die $\leq_1$-Reduzierbarkeit auf mehrwertigen Funktionen

Wir haben die Korrektheit des Algorithmus mit der Regel 9 zu beweisen. Wieder hat das Labeling des Baums den Zweck, die stetige Funktion B zu berechnen.

Wir beweisen wieder zwei Lemmas:

Lemma 13. *Wenn $f \leq_1 g$, dann findet der Algorithmus ein Labeling des Baums, das die Regel 9 erfüllt.*

Lemma 14. *Wenn der Algorithmus ein Labeling des Baums findet, das die Regel 9 erfüllt, dann gilt $f \leq_1 g$.*

Beweis von Lemma 13.
Wie in dem Beweis von Lemma 3 und 5 können wir beweisen, dass es stetige und totale Funktionen $B' : \mathbb{B}^n \longrightarrow \mathbb{B}^m$ und $A' : \mathbb{B}^n \times \mathbb{N} \longrightarrow \mathbb{N}$ gibt mit den Eigenschaften:

1. $B'(p_1, \ldots, p_n) = B'(q_1, \ldots, q_n)$ für alle $p_1, \ldots, p_n$ und $q_1, \ldots, q_n$ mit $arr(p_1, \ldots, p_n) = arr(q_1, \ldots, q_n)$ (B' hängt nur von arr ab)
2. $A \circ g \circ B(p_1, \ldots, p_n) = A' \circ g \circ B'(p_1, \ldots, p_n)$ für alle $p_1, \ldots, p_n$ mit $div \notin f(p_1, \ldots, p_n)$ (A' und B' zusammen verhalten sich wie A und B)
3. Für alle $p_1, \ldots, p_n$ mit $div \notin f(p_1, \ldots, p_n)$ gibt es $p_1^*, \ldots, p_n^*$ mit $arr(p_1^*, \ldots, p_n^*) = arr(p_1, \ldots, p_n)$ und $cp(B(p_1^*, \ldots, p_n^*)) = cp(B'(p_1^*, \ldots, p_n^*))$ (es gibt eine Eingabe, für die sich B' wie B verhält)

Die Funktion B' kann für das Labeling des Baums benutzt werden.
Und die Definition von A' impliziert einige Restriktionen.
Falls $div \notin f(p_1, \ldots, p_n)$, dann gilt $div \notin A \circ g \circ B(p_1, \ldots, p_n)$.
Dann gilt $div \notin A' \circ B'(p_1, \ldots, p_n)$ wegen Eigenschaft 2.
Dann gilt $div \notin B'(p_1, \ldots, p_n)$.
Das ist ein Teil der Regel des Algorithmus.

Sei i ein Knoten im Baum. Sei $x \in labelset_g(i)$.
Sei $M := \{j | x \in labelset_g(j), div \notin f(j)\}$. Dann gilt

$$\bigcap_{j \in M} f(j) \neq \emptyset$$

Beweis: Sei $g \circ B'(p_1, \ldots, p_n) = x$ und $A'(x) = a \in f(i)$ für alle $p_1, \ldots, p_n$ mit $arr(p_1, \ldots, p_n) = i$.
Aus $A'(x) = A(x)$ für alle $p_1, \ldots, p_n$ und $x \in g \circ B(p_1, \ldots, p_n)$ folgt $a \in f(j)$ für alle j mit $x \in g \circ B'(p_1, \ldots, p_n)$ mit $arr(p_1, \ldots, p_n) = j$.
Das impliziert

$$a \in \bigcap_{j \in M} f(j)$$

Somit ist der zweite Teil der Regel erfüllt. □

Beweis von Lemma 14.
Die Argumentation folgt den Beweisen von Lemma 4 und 6. Wenn der Algorithmus ein Labeling des Baums erzeugt, das die Regel erfüllt, dann ist es möglich, stetige Funktionen B und A zu konstruieren mit

$$f(p_1, \dots, p_n) \in A \circ g \circ B(p_1, \dots, p_n).$$

Der Algorithmus zur Berechnung der stetigen Funktion B ist derselbe wie im Beweis von Lemma 6. Er basiert auf einem Labeling, das die Regel erfüllt.

Es ist ebenso möglich, die stetige Funktion A zu berechnen. Sie hängt von einem Labeling ab, das die Regel erfüllt.

Definiere A durch

$$A(k) := \begin{cases} \min H & \text{falls } H := \{y | (\exists i) y = g \circ B'(i)\} \neq \emptyset \\ div & \text{sonst.} \end{cases}$$

Dann gilt $A(x) \in f(p_1, \dots, p_n)$ für alle $x \in g \circ B(p_1, \dots, p_n)$.
Somit gilt $f \leq_2 g$. $\square$

Lemma 13 und Lemma 14 zusammen ergeben den Beweis für die Korrektheit der Regel 9.

3.5.4 Strenge Reduzierbarkeit zwischen partiellen Funktionen

Nun soll die strenge Reduzierbarkeit (siehe Def. 46) zwischen partiellen Funktionen betrachtet werden.

Es gibt 2 Äquivalenzklassen für 0-stellige Funktionen für strenge Reduzierbarkeit:

Klasse	Typ
1	0
2	-

Diese Klassen sind nach dem Grad der Unstetigkeit geordnet.

Partielle 1-stellige Funktionen:
Es gibt 4 Äquivalenzklassen für strenge Reduzierbarkeit:

Klasse	Typ	
1	01	
2	0-	
3	00	-1
4	--	

Die Klassen sind wieder nach dem Grad der Unstetigkeit geordnet.

Partielle 2-stellige Funktionen:

Es gibt 8 Äquivalenzklassen für strenge Reduzierbarkeit:

Klasse	Typ								
1	0123	0113	0122	0023	0103	0121	01-3	0-23	
2	0100	0110	0020	0120	0-20	01-0			
3	01--	011-	002-	010-	012-	0-2-			
4	0101	0022	0111	0003	-122	--23	-113	-121	-123
	0--3	0-22	00-3	0-03	01-1	-1-3			
5	00-0	0-00	0--0						
6	-12-	-1--	--2-	-11-	0---	000-	0-0-	00--	
7	0000	---3	--22	-111	-1-1				
8	----								

Die Reduzierbarkeitsrelation sieht folgendermaßen aus:

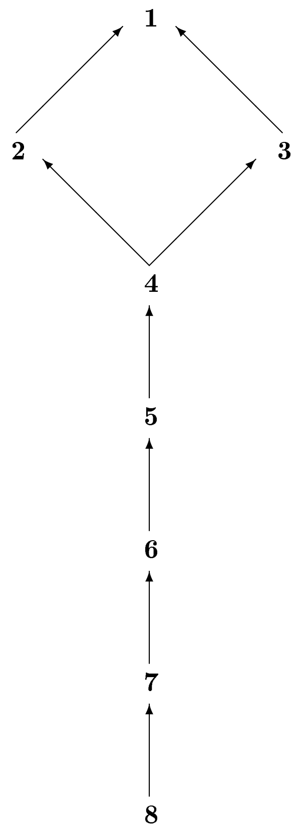

Abbildung 11: Strenge Reduzierbarkeit auf partiellen Funktionen

Der Algorithmus zum Testen der strengen Reduzierbarkeit für partielle Funktionen ist der gleiche, nur die Regel muss abgeändert werden:
Ein Labeling muss gefunden werden, das die Regel für alle Knoten i und j erfüllt.
Das muss durch den Algorithmus getestet werden. Die Regel ist die folgende:

Regel 10 (Strenge Reduzierbarkeit $\leq_2^s$)**.**
$f(i) \neq div \Longrightarrow div \notin labelset_g(i)$ und
$f(j) \neq div \Longrightarrow div \notin labelset_g(j)$ und
$(i \preceq j$ und $f(j) \neq div$ und $f(i) \neq f(j)) \Longrightarrow (labelset_g(i) \cap labelset_g(j) = \emptyset)$

Der Hauptunterschied zu der Regel für $\leq_2$ für partielle Funktionen ist, dass in der letzten Implikation nur „$f(j) \neq div$" in der Bedingung enthalten ist statt „$f(i) \neq div$ und $f(j) \neq div$".
Der Beweis wird hier ausgelassen, aber liegt nach den Beweisen der anderen

verwandten Sätze auf der Hand. Die Bedingung „$f(i) \neq div$ und $f(j) \neq div$" in Regel 2 bedeutet, dass die ganze Bedingung nicht überprüft werden muss, falls $f(i) = div$. Dann spielt es keine Rolle, welchen Wert $f(i)$ durch den Algorithmus gegeben wird.
In der Regel 10 ist der Teil „$f(i) \neq div$" ausgelassen. Somit muss die ganze Bedingung getestet werden, auch wenn $f(i) = div$.
Falls $f(i) = div$ und $i \preceq j$ und $f(j) \neq div$, dann ist $f(j) \neq div$ erfüllt. Die Bedingung fordert dann, dass $labelset_g(i) \cap labelset_g(j) = \emptyset$. Und das ist notwendig. Nur wenn $labelset_g(i) \cap labelset_g(j) = \emptyset$, kann der Algorithmus entscheiden, ob $f(i)$ ein Funktionswert gegeben werden muss oder nicht (div).
In Regel 2 war das nicht notwendig, da im Falle von $f(i) = div$ trotzdem ein Funktionswert zugeordnet werden darf.

3.6 Reduzierbarkeit auf Mengen von Funktionen

In [Wei92a] hat Klaus Weihrauch die $LLPO$-Hierarchie definiert, indem er Funktionsmengen verwendete. Wir haben dagegen mehrwertige Funktionen benutzt. Bei Weihrauch wurde $LLPO_2$ beschrieben durch

$$LLPO_2 := \{f_1, f_2\}$$

mit $cp(f_1) := (1, 1, 2, d)$ und $cp(f_2) := (2, 1, 2, d)$.
Wir beschreiben $LLPO_2$ als mehrwertige Funktion:

$$cp(LLPO_2) := (\{1, 2\}, \{1\}, \{2\}, d).$$

Es gibt Beziehungen zwischen beiden Beschreibungen. Jede mehrwertige Funktion kann als eine Menge von Funktionen mit unterschiedlichen Funktionswerten dargestellt werden. Aber eine Menge von Funktionen kann nicht immer als mehrwertige Funktion beschrieben werden.

Die mehrwertige Funktion f mit $cp(f) = (M_0, M_1, \ldots, M_{n-1})$ kann durch die folgende Menge von Funktionen ausgedrückt werden:

$$M = \{g | cp(g) = (x_0, \ldots, x_{n-1}), x_i \in M_i, 0 \leq i \leq n-1\}.$$

Aber die Funktionsmenge $M' := \{f_1, f_2\}$ mit $cp(f_1) := (0, 1, 1, 1)$ und $cp(f_2) := (1, 0, 0, 0)$ kann nicht durch eine mehrwertige Funktion ausgedrückt werden.
Falls wir die obige Funktionsmenge durch $cp(f') := (\{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\})$ und g durch $cp(g) := (0, 0, 0, 0)$ repräsentieren, dann gilt $f' \leq_2 g$ und $\{f_1, f_2\} \not\leq_2 g$.

Es ist wahrscheinlich möglich, auch die Reduzierbarkeit auf Funktionsmengen zu berechnen. Für $LLPO$ und verwandte Probleme haben wir den Weg über mehrwertige Funktion eingeschlagen, da er einfacher zu sein scheint.

4 Vollständigkeit in der C-Hierarchie

4.1 Die C-Hierarchie

In [St89] und in [Myl92] wurden unstetige Funktionen höheren Grades untersucht.

Die Funktionen wurden mit den Funktionen Ω und C verglichen.

$\Omega : \mathbb{B} \longrightarrow \mathbb{N}$ wurde schon in Definition 3 definiert durch

$$\Omega(p) := \begin{cases} 0 & \text{falls } (\forall n)p(n) = 0 \\ 1 & \text{sonst.} \end{cases}$$

Somit sagt uns Ω, ob es eine Zahl ungleich Null in einer Folge gibt. Das ist äquivalent zu der Funktion, die für eine beliebige Zahl feststellt, ob sie in der Folge vorkommt.
Die Funktion C sagt uns dies für jede Zahl:

Definition 47 (C)**.**
$C : \mathbb{B} \longrightarrow \mathbb{B}$ ist definiert durch

$$C(p)(i) := \begin{cases} 0 & \textit{falls } (\exists n)p(n) = i + 1 \\ 1 & \textit{sonst.} \end{cases}$$

In den folgenden Abschnitten werden wir Funktionen mit Ω und C vergleichen. Dafür sind folgende Definitionen nützlich:

Definition 48 (Ω-stetig)**.**
Für $X = \mathbb{B}$ oder $X = \mathbb{N}$ heißt $f :\subseteq \mathbb{B} \longrightarrow X$ Ω-stetig, gdw.

$$f \leq_2 \Omega$$

In diesem Kapitel wird $\leq_2$ definiert durch $f \leq_2 g :\Leftrightarrow f(p) = A(p, g \circ B(p))$ für alle $p \in Def(f)$ mit stetigen Funktionen A und B.

Definition 49 (C^n-stetig)**.**
Sei $X = \mathbb{B}$ oder $X = \mathbb{N}$.
$f :\subseteq \mathbb{B} \longrightarrow X$ heißt C^0-stetig, gdw. f stetig ist.
$f :\subseteq \mathbb{B} \longrightarrow X$ heißt C^n-stetig für $n > 0$, gdw. es stetige Funktionen $A_n :\subseteq \mathbb{B} \to X$ und $A_{n-1}, \ldots, A_0 :\subseteq \mathbb{B} \to \mathbb{B}$ gibt mit

$$f(p) = A_n \circ C \circ \ldots A_1 \circ C \circ A_0(p)$$

für alle $p \in Def(f)$. Eine C^1-stetige Funktion f wird auch C-stetig genannt.

Definition 50 ($[\mathbb{B} \longrightarrow X]^n$).
Für $X = \mathbb{B}$ oder $X = \mathbb{N}$ sei $[\mathbb{B} \longrightarrow X]^n$ die Menge aller C^n-stetigen Funktionen $f :\subseteq \mathbb{B} \longrightarrow X$.

Es gibt unstetige Funktionen, die auf Ω reduzierbar sind[13]:

Definition 51 (Test auf Gleichheit mit Null: O).
$O :\subseteq \mathbb{B} \longrightarrow \mathbb{N}$, definiert durch

$$O(p) := \begin{cases} 0 & \textit{falls } \rho(p) = 0 \\ 1 & \textit{sonst} \end{cases} = \begin{cases} 0 & \textit{falls } (\forall n)|\nu_Q(p(n))| \leq 2^{-n} \\ 1 & \textit{sonst} \end{cases}$$

ist der Test auf Gleichheit mit Null und $\Omega \equiv_2 O$.[14]

Definition 52 (Test auf Gleichheit mit der Null-Funktion: V).
$V :\subseteq \mathbb{B} \longrightarrow \mathbb{N}$, definiert durch

$$V(p) := \begin{cases} 0 & \textit{falls } \delta_\alpha(p) = \overline{0} \\ 1 & \textit{sonst} \end{cases} = \begin{cases} 0 & \textit{falls } (\forall n)|\alpha(p(n))| \leq 2^{-n} \\ 1 & \textit{sonst} \end{cases}$$

ist der Test auf Gleichheit mit der Null-Funktion und $\Omega \equiv_2 V$.[15]

Definition 53 (Test auf Nullstellen einer Funktion: Z).
$Z :\subseteq \mathbb{B} \longrightarrow \mathbb{N}$, definiert durch

$$Z(p) := \begin{cases} 0 & \textit{falls } \delta_\alpha(p) \textit{hat eine Nullstelle} \\ 1 & \textit{sonst} \end{cases} = \begin{cases} 0 & \textit{falls } (\forall n) \inf(|\alpha(p(n))|) \leq 2^{-n} \\ 1 & \textit{sonst} \end{cases}$$

ist der Test, ob eine stetige Funktion eine Nullstelle hat, und $\Omega \equiv_2 Z$.

Andere Funktionen sind schwieriger und nicht auf Ω reduzierbar, aber auf C:

Definition 54 (Translation von ρ_n nach ρ).
Es gibt eine Translation F von ρ_n nach ρ mit der Eigenschaft

$$\rho(F(p)) = \rho_n(p)$$

und $C \equiv_2 F$.

Definition 55 (Ableitung einer Funktion in einem Punkt: D').
Es gibt eine Ableitung D' für $\delta_\alpha(p)$ im Punkt $\rho_n(q)$ mit der Eigenschaft

$$\rho(D'\langle p, q\rangle) = (\delta_\alpha(p))'(\rho_n(q))$$

und $C \equiv_2 D'$.

[13] Die folgenden Funktionen wurden in [St89] und [Myl92] definiert und untersucht
[14] ρ und ν_Q wurden in Abschnitt 2 definiert
[15] δ_α, $\overline{0}$ und $\alpha(p(n))$ wurden in Abschnitt 2 definiert

Definition 56 (Ableitung einer Funktion: A).
Es gibt eine Ableitung A für $\delta_\alpha(p)$ mit der Eigenschaft

$$\delta_\alpha(A(p)) = (\delta_\alpha(p))'$$

und $C \equiv_2 A$.

Definition 57 (Höhere Ableitungen einer Funktion: A^k).
Für jedes $k \in \mathbb{N}$ gibt es eine Ableitung A^k für $\delta_\alpha(p)$ mit der Eigenschaft

$$\delta_\alpha(A^k(p)) = (\delta_\alpha(p))^{(k)}$$

und $C \equiv_2 A^k$.

In [Myl92] wurde gezeigt, dass es Funktionen gibt, die noch höher in der C-Hierarchie eingeordnet werden müssen. Hier sind einige Beispiele:

Definition 58 (Konvergenz einer Folge von natürlichen Zahlen: KON).
$KON : \mathbb{B} \longrightarrow \mathbb{N}$, definiert durch

$$KON(p) := \begin{cases} 0 & \textit{falls } \lim_{n\to\infty} p(n) \textit{ existiert} \\ 1 & \textit{sonst} \end{cases} = \begin{cases} 0 & \textit{falls } (\exists l)(\exists k)(\forall m > l) p(m) = k \\ 1 & \textit{sonst} \end{cases}$$

ist der Test auf Konvergenz, und KON ist C^2-stetig, aber nicht C-stetig.

Definition 59 (Grenzwert einer Folge von natürlichen Zahlen: $KON2$).
$KON2 : \mathbb{B} \longrightarrow \mathbb{N}$, definiert durch

$$KON2(p) := \begin{cases} m+1 & \textit{falls } \lim_{n\to\infty} p(n) = m \\ 0 & \textit{sonst} \end{cases}$$

$$= \begin{cases} m+1 & \textit{falls } (\exists l)(\forall k > l) p(k) = m \\ 0 & \textit{sonst} \end{cases}$$

berechnet den Grenzwert einer Folge, und $KON2$ ist C^2-stetig, aber nicht C-stetig.

Definition 60 (Berechnung des Maximums: $MAX2$).
$MAX2 : \mathbb{B} \longrightarrow \mathbb{N}$, definiert durch

$$MAX2(p) := \begin{cases} m+1 & \textit{falls } m = \max\{p(i) | i \in \mathbb{N}\} \textit{ existiert} \\ 0 & \textit{sonst} \end{cases}$$

berechnet das Maximum einer Folge, und $MAX2$ ist C^2-stetig, aber nicht C-stetig.

Definition 61 (Existenz einer konvergenten Teilfolge: T).
$T : \mathbb{B} \longrightarrow \{0, 1\}$, definiert durch

$$T(p) := \begin{cases} 0 & \textit{falls es eine konstante Teilfolge in } p \textit{ gibt} \\ 1 & \textit{sonst} \end{cases}$$

$$= \begin{cases} 0 & \textit{if } (\exists n)(\forall m)(\exists k > m) p(k) = n \\ 1 & \textit{sonst} \end{cases}$$

ist der Test auf die Existenz einer konstanten Teilfolge, und T ist C^3-stetig, aber nicht C^2-stetig.

Definition 62 (Konvergenz von rationalen Zahlen: $QKON$).
$QKON : \mathbb{B} \longrightarrow \{0,1\}$, definiert durch

$$QKON(p) := \begin{cases} 0 & \textit{falls } \lim_{n\to\infty} \nu_Q(p(n)) \textit{ existiert} \\ 1 & \textit{sonst} \end{cases}$$

ist der Test auf die Konvergenz von rationalen Zahlen, und $QKON$ ist C^3-stetig, aber nicht C^2-stetig.

Definition 63 (Existenz einer konvergenten Teilfolge von rationalen Zahlen: $TQKON$).
$TQKON : \mathbb{B} \longrightarrow \{0,1\}$, definiert durch

$$TQKON(p) := \begin{cases} 0 & \textit{falls } (\nu_Q(p(n)))_{n\in\mathbb{N}} \textit{ hat eine konvergente Teilfolge} \\ 1 & \textit{sonst} \end{cases}$$

$$= \begin{cases} 0 & \textit{falls } (\exists n)(\exists m)(\forall k)(\exists i > k)\nu_Q(n) \leq \nu_Q(p(i)) \leq \nu_Q(m) \\ 1 & \textit{sonst} \end{cases}$$

ist der Test auf die Existenz einer konvergenten Teilfolge von rationalen Zahlen, und $TQKON$ ist C^3-stetig, aber nicht C^2-stetig.

Definition 64 (Konvergenz von reellen Zahlen: $RKON$).
$RKON :\subseteq \mathbb{B} \longrightarrow \{0,1\}$, definiert durch

$$RKON(p) := \begin{cases} 0 & \textit{falls } \lim_{n\to\infty} \rho(p_n) \textit{ existiert} \\ 1 & \textit{sonst} \end{cases}$$

ist der Test auf die Konvergenz von reellen Zahlen, und $RKON$ ist C^3-stetig, aber nicht C^2-stetig.

Definition 65 (Existenz einer konvergenten Teilfolge von reellen Zahlen: $TRKON$).

$TRKON :\subseteq \mathbb{B} \longrightarrow \{0,1\}$, definiert durch

$$TRKON(p) := \begin{cases} 0 & \textit{falls } (\rho(p_n))_{n\in\mathbb{N}} \textit{ hat eine konvergente Teilfolge} \\ 1 & \textit{sonst} \end{cases}$$

ist der Test auf die Existenz einer konvergenten Teilfolge von reellen Zahlen, und $TRKON$ ist C^3-stetig, aber nicht C^2-stetig.

Die nächsten Definitionen verwenden die universellen Funktionen U_χ^n und U_ψ^n für die C^n-stetigen Funktionen von $\mathbb{B}$ nach $\mathbb{N}$ bzw. von $\mathbb{B}$ nach $\mathbb{B}$. Siehe dazu Definition 71 oder [St89] Seite 61.

Definition 66 (Selbstanwendbarkeitsproblem: K_χ^n).
Für $n \geq 0$ ist $K_\chi^n : \mathbb{B} \longrightarrow \{0,1\}$, definiert durch

$$K_\chi^n(p) := \begin{cases} 0 & \textit{falls } U_\chi^n(p,p) \textit{ existiert} \\ 1 & \textit{sonst} \end{cases}$$

das Selbstanwendbarkeitsproblem für die universelle Funktion U_χ^n und ist C^{n+1}-stetig, aber nicht C^n-stetig.

Definition 67 (Selbstanwendbarkeitsproblem: K^n_ψ).
Für $n \geq 0$ ist $K^n_\psi : \mathbb{B} \longrightarrow \{0,1\}$, definiert durch

$$K^n_\psi(p) := \begin{cases} 0 & \textit{falls } U^n_\psi(p,p) \textit{ existiert} \\ 1 & \textit{sonst} \end{cases}$$

das Selbstanwendbarkeitsproblem für die universelle Funktion U^n_ψ und ist C^{n+2}-stetig, aber nicht C^{n+1}-stetig.

Definition 68 (Halteproblem: H^n_χ).
Für $n \geq 0$ ist $H^n_\chi : \mathbb{B} \longrightarrow \{0,1\}$, definiert durch

$$H^n_\chi\langle p,q\rangle := \begin{cases} 0 & \textit{falls } U^n_\chi(p,q) \textit{ existiert} \\ 1 & \textit{sonst} \end{cases}$$

das Halteproblem für die universelle Funktion U^n_χ und ist C^{n+1}-stetig, aber nicht C^n-stetig.

Definition 69 (Halteproblem: H^n_ψ).
Für $n \geq 0$ ist $H^n_\psi : \mathbb{B} \longrightarrow \{0,1\}$, definiert durch

$$H^n_\psi\langle p,q\rangle := \begin{cases} 0 & \textit{falls } U^n_\psi(p,q) \textit{ existiert} \\ 1 & \textit{sonst} \end{cases}$$

das Halteproblem für die universelle Funktion U^n_ψ und ist C^{n+2}-stetig, aber nicht C^{n+1}-stetig.

4.2 Vollständige Funktionen

4.2.1 Vollständige Funktionen für $[\mathbb{B} \to X]^n$

In [St89] und in [Myl92] wurden Funktionen höheren Unstetigkeitsgrades untersucht.

Es wurde bewiesen, dass einige Funktionen Ω^n-stetig oder C^n-stetig sind für $n \in \mathbb{N}$. In vielen Fällen wurde auch bewiesen, dass einige der Funktionen äquivalent zu einander sind.

Aber es wurde meistens nicht geklärt, welche Funktionen vollständig für bestimmte Klassen sind und ob es solche Funktionen gibt.

Es ist klar, dass C vollständig für $[\mathbb{B} \to \mathbb{B}]^1$ ist und dass Ω vollständig für die Ω-stetigen Funktionen ist. Aber $KON2$ ist zum Beispiel C^2-stetig und nicht auf C reduzierbar. Und es ist nicht klar, ob $KON2$ vollständig für $[\mathbb{B} \to \mathbb{N}]^2$ ist.

Wir werden nun die Vollständigkeit für einige Funktionen und Klassen beweisen.

Zu diesem Zweck werden wir Funktionen definieren, die vollständig für $[\mathbb{B} \to \mathbb{B}]^n$, $[\mathbb{B} \to \mathbb{N}]^n$ und $[\mathbb{B} \to \{0,1\}]^n$ sind. Dann werden wir zeigen, dass die meisten der Funktionen, die in [St89] und [Myl92] definiert wurden, nicht vollständig sind.

Definition 70 (Vollständigkeit für $[\mathbb{B} \to X]^n$).
Sei $X = \mathbb{B}$, $X = \mathbb{N}$ *oder* $X = \{0,1\}$.
$f : \mathbb{B} \to X$ *heißt* $[\mathbb{B} \longrightarrow X]^n - i-$*hart oder* $i-$*hart für* $[\mathbb{B} \longrightarrow X]^n$, *gdw.*

$$g \leq_i f$$

für alle $g \in [\mathbb{B} \longrightarrow X]^n$, $i \in \{0,1,2\}$.
$f : \mathbb{B} \to X$ *heißt* $[\mathbb{B} \longrightarrow X]^n - i-$*vollständig oder* $i-$*vollständig für* $[\mathbb{B} \longrightarrow X]^n$, *gdw.*

$$f \in [\mathbb{B} \longrightarrow X]^n \text{ und } f \text{ ist } i\text{−hart für } [\mathbb{B} \longrightarrow X]^n.$$

Manchmal sagen wir statt $2-$hart oder $2-$vollständig auch einfach hart oder vollständig, wenn die Bedeutung unzweifelhaft ist.

4.2.2 Universelle Funktionen

Ein Beispiel für vollständige Funktionen sind die universellen Funktionen.

Definition 71 (Universelle Funktionen U_χ^n und U_ψ^n)**.**
Die universellen Funktionen $U_\chi^n :\subseteq \mathbb{B} \longrightarrow \mathbb{N}$ *und* $U_\psi^n :\subseteq \mathbb{B} \longrightarrow \mathbb{B}$ *sind definiert durch*

$$U_\chi^n\langle p, q\rangle := \chi_p^n(q)$$

und

$$U_\psi^n\langle p, q\rangle := \psi_p^n(q).$$

χ^n und ψ^n sind dabei Repräsentationen der C^n-stetigen Funktionen aus $[\mathbb{B} \longrightarrow \mathbb{N}]^n$ und $[\mathbb{B} \longrightarrow \mathbb{B}]^n$.

Satz 60 (Vollständige Funktion für $[\mathbb{B} \longrightarrow \mathbb{N}]^n$)**.**
U_χ^n *ist* 2*-vollständig für die Klasse* $[\mathbb{B} \longrightarrow \mathbb{N}]^n$*.*

Beweis. Sei f eine Funktion aus $[\mathbb{B} \longrightarrow \mathbb{N}]^n$.
Dann gibt es ein $p \in \mathbb{B}$ mit $f(q) = \chi_p^n(q)$. Definiere g durch $g(q) := \langle p, q\rangle$.
Dann folgt $f(q) = \chi_p^n(q) = U_\chi^n\langle p, q\rangle = U_\chi^n \circ g(q)$. Somit gilt $f \leq_2 U_\chi^n$.
Ferner gilt $U_\chi^n \in [\mathbb{B} \longrightarrow \mathbb{N}]^n$ nach dem utm-Theorem für χ.[16] □

Satz 61 (Vollständige Funktion für $[\mathbb{B} \longrightarrow \mathbb{B}]^n$)**.**
U_ψ^n *ist* 2*-vollständig für die Klasse* $[\mathbb{B} \longrightarrow \mathbb{B}]^n$*.*

Beweis. Sei f eine Funktion aus $[\mathbb{B} \longrightarrow \mathbb{B}]^n$.
Dann gibt es ein $p \in \mathbb{B}$ mit $f(q) = \psi_p^n(q)$. Definiere g durch $g(q) := \langle p, q\rangle$.
Dann folgt $f(q) = \psi_p^n(q) = U_\psi^n\langle p, q\rangle = U_\psi^n \circ g(q)$. Somit gilt $f \leq_2 U_\psi^n$.
Weiter gilt $U_\psi^n \in [\mathbb{B} \longrightarrow \mathbb{B}]^n$ nach dem utm-Theorem für ψ.[17] □

Somit gibt es für alle $n \in \mathbb{N}$ 2-vollständige Funktionen für $[\mathbb{B} \longrightarrow \mathbb{B}]^n$ und $[\mathbb{B} \longrightarrow \mathbb{N}]^n$. Aber diese Funktionen sind „Low-Level-Funktionen". Sie sind maschinenorientiert, sie sind definiert auf Grundlage von Typ-II-Turingmaschinen. Und für $n \geq 2$ sind sie definiert durch eine Komposition von stetigen Funktionen und Anwendungen von C. In den nächsten Abschnitten werden wir Funktionen einführen, die in mathematischen Begriffen definiert sind und nicht durch wiederholte Anwendung der Funktion C.

[16] [St89] Seite 61
[17] [St89] Seite 61

4.2.3 Einige Lemmata für die Funktion C

Die nächsten Lemmata werden in Beweisen benötigt.

Lemma 15. *Für stetige Funktionen $A :\subseteq \mathbb{B} \times \mathbb{B} \to \mathbb{N}$ und $B :\subseteq \mathbb{B} \to \mathbb{B}$ gibt es stetige Funktionen $A' :\subseteq \mathbb{B} \to \mathbb{N}$ und $B' :\subseteq \mathbb{B} \to \mathbb{B}$ mit*

$$A(p, C \circ B(p)) = A' \circ C \circ B'(p)$$

für alle $p \in \mathbb{B}$.

Lemma 16. *Für stetige Funktionen $A :\subseteq \mathbb{B} \times \mathbb{B} \to \mathbb{B}$ und $B :\subseteq \mathbb{B} \to \mathbb{B}$ gibt es stetige Funktionen $A' :\subseteq \mathbb{B} \to \mathbb{B}$ und $B' :\subseteq \mathbb{B} \to \mathbb{B}$ mit*

$$A(p, C \circ B(p)) = A' \circ C \circ B'(p)$$

für alle $p \in \mathbb{B}$.

Das bedeutet, dass $f \leq_1 C$, gdw. $f \leq_2 C$. Der Beweis steht in [St89].

Lemma 17. *Für C-stetige Funktionen $f :\subseteq \mathbb{B} \to \mathbb{N}$ und $g :\subseteq \mathbb{B} \to \mathbb{N}$ gibt es eine C-stetige Funktion $h :\subseteq \mathbb{B} \to \mathbb{N}$ mit*

$$h(p) = \langle f(p), g(p) \rangle$$

für alle $p \in \mathbb{B}$.

Beweis. Es gibt stetige Funktionen $A :\subseteq \mathbb{B} \to \mathbb{N}$, $A' :\subseteq \mathbb{B} \to \mathbb{N}$, $B :\subseteq \mathbb{B} \to \mathbb{B}$ und $B' :\subseteq \mathbb{B} \to \mathbb{B}$ mit $f(p) = A \circ C \circ B(p)$ und $g(p) = A' \circ C \circ B'(p)$ für alle $p \in \mathbb{B}$.
Definiere $B^* : \mathbb{B} \to \mathbb{B}$ durch

$$B^*(p)(2n) := \begin{cases} 0 & \text{falls } B(p)(n) = 0 \\ 2B(p)(n) & \text{sonst} \end{cases}$$

und

$$B^*(p)(2n+1) := \begin{cases} 0 & \text{falls } B'(p)(n) = 0 \\ 2B'(p)(n) + 1 & \text{sonst} \end{cases}$$

und $A^* :\subseteq \mathbb{B} \to \mathbb{N}$ durch

$$A^*(q) := \langle a, b \rangle$$

mit

$$a := A(q(1), q(3), q(5), \ldots)$$

und

$$b := A'(q(2), q(4), q(6), \ldots)$$

Definiere $h := A^* \circ C \circ B^*$. Dann gilt $h(p) = \langle f(p), g(p) \rangle$ für alle $p \in \mathbb{B}$.
<u>Beweis</u>:
Definiere $q = C \circ B^*(p)$.
$q(2i+1) = 0 \iff C \circ B^*(p)(2i+1) = 0 \iff (\exists k) B^*(p)(k) = 2i+2$
$\iff (\exists k) B(p)(k) = i+1 \iff C \circ B(p)(i) = 0$ und
$q(2i+2) = 0 \iff C \circ B^*(p)(2i+2) = 0 \iff (\exists k) B^*(p)(k) = 2i+3$
$\iff (\exists k) B'(p)(k) = i+1 \iff C \circ B'(p)(i) = 0$
Somit gilt $C \circ B(p) = (q(1), q(3), q(5), \ldots)$ und $C \circ B'(p) = (q(2), q(4), q(6), \ldots)$.
Daraus folgt $A^* \circ C \circ B^*(p) = \langle A(q(1), q(3), q(5), \ldots), A'(q(2), q(4), q(6), \ldots) \rangle$
$= \langle f(p), g(p) \rangle$. □

Lemma 18. *Für C-stetige Funktionen $f :\subseteq \mathbb{B} \to \mathbb{B}$ und $g :\subseteq \mathbb{B} \to \mathbb{B}$ gibt es eine C-stetige Funktion $h :\subseteq \mathbb{B} \to \mathbb{B}$ mit*

$$h(p) = (\langle f(p)(0), g(p)(0)\rangle, \langle f(p)(1), g(p)(1)\rangle, \ldots)$$

für alle $p \in \mathbb{B}$.

Beweis. Der Beweis verläuft ähnlich wie der vorige. □

Lemma 19. *Für eine C^n-stetige Funktion $D :\subseteq \mathbb{B} \to \mathbb{B}$ definiere $D' :\subseteq \mathbb{B} \to \mathbb{B}$ durch $D'(p) = D'\langle p_0, p_1, \ldots\rangle := \langle D(p_0), D(p_1), \ldots\rangle$.*
Dann ist D' C^n-stetig.

Beweis. Durch Induktion über n:
$\underline{n = 0}$:
Falls $n = 0$, dann ist D stetig. Sei $D'\langle p_0, p_1, \ldots\rangle := \langle D(p_0), D(p_1), \ldots\rangle$.
Es ist klar, dass D' stetig ist.
$\underline{n \longrightarrow n + 1}$:
Sei D C^{n+1}-stetig. Dann gilt $D = A \circ C \circ B$ mit einer stetigen Funktion $A :\subseteq \mathbb{B} \to \mathbb{B}$ und einer C^n-stetigen Funktion $B :\subseteq \mathbb{B} \to \mathbb{B}$.
Dann gilt $D'(p) = \langle D(p_0), D(p_1), \ldots\rangle$
$= \langle A \circ C \circ B(p_0), A \circ C \circ B(p_1), \ldots\rangle$.
Es gibt eine stetige Funktion A' mit
$D'(p) = \langle D(p_0), D(p_1), \ldots\rangle$
$= \langle A \circ C \circ B(p_0), A \circ C \circ B(p_1), \ldots\rangle$
$= A'\langle C \circ B(p_0), C \circ B(p_1), \ldots\rangle$.
Nach Lemma 20 gibt es eine C-stetige Funktion C' mit
$D'(p) = \langle D(p_0), D(p_1), \ldots\rangle$
$= A'\langle C \circ B(p_0), C \circ B(p_1), \ldots\rangle$
$= A' \circ C'\langle B(p_0), B(p_1), \ldots\rangle$.
Nach Induktionsannahme gibt es eine C^n-stetige Funktion B' mit
$D'(p) = \langle D(p_0), D(p_1), \ldots\rangle$
$= A'\langle C \circ B(p_0), C \circ B(p_1), \ldots\rangle$
$= A' \circ C'\langle B(p_0), B(p_1), \ldots\rangle$
$= A' \circ C' \circ B'\langle p_0, p_1, \ldots\rangle$
$= A' \circ C' \circ B'(p)$.
Somit ist $D' = A' \circ C' \circ B'$ C^{n+1}-stetig. □

Lemma 20. *Definiere $C' :\subseteq \mathbb{B} \to \mathbb{B}$ durch $C'\langle p_0, p_1, \ldots\rangle := \langle C(p_0), C(p_1), \ldots\rangle$.*
Dann ist C' C-stetig.

Beweis. Definiere $f :\subseteq \mathbb{B} \to \mathbb{B}$ durch $f(p)\langle i, j\rangle := \langle i, p\langle i, j\rangle\rangle + 1$ und $g :\subseteq \mathbb{B} \to \mathbb{B}$ durch
$g(p)\langle i, j\rangle := p\langle i, j + 1\rangle$.
f und g sind stetig und $C'\langle p_0, p_1, \ldots\rangle = g \circ C \circ f\langle p_0, p_1, \ldots\rangle$.
<u>Beweis</u>:
$C'\langle p_0, p_1, \ldots\rangle\langle i, j\rangle = 0$
$\Longrightarrow \langle C(p_0), C(p_1), \ldots\rangle\langle i, j\rangle = 0$
$\Longrightarrow C(p_i)(j) = 0$
$\Longrightarrow (\exists m) p_i(m) = j + 1$
$\Longrightarrow (\exists m) f(p)\langle i, m\rangle = \langle i, p\langle i, m\rangle\rangle + 1 = \langle i, j + 1\rangle + 1$
$\Longrightarrow (\exists k) f(p)(k) = \langle i, j + 1\rangle + 1$

$\Longrightarrow C \circ f(p)\langle i, j+1\rangle = 0$
$\Longrightarrow g \circ C \circ f(p)\langle i, j\rangle = 0$
und
$C'\langle p_0, p_1, \ldots\rangle\langle i, j\rangle = 1$
$\Longrightarrow \langle C(p_0), C(p_1), \ldots\rangle\langle i, j\rangle = 1$
$\Longrightarrow C(p_i)(j) = 1$
$\Longrightarrow (\forall m) p_i(m) \neq j+1$
$\Longrightarrow (\forall m) f(p)\langle i, m\rangle = \langle i, p\langle i, m\rangle\rangle + 1 \neq \langle i, j+1\rangle + 1$
$\Longrightarrow (\forall k) f(p)(k) \neq \langle i, j+1\rangle + 1$ [18]
$\Longrightarrow C \circ f(p)\langle i, j+1\rangle = 1$
$\Longrightarrow g \circ C \circ f(p)\langle i, j\rangle = 1.$ □

[18] Wenn es eine Zahl $k = \langle a, b\rangle$ gäbe mit $f(p)(k) = \langle i, j+1\rangle + 1$, dann würde gelten: $f(p)(k) = f(p)\langle a, b\rangle = \langle a, p\langle a, b\rangle\rangle + 1 = \langle i, j+1\rangle + 1$ impliziert $a = i$ und damit $p\langle i, b\rangle = j+1$. Dann gilt $p_i(b) = j+1$. Das ist ein Widerspruch zu $(\forall m) p_i(m) \neq j+1$.

4.2.4 $KDIV^{\mathbb{N}}$

In [Myl92] wurde eine Funktion $KON2 : \mathbb{B} \to \mathbb{N}$ definiert, mit der der Grenzwert einer Folge berechnet wird.

$$KON2(p) := \begin{cases} n+1 & \text{falls } \lim_{m\to\infty}(p(m)) = n \\ 0 & \text{sonst} \end{cases}$$
$$= \begin{cases} n+1 & \text{falls } (\exists k)(\forall m > k) p(m) = n \\ 0 & \text{sonst} \end{cases}$$

$KON2$ ist nicht C-stetig, aber C^2-stetig und, wie wir später sehen, nicht vollständig für $[\mathbb{B} \to \mathbb{N}]^2$. $KON2$ berechnet nicht nur den Grenzwert einer Folge, sondern entscheidet darüber hinaus auch, ob die Folge konvergiert oder nicht. Das Problem ist schwieriger, als nur den Grenzwert zu berechnen, falls er existiert.

Für unsere Zwecke ist es besser, eine leichtere Funktion zu definieren, die den Grenzwert einer Folge berechnet und die divergiert, wenn kein Grenzwert existiert. Diese Funktion ist, wie wir sehen werden, C-stetig und erlaubt die Definition einer Reihe von Funktionen, die vollständig für die Klassen in der C-Hierarchie sind.

Definition 72 ($KDIV^{\mathbb{N}}$)**.**
Sei $KDIV^{\mathbb{N}} :\subseteq \mathbb{B} \to \mathbb{N}$ definiert durch

$$KDIV^{\mathbb{N}}(p) := \begin{cases} m & \textit{falls } \lim_{k\to\infty} p(k) = m \textit{ existiert} \\ div & \textit{sonst} \end{cases}$$
$$= \begin{cases} m & \textit{falls } (\exists x)(\forall k > x) p(k) = m \\ div & \textit{sonst} \end{cases}$$

$KDIV^{\mathbb{N}}$ berechnet den Grenzwert einer Folge aus $\mathbb{B}$. Aber in Gegensatz zu $KON2$ divergiert diese Funktion, wenn p keinen Grenzwert hat.

Satz 62. *$KDIV^{\mathbb{N}}$ ist C-stetig.*

Beweis. Definiere $f : \mathbb{B} \to \mathbb{B}$ durch $f(p)(0) := \langle 0, 0\rangle$ und

$$f(p)(n+1) := \begin{cases} \langle p(n+1)+1, \pi_2^{(2)}(f(p)(n))\rangle & \text{falls } p(n) = p(n+1) \\ \langle 0, \pi_2^{(2)}(f(p)(n)) + 1\rangle & \text{falls } p(n) \neq p(n+1) \end{cases}$$

f wird benutzt, um eine Position zu entdecken, ab der die Folge konstant bleibt. Falls der n-te Wert sich vom Wert davor unterscheidet, gilt $f(p)(n) = \langle 0, b\rangle$, wobei b ein Zähler ist, der mit jedem Wechsel in p inkrementiert wird. Die Existenz eines maximalen Wertes für b zeigt an, dass p konvergiert. Falls der n-te Wert sich nicht vom Wert davor unterscheidet, gilt $f(p)(n) = \langle a, b\rangle$, wobei a den Wert von $p(n)$ enthält, um 1 inkrementiert.

Definiere $g :\subseteq \mathbb{B} \to \mathbb{B}$ durch $g(p) := \langle min\{n | p\langle 0, n\rangle = 1\}, p\rangle$ und

$h :\subseteq \mathbb{B} \to \mathbb{N}$ durch $h\langle a, p\rangle := min\{n > 0 | p\langle n, a\rangle = 0\} - 1$.
Dann gilt $h \circ g \circ C \circ INC \circ f(p) = KDIV^{\mathbb{N}}(p)$ für alle $p \in \mathbb{B}$.
Beweis:
Sei p divergent. Dann existiert für jedes n eine Zahl m
mit $f(p)(m) = \langle 0, n\rangle$. Somit gilt $C \circ INC \circ f(p)\langle 0, n\rangle = 0$ für alle $n \in \mathbb{N}$.
Dann gilt $g \circ C \circ INC \circ f(p) = div$ und $h \circ g \circ C \circ INC \circ f(p) = div = KDIV^{\mathbb{N}}(p)$.
Sei p konvergent. Dann gibt es eine Zahl n mit $p(m) = n$ für fast alle $m \in \mathbb{N}$.
Dann gibt es ein minimales y mit $f(p)(k) \neq \langle 0, y\rangle$ für alle $k \in \mathbb{N}$.
Dann folgt $C \circ INC \circ f(p)\langle 0, y\rangle = 1$ und $C \circ INC \circ f(p)\langle 0, z\rangle = 0$ für alle $z < y$.
Das impliziert $g \circ C \circ INC \circ f(p) = \langle y, C \circ INC \circ f(p)\rangle$ und
$h \circ g \circ C \circ INC \circ f(p) = h\langle y, C \circ INC \circ f(p)\rangle$
$= min\{m > 0 | C \circ INC \circ f(p)\langle m, y\rangle = 0\} - 1$
$= min\{m > 0 | (\exists k) f(p)(k) = \langle m, y\rangle\} - 1 = n$
$= KDIV^{\mathbb{N}}(p)$. □

Lemma 21. *Für alle stetigen Funktion $A :\subseteq \mathbb{B} \longrightarrow \mathbb{N}$ und $B :\subseteq \mathbb{B} \longrightarrow \mathbb{B}$ gibt es eine stetige Funktion $D : \mathbb{B} \longrightarrow \mathbb{B}$ mit $A \circ C \circ B(p) = KDIV^{\mathbb{N}} \circ D(p)$ für alle $p \in Def(A \circ C \circ B)$.*

Beweis. Für jede stetige Funktion $B :\subseteq \mathbb{B} \longrightarrow \mathbb{B}$ sei $\overline{B} : \mathbb{N}^* \longrightarrow \mathbb{N}^*$ die assoziierte isotone Funktion.[19]
Definiere die totale Funktion $B^* : \mathbb{N}^* \to \mathbb{N}^*$ durch

$$B^*(()) := P(\overline{B}(()))$$

und

$$B^*(wx) := \begin{cases} B^*(w) \cdot P(v) & \text{falls es ein } v \neq () \text{ gibt mit } \overline{B}(wx) = \overline{B}(w) \cdot v \\ B^*(w) \cdot 0 & \text{sonst} \end{cases}$$

für alle $w \in \mathbb{N}^*$ und $x \in \mathbb{N}$,
wobei $P : \mathbb{N}^* \longrightarrow \mathbb{N}^*$ definiert ist durch $P(n_1, \ldots, n_k) := (n_1 + 1, \ldots, n_k + 1)$.
Definiere die totale Funktion $B' : \mathbb{B} \to \mathbb{B}$ durch

$$B'(p) := \sup\{B^*(w) | w \sqsubseteq p\}$$

für alle $p \in \mathbb{B}$.
$B'(p)$ enthält alle Elemente von $B(p)$ in derselben Reihenfolge, um 1 erhöht, mit zusätzlichen Nullen in der Folge. Wenn B stetig ist, dann ist es B' auch.
Definiere $C' : \mathbb{B} \to \mathbb{B}$ durch

$$C'(q)\langle i, n\rangle := \begin{cases} 0 & \text{falls } (\exists k \leq i) q(k) = n + 2 \\ 1 & \text{sonst} \end{cases}$$

Für jede stetige Funktion $A : \mathbb{B} \longrightarrow \mathbb{N}$ definiere A'' durch

$$A''(q)\langle i\rangle := \begin{cases} \overline{A}(q\langle i, 0\rangle, \ldots, q\langle i, i\rangle) & \text{falls } \overline{A}(q\langle i, 0\rangle, \ldots, q\langle i, i\rangle) \text{ existiert} \\ i & \text{sonst} \end{cases}$$

[19] Für jede stetige Funktion $f :\subseteq \mathbb{B} \to \mathbb{B}$ gibt es eine isotone Funktion $\overline{f} : \mathbb{N}^* \to \mathbb{N}^*$ mit der Eigenschaft $\overline{f}([[p]]_n) \sqsubseteq \overline{f}([[p]]_m)$ für alle $n \leq m$ und $\lim_{n\to\infty} \overline{f}([[p]]_n) = f(p)$. Der Beweis dieses Satzes ähnelt dem Beweis des „Warteschleifentheorems" in [St89], Seite 63.

Definiere D durch $D := A'' \circ C' \circ B'$.
Dann gilt $A \circ C \circ B(p) = k$
$\iff \lim_{i\to\infty} A'' \circ C' \circ B'(p)(i) = k$
$\iff KDIV^{\mathbb{N}} \circ A'' \circ C' \circ B'(p) = k$
$\iff KDIV^{\mathbb{N}} \circ D(p) = k$
und
$A \circ C \circ B(p) = div$
$\iff \lim_{i\to\infty} A'' \circ C' \circ B'(p)(i) = div$
$\iff KDIV^{\mathbb{N}} \circ A'' \circ C' \circ B'(p) = div$
$\iff KDIV^{\mathbb{N}} \circ D(p) = div$ □

Lemma 22. *Für alle stetigen Funktionen $A :\subseteq \mathbb{B} \longrightarrow \mathbb{B}$ und $B :\subseteq \mathbb{B} \longrightarrow \mathbb{B}$ gibt es eine stetige Funktion $D : \mathbb{B} \longrightarrow \mathbb{B}$ mit*
$A \circ C \circ B(p) = (KDIV^{\mathbb{N}}(D(p)(0)), KDIV^{\mathbb{N}}(D(p)(1)), \ldots)$.

Beweis. Für jede stetige Funktion $B :\subseteq \mathbb{B} \longrightarrow \mathbb{B}$ sei $\overline{B} : \mathbb{N}^* \longrightarrow \mathbb{N}^*$ die assoziierte isotone Funktion.
Definiere die Funktionen $B^* : \mathbb{N}^* \to \mathbb{N}^*$ und $B' : \mathbb{B} \to \mathbb{B}$ wie in dem vorigen Beweis.
$B'(p)$ enthält wieder alle Elemente von $B(p)$ in derselben Reihenfolge und um 1 inkrementiert, mit zusätzlichen Nullen in der Ausgabefolge. Wenn B stetig ist, dann ist es auch B'.
$C' : \mathbb{B} \to \mathbb{B}$ ist definiert wie in dem Beweis zuvor, aber die Definition von A'' unterscheidet sich von der vorigen Definition:
Für jede stetige Funktion $A :\subseteq \mathbb{B} \longrightarrow \mathbb{B}$ definiere A'' durch

$$A''(q)\langle i, j\rangle := \begin{cases} n & \text{falls } n = \overline{A}(q\langle i, 0\rangle, \ldots, q\langle i, i\rangle)(j) \text{ existiert} \\ i & \text{sonst} \end{cases}$$

Definiere D durch $D := A'' \circ C' \circ B'$.
Dann gilt $(A \circ C \circ B(p))(j) = k$
$\iff (\lim_{i\to\infty} A'' \circ C' \circ B'(p)(i))(j) = k$
$\iff \lim_{i\to\infty}(A'' \circ C' \circ B'(p)(i)(j)) = k$
$\iff KDIV^{\mathbb{N}}(A'' \circ C' \circ B'(p)(j)) = k$
$\iff KDIV^{\mathbb{N}}(D(p)(j)) = k$
und
$(A \circ C \circ B(p))(j) = div$
$\iff (\lim_{i\to\infty} A'' \circ C' \circ B'(p)(i))(j) = div$
$\iff \lim_{i\to\infty}(A'' \circ C' \circ B'(p)(i)(j)) = div$
$\iff KDIV^{\mathbb{N}}(A'' \circ C' \circ B'(p)(j)) = div$
$\iff KDIV^{\mathbb{N}}(D(p)(j)) = div.$ □

Satz 63. *$f \leq_0 KDIV^{\mathbb{N}}$ für jede C-stetige Funktion $f : \mathbb{B} \longrightarrow \mathbb{N}$.*

Beweis. Sei $f = A \circ C \circ B$. Dann gibt es nach Lemma 21 eine stetige Funktion D mit $f(p) = A \circ C \circ B(p) = KDIV^{\mathbb{N}} \circ D(p)$ für alle $p \in Def(f)$.
Somit gilt $f \leq_0 KDIV^{\mathbb{N}}$ für alle C-stetigen Funktionen $f : \mathbb{B} \longrightarrow \mathbb{N}$. □

Die letzten beiden Sätze implizieren

Satz 64. *$KDIV^{\mathbb{N}}$ ist 0-vollständig für $[\mathbb{B} \longrightarrow \mathbb{N}]$.*

$KDIV^{\mathbb{N}}$ ist eine Funktion mit Bildbereich $\mathbb{N}$. Wir definieren eine ähnliche Funktion mit Bildbereich $\{0,1\}$:

Definition 73 ($KDIV^{\{0,1\}}$).
Sei $KDIV^{\{0,1\}} :\subseteq \mathbb{B} \to \{0,1\}$ *eine Funktion mit*

$$KDIV^{\{0,1\}}(p) := \begin{cases} 0 & \textit{falls } \lim_{k\to\infty} p(k) = 0 \\ 1 & \textit{falls } \lim_{k\to\infty} p(k) > 0 \\ div & \textit{sonst} \end{cases}$$

$$= \begin{cases} 0 & \textit{falls } (\exists x)(\forall k > x)p(k) = 0 \\ 1 & \textit{falls } (\exists m > 0)(\exists x)(\forall k > x)p(k) = m \\ div & \textit{sonst} \end{cases}$$

Aus dieser Definition folgt unmittelbar:

Satz 65. $KDIV^{\{0,1\}}$ *ist* C*-stetig.*

Beweis. Definiere $f : \mathbb{N} \to \mathbb{N}$ durch

$$f(n) := \begin{cases} 0 & \text{falls } n = 0 \\ 1 & \text{falls } n > 0 \end{cases}$$

für alle $n \in \mathbb{N}$.
Dann gilt $KDIV^{\{0,1\}}(p) = f \circ KDIV^{\mathbb{N}}(p)$.
Da $KDIV^{\mathbb{N}}$ C-stetig ist, ist auch $KDIV^{\{0,1\}}$ C-stetig. □

Satz 66. $f \leq_0 KDIV^{\{0,1\}}$ *für jede* C*-stetige Funktion* $f :\subseteq \mathbb{B} \longrightarrow \{0,1\}$.

Beweis. Der Beweis ähnelt dem Beweis von Satz 63. □

Das impliziert

Satz 67. $KDIV^{\{0,1\}}$ *ist* 0*-vollständig für* $[\mathbb{B} \longrightarrow \{0,1\}]$.

4.2.5 $KDIV_n^{\mathbb{N}}$

Die nächste Definition definiert eine Funktion, die die Grenzwerte von unendlich vielen Teilfolgen berechnet und dann den Grenzwert von all diesen Werten berechnet, falls er existiert.

Definition 74 ($KDIV_2^{\mathbb{N}}$).
Sei $KDIV_2^{\mathbb{N}} :\subseteq \mathbb{B} \to \mathbb{N}$ definiert durch

$$\begin{aligned} KDIV_2^{\mathbb{N}}(p) :&= \begin{cases} n & \textit{falls } \lim_{l\to\infty} \lim_{m\to\infty} p\langle l, m\rangle = n \textit{ existiert} \\ div & \textit{sonst} \end{cases} \\ &= \begin{cases} n & \textit{falls } (\exists r)(\forall l > r)(\exists k)(\forall m > k) p\langle l, m\rangle = n \\ div & \textit{sonst} \end{cases} \end{aligned}$$

Mit einer rekursiven Definition ist es möglich, $KDIV^{\mathbb{N}}$ zu verallgemeinern. Durch $KDIV_n^{\mathbb{N}}$ wird der Grenzwert n-mal iteriert.

Definition 75 ($\lim_{(k_1,\ldots,k_n)\to\infty}$).
Definiere $\lim_{(k_1,\ldots,k_n)\to\infty} :\subseteq \mathbb{B} \to \mathbb{N}$ durch
$\lim_{()\to\infty}(p) := 0$,
$\lim_{(k_1)\to\infty}(p) := \lim_{k_1\to\infty}(p(k_1))$ *und*
$\lim_{(k_1,\ldots,k_{n+1})\to\infty}(p)$
$:= \lim_{k_1\to\infty}(\lim_{(k_2,\ldots,k_{n+1})\to\infty}(\langle p\rangle_0), \lim_{(k_2,\ldots,k_{n+1})\to\infty}(\langle p\rangle_1), \ldots)$.

Definition 76 ($KDIV_n^{\mathbb{N}}$).
Definiere $KDIV_n^{\mathbb{N}} :\subseteq \mathbb{B} \to \mathbb{N}$ durch

$$KDIV_n^{\mathbb{N}}(p) := \lim_{(k_1,\ldots,k_n)\to\infty}(p)$$

für alle $p \in \mathbb{B}$.

Analog verallgemeinern wir $KDIV^{\{0,1\}}$:

Definition 77. *Definiere $KDIV_n^{\{0,1\}} :\subseteq \mathbb{B} \to \{0,1\}$ durch*

$$\begin{aligned} KDIV_0^{\{0,1\}}(p) &:= 0 \\ KDIV_1^{\{0,1\}}(p) &:= KDIV^{\{0,1\}}(p) \textit{ und} \\ KDIV_n^{\{0,1\}}(p) :&= \begin{cases} 0 & \textit{falls } \lim_{(k_1,\ldots,k_n)\to\infty}(p) = 0 \\ 1 & \textit{falls } \lim_{(k_1,\ldots,k_n)\to\infty}(p) > 0 \\ div & \textit{sonst} \end{cases} \end{aligned}$$

für alle $n \geq 2$.

Die nächsten Lemmata werden für den Beweis des nächsten Satzes gebraucht:

Lemma 23 (Berechnung einer Folge mit dem Grenzwert).
Es gibt eine C-stetige Funktion $h : \mathbb{B} \to \mathbb{B}$ mit $h(p) = q$ und

1. $q = 0^\omega$ *falls* $\lim_{m\to\infty} p(m) = div$

2. $(\exists i) q = 0^i(n+1)0^\omega$ *falls* $\lim_{m\to\infty} p(m) = n$ *existiert*

Beweis. Definiere $f : \mathbb{B} \to \mathbb{B}$ durch $f(p)(0) := \langle 0, 0\rangle$ und

$$f(p)(n+1) := \begin{cases} \langle p(n)+1, \pi_2^{(2)}(f(p)(n))\rangle & \text{falls } p(n) = p(n+1) \\ \langle 0, \pi_2^{(2)}(f(p)(n)) + 1\rangle & \text{falls } p(n) \neq p(n+1) \end{cases}$$

Definiere $g : \mathbb{B} \to \mathbb{B}$ durch

$$g(p)\langle a, n\rangle := \begin{cases} a & \text{falls } p\langle 0, n\rangle = 0 \text{ und } p\langle 0, n+1\rangle = 1 \\ & \text{und } p\langle a, n\rangle = 0 \text{ und } a > 0 \\ 0 & \text{sonst} \end{cases}$$

f und g sind stetig.
Definiere $h : \mathbb{B} \to \mathbb{B}$ durch $h(p) := g \circ C \circ INC \circ f(p)$ für alle $p \in \mathbb{B}$.
Dann hat h die Eigenschaften des Lemmas.
Beweis:
Sei p divergent. Dann gibt es für jedes n eine Zahl m mit $f(p)(m) = \langle 0, n\rangle$. Somit gilt $C \circ INC \circ f(p)\langle 0, n\rangle = 0$ für alle $n \in \mathbb{N}$.
Dann gilt $h(p) = g \circ C \circ INC \circ f(p) = 0^\omega$.
Sei p konvergent. Dann gibt es eine Zahl n mit $p(m) = n$ für fast alle $m \in \mathbb{N}$.
Dann gibt es ein minimales y mit $f(p)(k) \neq \langle 0, y\rangle$ für alle $k \in \mathbb{N}$.
Dann folgt $C \circ INC \circ f(p)\langle 0, y\rangle = 1$ und $C \circ INC \circ f(p)\langle 0, z\rangle = 0$ für alle $z < y$.
Dann gilt $h(p) = g \circ C \circ INC \circ f(p) = 0^{\langle n,y\rangle}(n+1)0^\omega$. □

Lemma 24 (Berechnung einer Folge mit dem Grenzwert von positiven Werten)**.**

Es gibt eine C-stetige Funktion $h' : \mathbb{B} \to \mathbb{B}$ *mit* $h'(p) = q$ *und*

$$(\exists i) q = 0^i(n+1)0^\omega, \textit{ falls } \lim_{m\to\infty, p(m)>0} p(m) = n \textit{ existiert}$$

Beweis. Definiere $B : \mathbb{B} \to \mathbb{B}$ durch $B(p)(0) := p(0)$ und

$$B(p)(n+1) := \begin{cases} B(p)(n) & \text{falls } p(n+1) = 0 \\ p(n+1) & \text{sonst} \end{cases}$$

Dann hat $h' := h \circ B$ mit der Funktion h aus Lemma 23 die geforderte Eigenschaft und ist C-stetig. □

Lemma 25 (Berechnung von Folgen mit den Grenzwerten)**.**
Es gibt eine C-stetige Funktion $h : \mathbb{B} \to \mathbb{B}$ *mit* $f(p) = q$ *und*

1. $\langle q\rangle_j = 0^\omega$, *falls* $\lim_{m\to\infty} p\langle j, m\rangle = div$

2. $(\exists i)\langle q\rangle_j = 0^i(n+1)0^\omega$, *falls* $\lim_{m\to\infty} p\langle j, m\rangle = n$ *existiert*

für alle $j \in \mathbb{N}$.

Beweis. Definiere $f : \mathbb{B} \to \mathbb{B}$ durch $f(p)\langle i, 0\rangle := \langle i, 0, 0\rangle$ und

$$f(p)\langle i, j+1\rangle := \begin{cases} \langle i, p\langle i, j\rangle + 1, \pi_3^{(3)}(f(p)\langle i, j\rangle)\rangle & \text{falls } p\langle i, j\rangle = p\langle i, j+1\rangle \\ \langle i, 0, \pi_3^{(3)}(f(p)\langle i, j\rangle) + 1\rangle & \text{sonst} \end{cases}$$

Definiere $g : \mathbb{B} \to \mathbb{B}$ durch

$$g(p)\langle i, a, n\rangle := \begin{cases} a & \text{falls } p\langle i, 0, n\rangle = 0 \text{ und } p\langle i, 0, n+1\rangle = 1 \\ & \text{und } p\langle i, a, n\rangle = 0 \text{ und } a > 0 \\ 0 & \text{sonst} \end{cases}$$

f und g sind stetig.
Definiere $h : \mathbb{B} \to \mathbb{B}$ durch $h(p) := g \circ C \circ INC \circ f(p)$ für alle $p \in \mathbb{B}$.
Dann hat h die Eigenschaften des Lemmas.
Beweis:
Sei p_i divergent. Dann existiert für jedes n eine Zahl m mit $f(p)\langle i, m\rangle = \langle i, 0, n\rangle$. Somit gilt $C \circ INC \circ f(p)\langle i, 0, n\rangle = 0$ für alle $n \in \mathbb{N}$.
Somit folgt $\langle h(p)\rangle_i = \langle g \circ C \circ INC \circ f(p)\rangle_i = 0^\omega$.
Sei p_i konvergent. Dann gibt es eine Zahl n mit $p\langle i, m\rangle = n$ für fast alle $m \in \mathbb{N}$.
Dann gibt es ein minimales y mit $f(p)\langle i, k\rangle \neq \langle i, 0, y\rangle$ für alle $k \in \mathbb{N}$.
Dann gilt $C \circ INC \circ f(p)\langle i, 0, y\rangle = 1$ und $C \circ INC \circ f(p)\langle i, 0, z\rangle = 0$ für alle $z < y$.
Dann gilt $\langle h(p)\rangle_i = \langle g \circ C \circ INC \circ f(p)\rangle_i = 0^{\langle n, y\rangle}(n+1)0^\omega$. □

Lemma 26 (Gesamtfolge konvergiert)**.**
Sei $p \in \mathbb{B}$ eine Folge mit $(\forall m)\langle p\rangle_m \in 0^*\mathbb{N}0^\omega \cup 0^\omega$ und $\lim_{m\to\infty} \max(\langle p\rangle_m) = n > 0$ existiert. Dann gilt

$$\lim_{i\to\infty, p(i)\neq 0} p(i) = n.$$

Beweis. Falls

$$\lim_{i\to\infty} \max(\langle p\rangle_i) = n,$$

existiert, dann müssen die Werte ungleich 0 in den Folgen $\langle p\rangle_i$ von einer gewissen Stelle an alle gleich n sein. Das gilt dann auch für die gesamte Folge p. □

Lemma 27 (Berechnen eines Wertes aus einer Folge)**.**
Es gibt eine stetige Funktion $f :\subseteq \mathbb{B} \to \mathbb{B}$ mit der Eigenschaft:

$$f(p) = \begin{cases} a & \textit{falls } p \in 0^*(a+1)\mathbb{B} \\ div & \textit{sonst} \end{cases}$$

Beweis. Der Beweis ist trivial. □

Definition 78 ($KDIV'_n$)**.**
Definiere $KDIV'_n$ als die Menge von Funktionen $f_n :\subseteq \mathbb{B} \to \mathbb{B}$ mit

$$(\forall p \in \mathbb{B}) f_0(p) = p \textit{ und}$$

$$(\forall p \in \mathbb{B})(a = \lim_{(k_1,\ldots,k_n)\to\infty} p \Rightarrow (\exists i) f_n(p) = 0^i a 0^\omega)$$

Lemma 28. *Für alle $n \in \mathbb{N}$ gibt es eine C^n-stetige Funktion $f \in KDIV'_n$.*

Beweis. Der Beweis wird mit vollständiger Induktion durchgeführt.
$\underline{n = 0}$:
$f \in KDIV'_0 \Rightarrow f = id$. Dann ist f stetig und somit C^0-stetig.
$\underline{n \to n+1}$:
Sei

$$\lim_{(k_1,\ldots,k_{n+1})\to\infty} (p) = a$$

Wenn $n = 0$, dann gibt es nach Lemma 23 eine C-stetige Funktion $f \in KDIV'_1$.
Wenn $n > 0$ dann gibt es nach Induktionsannahme eine C^n-stetige Funktion $g \in KDIV'_n$.
Sei $g' :\subseteq \mathbb{B} \longrightarrow \mathbb{B}$ definiert durch $g'(p_0, p_1, \ldots) := (g(p_0), g(p_1), \ldots)$.
Nach Lemma 19 ist mit g auch g' C^n-stetig.
$q := g'(p) = (q_0, q_1, \ldots)$ ist eine Folge von Folgen mit der Eigenschaft

$$\lim_{(k_2,\ldots,k_{n+1})\to\infty} (p_i) = b \Rightarrow (\exists j) q_i = 0^j(b+1)0^\omega$$

Nach Lemma 26 gilt

$$\lim_{i\to\infty, q(i)>0} q(i) = a + 1$$

Nach Lemma 24 gibt es eine C-stetige Funktion $h : \mathbb{B} \to \mathbb{B}$ mit $h(q) = q'$ mit

$$(\exists j) q' = 0^j(a+1)0^\omega$$

Definiere f durch $f := h \circ g'$. Dann gilt $f \in KDIV'_{n+1}$ und f ist C^{n+1}-stetig. □

Satz 68. *$KDIV_n^{\mathbb{N}}$ ist C^n-stetig.*

Beweis. Es gibt eine C^n-stetige Funktion $h \in KDIV'_n$.
$KDIV_n^{\mathbb{N}}(p) = f \circ h(p)$ für alle $p \in Def(KDIV_n^{\mathbb{N}})$ mit der stetigen Funktion f von Lemma 27.
Dann ist $KDIV_n^{\mathbb{N}}$ auch C^n-stetig. □

Lemma 29. *Falls $f :\subseteq \mathbb{B} \to \mathbb{B}$, $f = A_n \circ C \circ \ldots A_1 \circ C \circ A_0$, mit stetigen Funktionen $A_0, \ldots, A_n$, dann gibt es eine stetige Funktion B mit*

$$f(p)(i) = \lim_{(k_n,\ldots,k_1)\to\infty} B(p)\langle i, k_n, \ldots, k_1\rangle.$$

Beweis. Der Beweis wird durch vollständige Induktion geführt.
$\underline{n = 0}$: f ist stetig und $f(p)(i) = \lim_{()\to\infty} f(p)\langle i\rangle$.
$\underline{n = 1}$: Dies entspricht Lemma 22.
$\underline{n \to n+1}$: Sei $f = A_{n+1} \circ C \circ \ldots A_1 \circ C \circ A_0$.
Dann ist $g = A_n \circ C \circ \ldots A_1 \circ C \circ A_0$ C^n-stetig. Nach Induktionsannahme gibt es eine stetige Funktion B mit

$$g(p)(i) = \lim_{(k_n,\ldots,k_1)\to\infty} B(p)\langle i, k_n, \ldots, k_1\rangle.$$

Definiere $C' : \mathbb{B} \to \mathbb{B}$ durch

$$C'(q)\langle i,j\rangle := \begin{cases} 0 & \text{falls } (\exists k \leq j) q(k) = i+1 \\ 1 & \text{sonst} \end{cases}$$

Definiere A' durch

$$A'(q)\langle i,j\rangle := A_{n+1}(q\langle 0,j\rangle, q\langle 1,j\rangle, \ldots)(i)$$

Definiere $B' := A' \circ C' \circ B$. Dann gilt

$$\begin{aligned} f(p)(i) &= \lim_{j\to\infty} (A' \circ C' \circ g(p))\langle i,j\rangle \\ &= \lim_{(k_{n+1},\ldots,k_1)\to\infty} (A' \circ C' \circ B(p))\langle i, k_{n+1}, \ldots, k_1\rangle \\ &= \lim_{(k_{n+1},\ldots,k_1)\to\infty} (B'(p))\langle i, k_{n+1}, \ldots, k_1\rangle \end{aligned}$$

□

Satz 69. *Jede C^n-stetige Funktion $f :\subseteq \mathbb{B} \to \mathbb{N}$ ist 0-reduzierbar auf $KDIV_n^{\mathbb{N}}$.*

Beweis. Sei $f :\subseteq \mathbb{B} \to \mathbb{N}$ eine C^n-stetige Funktion.
Dann gibt es stetige Funktionen $A_0, \ldots, A_{n-1} :\subseteq \mathbb{B} \to \mathbb{B}$ und $A_n :\subseteq \mathbb{B} \to \mathbb{N}$ mit

$$f(p) = A_n \circ C \circ \ldots \circ A_1 \circ C \circ A_0(p).$$

Definiere $g :\subseteq \mathbb{B} \to \mathbb{B}$ durch

$$g(p) = id \circ C \circ \ldots \circ A_1 \circ C \circ A_0(p).$$

Dann gibt es nach Lemma 29 eine stetige Funktion B mit

$$g(p)(i) = \lim_{(k_n,\ldots,k_1)\to\infty} (B(p))\langle i, k_n, \ldots, k_1\rangle$$

Dann gilt

$$f(p) = \lim_{(k_n,\ldots,k_1)\to\infty} (A_n \circ B(p))\langle k_n, \ldots, k_1\rangle$$

Aus $f(p) \neq div$ folgt dann $f(p) = KDIV_n^{\mathbb{N}} \circ A_n \circ B(p)$.
Damit gilt $f \leq_0 KDIV_n^{\mathbb{N}}$. □

Die letzten beiden Sätze implizieren

Satz 70. *$KDIV_n^{\mathbb{N}}$ ist 0-vollständig für $[\mathbb{B} \longrightarrow \mathbb{N}]^n$.*

Analog können wir beweisen:

Satz 71. *$KDIV_n^{\{0,1\}}$ ist 0-vollständig für $[\mathbb{B} \longrightarrow \{0,1\}]^n$ für alle $n \in \mathbb{N}$.*

4.2.6 $KDIV^\infty$

Definition 79. *Sei* $KDIV^\infty :\subseteq \mathbb{B} \longrightarrow \mathbb{B}$ *definiert durch*

$$KDIV^\infty(p_0, p_1, \ldots) := (KDIV^{\mathbb{N}}(p_0), KDIV^{\mathbb{N}}(p_1), \ldots)$$

Satz 72. $KDIV^\infty$ *ist* C*-stetig.*

Beweis. Der Satz folgt aus Satz 68 und Lemma 19. □

Satz 73. $C \leq_0 KDIV^\infty$.

Beweis. Definiere $C' : \mathbb{B} \to \mathbb{B}$ durch

$$C'(p)\langle i, j\rangle := \begin{cases} 0 & \text{falls } (\exists n < j) p(n) = i + 1 \\ 1 & \text{sonst} \end{cases}$$

Dann ist C' stetig und für alle $i \in \mathbb{N}$ gilt

$$C(p)(i) = \lim_{n\to\infty} C'(p)\langle i, n\rangle$$

Somit gilt $C(p) = KDIV^\infty \circ C'(p)$ und damit $C \leq_0 KDIV^\infty$. □

Satz 74. $f \leq_1 KDIV^\infty$ *für jede* C*-stetige Funktion* $f :\subseteq \mathbb{B} \longrightarrow \mathbb{B}$.

Beweis. Falls $f :\subseteq \mathbb{B} \longrightarrow \mathbb{B}$ C-stetig ist, dann gibt es stetige Funktionen A und B mit $f(p) = A \circ C \circ B(p)$ für alle $p \in \mathbb{B}$.
Da C auf $KDIV^\infty$ 0-reduzierbar ist, folgt $f \leq_1 KDIV^\infty$ unmittelbar. □

Das impliziert

Satz 75. $KDIV^\infty$ *ist* 1*-vollständig für* $[\mathbb{B} \longrightarrow \mathbb{B}]^1$.

4.2.7 $KDIV_n^\infty$

Wir definieren $KDIV_n^\infty$ als Verallgemeinerung von $KDIV^\infty$ und betrachten Beziehungen zu $KDIV_n^{\mathbb{N}}$.

Definition 80 ($KDIV_n^\infty$)**.**
Definiere $KDIV_n^\infty :\subseteq \mathbb{B} \to \mathbb{B}$ *durch*
$KDIV_0^\infty(p) := p$ *und* $KDIV_1^\infty(p) := KDIV^\infty(p)$ *und*
$KDIV_n^\infty(p) := (KDIV_n^{\mathbb{N}}(p_0), KDIV_n^{\mathbb{N}}(p_1), \ldots)$ *für* $n > 1$.

Satz 76. *$KDIV_n^\infty$ ist C^n-stetig für alle $n \in \mathbb{N}$.*

Beweis. $KDIV_0^\infty$ ist stetig und $KDIV_1^\infty$ ist C-stetig.
$KDIV_n^\infty(p) = (KDIV_n^{\mathbb{N}}(p_0), KDIV_n^{\mathbb{N}}(p_1), \ldots)$ für $n > 1$.
Nach Satz 68 ist $KDIV_n^{\mathbb{N}}$ C^n-stetig.
Nach Lemma 19 ist $KDIV_n^\infty$ ebenfalls C^n-stetig. □

Lemma 30. *Für alle stetigen Funktionen $A :\subseteq \mathbb{B} \longrightarrow \mathbb{B}$ und $B :\subseteq \mathbb{B} \longrightarrow \mathbb{B}$ gibt es eine stetige Funktion $B^* : \mathbb{B} \longrightarrow \mathbb{B}$ mit*
$A \circ C \circ B(p)(i) = \lim_{n\to\infty} B^*(p)\langle i, n\rangle$.

Beweis. Für jede stetige Funktion $A :\subseteq \mathbb{B} \longrightarrow \mathbb{B}$ sei $\overline{A} : \mathbb{N}^* \longrightarrow \mathbb{N}^*$ die isotone Funktion.
Definiere die totale Funktion A' durch

$$A'(\varepsilon) := P(\overline{A}(\varepsilon))$$

und

$$A'(wx) := \begin{cases} A'(w) \cdot P(v) & \text{falls es ein } v \neq \varepsilon \text{ gibt mit } \overline{A}(wx) = \overline{A}(w) \cdot v \\ 0 & \text{sonst} \end{cases}$$

mit $P : \mathbb{N}^* \longrightarrow \mathbb{N}^*$ definiert durch $P(n_1, \ldots, n_k) := (n_1 + 1, \ldots, n_k + 1)$.
$A'(p)$ enthält alle Elemente von $A(p)$ in derselben Reihenfolge und um 1 erhöht, mit zusätzlichen Nullen in der Folge.
Definiere C' durch

$$C'(q)\langle i, n\rangle := \begin{cases} 0 & \text{falls } (\exists k \leq i) q(k) = n + 2 \\ 1 & \text{sonst} \end{cases}$$

Für jede stetige Funktion $A : \mathbb{B} \longrightarrow \mathbb{B}$ definiere A'' durch

$$A''(q)\langle i, n\rangle := \begin{cases} (\overline{A}(q\langle n, 0\rangle, \ldots))(i) & \text{falls } (\overline{A}(q\langle n, 0\rangle, \ldots))(i) \text{ existiert} \\ n & \text{sonst} \end{cases}$$

Definiere B^* durch $B^* := A'' \circ C' \circ B'$.
Dann gilt $A \circ C \circ B(p)(i) = k$
$\Longleftrightarrow \lim_{n\longrightarrow\infty} A'' \circ C' \circ B'(p)\langle i, n\rangle = k$
$\Longleftrightarrow \lim_{n\longrightarrow\infty} B^*(p)\langle i, n\rangle = k$. □

Satz 77. *Sei $f :\subseteq \mathbb{B} \to \mathbb{B}$ eine C^n-stetige Funktion.*
Dann gibt es eine stetige Funktion B mit
$f(p)(i) = \lim_{i_n\to\infty} \ldots \lim_{i_1\to\infty}(B(p)\langle i, i_n, \ldots, i_1\rangle)$.

Beweis. Der Beweis wird durch Induktion über n geführt.
$\underline{n = 0}$:
Falls f C^0-stetig ist, dann ist f stetig und
$f(p)(i) = \lim_{i_0 \to \infty} \ldots \lim_{i_1 \to \infty} B(p)(i)$ mit der stetigen Funktion $B := f$.
$\underline{n = 1}$:
Falls f C-stetig ist, dann gilt $f \leq_2 KDIV^\infty = KDIV_1^\infty$ nach Lemma 30.
$\underline{n \longrightarrow n + 1}$:
Sei f C^{n+1}-stetig. Dann gibt es eine C^1-stetige Funktion g_1 und eine C^n-stetige Funktion g_2 mit $f = g_1 \circ g_2$.
Es gibt eine stetige Funktion h_1 mit
$g_1(p)(i) = \lim_{i_1 \to \infty} h_1(p)\langle i, i_1 \rangle$.
Nach Annahme gibt es eine stetige Funktion h_2 mit
$h_1 \circ g_2(p)(j) = \lim_{i_n \to \infty} \ldots \lim_{i_1 \to \infty} h_2(p)\langle i, i_n, \ldots, i_1 \rangle$.
Dann gilt $f(p)(i)$
$= g_1 \circ g_2(p)(i)$
$= \lim_{i_{n+1} \to \infty} h_1 \circ g_2(p)\langle i, i_{n+1} \rangle$
$= \lim_{i_{n+1} \to \infty} \ldots \lim_{i_1 \to \infty} h_2(p)\langle \langle i, i_{n+1} \rangle, i_n, \ldots, i_1 \rangle$
$= \lim_{i_{n+1} \to \infty} \ldots \lim_{i_1 \to \infty} h_2(p)\langle i, i_{n+1}, i_n, \ldots, i_1 \rangle$.

□

Hieraus folgt unmittelbar der nächste Satz:

Satz 78. *$f \leq_1 KDIV_n^\infty$ für jede C^n-stetige Funktion f für alle $n \in \mathbb{N}$.*

Aus diesen Sätzen folgt:

Satz 79. *$KDIV_n^\infty$ ist 1-vollständig für $[\mathbb{B} \longrightarrow \mathbb{B}]^n$ für alle $n \in \mathbb{N}$.*

4.2.8 Logische Beschreibung von C^n-stetigen Funktionen

Satz 80. *Für jede C^m-stetige Funktion $f :\subseteq \mathbb{B} \to \mathbb{N}$ gibt es eine Funktion $g :\subseteq \mathbb{B} \to \mathbb{N}$ mit $f(p) = g(p)$ für alle $p \in Def(f)$ und*

$$g(p) = \begin{cases} n & \textit{falls } (\exists l_1)(\forall k_1 > l_1) \dots (\exists l_m)(\forall k_m > l_m) A(p)\langle k_1, \dots, k_m\rangle = n \\ div & \textit{sonst} \end{cases}$$

mit einer totalen stetigen Funktion $A : \mathbb{B} \to \mathbb{B}$.

Beweis. Nach Satz 70 ist $KDIV_n^{\mathbb{N}}$ 0-vollständig für $[\mathbb{B} \longrightarrow \mathbb{N}]^n$. Somit gibt es eine stetige Funktion $A : \mathbb{B} \longrightarrow \mathbb{B}$ mit

$$f(p) = KDIV_n^{\mathbb{N}} \circ A(p)$$

für alle $p \in Def(f)$. Das impliziert

$$f(p) = \begin{cases} n & \text{falls } (\exists l_1)(\forall k_1 > l_1) \dots (\exists l_m)(\forall k_m > l_m) A(p)\langle k_1, \dots, k_m\rangle = n \\ div & \text{sonst} \end{cases}$$

für alle $p \in Def(f)$. □

Satz 81. *Sei $A : \mathbb{B} \to \mathbb{B}$ eine totale stetige Funktion.*
Definiere $f :\subseteq \mathbb{B} \to \mathbb{N}$ durch

$$f(p) := \begin{cases} n & \textit{falls } (\exists l_1)(\forall k_1 > l_1) \dots (\exists l_m)(\forall k_m > l_m) A(p)\langle k_1, \dots, k_m\rangle = n \\ div & \textit{sonst} \end{cases}$$

Dann ist f C^m-stetig.

Beweis. Sei $f :\subseteq \mathbb{B} \to \mathbb{N}$ definiert durch

$$f(p) := \begin{cases} n & \text{falls } (\exists l_1)(\forall k_1 > l_1) \dots (\exists l_m)(\forall k_m > l_m) A(p)\langle k_1, \dots, k_m\rangle = n \\ div & \text{sonst} \end{cases}$$

Dann gilt $f = KDIV_m^{\mathbb{N}} \circ A$. Da $KDIV_m^{\mathbb{N}}$ C^m-stetig ist, folgt das auch für f. □

Man kann sagen, dass alle C^m-stetigen Funktionen durch diese logischen Ausdrücke beschrieben werden können. Nur die Definitionsbereiche der Funktionen können verschieden sein.

Diese Resultate können analog auf die anderen Arten von Funktionen übertragen werden:

Satz 82. *Jede Funktion $f :\subseteq \mathbb{B} \to \{0,1\}$ ist C^m-stetig, gdw. sie ausgedrückt werden kann durch*

$$f(p) = \begin{cases} 0 & \textit{falls } (\exists l_1)(\forall k_1 > l_1)\dots(\exists l_m)(\forall k_m > l_m) \\ & A(p)\langle k_1, \dots, k_m\rangle = 0 \\ 1 & \textit{falls } (\exists l_1)(\forall k_1 > l_1)\dots(\exists l_m)(\forall k_m > l_m) \\ & A(p)\langle k_1, \dots, k_m\rangle = n > 0 \\ div & \textit{sonst} \end{cases}$$

mit einer totalen stetigen Funktion $A : \mathbb{B} \to \mathbb{B}$.

Satz 83. *Jede Funktion $f :\subseteq \mathbb{B} \to \mathbb{B}$ ist C^m-stetig, gdw. sie ausgedrückt werden kann durch*

$$f(p)(i) = \begin{cases} n & \textit{falls } (\exists k_1)(\forall l_1 > k_1)\dots(\exists k_m)(\forall l_m > k_m) \\ & A(p)\langle i, l_1, \dots, l_m\rangle = n \\ div & \textit{sonst} \end{cases}$$

mit einer totalen stetigen Funktion $A : \mathbb{B} \to \mathbb{N}$.

Die Beweise sind ähnlich zu den Beweisen zuvor.

4.2.9 Weitere Klassen in der C^n-Hierarchie

Die logische Beschreibung der Klassen der C^n-stetigen Funktionen führt zu der Idee, andere logische Definitionen zu probieren und sie miteinander zu vergleichen.

Wir werden nur logische Ausdrücke mit dem $\forall$-Quantor am Ende benutzen. Zum Beispiel definiert

$$f(p) := \begin{cases} n & \text{falls } (\exists k)(\forall l > k)p(l) = n \\ div & \text{sonst} \end{cases}$$

eine Funktion mit einem eindeutigen Funktionswert, dem Grenzwert der Folge p, falls er existiert.
Falls wir den $\exists$-Quantor am Ende verwenden, erhalten wir mehrwertige Funktionen.

$$g(p) := \begin{cases} n & \text{falls } (\forall k)(\exists l > k)p(l) = n \\ div & \text{sonst} \end{cases}$$

berechnet z.B. die Menge aller Zahlen, die unendlich oft in p vorkommen. Es kann mehr als eine Zahl mit dieser Eigenschaft geben.

Definition 81.
Sei $C^0_{\forall,div}$ die Menge aller Funktionen $f :\subseteq \mathbb{B} \to \mathbb{N}$ mit

$$f(p) := \begin{cases} m+1 & \textit{falls } A(p) = m \\ div & \textit{sonst} \end{cases}$$

mit einer stetigen Funktion $A : \mathbb{B} \longrightarrow \mathbb{N}$.

Definition 82.
Sei $C^0_{\forall,tot}$ die Menge aller Funktionen $f : \mathbb{B} \to \mathbb{N}$ mit

$$f(p) := \begin{cases} m+1 & \textit{falls } A(p) = m \\ 0 & \textit{sonst} \end{cases}$$

mit einer stetigen Funktion $A : \mathbb{B} \longrightarrow \mathbb{N}$.

Definition 83.
Für alle geraden $n > 0$ sei $C^n_{\forall,div}$ die Menge aller Funktionen $f :\subseteq \mathbb{B} \to \mathbb{N}$ mit

$$f(p) := \begin{cases} m+1 & \textit{falls } (\exists k_1)(\forall k_2 > k_1)\dots(\exists k_{n-1})(\forall k_n > k_{n-1}) \\ & A(p)\langle k_2, k_4, \dots, k_n\rangle = m \\ div & \textit{sonst} \end{cases}$$

mit einer stetigen Funktion $A : \mathbb{B} \longrightarrow \mathbb{B}$.

Definition 84.
Für alle geraden $n > 0$ sei $C^n_{\forall,tot}$ die Menge aller Funktionen $f : \mathbb{B} \to \mathbb{N}$ mit

$$f(p) := \begin{cases} m+1 & \textit{falls } (\exists k_1)(\forall k_2 > k_1)\dots(\exists k_{n-1})(\forall k_n > k_{n-1}) \\ & A(p)\langle k_2, k_4, \dots, k_n\rangle = m \\ 0 & \textit{sonst} \end{cases}$$

mit einer stetigen Funktion $A : \mathbb{B} \longrightarrow \mathbb{B}$.

Definition 85.
Für alle ungeraden n sei $C^n_{\forall,div}$ die Menge aller Funktionen $f :\subseteq \mathbb{B} \to \mathbb{N}$ mit

$$f(p) := \begin{cases} m+1 & \textit{falls } (\forall k_1)(\exists k_2)(\forall k_3 > k_2)\ldots(\exists k_{n-1})(\forall k_n > k_{n-1}) \\ & A(p)\langle k_1, k_3, \ldots, l_n\rangle = m \\ div & \textit{sonst} \end{cases}$$

mit einer stetigen Funktion $A : \mathbb{B} \longrightarrow \mathbb{B}$.

Definition 86.
Für alle ungeraden n sei $C^n_{\forall,tot}$ die Menge aller Funktionen $f : \mathbb{B} \to \mathbb{N}$ mit

$$f(p) := \begin{cases} m+1 & \textit{falls } (\forall k_1)(\exists k_2)(\forall k_3 > k_2)\ldots(\exists k_{n-1})(\forall k_n > k_{n-1}) \\ & A(p)\langle k_1, k_3, \ldots, l_n\rangle = m \\ 0 & \textit{sonst} \end{cases}$$

mit einer stetigen Funktion $A : \mathbb{B} \longrightarrow \mathbb{B}$.

Satz 84.
Eine Funktion $f :\subseteq \mathbb{B} \longrightarrow \mathbb{N}$ ist stetig, gdw. $f \leq_2 g$ für eine Funktion $g \in C^0_{\forall,div}$.

Beweis.
$\Longrightarrow$: Sei f stetig. Dann ist die Funktion g, definiert durch

$$g(p) := \begin{cases} n+1 & \text{falls } f(p) = n \\ div & \text{sonst,} \end{cases}$$

in $C^0_{\forall,div}$. Dann gilt $f = DEC \circ g$ und damit $f \leq_2 g$.
$\Longleftarrow$: Sei $f \leq_2 g$ für $g \in C^0_{\forall,div}$. Dann ist g stetig und damit auch f. □

Satz 85.
Eine Funktion $f :\subseteq \mathbb{B} \longrightarrow \mathbb{N}$ ist Ω-stetig, gdw. $f \leq_2 g$ für eine Funktion $g \in C^0_{\forall,tot}$.

Beweis.
$\Longrightarrow$: Sei f Ω-stetig. Dann gilt $f(p) = A(p, \Omega \circ B(p))$ mit stetigen Funktionen A und B.
Definiere D durch

$$D(p) := \begin{cases} 0 & \text{falls } (\exists k)p(k) \neq 0 \\ div & \text{sonst} \end{cases}$$

D ist stetig.
Dann ist g, definiert durch

$$g(p) := \begin{cases} 1 & \text{falls } D(p) = 0 \\ 0 & \text{sonst,} \end{cases}$$

in $C^0_{\forall,tot}$.
Dann gilt $\Omega(p) = 0 \Rightarrow (\forall k)p(k) = 0 \Rightarrow D(p) = div \Rightarrow g(p) = 0$
und $\Omega(p) = 1 \Rightarrow (\exists k)p(k) \neq 0 \Rightarrow D(p) = 0 \Rightarrow g(p) = 1$.
Somit gilt $\Omega(p) = g(p)$ und das impliziert $f \leq_2 g$.

$\Longleftarrow$: Sei $f \leq_2 g$ für $g \in C^0_{\forall,tot}$. Dann gilt $f(p) = A(p, g \circ B(p))$ mit stetigen Funktionen A und B und

$$g(p) = \begin{cases} m+1 & \text{falls } A'(p) = m \\ 0 & \text{sonst} \end{cases}$$

mit einer stetigen Funktion A'.
Nach dem Warteschleifenlemma[20] gibt es eine stetige Funktion D mit

$$D(p) = 0^\omega \Leftrightarrow A'(p) = div.$$

Dann gilt $\Omega \circ D(p) = 0 \Leftrightarrow g(p) = 0$.
Definiere X durch

$$X(p,k) := \begin{cases} A(p,0) & \text{falls } k = 0 \\ A(p, A' \circ B(p) + 1) & \text{sonst} \end{cases}$$

Dann gilt $g \circ B(p) = 0$
$\Rightarrow A' \circ B(p) = div$
$\Rightarrow D \circ B(p) = 0^\omega$
$\Rightarrow \Omega \circ D \circ B(p) = 0$
$\Rightarrow X(p, \Omega \circ D \circ B(p)) = X(p,0) = A(p,0) = A(p, g \circ B(p))$
und $g \circ B(p) = m+1$
$\Rightarrow A' \circ B(p) = m$
$\Rightarrow D \circ B(p) \neq 0^\omega$
$\Rightarrow \Omega \circ D \circ B(p) = 1$
$\Rightarrow X(p, \Omega \circ D \circ B(p)) = X(p,1) = A(p, A' \circ B(p) + 1) = A(p, g \circ B(p))$.
Somit gilt $f(p) = A(p, g \circ B(p)) = X(p, \Omega \circ D \circ B(p))$. Also ist f Ω-stetig. □

Satz 86.
Eine Funktion $f :\subseteq \mathbb{B} \longrightarrow \mathbb{N}$ ist C-stetig, gdw. $f \leq_2 g$ für eine Funktion $g \in C^2_{\forall,div}$.

Beweis.
$\Longrightarrow$: Sei f C-stetig. Nach Satz 80 gibt es eine totale stetige Funktion A mit

$$f(p) = \begin{cases} n & \text{falls } (\exists k)(\forall l > k) A(p)(l) = n \\ div & \text{sonst} \end{cases}$$

Definiere die Funktion g durch

$$g(p) = \begin{cases} n+1 & \text{falls } (\exists k)(\forall l > k) A(p)(l) = n \\ div & \text{sonst} \end{cases}$$

Dann gilt $f = DEC \circ g$.
$\Longleftarrow$:
Sei g definiert durch

$$g(p) = \begin{cases} n+1 & \text{falls } (\exists k)(\forall l > k) A'(p)(l) = n \\ div & \text{sonst} \end{cases}$$

[20] [St89] Seite 63

mit einer stetigen Funktion A', und gelte $f(p) = A(p, g \circ B(p))$ mit stetigen Funktionen A und B.
Nach Satz 80 ist die Funktion h, definiert durch

$$h(p) = \begin{cases} n & \text{falls } (\exists k)(\forall l > k) A'(p)(l) = n \\ div & \text{sonst} \end{cases}$$

C-stetig. Also gilt $f(p) = A(p, g \circ B(p)) = A(p, 1 + h \circ B(p))$. Damit ist die Funktion f C-stetig. □

Für die nächsten Sätze brauchen wir Variationen der Grenzwertberechnung.

Definition 87. *Für $n > 0$ definiere lim_n durch $lim_n := KDIV_n^{\mathbb{N}} + 1$,*

$$lim_n(p) := \begin{cases} m + 1 & \textit{falls } lim_{k_1 \to \infty} \dots lim_{k_n \to \infty} p(n) = m \\ div & \textit{sonst} \end{cases},$$

$\overline{lim_n}$ durch

$$\overline{lim_n}(p) := \begin{cases} m & \textit{falls } lim_n(p) = m \\ 0 & \textit{sonst} \end{cases},$$

Lim_n durch

$$Lim_n(p) := \begin{cases} m & \textit{falls } lim_n(p_0) = lim_n(p_1) = \dots = m \\ div & \textit{sonst} \end{cases}$$

und $\overline{Lim_n}$ durch

$$\begin{aligned} \overline{Lim_n}(p) &= \begin{cases} m & \textit{falls } lim_n(p_0) = lim_n(p_1) = \dots = m \\ 0 & \textit{sonst} \end{cases} \\ &= \begin{cases} m & \textit{falls } Lim_n(p) = m \\ 0 & \textit{sonst} \end{cases} \end{aligned}$$

Für $n = 0$ definiere lim_0 durch

$$lim_0(p) := p(0) + 1$$

$\overline{lim_0}$ durch

$$\overline{lim_0}(p) := p(0) + 1$$

Lim_0 durch

$$Lim_0(p) := \begin{cases} m + 1 & \textit{falls } p(0) = p(1) = \dots = m \\ div & \textit{sonst} \end{cases}$$

und $\overline{Lim_0}$ durch

$$\begin{aligned} \overline{Lim_0}(p) &= \begin{cases} m + 1 & \textit{falls } p(0) = p(1) = \dots = m \\ 0 & \textit{sonst} \end{cases} \\ &= \begin{cases} m & \textit{falls } Lim_0(p) = m \\ 0 & \textit{sonst} \end{cases} \end{aligned}$$

Satz 87 (Funktionen in $C^n_{\forall,div}$ oder $C^n_{\forall,tot}$).
lim_n ist 0-vollständig für $C^{2n}_{\forall,div}$ für alle n.
$\overline{lim}_n$ ist 0-vollständig für $C^{2n}_{\forall,tot}$ für alle n.
Lim_n ist 0-vollständig für $C^{2n+1}_{\forall,div}$ für alle n.
$\overline{Lim}_n$ ist 0-vollständig für $C^{2n+1}_{\forall,tot}$ für alle n.

Beweis. Der Beweis ist einfach und wird deshalb ausgelassen. □

Satz 88.
Eine Funktion $f :\subseteq \mathbb{B} \longrightarrow \mathbb{N}$ ist C^n-stetig, gdw. $f \leq_2 lim_n$.

Beweis.
Es ist leicht zu sehen, dass $lim_n \equiv_2 KDIV^{\mathbb{N}}_n$. Somit ist lim_n 2-vollständig für die Klasse der C^n-stetigen Funktionen. □

Die Beziehungen zwischen lim_n, Lim_n, $\overline{lim}_n$ und $\overline{Lim}_n$ sind hier graphisch dargestellt und sollen im Folgenden bewiesen werden.

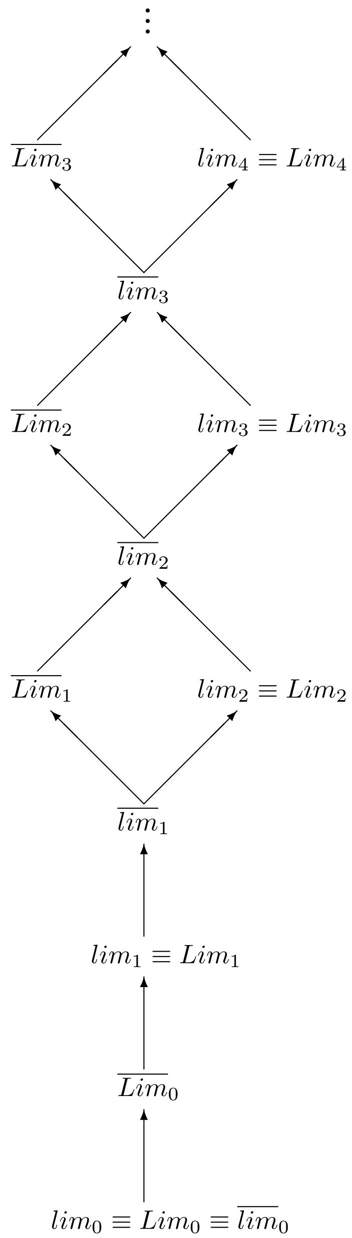

Abbildung 12: $\leq_2$-Reduzierbarkeit zwischen lim_n, Lim_n, $\overline{lim_n}$ und $\overline{Lim_n}$

Satz 89. *$lim_n \equiv_2 Lim_n$ für alle $n \in \mathbb{N}$.*

Beweis.
$\underline{lim_n \leq_2 Lim_n}$:
Definiere B durch $B(p) := q$ mit $q_i := p$ für alle $i \in \mathbb{N}$. Dann gilt $lim_n(p) = Lim_n \circ B(p)$ für alle $p \in \mathbb{B}$.

$\underline{Lim_n \leq_2 lim_n}$:
Definiere B durch $B(p) := p_0$. Wenn $Lim_n(p) = m$, dann gilt $lim_n(p_0) = m$ für alle $p \in \mathbb{B}$.
Somit gilt $Lim_n(p) = lim_n \circ B(p)$ für alle $p \in Def(Lim_n)$. ☐

Satz 90. *$lim_n \lneqq_2 \overline{lim}_n$ für alle $n \in \mathbb{N}$.*

Beweis.
$\underline{lim_n \leq_2 \overline{lim}_n}$: Der Beweis ist einfach.
$\underline{\overline{lim}_n \not\leq_2 lim_n}$:
Zuerst zeigen wir $K^n_\chi \leq_2 \overline{lim}_n$[21]. Die universelle Funktion U^n_χ ist C^n-stetig. Also ist die Funktion $U0^n_\chi$, definiert durch

$$U0^n_\chi(p) := \begin{cases} 0 & \text{falls } U^n_\chi(p) \text{ existiert} \\ div & \text{sonst,} \end{cases}$$

auch C^n-stetig.
Nach Satz 82 gibt es eine stetige Funktion A mit

$$U0^n_\chi(p) = \begin{cases} 0 & \text{falls } (\exists l_1)(\forall k_1 > l_1)\ldots(\exists l_n)(\forall k_n > l_n)A(p)\langle k_1, \ldots, k_m\rangle = 0 \\ div & \text{sonst} \end{cases}$$

Dann gilt

$$U0^n_\chi(p)+1 = \begin{cases} 1 & \text{falls } (\exists l_1)(\forall k_1 > l_1)\ldots(\exists l_n)(\forall k_n > l_n)A(p)\langle k_1, \ldots, k_m\rangle = 0 \\ div & \text{sonst} \end{cases}$$

Somit gilt $U0^n_\chi(p)+1 = lim_n(A(p))$. Das impliziert $K^n_\chi(p) = NEG \circ \overline{lim}_n \circ A(p)$.
Dann würde die Annahme $\overline{lim}_n \leq_2 lim_n$ implizieren, dass $K^n_\chi \leq_2 \overline{lim}_n \leq_2 lim_n$ gilt.
Das würde implizieren, dass K^n_χ C^n-stetig ist. Aber das ist nicht wahr nach Satz 5.1 in [Myl92][22]. ☐

Satz 91. *$\overline{lim}_n \lneqq_2 lim_{n+1}$ für alle $n \in \mathbb{N}$.*

Beweis.
$\underline{\overline{lim}_n \leq_2 lim_{n+1}}$:
Da lim_n C^n-stetig ist, gibt es nach dem Warteschleifenlemma[23] eine C^n-stetige Funktion D mit

$$D(p) = \begin{cases} 0^\omega & \text{falls } lim_n(p) = div \\ 0^i(m+1)0^\omega & \text{falls } lim_n(p) = m \text{ für ein } i \in \mathbb{N} \end{cases}$$

[21] siehe Definition 66
[22] [Myl92] Seie 62
[23] [St89] Seite 63

Definiere B durch

$$B(p)(2n) = \begin{cases} 1 & \text{falls } p(n) > 0 \\ 0 & \text{sonst} \end{cases}$$

und

$$B(p)(2n+1) = \begin{cases} p(n)+1 & \text{falls } p(n) > 0 \\ 0 & \text{sonst} \end{cases}$$

Definiere A durch

$$A(q) = \begin{cases} 0 & \text{falls } p(0) = 1 \\ p(k)-1 & \text{falls } k = \min\{i \mid i > 0, p(i) > 0\} \text{ existiert} \\ div & \text{sonst} \end{cases}$$

Dann gilt $\overline{lim}_n = A \circ C \circ B \circ D$.
Beweis:
$\overline{lim}_n(p) = 0$
$\Rightarrow lim_n(p) = div$
$\Rightarrow D(p) = 0^\omega$
$\Rightarrow B \circ D(p) = 0^\omega$
$\Rightarrow C \circ B \circ D(p) = 1^\omega$
$\Rightarrow A \circ C \circ B \circ D(p) = 0$
und $\overline{lim}_n(p) = m+1$
$\Rightarrow lim_n(p) = m+1$
$\Rightarrow (\exists i) D(p) = 0^i(m+2)0^\omega$
$\Rightarrow (\exists i) B \circ D(p) = 0^{2i}1(m+3)0^\omega$
$\Rightarrow C \circ B \circ D(p)(0) = 0 = C \circ B \circ D(p)(m+2)$
und $(\forall j \notin \{0, m+2\}) C \circ B \circ D(p)(j) = 1$
$\Rightarrow A \circ C \circ B \circ D(p) = m+1$
Da D C^n-stetig ist, ist $A \circ C \circ B \circ D$ C^{n+1}-stetig. Somit ist $\overline{lim}_n$ C^{n+1}-stetig.
Das impliziert $\overline{lim}_n \leq_2 lim_{n+1}$.
$lim_{n+1} \not\leq_2 \overline{lim}_n$:
Annahme: $lim_{n+1} \leq_2 \overline{lim}_n$.
Dann gibt es stetige Funktionen A und B mit

$$lim_{n+1}(p) = A(p, \overline{lim}_n \circ B(p)) \text{ für alle } p \in \mathbb{B}.$$

Wenn es keine Folge p gibt mit $\overline{lim}_n \circ B(p) = 0$, dann gilt $lim_{n+1}(p) = A(p, lim_n \circ B(p))$. Dies gilt, weil $lim_n(p) = \overline{lim}_n(p)$ für alle konvergierenden Folgen p. Aber das impliziert $lim_{n+1} \leq_2 lim_n$. Da lim_{n+1} C^{n+1}-vollständig und lim_n C^n-stetig ist, ist das nicht möglich.
Somit gibt es eine Folge p mit $\overline{lim}_n \circ B(p) = 0$.
Falls $\overline{lim}_n \circ B(p) = 0$ nur für Folgen p mit $lim_{n+1}(p) = div$ gelten würde, dann würde auch $lim_{n+1}(p) = A(p, lim_n \circ B(p))$ gelten.
Also muss es Folgen p geben mit $lim_{n+1}(p) \neq div$ und $\overline{lim}_n \circ B(p) = 0$.
Nehme an, es gibt eine Zahl x und eine Folge $\overline{p}$ mit

$$lim_{n+1}(\overline{p}) = x \text{ und } \overline{lim}_n \circ B(\overline{p}) = 0.$$

Dann gilt

$$lim_{n+1}(\overline{p}) = x = A(\overline{p}, \overline{lim}_n \circ B(\overline{p})) = A(\overline{p}, 0)$$

Da A stetig ist, gibt es eine Zahl K mit

$$A(p,0) = x \text{ für alle } p \text{ mit } [[\overline{p}]]_K \sqsubseteq p$$

Definiere $* : \mathbb{N}^* \times \mathbb{N}^\omega \longrightarrow \mathbb{N}^\omega$ durch

$$w * p(n) = \begin{cases} w(n) & \text{falls } n < lg(w) \\ p(n) & \text{sonst} \end{cases}$$

Definiere $w := [[\overline{p}]]_K$ und A' durch $A'(p,a) := A(w*p,a)$ und B' durch $B'(p) := w*p$.
Dann gilt $A'(p, \overline{lim}_n \circ B \circ B'(p)) = A(w*p, \overline{lim}_n \circ B(w*p)) = A(p, \overline{lim}_n \circ B(p))$, da die Ersetzung eines endlichen Präfixes nicht das Konvergenzverhalten der Folge ändern kann.
Definiere weiter B'' durch $B''(p)(i) := p(i) + x + 1$ und A^* durch $A^*(p,a) := A(w*q,a) - x - 1$ mit $q := (p(0)+x+1, p(1)+x+1, \ldots)$.
Definiere B^* durch $B^* := B \circ B' \circ B''$.
Dann gilt $A^*(p, \overline{lim}_n \circ B^*(p))$
$= A(w*q, \overline{lim}_n \circ B \circ B' \circ B''(p)) - x - 1$
$= A(w*q, \overline{lim}_n \circ B \circ B'(q)) - x - 1$
$= A(w*q, \overline{lim}_n \circ B(w*q)) - x - 1$
$= A(q, \overline{lim}_n \circ B(q)) - x - 1$ (ein endliches Präfix ändert nicht den Grenzwert)
$= A(p, \overline{lim}_n \circ B(p))$ (da $q(i) = p(i) + x + 1$ für alle i).
Die Hauptsache ist, dass $\overline{lim}_n \circ B^*(p) = 0$ nicht möglich ist. Also gilt $lim_{n+1}(p) = A^*(p, \overline{lim}_n \circ B^*(p)) = A^*(p, lim_n \circ B^*(p))$.
Somit würde wieder $lim_{n+1} \leq_2 lim_n$ impliziert. Und das ist ein Widerspruch. □

Lemma 31. *$K_\psi^n \leq_2 \overline{Lim}_n$ für alle $n \geq 0$.*

Proof. $K_\psi^n(p) = 0$
$\Longleftrightarrow U_\psi^n\langle p,p\rangle$ existiert
$\Longleftrightarrow (\forall i) U_\psi^n\langle p,p\rangle(i)$ existiert
$\Longleftrightarrow (\forall i)(\exists m) U_\psi^n\langle p,p\rangle(i) = m$
$\Longleftrightarrow (\forall i)(\exists m)(\exists k_1)(\forall l_1 > k_1)\ldots(\exists k_n)(\forall l_n > k_n) A(p)\langle i, l_1, \ldots, l_n\rangle = m$ [24]
$\Longleftrightarrow (\forall i)(\exists k_1)(\forall l_1 > k_1)\ldots(\exists k_n)(\forall l_n > k_n) A(p)\langle i, l_1, \ldots, l_n\rangle = \pi_1(k_1)$
$\Longleftrightarrow (\forall i)(\exists k_1)(\forall l_1 > k_1)\ldots(\exists k_n)(\forall l_n > k_n) A'(p)\langle i, l_1, \ldots, l_n\rangle = 0$
$\Longleftrightarrow \overline{Lim}_n \circ A'(p) = 1$
$\Longleftrightarrow NEG \circ \overline{Lim}_n \circ A'(p) = 0$,
wobei A' definiert ist durch

$$A'(p)\langle i, l_1, \ldots, l_n\rangle := \begin{cases} 0 & \text{falls } A(p)\langle i, l_1, \ldots, l_n\rangle \text{ existiert} \\ div & \text{sonst} \end{cases}$$

und damit $K_\psi^n(p) = 1 \Leftrightarrow NEG \circ \overline{Lim}_n \circ A'(p) = 1$. □

Satz 92. $\overline{Lim}_0 \leq_2 lim_1$.

[24] mit stetigem A

Beweis.
Definiere $DEC : \mathbb{N} \longrightarrow \mathbb{N}$ durch

$$DEC(k) := \begin{cases} k-1 & \text{falls } k > 0 \\ 0 & \text{sonst} \end{cases}$$

und $B : \mathbb{B} \longrightarrow \mathbb{B}$ durch $B(p)(0) := p(0) + 1$ und

$$B(p)(n+1) := \begin{cases} 0 & \text{falls } B(p)(n) = 0 \text{ oder } p(n) \neq p(n+1) \\ B(p)(n) & \text{sonst} \end{cases}$$

Dann gilt $\overline{Lim}_0(p) = m + 1$
$\Longrightarrow (\forall k)p(k) = m$
$\Longrightarrow (\forall k)B(p)(k) = m + 1$
$\Longrightarrow lim_1 \circ B(p)(k) = m + 2$
$\Longrightarrow DEC \circ lim_1 \circ B(p) = m + 1$
und $\overline{Lim}_0(p) = 0$
$\Longrightarrow (\exists k)p(k) \neq p(k+1)$
$\Longrightarrow (\exists k)(\forall l > k)B(p)(l) = 0$
$\Longrightarrow lim_1 \circ B(p) = 1$
$\Longrightarrow DEC \circ lim_1 \circ B(p) = 0.$
Also gilt $\overline{Lim}_0 = DEC \circ lim_1 \circ B$. Das impliziert $\overline{Lim}_0 \leq_2 lim_1$. □

Satz 93. *$\overline{Lim}_n \not\leq_2 lim_{n+1}$ für alle $n > 0$.*

Beweis.
$K_\psi^n \leq_2 \overline{Lim}_n$ nach Lemma 31. K_ψ^n ist C^{n+2}-stetig, aber nicht C^{n+1}-stetig. Also würde $\overline{Lim}_n \leq_2 lim_{n+1} \Longrightarrow K_\psi^n \leq_2 lim_{n+1}$ gelten. Aber das impliziert, dass lim_{n+1} nicht C^{n+1}-stetig ist oder K_ψ^n C^{n+1}-stetig ist. Das ist ein Widerspruch. □

Satz 94. *$lim_{n+1} \not\leq_2 \overline{Lim}_n$ für alle $n \in \mathbb{N}$.*

Beweis.
Wenn $lim_{n+1} \leq_2 \overline{Lim}_n$, dann gibt es stetige Funktionen A und B mit

$$lim_{n+1}(p) = A(p, \overline{Lim}_n \circ B(p))$$

Wegen $lim_{n+1}(p) \neq div$ impliziert $\overline{Lim}_n \circ B(p) \neq 0$ (das Argument ist dasselbe wie im Beweis des Satzes 91), ist dies äquivalent zu

$$lim_{n+1}(p) = A(p, Lim_n \circ B(p))$$

Und wegen $lim_n \equiv_2 Lim_n$ impliziert das $lim_{n+1} \leq_2 lim_n$.
Das ist ein Widerspruch. Also gilt $lim_{n+1} \not\leq_2 \overline{Lim}_n$. □

Satz 95. *$\overline{lim}_n \leq_2 \overline{Lim}_n$ für alle $n \in \mathbb{N}$.*

Beweis. Definiere A durch $A(p) := \langle p, p, \ldots \rangle$.
Dann gilt $\overline{lim}_n(p) = \overline{Lim}_n \circ A(p)$ für alle $p \in \mathbb{B}$. □

Satz 96. *$\overline{Lim}_n \not\leq_2 \overline{lim}_n$ für alle $n > 0$.*

Beweis. Aus $\overline{Lim}_n \leq_2 \overline{lim}_n$ würde $\overline{Lim}_n \leq_2 lim_{n+1}$ folgen nach Satz 91. Das ist ein Widerspruch zu Satz 93. □

Satz 97. *$\overline{Lim}_0 \not\leq_2 \overline{lim}_0$.*

Beweis. Definiere A durch

$$A(p,k) := \begin{cases} 0 & \text{falls } k = 1 \\ 1 & \text{sonst} \end{cases}$$

Dann gilt $\Omega(p) = A(p, \overline{Lim}_0(p))$ für alle $p \in \mathbb{B}$.
<u>Beweis</u>:
$\Omega(p) = 0$
$\Longrightarrow (\forall k) p(k) = 0$
$\Longrightarrow \overline{Lim}_0(p) = 1$
$\Longrightarrow A(p, \overline{Lim}_0 \circ B(p)) = 0$ und
$\Omega(p) = 1$
$\Longrightarrow (\exists k) p(k) \neq 0$
$\Longrightarrow \overline{Lim}_0 \circ B(p) = 0$
$\Longrightarrow A(p, \overline{Lim}_0 \circ B(p)) = 1$.
Aus $\overline{Lim}_0 \leq_2 \overline{lim}_0$ und $\Omega \leq_2 \overline{Lim}_0$ folgt $\Omega \leq_2 \overline{lim}_0$.
Das ist aber nicht möglich, da $\overline{lim}_0$ stetig ist.
Also gilt $\overline{Lim}_0 \not\leq_2 \overline{lim}_0$. □

Satz 98. *$\overline{Lim}_n \lneqq_2 \overline{lim}_{n+1}$ für alle $n \in \mathbb{N}$.*

Beweis.
<u>$\overline{Lim}_n \leq_2 \overline{lim}_{n+1}$</u>:
Definiere f durch $f(p) := \langle p, p, \ldots \rangle$. Dann gilt $\overline{Lim}_n(p) = \overline{lim}_{n+1} \circ f(p)$.
<u>$\overline{lim}_{n+1} \not\leq_2 \overline{Lim}_n$</u>:
$\overline{lim}_{n+1} \leq_2 \overline{Lim}_n$ impliziert $lim_{n+1} \leq_2 \overline{Lim}_n$. Das ist ein Widerspruch zu Satz 94. □

Die Reduzierbarkeitsrelationen zwischen lim_n, $\overline{lim_n}$, Lim_n und $\overline{Lim_n}$ können benutzt werden um die Relationen der Mengen $C^n_{\forall,tot}$ und $C^n_{\forall,div}$ zu untersuchen. Ein Beispiel dafür gibt der nächste Satz.

Satz 99. *$C^{2n}_{\forall,div} \subsetneqq C^{2n}_{\forall,tot}$ für alle $n \geq 1$.*

Beweis. $f \in C^{2n}_{\forall,div} \Leftrightarrow f \leq_0 lim_n$ und $lim_n \leq_0 \overline{lim_n}$.
Also folgt wegen der Transitivität von $\leq_0$, dass $f \leq_0 \overline{lim_n}$ und damit $f \in C^{2n}_{\forall,tot}$.
$\overline{lim_n} \in C^{2n}_{\forall,tot}$. Falls $lim_n \in C^{2n}_{\forall,div}$, dann $\overline{lim_n} \leq_0 lim_n$. Aber das würde $\overline{lim_n} \leq_2 lim_n$ implizieren. Das ist aber nach Satz 90 nicht möglich. □

Vasco Brattka hat eine ähnliche Hierarchie beschrieben, die mit unserer nicht übereinstimmt. In [Bra03] definiert er die folgenden Funktionen:

Definition 88. *Definiere $C_k : \mathbb{B} \longrightarrow \mathbb{B}$ durch*

$$C_k(p)(n) := \begin{cases} 0 & \textit{falls } (\exists n_k)(\forall n_{k-1}) \ldots p\langle n, n_1, \ldots, n_k\rangle \neq 0 \\ 1 & \textit{sonst} \end{cases}$$

In dieser Definition ist der erste Quantor $(\exists n_k)$ und der letzte Quantor ist $(\forall n_1)$, falls k gerade ist, und $(\exists n_1)$, falls k ungerade ist.

Die Funktionen C_k dürfen nicht mit den Funktionen verwechselt werden, die wir in Definition 15 beschrieben haben. Allem Anschein nach sind diese Funktionen nicht vollständig für die Klassen der C-Hierarchie.[25]

Somit unterscheidet sich die Hierarchie der Funktionen C_k von der Hierarchie der C^k-vollständigen Funktionen. Aber sie können in der letzteren Hierarchie eingeordnet werden:

Lemma 32.
Definiere $NEG' : \mathbb{B} \longrightarrow \mathbb{B}$ durch

$$NEG'(p)(n) := \begin{cases} 0 & \textit{falls } p(n) > 0 \\ 1 & \textit{sonst} \end{cases}$$

Dann gilt

$$C_{n+1}(p)(k) = \begin{cases} 0 & \textit{falls } (\exists l) C_n \circ NEG'(p)\langle k, l\rangle = 1 \\ 1 & \textit{sonst} \end{cases}$$

für alle $n > 1$.

[25] Es ist leicht zu beweisen, dass C_0 2-vollständig für die Klasse der stetigen Funktionen und C_1 2-vollständig für die Klasse der C-stetigen Funktionen ist. Ferner ist C_k nicht C^2-stetig für alle $k \geq 3$. C_2 ist wohl nicht vollständig für die Klasse der C^2-stetigen Funktionen. Dann gibt es kein k mit der Eigenschaft, dass C_k 2-vollständig für die Klasse der C^2-stetigen Funktionen ist, aber das haben wir bisher noch nicht endgültig bewiesen.

Beweis. Für $n > 1$ gilt
$C_{n+1}(p)(k) = 0$
$\Leftrightarrow (\exists l)(\forall l_1) \dots p\langle k, l, l_1, \dots, l_n \rangle \neq 0$
$\Leftrightarrow (\exists l)(\forall l_1) \dots NEG'(p)\langle k, l, l_1, \dots, l_n \rangle = 0$
$\Leftrightarrow (\exists l) C_n \circ NEG'(p)\langle k, l \rangle = 1.$ □

Satz 100. *C_n ist C^n-stetig für alle $n > 0$.*

Beweis. Beweis durch Induktion:
<u>$n = 1$</u>:
Definiere f durch

$$f(p)\langle k, l \rangle := \begin{cases} k+1 & \text{falls } p\langle k, l \rangle \neq 0 \\ 0 & \text{sonst} \end{cases}$$

Dann gilt $C_1 = C \circ f$.
Beweis: $C_1(p)(k) = 0$
$\Leftrightarrow (\exists l) p\langle k, l \rangle \neq 0$
$\Leftrightarrow (\exists l) f(p)\langle k, l \rangle = k + 1$
$\Leftrightarrow C \circ f(p)(k) = 0$ und
$C_1(p)(k) = 1$
$\Leftrightarrow (\forall l) p\langle k, l \rangle = 0$
$\Leftrightarrow (\forall l) f(p)\langle k, l \rangle = 0$
$\Leftrightarrow C \circ f(p)(k) = 1.$
Somit ist C_1 C-stetig.
<u>$n \to n + 1$</u>:
Nach Lemma 32 gilt $C_{n+1}(p)(k) = 0 \Leftrightarrow (\exists l) C_n \circ NEG'(p)\langle k, l \rangle = 1$.
Nach Induktionsannahme ist C_n eine C^n-stetige Funktion.
Definiere g durch

$$g(p)\langle k, l \rangle := \begin{cases} k+1 & \text{falls } p\langle k, l \rangle = 1 \\ 0 & \text{sonst} \end{cases}$$

Dann gilt $C_{n+1} = C \circ g \circ C_n \circ NEG'$. Beweis:
$C_{n+1}(p)(k) = 0$
$\Leftrightarrow (\exists l) C_n \circ NEG'(p)\langle k, l \rangle = 1$
$\Leftrightarrow (\exists l) g \circ C_n \circ NEG'(p)\langle k, l \rangle = k + 1$
$\Leftrightarrow C \circ g \circ C_n \circ NEG'(p)(k) = 0$ und
$C_{n+1}(p)(k) = 1$
$\Leftrightarrow (\forall l) C_n \circ NEG'(p)\langle k, l \rangle = 0$
$\Leftrightarrow (\forall l) g \circ C_n \circ NEG'(p)\langle k, l \rangle = 0$
$\Leftrightarrow C \circ g \circ C_n \circ NEG'(p)(k) = 1.$
Somit ist C_{n+1} C^{n+1}-stetig.

Dieser Beweis zeigt, dass wir C_n in der folgenden Form schreiben können:
$C_1 = C \circ f$,
$C_2 = C \circ g \circ C \circ f \circ NEG'$,
$C_3 = C \circ g \circ C \circ g \circ C \circ f \circ NEG' \circ NEG'$,
$C_4 = C \circ g \circ C \circ g \circ C \circ g \circ C \circ f \circ NEG' \circ NEG' \circ NEG'$,
Im allgemeinen gilt:
$C_{n+1} = (C \circ g)^n \circ C \circ f \circ (NEG')^n$ für alle $n \in \mathbb{N}$. □

Es ist möglich, die Beziehung zwischen der C^n-Hierarchie und der C_n-Hierarchie genauer zu beschreiben.

Definition 89. *Definiere* $C'_n :\subseteq \mathbb{B} \longrightarrow \mathbb{N}$ *durch*

$$C'_n(p) := \begin{cases} 0 & \text{falls } (\exists k_n) \dots (\forall k_1) p\langle k_1, \dots, k_n\rangle \neq 0 \\ 1 & \text{sonst} \end{cases}$$

für gerades $n \in \mathbb{N}$ *und*

$$C'_n(p) := \begin{cases} 0 & \text{falls } (\exists k_n) \dots (\exists k_1) p\langle k_1, \dots, k_n\rangle \neq 0 \\ 1 & \text{sonst} \end{cases}$$

für ungerades $n \in \mathbb{N}$.

Definition 90. *Definiere* $H^n_{\chi,\infty} : \mathbb{B} \to \mathbb{B}$ *durch*

$$H^n_{\chi,\infty}(p,q)(i) := \begin{cases} 0 & \text{falls } \chi^n_{p_i}(q) \text{ existiert} \\ 1 & \text{sonst} \end{cases}$$

und $K^n_{\chi,\infty} : \mathbb{B} \to \mathbb{B}$ *durch*

$$K^n_{\chi,\infty}(p)(i) := \begin{cases} 0 & \text{falls } \chi^n_{p_i}(p) \text{ existiert} \\ 1 & \text{sonst} \end{cases}$$

Lemma 33. $K^j_\chi \leq_2 C'_{2j}$ *für alle* $j \in \mathbb{N}$.

Beweis. Definiere f durch

$$f(p) := \begin{cases} 0 & \text{falls } U^j_\chi(p,p) \text{ existiert} \\ div & \text{sonst} \end{cases}$$

Da U^j_χ C^j-stetig ist, gibt es eine stetige und totale Funktion A mit

$$f(p) = \begin{cases} 0 & \text{falls } (\exists k_1)(\forall l_1 > k_1) \dots (\exists k_j)(\forall l_j > k_j) A(p)\langle l_1, \dots, l_j\rangle = 0 \\ div & \text{sonst} \end{cases}$$

Definiere B durch

$$B(p)\langle k_1, l_1, \dots, k_j, l_j\rangle := p\langle l_1, \dots, l_j\rangle$$

Definiere B' durch

$$B'(p)\langle k_1, l_1, \dots, k_j, l_j\rangle := \begin{cases} 1 & \text{falls } l_1 \leq k_1 \text{ oder } \dots \text{ oder } l_j \leq k_j \\ & \text{oder } A(p)\langle l_1, \dots, l_j\rangle = 0 \\ 0 & \text{sonst} \end{cases}$$

Dann gilt $K^j_\chi(p) = C'_{2j} \circ B' \circ B(p)$.
Beweis:
$K^j_\chi(p) = 0$
$\Rightarrow U^j_\chi(p,p)$ existiert

$\Rightarrow f(p) = 0$
$\Rightarrow (\exists k_1)(\forall l_1 > k_1)\dots(\exists k_j)(\forall l_j > k_j)A(p)\langle l_1,\dots,l_j\rangle = 0$
$\Rightarrow (\exists k_1)(\forall l_1)\dots(\exists k_j)(\forall l_j)B' \circ B(p)\langle k_1, l_1,\dots,k_j,l_j\rangle = 1$
$\Rightarrow (\exists k_1)(\forall l_1)\dots(\exists k_j)(\forall l_j)B' \circ B(p)\langle k_1, l_1,\dots,k_j,l_j\rangle \neq 0$
$\Rightarrow C'_{2j} \circ B' \circ B(p) = 0$ und
$K^j_\chi(p) = 1$
$\Rightarrow U^j_\chi(p,p)$ existiert nicht
$\Rightarrow f(p) = div$
$\Rightarrow \neg(\exists k_1)(\forall l_1 > k_1)\dots(\exists k_j)(\forall l_j > k_j)A(p)\langle l_1,\dots,l_j\rangle = 0$
$\Rightarrow \neg(\exists k_1)(\forall l_1)\dots(\exists k_j)(\forall l_j)B' \circ B(p)\langle k_1, l_1,\dots,k_j,l_j\rangle = 1$
$\Rightarrow \neg(\exists k_1)(\forall l_1)\dots(\exists k_j)(\forall l_j)B' \circ B(p)\langle k_1, l_1,\dots,k_j,l_j\rangle \neq 0$
$\Rightarrow C'_{2j} \circ B' \circ B(p) = 1.$ □

Lemma 34. *$C'_{2j} \leq_2 H^j_\chi$ für alle $j \in \mathbb{N}$.*

Beweis. Definiere f durch

$$f(p) = \begin{cases} 0 & \text{falls } (\exists k_1)(\forall l_1)\dots(\exists k_j)(\forall l_j)p\langle k_1, l_1,\dots,k_j,l_j\rangle \neq 0 \\ div & \text{sonst} \end{cases}$$

f ist C^j-stetig. Also gibt es ein $q \in \mathbb{B}$ mit $f(p) = U^j_\chi(q,p)$.
Definiere B durch $B(p) := \langle q,p\rangle$.
Dann gilt $C'_{2j}(p) = H^j_\chi \circ B(p) = H^j_\chi \circ \langle q,p\rangle$. □

Satz 101. $C'_{2j} \equiv_2 H^j_\chi \equiv_2 K^j_\chi$.

Beweis. Das folgt aus den beiden letzten Lemmas und der Tatsache, dass $H^j_\chi \equiv_2 K^j_\chi$. □

Lemma 35. *Seien $f :\subseteq \mathbb{B} \longrightarrow \mathbb{N}$ und $g :\subseteq \mathbb{B} \longrightarrow \mathbb{N}$ Funktionen mit $f \leq_2 g$. Seien $f' :\subseteq \mathbb{B} \longrightarrow \mathbb{B}$ und $g' :\subseteq \mathbb{B} \longrightarrow \mathbb{B}$ definiert durch $f'(p)(k) := f(p_k)$ und $g'(p)(k) := g(p_k)$.*
Dann gilt $f' \leq_2 g'$.

Beweis. Annahme: $f(p) = A(p, g \circ B(p))$ mit stetigen Funktionen A und B.
Definiere B' durch

$$B'(p)\langle k,i\rangle := B(p_k)(i)$$

und A' durch

$$A'(p,q)(k) := A(p_k, q(k))$$

Dann gilt $f'(p) = A'(p, g' \circ B'(p))$ für alle $p \in \mathbb{B}$.
Beweis:
$A'(p, g' \circ B'(p))(k)$
$= A(p_k, g' \circ B'(p)(k))$
$= A(p_k, g \circ B(p_k))$
$= f'(p_k)$.
Also gilt $f' \leq_2 g'$. □

Satz 102. $C_{2j} \equiv_2 H^j_{\chi,\infty} \equiv_2 K^j_{\chi,\infty}$.

Beweis. $C'_{2j} \leq_2 H^j_\chi$ nach Lemma 34. Dann gilt $C_{2j} \leq_2 H^j_{\chi,\infty}$ nach Lemma 35.
$K^j_\chi \leq_2 C'_{2j}$ nach Lemma 33. Dann gilt $K^j_{\chi,\infty} \leq_2 C_{2j}$ nach Lemma 35.
Ferner ist es einfach zu zeigen, dass $K^j_{\chi,\infty} \equiv_2 H^j_{\chi,\infty}$. □

Satz 103. *Sei $f :\subseteq \mathbb{B} \longrightarrow \mathbb{N}$ eine C^n-stetige Funktion.*
Definiere $f' :\subseteq \mathbb{B} \longrightarrow \mathbb{B}$ durch $f'(p)(k) := f(p_k)$.
Dann ist f' ebenfalls C^n-stetig.

Beweis. Sei f C^n-stetig. Dann gibt es stetige Funktionen $A_0, \ldots, A_n$ mit $f(p) = A_n \circ C \circ \ldots \circ C \circ A_0(p)$ für alle $p \in \mathbb{B}$.
Definiere A'_0 durch

$$A'_0(p)\langle k, i\rangle := \langle k, A_0(p_k)(i)\rangle + 1$$

Definiere A'_m durch

$$A'_m(q)\langle k, i\rangle := \langle k, A_m(q_k)\rangle + 1$$

für $1 \leq m \leq n-1$.
Definiere A'_m durch

$$A'_n(q)(k) := A_n(q_k)$$

Dann gilt $f'(p) = A'_n \circ C \circ \ldots \circ C \circ A'_0(p)$ für alle $p \in \mathbb{B}$. Also ist f' C^n-stetig. □

Satz 104. *C_{2j} und C_{2j-1} sind C^{j+1}-stetig.*

Beweis. H^j_χ ist C^{j+1}-stetig. Nach Lemma 103 ist $H^j_{\chi,\infty}$ ebenfalls C^{j+1}-stetig. $H^j_{\chi,\infty} \equiv_2 C_{2j}$ impliziert dann, dass C_{2j} ebenfalls C^{j+1}-stetig ist. □

4.2.10 Grenzwertberechnung für Folgen reeller Zahlen

Definition 91 (Grenzwert von Folgen reeller Zahlen)**.** *Sei $RKONVDIV$ die Menge der Funktionen $f :\subseteq \mathbb{B} \to \mathbb{B}$ mit*

$$\rho(f(p)) = \lim_{n\to\infty} \rho(p_n)$$

Die Funktionen in $RKONVDIV$ berechnen den Grenzwert von Folgen von Darstellungen reeller Zahlen.

Satz 105 (Vollständigkeit von $RKONVDIV$)**.** *Es gibt eine C-vollständige Funktion $f :\subseteq \mathbb{B} \longrightarrow \mathbb{B}$ mit $f \in RKONVDIV$.*

Beweis. Es gibt eine $C-$vollständige Funktion $h \in F$, wobei F die Menge aller Funktionen ist, die ρ_n-Darstellungen in ρ-Darstellungen übersetzen (schnellkonvergierende Repräsentationen von reellen Zahlen)[26].
Definiere B durch $B(p)(n) := \langle p\rangle_n(n) = p\langle n, n\rangle$.
Definiere f durch $f := h \circ B$.
Dann ist f C-stetig und $f \in RKONVDIV$.
Beweis:
f ist C-stetig, weil h C-stetig und B stetig ist.
Wenn $\lim_{n\to\infty} \rho(p_n) = x \in \mathbb{R}$ existiert, dann gilt
$\lim_{n\to\infty} \rho_n(B(p)) = \lim_{n\to\infty} \rho_n(p\langle n, n\rangle) = x$. Dann gilt $\rho(h \circ B(p)) = x$.
f ist auch C-vollständig.
Beweis:
Sei $h \in F$ wieder eine $C-$vollständige Funktion. Definiere B' durch $B'(p)\langle i, j\rangle := p(j)$ für alle $i, j \in \mathbb{N}$. Dann gilt $h(p) = h \circ B \circ B'(p) = f \circ B'(p)$ für alle $p \in \mathbb{B}$. Also ist eine C-vollständige Funktion 2-reduzierbar auf f, somit ist f 2-vollständig für $[\mathbb{B} \longrightarrow \mathbb{B}]^1$. □

Die Definition von $RKONVDIV$ kann verallgemeinert werden.

Definition 92 (Grenzwert von iterierten Folgen von reellen Zahlen)**.**
Sei $lim^{\mathbb{R}}_n :\subseteq \mathbb{B} \longrightarrow \mathbb{R}$ rekursiv definiert durch
$lim^{\mathbb{R}}_0(p) := 0$,
$lim^{\mathbb{R}}_1(p) := \lim_{i\to\infty} \rho(p_i)$ *und*
$lim^{\mathbb{R}}_{n+1}(p) := \lim_{i\to\infty}(lim^{\mathbb{R}}_n(p_i))$.
Definiere $RKONVDIV^n$ durch
$RKONVDIV^0$ ist die Menge $\{f\}$ mit $f(p) := p$,
$RKONVDIV^1 := RKONVDIV$ und
$RKONVDIV^n$ ist die Menge der Funktionen $f :\subseteq \mathbb{B} \to \mathbb{B}$ mit

$$\rho(f(p)) = lim^{\mathbb{R}}_n(p)$$

für alle $n > 1$.

Der folgende Satz ist eine Verallgemeinerung des vorigen Satzes.

Satz 106 (Vollständigkeit von $RKONVDIV^n$)**.** *Für alle $n \in \mathbb{N}$ gibt es eine Funktion $f \in RKONVDIV^n$, die 2-vollständig für $[\mathbb{B} \longrightarrow \mathbb{B}]^n$ ist.*

[26] siehe Definition 54 und [St89] Seite 35

Beweis.
$\underline{n=0}$: $RKONVDIV^0 = \{f\}$ mit $f(p) := p$. f ist C^0-stetig und 2-hart für $[\mathbb{B} \longrightarrow \mathbb{B}]^0$.
Wie in dem Beweis zuvor gibt es eine $C-$vollständige Funktion $h \in F$, wobei F die Menge der Funktionen ist, die ρ_n-Repräsentationen in ρ-Repräsentationen übersetzen.
$\underline{n>0}$: Definiere B durch

$$B(p)\langle i_1, \ldots, i_{n+1}\rangle := p\langle i_1, \ldots, i_n\rangle$$

B ist stetig.
Definiere B' durch $B'(p)(i) := p\langle i, i\rangle$.
Definiere weiter h_n rekursiv durch $h_1(p) := h \circ B'(p)$ und

$$h_{n+1}(p) := h \circ B'(h_n(p_0), h_n(p_1), \ldots)$$

Es ist leicht zu zeigen, dass h_n C^n-stetig ist.
Dann gilt

$$lim_n(p) = h_n \circ B(p)$$

für alle $p \in \mathbb{B}$ und $h_n \in RKONVDIV^n$.
Somit gibt es eine C^n-stetige Funktion $h_n \in RKONVDIV^n$, die 2-hart für $[\mathbb{B} \longrightarrow \mathbb{B}]^n$ ist. $\square$

Also können wir die Berechnung der Grenzwerte von Folgen reeller Zahlen als Beispiele für C^n-vollständige Funktionen von $\mathbb{B}$ nach $\mathbb{B}$ ansehen. In dem obigen Beweis können wir für f die einfachste Funktion in $RKONVDIV^n$ benutzen.

Somit kann der obige Satz in der folgenden Weise formuliert werden:

Satz 107 (Vollständigkeit von $RKONVDIV^n$)**.** ***Für alle $n \in \mathbb{N}$ gibt es eine C^n-vollständige Funktion $f \in RKONVDIV^n$.***

4.2.11 Integration über unendlichen Intervallen

Wie Klaus Weihrauch gezeigt hat, ist die Integration von berechenbaren reellen Funktionen über endlichen Intervallen berechenbar. Auf der anderen Seite ist die Differentiation von berechenbaren reellen Funktionen im allgemeinen nicht berechenbar. In knappen Worten: Integration ist numerisch einfach und symbolisch schwierig, und Differentiation ist numerisch schwierig und symbolisch einfach.

Aber die numerische Berechnung der Integration kann ebenfalls schwierig sein, wenn das Intervall unendlich ist.

Definition 93. *Sei Int die Menge der Funktionen $f :\subseteq \mathbb{B}^3 \longrightarrow \mathbb{B}$ mit*

$$\rho(f(p,a,b)) = \int_{\rho(a)}^{\rho(b)} \delta_p(x)d(x)$$

Satz 108. *Es gibt stetige Funktionen $f \in Int$.*

Beweis. Der Beweis ist in [Wei00] Seite 182 zu finden. □

Definition 94. *Sei β eine effektive Repräsentation von stetigen Funktionen auf $\mathbb{R}$, die durch Polygone auf den Intervallen $[-a,a]$ mit $a \in \mathbb{Z}$ approximiert werden können: für alle i gibt es eine Zahl $a \in \mathbb{Z}$ mit $\beta(p(i))$ ist ein Polygon auf dem Intervall $[-a,a]$ und $\beta(p(i))(x) = 0$ für alle $x \leq -a$ und $x \geq a$.*
Dann gilt $\beta(p) = \lim_{i\to\infty} \beta(p(i))$.
Definiere eine Repräsentation δ_∞ von stetigen Funktionen $\mathbb{R} \to \mathbb{R}$ durch

$$\delta_\infty(p) := \begin{cases} f & \text{falls } \beta(p)(n) \text{ konvergiert und} \\ & (\forall m)(\forall n > m)|\beta(p(m)) - \beta(p(n))| < 2^{-n} \\ & \text{und } (\forall i)\beta(p(i)) \text{ ist ein Polygon auf } [-i,i] \\ div & \text{sonst} \end{cases}$$

Definition 95. *Sei $Int_{0,\infty}$ die Menge der Funktionen $f :\subseteq \mathbb{B} \longrightarrow \mathbb{B}$ mit*

$$\rho(f(p)) = \int_0^\infty \delta_\infty(p)(x)d(x)$$

Satz 109. *In $Int_{0,\infty}(p)$ gibt es C-vollständige Funktionen f für $[\mathbb{B} \longrightarrow \mathbb{B}]$.*

Beweis. Für $a < b$, $a, b \in \mathbb{Q}$, und $q \in \mathbb{Q}$ definiere $P(a,b,q) : \mathbb{R} \to \mathbb{R}$ durch

$$P(a,b,q)(x) := \begin{cases} 0 & \text{falls } x < a \text{ oder } x > b \\ 2q\frac{x-a}{b-a} & \text{falls } a \leq x \leq \frac{b+a}{2} \\ 2q\frac{b-x}{b-a} & \text{falls } \frac{b+a}{2} < x \leq b \end{cases}$$

Definiere die Funktion f_n durch

$$f_n := \begin{cases} P(0,1,2\cdot p(0)) & \text{falls } n = 0 \\ P(n,n+1,2\cdot(p(n)-p(n-1))) & \text{falls } n > 0 \end{cases}$$

Definiere g_n durch

$$g_n := \sum_{i=0}^{n} f_i$$

und

$$g := \lim_{i \to \infty} g_i$$

Es ist leicht, eine stetige Funktion B zu definieren mit $\beta(B(p)) = g$.
Dann gilt nach Definition von g

$$Int_{0,\infty}(g) = \lim_{i \to \infty} p(i)$$

Somit gilt $lim_1(p) = Int_{0,\infty} \circ B(p)$, also $lim_1 \leq_2 Int_{0,\infty}$.

Auf der anderen Seite gilt

$$Int_{0,\infty}(p) = \int_0^\infty \delta_\infty(p) = \lim_{i \to \infty} \int_0^{i+1} \delta_\infty(p) = \lim_{i \to \infty} Int(0, i+1)(p)$$

Weil Int eine berechenbare und damit eine stetige Funktion ist, gibt es eine stetige Funktion B mit $B(p)(i) = Int(0, i+1, p)$ Also gilt

$$Int_{0,\infty}(p) = \lim_{i \to \infty} B(p)(i)$$

Daraus folgt $Int_{0,\infty} = lim_1 \circ B$, und das bedeutet $Int_{0,\infty} \leq_2 lim_1$. $\square$

4.3 Beispiele für unvollständige Funktionen

Viele früher definierte Funktionen stellen sich als unvollständig heraus. Hier sollen einige Beispiele behandelt werden.

4.3.1 $MAX2$ ist nicht vollständig für $[\mathbb{B} \to \mathbb{N}]^2$

Satz 110. *MAX2 ist nicht* 1*-vollständig für* $[\mathbb{B} \to \mathbb{N}]^2$.

Beweis. Wir zeigen: $KDIV_2^{\mathbb{N}} \not\leq_1 MAX2$.
Annahme: $KDIV_2^{\mathbb{N}} \leq_1 MAX2$. Dann gibt es stetige Funktionen A und B mit $KDIV_2^{\mathbb{N}}(p) = A \circ MAX2 \circ B(p)$ für alle $p \in \mathbb{B}$.

Es gibt höchstens eine Zahl n mit $KDIV_2^{\mathbb{N}}(p) = n$ und $MAX2 \circ B(p) = 0$ für ein $p \in \mathbb{B}$.
Wenn es zwei Zahlen n_1 und n_2 gibt, dann folgt $KDIV_2^{\mathbb{N}}(p_1) = n_1 = A \circ MAX2 \circ B(p_1) = A(0) = A \circ MAX2 \circ B(p_2) = KDIV_2^{\mathbb{N}}(p_2) = n_2$.

Sei N definiert durch

$$N := \begin{cases} \text{die eindeutige Zahl } n & \text{falls } (\exists p) KDIV_2^{\mathbb{N}}(p) = n \\ & \text{und } MAX2 \circ B(p) = 0 \\ 0 & \text{sonst} \end{cases}$$

Definiere p_0 durch $p_0\langle i,j\rangle := N+1$ für alle i und j. Dann gilt $KDIV_2^{\mathbb{N}}(p_0) = N+1 = A \circ MAX2 \circ B(p_0)$. Es gibt ein maximales k_0 in $B(p_0)$ und ein Präfix der Länge l_0 von p_0 mit $k_0 \leq max([[p_0]]_{l_0} r)$ für alle $r \in \mathbb{B}$.
Definiere p_1 durch

$$p_1\langle i,j\rangle := \begin{cases} N+1 & \text{falls } i \leq l_0 \text{ und } j \leq l_0 \\ 0 & \text{falls } i \leq l_0 \text{ und } j > l_0 \\ N+2 & \text{sonst} \end{cases}$$

Dann gilt $KDIV_2^{\mathbb{N}}(p_1) = N+2 = A \circ MAX2 \circ B(p_1)$. Es gibt ein maximales $k_1 > k_0$ in $B(p_1)$ und ein Präfix der Länge l_1 von p_1 mit $k_1 \leq max([[p_1]]_{l_1} r)$ für alle $r \in \mathbb{B}$.
Definiere rekursiv p_n durch

$$p_{n+1}\langle i,j\rangle := \begin{cases} p_n\langle i,j\rangle & \text{falls } i \leq l_{n-1} \\ N+n & \text{falls } l_{n-1} < i \leq l_n \text{ und } j \leq l_n \\ 0 & \text{falls } l_{n-1} < i \leq l_n \text{ und } j > l_n \\ N+n+1 & \text{sonst} \end{cases}$$

und k_n als das Maximum in $B(p_n)$ und l_n als die Länge des kürzesten Präfixes von p_n mit $k_n \leq max([[p_n]]_{l_n} r)$ für alle $r \in \mathbb{B}$.

Definiere p durch $p := lim_{n\to\infty}(p_n)$. Dann gilt $KDIV_2^{\mathbb{N}}(p) = 0 = A \circ MAX2 \circ B(p) = A(0)$.

Definiere q_0 durch $q_0\langle i,j\rangle := N+1$ für alle i und j. Dann gilt $KDIV_2^{\mathbb{N}}(q_0) = N+1 = A \circ MAX2 \circ B(q_0)$. Es gibt ein maximales k_0 in $B(q_0)$ und ein Präfix der Länge l_0 von q_0 mit $k_0 \leq max([[q_0]]_{l_0} r)$ für alle $r \in \mathbb{B}$.
Definiere q_1 durch

$$q_1\langle i,j\rangle := \begin{cases} N+1 & \text{falls } i \leq l_0 \text{ und } j \leq l_0 \\ 1 & \text{falls } i \leq l_0 \text{ und } j > l_0 \\ N+2 & \text{sonst} \end{cases}$$

Dann gilt $KDIV_2^{\mathbb{N}}(q_1) = N+2 = A \circ MAX2 \circ B(q_1)$. Es gibt ein maximales $k_1 > k_0$ in $B(q_1)$ und ein Präfix der Länge l_1 von q_1 mit $k_1 \leq max([[q_1]]_{l_1} r)$ für alle $r \in \mathbb{B}$.
Definiere rekursiv q_n durch

$$q_{n+1}\langle i,j\rangle := \begin{cases} q_n\langle i,j\rangle & \text{falls } i \leq l_{n-1} \\ N+n & \text{falls } l_{n-1} < i \leq l_n \text{ und } j \leq l_n \\ 1 & \text{falls } l_{n-1} < i \leq l_n \text{ und } j > l_n \\ N+n+1 & \text{sonst} \end{cases}$$

und k_n als das Maximum in $B(q_n)$ und l_n als die Länge des kürzesten Präfixes von q_n mit $k_n \leq max([[q_n]]_{l_n} r)$ für alle $r \in \mathbb{B}$.

Definiere q durch $q := lim_{n\to\infty}(q_n)$. Dann gilt $KDIV_2^{\mathbb{N}}(q) = 1 = A \circ MAX2 \circ B(p) = A(0)$.

Das impliziert $KDIV_2^{\mathbb{N}}(p) = 0 = A(0) = KDIV_2^{\mathbb{N}}(q) = 1$. Das ist ein Widerspruch und somit gilt $KDIV_2^{\mathbb{N}} \not\leq_1 MAX2$. $\square$

Mit diesem Beweis ist nur gezeigt worden, dass $MAX2$ nicht 1-vollständig für $[\mathbb{B} \to \mathbb{N}]^2$ ist.

Die Idee des Beweises wird in der nächsten Abbildung illustriert. Die Teilfolgen von p und q sind dabei waagerecht eingetragen.

Der folgende Satz sagt, dass $MAX2$ ebenfalls nicht 2-vollständig für $[\mathbb{B} \to \mathbb{N}]^2$ ist.

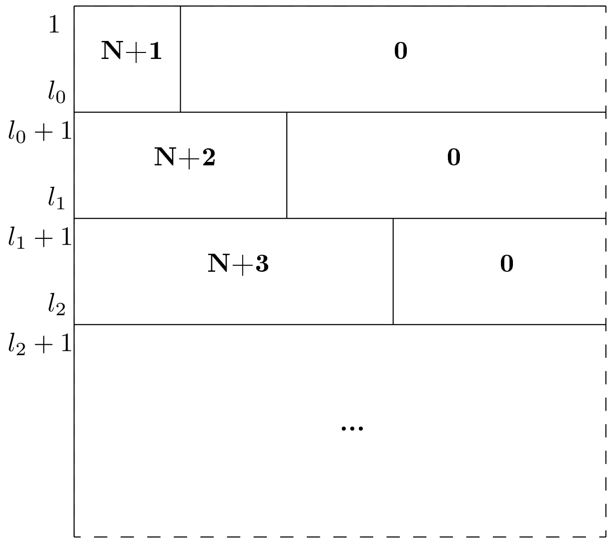

Abbildung 13: Illustration von p

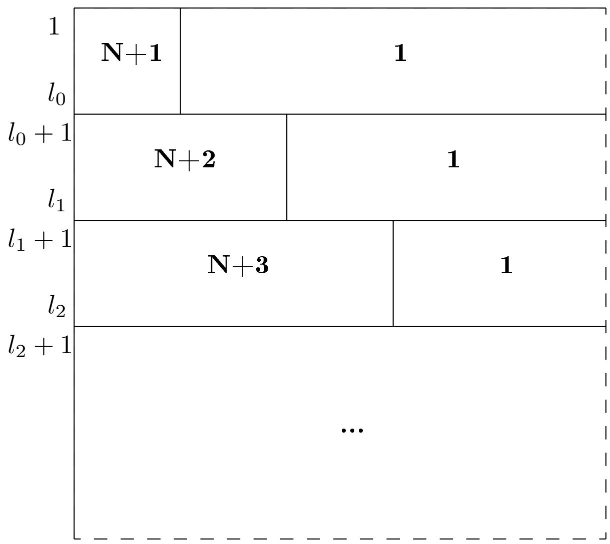

Abbildung 14: Illustration von q

Satz 111. *$MAX2$ ist nicht 2-vollständig für $[\mathbb{B} \to \mathbb{N}]^2$.*

Beweis. Wir zeigen: $KDIV_2^{\mathbb{N}} \not\leq_2 MAX2$.
Annahme: $KDIV_2^{\mathbb{N}} \leq_2 MAX2$. Dann gibt es stetige Funktionen A und B mit $KDIV_2^{\mathbb{N}}(p) = A(p, MAX2 \circ B(p))$ für alle $p \in \mathbb{B}$.

Weil A stetig ist, gibt es für alle $p \in \mathbb{B}$ und $i \in \mathbb{N}$ mit $A(p,i) \neq div$ eine Zahl $k(p,i)$ mit $A([[p]]_{k(p,i)}r, i) = A(p,i)$ für alle $r \in \mathbb{B}$.

Definiere q_0 durch $q_0 := 0^\omega$.

Dann gilt $KDIV_2^{\mathbb{N}}(q_0) = 0 = A(q_0, MAX2 \circ B(q_0))$.

Falls $MAX2 \circ B(q_0) > 0$, dann gibt es eine Zahl s_0 mit $MAX2 \circ B(q_0) \leq max(\overline{B}([[q_0]]_{s_0}))$ und $A([[q_0]]_{s_0}r, MAX2 \circ B(q_0)) = A(q_0, MAX2 \circ B(q_0))$ für alle $r \in \mathbb{B}$.

Falls $MAX2 \circ B(q_0) = 0$, dann gibt es eine Zahl s_0 mit $A([[q_0]]_{s_0}r, 0) = A(q_0, 0)$ für alle $r \in \mathbb{B}$.

Und es gibt eine Zahl l_0 mit $l_0 = max(\overline{B}([[q_0]]_{s_0}))$.

Definiere q_1 durch

$$q_1\langle i,j\rangle := \begin{cases} q_0\langle i,j\rangle & \text{falls } i \leq s_0 \text{ und } j \leq s_0 \\ 0 & \text{falls } i \leq s_0 \text{ und } j > s_0 \\ 1 & \text{sonst} \end{cases}$$

Dann gilt $KDIV_2^{\mathbb{N}}(q_1) = 1 = A(q_1, MAX2 \circ B(q_1))$.

Falls $MAX2 \circ B(q_1) > 0$, dann gibt es eine Zahl $s_1 > s_0$ mit $MAX2 \circ B(q_1) \leq max(\overline{B}([[q_1]]_{s_1}))$ und $A([[q_1]]_{s_1}r, MAX2 \circ B(q_1)) = A(q_1, MAX2 \circ B(q_1))$ für alle $r \in \mathbb{B}$.
Dann sei $l_1 := max(\overline{B}([[q_1]]_{s_1}))$.

Falls $MAX2 \circ B(q_1) = 0$, dann gibt es eine Zahl s_1 mit $A([[q_1]]_{s_1}r, 0) = A(q_1, 0)$ für alle $r \in \mathbb{B}$ und $l_1 := max(\overline{B}([[q_1]]_{s_1})) > l_0$.

Dann gilt $l_1 > l_0$.

Definiere rekursiv q_n durch

$$q_{n+1}\langle i,j\rangle := \begin{cases} q_n\langle i,j\rangle & \text{falls } i \leq s_{n-1} \\ n & \text{falls } s_{n-1} < i \leq s_n \text{ und } j \leq s_n \\ 0 & \text{falls } s_{n-1} < i \leq s_n \text{ und } j > s_n \\ n+1 & \text{sonst} \end{cases}$$

und l_n und s_n in der gleichen Weise.

Definiere q durch $q := lim_{n\to\infty}(q_n)$. Dann gilt $KDIV_2^{\mathbb{N}}(q) = 0 = A(q, MAX2 \circ B(q)) = A(q, 0)$.

Da A stetig ist, gibt es eine Zahl x mit $A([[q]]_x r, 0) = A(q, 0) = 0$ für alle $r \in \mathbb{B}$.

Sei p_0 definiert durch

$$p_0\langle i, j\rangle := \begin{cases} q\langle i, j\rangle & \text{falls } i \leq x \text{ und } j \leq x \\ 0 & \text{sonst} \end{cases}$$

Dann gilt $KDIV_2^{\mathbb{N}}(p_0) = 0 = A(p_0, MAX2 \circ B(p_0))$.

Falls $MAX2 \circ B(p_0) > 0$, dann gibt es eine Zahl r_0 mit $MAX2 \circ B(p_0) \leq max(\overline{B}([[p_0]]_{r_0}))$ und $A([[p_0]]_{r_0} r, MAX2 \circ B(p_0)) = A(p_0, MAX2 \circ B(p_0))$ für alle $r \in \mathbb{B}$.

Fall $MAX2 \circ B(p_0) = 0$, dann gibt es eine Zahl r_0 mit $A([[p_0]]_{r_0} r, 0) = A(p_0, 0)$ für alle $r \in \mathbb{B}$.

Und es gibt eine Zahl k_0 mit $k_0 = max(\overline{B}([[p_0]]_{r_0}))$.

Definiere p_1 durch

$$p_1\langle i, j\rangle := \begin{cases} p_0\langle i, j\rangle & \text{falls } i \leq r_0 \text{ und } j \leq r_0 \\ 1 & \text{falls } i \leq r_0 \text{ und } j > r_0 \\ 1 & \text{sonst} \end{cases}$$

Dann gilt $KDIV_2^{\mathbb{N}}(p_1) = 1 = A(p_1, MAX2 \circ B(p_1))$.

Falls $MAX2 \circ B(p_1) > 0$, dann gibt es eine Zahl $r_1 > r_0$ mit $MAX2 \circ B(p_1) \leq max(\overline{B}([[p_1]]_{r_1}))$ und $A([[p_1]]_{r_1} r, MAX2 \circ B(p_1)) = A(p_1, MAX2 \circ B(p_1))$ für alle $r \in \mathbb{B}$.
Dann sei $k_1 := max(\overline{B}([[p_1]]_{r_1}))$.

Falls $MAX2 \circ B(p_1) = 0$, dann gibt es eine Zahl r_1 mit $A([[p_1]]_{r_1} r, 0) = A(p_1, 0)$ für alle $r \in \mathbb{B}$ und $k_1 := max(\overline{B}([[p_1]]_{r_1})) > k_0$.

Dann gilt $k_1 > k_0$.

Definiere p_2 durch

$$p_2\langle i, j\rangle := \begin{cases} p_1\langle i, j\rangle & \text{falls } i \leq r_1 \text{ und } j \leq r_1 \\ 1 & \text{falls } i \leq r_1 \text{ und } j > r_1 \\ 2 & \text{sonst} \end{cases}$$

Definiere rekursiv p_n durch

$$p_{n+1}\langle i,j\rangle := \begin{cases} p_n\langle i,j\rangle & \text{falls } i \leq r_n \text{ und } j \leq r_n \\ 1 & \text{falls } i \leq r_n \text{ und } j > r_n \\ n+1 & \text{sonst} \end{cases}$$

und k_n und r_n in derselben Weise.

Definiere p durch $p := lim_{n\to\infty}(p_n)$. Dann gilt $KDIV_2^{\mathbb{N}}(p) = 1 = A(p, MAX2 \circ B(p)) = A(p,0)$.

Das impliziert $KDIV_2^{\mathbb{N}}(q) = 0 = A(q,0) = A(p,0) = 1$. Das ist ein Widerspruch und somit folgt $KDIV_2^{\mathbb{N}} \not\leq_2 MAX2$. □

4.3.2 $KON2$ und KB sind nicht vollständig für $[\mathbb{B} \to \mathbb{N}]^2$

Es gibt weitere C^2-stetige Funktionen, die nicht vollständig für $[\mathbb{B} \to \mathbb{N}]^2$ sind.

Definition 96. *Definiere $KON2$ durch*

$$KON2(p) := \begin{cases} m+1 & \textit{falls } p \textit{ gegen } m \textit{ konvergiert} \\ 0 & \textit{sonst} \end{cases}$$

Definition 97. *Definiere KB durch*

$$KB(p) := \begin{cases} \min\{k|(\forall n \geq k)p(n) = p(k)\} + 1 & \textit{falls } p \textit{ konvergiert} \\ 0 & \textit{sonst} \end{cases}$$

Satz 112. *$MAX2 \equiv_2 KON2 \equiv_2 KB$.*

Beweis. $MAX2 \equiv_2 KON2$ [27] und $KB \leq_2 MAX2$ [28]. Der Beweis von $MAX2 \leq_2 KB$ ist einfach und wird deshalb ausgelassen. □

Satz 113. *$KON2$ und KB sind nicht vollständig für $[\mathbb{B} \to \mathbb{N}]^2$.*

Beweis. Nach dem vorigen Lemma gilt $KON2 \equiv_2 KB \equiv_2 MAX2$. Falls $KON2$ oder KB 2-vollständig für $[\mathbb{B} \to \mathbb{N}]^2$ wären, würde dies auch für $MAX2$ gelten. □

4.3.3 KON ist nicht vollständig für $[\mathbb{B} \to \{0,1\}]^2$

Weiter oben wurde schon die Funktion $KON : \mathbb{B} \longrightarrow \mathbb{N}$ definiert durch

$$KON(p) := \begin{cases} 0 & \text{falls } \lim_{n\to\infty} p(n) \text{ existiert} \\ 1 & \text{sonst} \end{cases}$$

KON ist C^2-stetig, aber nicht C-stetig. Es soll nun gezeigt werden, dass KON nicht vollständig für $[\mathbb{B} \to \{0,1\}]^2$ ist.

[27] [Myl92] Seite 56
[28] [Myl92] Seite 57

Lemma 36. *Sei* $KDIV_2^{\{0,1\}}(p) = A(p, KON \circ B(p))$ *für alle* $p \in \mathbb{B}$ *mit stetigen Funktionen* A *und* B. *Seien* $P \in \mathbb{B}$ *und* $k \in \mathbb{N}$.
Dann gibt es eine Folge $q \in \mathbb{B}$ *mit den folgenden Eigenschaften:*

1. $KON \circ B(q) = 1$
2. $KDIV_2^{\{0,1\}}(q) = 0$
3. $q_i = P_i$ *für alle* $0 \leq i \leq k$

Beweis. Sei $KDIV_2^{\{0,1\}}(p) = A(p, KON \circ B(p))$ für alle $p \in \mathbb{B}$ und $P \in \mathbb{B}$ und $k \in \mathbb{N}$.
Definiere p_0 durch

$$p_0\langle m, n\rangle := \begin{cases} P_i\langle n\rangle & \text{falls } m \leq k \\ 0 & \text{sonst} \end{cases}$$

Dann gilt

$$KDIV_2^{\{0,1\}}(p_0) = 0 = A(p_0, KON \circ B(p_0))$$

Dann folgt $KON \circ B(p_0) = 0$ oder $KON \circ B(p_0) = 1$.
Definiere $a_0 := KON \circ B(p_0)$.
Da A stetig ist, gibt es eine Zahl $r_0 > k$ mit

$$A([[p_0]]_{r_0} q, a_0) = 0$$

für alle $q \in \mathbb{B}$.

Definiere q_0 durch

$$q_0\langle m, n\rangle := \begin{cases} P\langle m, n\rangle & \text{falls } m \leq k \\ p_0\langle m, n\rangle & \text{falls } k < m \leq r_0 \text{ und } n \leq r_0 \\ 0 & \text{falls } k < m \leq r_0 \text{ und } n > r_0 \\ 1 & \text{sonst} \end{cases}$$

Wegen $[[p_0]]_{r_0} = [[q_0]]_{r_0}$ folgt, dass $b_0 := KON \circ B(q_0) \neq KON \circ B(p_0)$.
Da A stetig ist, gibt es eine Zahl s_0 mit

$$A([[q_0]]_{s_0} q, b_0) = 1$$

für alle $q \in \mathbb{B}$.
Da $a_0 \neq b_0$, gibt es einen Wechsel in der Folge $B(p_0)$ oder $B(q_0)$. Die Zahlen r_0 und s_0 können groß genug gewählt werden, so dass es es einen Wechsel in $\overline{B}([[q_0]]_{s_0})$ gibt.

Seien die Folgen p_i und q_i und die Zahlen r_i und s_i definiert. Definiere p_{i+1} durch

$$p_{i+1}\langle m, n\rangle := \begin{cases} P\langle m, n\rangle & \text{falls } m \leq k \\ q_i\langle m, n\rangle & \text{falls } k < m \leq s_i \text{ und } n \leq s_i \\ 0 & \text{sonst} \end{cases}$$

Dann gilt

$$KDIV_2^{\{0,1\}}(p_{i+1}) = 0 = A(p_{i+1}, KON \circ B(p_{i+1}))$$

Dann gilt $KON \circ B(p_{i+1}) = 0$ oder $KON \circ B(p_{i+1}) = 1$.
Definiere $a_{i+1} := KON \circ B(p_{i+1})$.
Da A stetig ist, gibt es eine Zahl r_{i+1} mit

$$A([[p_{i+1}]]_{r_{i+1}} q, a_{i+1}) = 0$$

für alle $q \in \mathbb{B}$.

Definiere q_{i+1} durch

$$q_{i+1}\langle m, n\rangle := \begin{cases} P\langle m, n\rangle & \text{falls } m \leq k \\ p_{i+1}\langle m, n\rangle & \text{falls } k < m \leq r_{i+1} \text{ und } n \leq r_{i+1} \\ 0 & \text{falls } k < m \leq r_{i+1} \text{ und } n > r_{i+1} \\ 1 & \text{sonst} \end{cases}$$

Wegen $[[p_{i+1}]]_{r_{i+1}} = [[q_{i+1}]]_{r_{i+1}}$ folgt, dass $b_{i+1} := KON \circ B(q_{i+1}) \neq KON \circ B(p_{i+1})$.
Da A stetig ist, gibt es eine Zahl s_{i+1} mit

$$A([[q_{i+1}]]_{s_{i+1}} q, b_{i+1}) = 1$$

für alle $q \in \mathbb{B}$.
Wegen $a_{i+1} \neq b_{i+1}$ gibt es einen $(i+1)$-ten Wechsel in der Folge $B(p_{i+1})$ oder $B(q_{i+1})$. Die Zahlen r_{i+1} und s_{i+1} können so groß gewählt werden, dass es $i+1$ Wechsel in $\overline{B}([[q_{i+1}]]_{s_{i+1}})$ gibt.

Definiere $q \in \mathbb{B}$ durch

$$q := \lim_{i \to \infty} p_i$$

Nach Konstruktion hat q die Eigenschaften

1. $KON \circ B(q) = 1$
2. $KDIV_2^{\{0,1\}}(q) = 0$
3. $q_i = P_i$ für alle $0 \leq i \leq k$

$\square$

Satz 114. $KDIV_2^{\{0,1\}} \not\leq_2 KON$.

Beweis. Nehme an, dass $KDIV_2^{\{0,1\}} \leq_2 KON$.
Dann gibt es stetige Funktionen A und B mit

$$KDIV_2^{\{0,1\}}(p) = A(p, KON \circ B(p))$$

für alle $p \in \mathbb{B}$.
Nach Lemma 36 gibt es eine Folge p_0 mit den Eigenschaften

1. $KON \circ B(p_0) = 1$

2. $KDIV_2^{\{0,1\}}(p_0) = 0$

Da A stetig ist, gibt es ein $r_0 \in \mathbb{N}$ mit

$$A([[p_0]]_{r_0} q, 1) = 0$$

für alle $q \in \mathbb{B}$.
Definiere q_0 durch

$$q_0\langle m, n\rangle := \begin{cases} p_0\langle m, n\rangle & \text{falls } m \leq r_0 \text{ und } n \leq r_0 \\ 1 & \text{sonst} \end{cases}$$

Dann gilt $KON \circ B(q_0) = 0$ und

$$KDIV_2^{\{0,1\}}(q_0) = 1 = A(q_0, KON \circ B(q_0)) = A(q_0, 0)$$

Da A stetig ist, gibt es eine Zahl $s_0 \in \mathbb{N}$ mit

$$A([[q_0]]_{s_0} q, 0) = 1$$

für alle $q \in \mathbb{B}$.
Wähle $s_0 > r_0$ so groß, dass es einen Wechsel in $\overline{B}([[q_0]]_{s_0})$ gibt.

Seien die Folgen p_i und q_i und die Zahlen r_i und s_i definiert.
Es gebe nach Induktionsannahme i Wechsel in $\overline{B}([[q_i]]_{s_i})$.
Nach Lemma 36 gibt es eine Folge p_{i+1} mit den Eigenschaften

1. $KON \circ B(p_{i+1}) = 1$
2. $KDIV_2^{\{0,1\}}(p_{i+1}) = 0$
3. $(p_{i+1})_j = (q_i)_j$ für alle $0 \leq j \leq s_i$

Da A stetig ist, gibt es ein $r_{i+1} \in \mathbb{N}$ mit

$$A([[p_{i+1}]]_{r_{i+1}} q, 1) = 0$$

für alle $q \in \mathbb{B}$.

Definiere q_{i+1} durch

$$q_{i+1}\langle m, n\rangle := \begin{cases} p_{i+1}\langle m, n\rangle & \text{falls } m \leq r_{i+1} \text{ und } n \leq r_{i+1} \\ 1 & \text{sonst} \end{cases}$$

Dann gilt $KON \circ B(q_{i+1}) = 0$ und

$$KDIV_2^{\{0,1\}}(q_{i+1}) = 1 = A(q_{i+1}, KON \circ B(q_{i+1})) = A(q_{i+1}, 0)$$

Da A stetig ist, gibt es eine Zahl $s_{i+1} \in \mathbb{N}$ mit

$$A([[q_{i+1}]]_{s_{i+1}} q, 0) = 1$$

für alle $q \in \mathbb{B}$.
Wähle $s_{i+1} > r_{i+1}$ so groß, dass es mehr als i Wechsel in $\overline{B}([[q_{i+1}]]_{s_{i+1}})$ gibt.
Definiere q durch

$$q := \lim_{i \to \infty} q_i$$

Nach Konstruktion hat q die Eigenschaften

$$KON \circ B(q) = 1 \text{ und } KDIV_2^{\{0,1\}}(q) = 1$$

p_0 hat die Eigenschaften

$$KON \circ B(p_0) = 1 \text{ und } KDIV_2^{\{0,1\}}(p_0) = 0$$

Aber

$$A([[p_0]]_{r_0} q', 1) = 0$$

für alle $q' \in \mathbb{B}$ und

$$[[p_0]]_{r_0} \sqsubseteq q$$

Das ist ein Widerspruch. □

Satz 115. *KON ist nicht 2-vollständig für $[\mathbb{B} \longrightarrow \{0,1\}]^2$.*

Beweis. Wenn KON 2-vollständig für $[\mathbb{B} \longrightarrow \{0,1\}]^2$ wäre,
dann müsste $KDIV_2^{\{0,1\}} \leq_2 KON$ gelten, da $KDIV_2^{\{0,1\}} \in [\mathbb{B} \longrightarrow \{0,1\}]^2$. □

Viele weitere Funktionen sind ebenfalls nicht vollständig. Das kann durch Äquivalenz mit KON bewiesen werden.

Definition 98. *Definiere MAX durch*

$$MAX(p) := \begin{cases} 0 & \textit{falls } p \textit{ ein Maximum hat} \\ 1 & \textit{sonst} \end{cases}$$

Satz 116. $KDIV_2^{\{0,1\}} \not\leq_2 MAX$.

Beweis. Der Satz wird bewiesen durch den Nachweis von $KON \equiv_2 MAX$.
$\underline{MAX \leq_2 KON}$:
Definiere B durch $B(p)(0) := p(0)$ und $B(p)(i+1) := \max\{B(p)(i), p(i+1)\}$.
Dann gilt $MAX(p) = 0 \Longleftrightarrow KON \circ B(p) = 0$ und
$MAX(p) = 1 \Longleftrightarrow KON \circ B(p) = 1$.
$\underline{KON \leq_2 MAX}$:
Definiere B durch $B(p)(0) := p(0)$ und

$$B(p)(i+1) := \begin{cases} B(p)(i) & \text{falls } p(i) = p(i+1) \\ B(p)(i) + 1 & \text{sonst} \end{cases}$$

Dann gilt $KON(p) = 0 \Longleftrightarrow MAX \circ B(p) = 0$ und
$KON(p) = 1 \Longleftrightarrow MAX \circ B(p) = 1$. □

Es gibt viele weitere C^2-stetige Funktionen, die äquivalent zu KON sind. Die Beweise werden ausgelassen.

Definition 99. *Definiere N durch*

$$N(p) := \begin{cases} 0 & \textit{falls } M_p = \mathbb{N} \\ 1 & \textit{sonst} \end{cases}$$

Definition 100. *Definiere U durch*

$$U(p) := \begin{cases} 0 & \textit{falls } M_p \textit{ unendlich ist} \\ 1 & \textit{sonst} \end{cases}$$

Definition 101. *Definiere das Halteproblem K durch*

$$K\langle p, q\rangle := \begin{cases} 0 & \textit{falls } \psi_p(q) \textit{ existiert} \\ 1 & \textit{sonst} \end{cases}$$

Definition 102. *Definiere H durch*

$$H(p) := \begin{cases} 0 & \textit{falls } 0 \textit{ Häufungspunkt ist von } (\nu_Q(p(n)))_{n\in\mathbb{N}} \\ 1 & \textit{sonst} \end{cases}$$

H berechnet, ob 0 Häufungspunkt einer Folge rationaler Zahlen ist.

Definition 103. *Definiere H' durch*

$$H'\langle p, q\rangle := \begin{cases} 0 & \textit{falls } \rho(q) \textit{ ist Häufungspunkt von } (\nu_Q(p(n)))_{n\in\mathbb{N}} \\ 1 & \textit{sonst} \end{cases}$$

H berechnet, ob $\rho(q)$ Häufungspunkt einer Folge rationaler Zahlen ist.

Definition 104. *Definiere RAT durch*

$$RAT(p) := \begin{cases} 0 & \textit{falls } \rho(p) \in \mathbb{Q} \\ 1 & \textit{sonst} \end{cases}$$

RAT berechnet, ob $\rho(p)$ eine rationale Zahl ist.

Definition 105. *Definiere $DENSE$ durch*

$$DENSE(p) := \begin{cases} 0 & \textit{falls } \{\nu_Q(p(n)) | n \in \mathbb{N}\} \textit{ dicht in } \mathbb{R} \textit{ ist} \\ 1 & \textit{sonst} \end{cases}$$

Satz 117. *N, U, K, H, H', RAT und $DENSE$ sind nicht vollständig für $KDIV_2^{\{0,1\}}$.*

Beweis. $KON \equiv_2 N \equiv_2 U \equiv_2 K \equiv_2 H \equiv_2 H' \equiv_2 RAT \equiv_2 DENSE$ nach [St89] und [Myl92]. □

4.3.4 T ist nicht vollständig für $[\mathbb{B} \to \{0,1\}]^3$

Definition 106. *Definiere* $T : \mathbb{B} \longrightarrow \{0,1\}$ *durch*

$$T(p) := \begin{cases} 0 & \textit{falls } p \textit{ hat eine konstante Teilfolge} \\ 1 & \textit{sonst} \end{cases}$$

$$= \begin{cases} 0 & \textit{falls } (\exists n)(\forall k)(\exists l > k)p(l) = n \\ 1 & \textit{sonst} \end{cases}$$

$KDIV_3^{\{0,1\}} :\subseteq \mathbb{B} \to \{0,1\}$ wurde schon definiert. Es gilt
$KDIV_3^{\{0,1\}}(p)$

$$= \begin{cases} 0 & \text{falls } \lim_{m_1\to\infty} \lim_{m_2\to\infty} \lim_{m_3\to\infty}(p)\langle m_1, m_2, m_3\rangle = 0 \\ 1 & \text{falls } \lim_{m_1\to\infty} \lim_{m_2\to\infty} \lim_{m_3\to\infty}(p)\langle m_1, m_2, m_3\rangle = 1 \\ div & \text{sonst} \end{cases}$$

Satz 118. $KDIV_3^{\{0,1\}} \not\leq_2 T$.

Beweis. Der Beweis verläuft in folgenden Schritten:

1. $T \leq_2 \overline{Lim}_1^{\{0,1\}}$
2. $lim_2^{\{0,1\}} \not\leq_2 \overline{Lim}_1^{\{0,1\}}$
3. $lim_2^{\{0,1\}} \leq_2 lim_3^{\{0,1\}}$
4. $lim_3^{\{0,1\}} \not\leq_2 \overline{Lim}_1^{\{0,1\}}$ (folgt aus 2 und 3)
5. $lim_3^{\{0,1\}} \not\leq_2 T$ (aus $lim_3^{\{0,1\}} \leq_2 T$ würde $lim_3^{\{0,1\}} \leq_2 \overline{Lim}_1^{\{0,1\}}$ folgen, was zu einem Widerspruch führt.)
6. $lim_3^{\{0,1\}} \equiv_2 KDIV_3^{\{0,1\}}$
7. $KDIV_3^{\{0,1\}} \not\leq_2 T$ (folgt aus 5 und 6)

Die fehlenden Behauptungen werden in den folgenden Lemmas bewiesen. □

Lemma 37. $T \leq_2 \overline{Lim}_1^{\{0,1\}}$.

Beweis.

$$T(p) = \begin{cases} 0 & \text{falls } (\exists k)(\forall i)(\exists j > i)p(j) = k \\ 1 & \text{falls } (\forall k)(\exists i)(\forall j > i)p(j) \neq k \end{cases}$$

und

$$\overline{Lim}_1^{\{0,1\}}(p) = \begin{cases} 1 & \text{falls } (\forall k)(\exists i)(\forall j > i)p\langle k, j\rangle = 0 \\ 2 & \text{falls } (\forall k)(\exists i)(\forall j > i)p\langle k, j\rangle = 1 \\ 0 & \text{sonst} \end{cases}$$

Definiere B durch

$$B(p)\langle k,j\rangle := \begin{cases} 1 & \text{falls } p(j) = k \\ 0 & \text{sonst} \end{cases}$$

und A durch

$$A(x) := \begin{cases} 1 & \text{falls } x = 1 \\ 0 & \text{sonst} \end{cases}$$

Dann gilt $T(p) = A \circ \overline{Lim}_1^{\{0,1\}} \circ B(p)$ für alle $p \in \mathbb{B}$.
<u>Beweis</u>:
$T(p) = 1$
$\Longrightarrow (\forall k)(\exists i)(\forall j > i) p(j) \neq k$
$\Longrightarrow (\forall k)(\exists i)(\forall j > i) B(p)\langle k,j\rangle = 0$
$\Longrightarrow \overline{Lim}_1^{\{0,1\}} \circ B(p) = 1$
$\Longrightarrow A \circ \overline{Lim}_1^{\{0,1\}} \circ B(p) = 1$ und
$T(p) = 0$
$\Longrightarrow (\exists k)(\forall i)(\exists j > i) p(j) = k$
$\Longrightarrow (\exists k)(\forall i)(\exists j > i) B(p)\langle k,j\rangle = 1$
$\Longrightarrow \neg(\forall k)(\exists i)(\forall j > i) B(p)\langle k,j\rangle \neq 1$
$\Longrightarrow \neg(\forall k)(\exists i)(\forall j > i) B(p)\langle k,j\rangle = 0$
$\Longrightarrow \overline{Lim}_1^{\{0,1\}} \circ B(p) \in \{0,2\}$
$\Longrightarrow A \circ \overline{Lim}_1^{\{0,1\}} \circ B(p) = 0$.
Also folgt $T \leq_2 \overline{Lim}_1^{\{0,1\}}$. □

Lemma 38. $lim_2^{\{0,1\}} \not\leq_2 \overline{Lim}_1^{\{0,1\}}$

Beweis. Der Beweis ähnelt dem Beweis von Satz 94.
Falls $lim_2^{\{0,1\}} \leq_2 \overline{Lim}_1^{\{0,1\}}$, dann gibt es stetige Funktionen A und B mit

$$lim_2^{\{0,1\}}(p) = A(p, \overline{Lim}_1^{\{0,1\}} \circ B(p))$$

Aus $lim_2^{\{0,1\}}(p) \neq div$ folgt $\overline{Lim}_1^{\{0,1\}} \circ B(p) \neq 0$ (das Argument ist dasselbe wie in dem Beweis von Satz 91). Deshalb ist die vorige Aussage äquivalent zu

$$lim_2^{\{0,1\}}(p) = A(p, Lim_1^{\{0,1\}} \circ B(p))$$

Und wegen $lim_1^{\{0,1\}} \equiv_2 Lim_1^{\{0,1\}}$ folgt daraus

$$lim_2^{\{0,1\}}(p) = A(p, lim_1^{\{0,1\}} \circ B(p))$$

und damit $lim_2^{\{0,1\}} \leq_2 lim_1^{\{0,1\}}$.
Das ist ein Widerspruch. Also gilt $lim_2^{\{0,1\}} \not\leq_2 \overline{Lim}_1^{\{0,1\}}$. □

Lemma 39. $lim_2^{\{0,1\}} \leq_2 lim_3^{\{0,1\}}$.

Beweis. Definiere B durch $B(p) := \langle p, p, p, \ldots\rangle$.
Dann gilt $lim_2^{\{0,1\}}(p) = lim_3^{\{0,1\}} \circ B(p)$ für alle $p \in \mathbb{B}$. □

Lemma 40. $lim_3^{\{0,1\}} \equiv_2 KDIV_3^{\{0,1\}}$.

Beweis. Die Behauptung ist wahr wegen $lim_3^{\{0,1\}}(p) = INC \circ KDIV_3^{\{0,1\}}(p)$ und $KDIV_3^{\{0,1\}}(p) = DEC \circ lim_3^{\{0,1\}}(p)$ für alle $p \in \mathbb{B}$. $\square$

5 Zusammenfassung

Im ersten Teil dieser Arbeit wurde die Schwierigkeit von speziellen Funktionen diskutiert, deren Funktionswerte nur davon abhängen, welche der Eingabefolgen einen Wert ungleich 0 enthalten. Das Hauptergebnis war, dass die Reduzierbarkeit solcher Funktionen durch Abbildungen zwischen den Bäumen, die diese Funktionen repräsentieren, berechnet werden kann.

So kann berechnet werden, und zwar im Sinne der Typ-I-Berechenbarkeitstheorie, ob eine Funktion f dieses Typs reduzierbar auf eine andere Funktion g ist. Es bleibt die Frage offen, wie schwierig diese Berechnung sein mag. Ich glaube, dass diese Berechnung mindestens $NP - hart$ ist, aber das wurde hier nicht untersucht. Wenn dem so wäre, gäbe es keinen leichten Weg, die Funktionen dieses Typs miteinander zu vergleichen.

Ferner wurden spezielle Probleme diskutiert. Zwei der wichtigsten Resultate:

- Es gibt Funktionenmengen und mehrwertige Funktionen, die noch einfacher wie $LLPO_\infty$ sind, aber nicht stetig sind bzw. keine stetigen Funktionen enthalten.
- Es können Funktionenmengen $LLPO_{n,m}$ als Verallgemeinerungen von $LLPO_n$ definiert werden, die miteinander verglichen werden können. Andere Mengen wie $LLPO_n$ oder $MLPO_n$ sind Spezialfälle von ihnen.

Im zweiten Teil wurde diskutiert, welche Funktionen aus der C-Hierarchie C^n-vollständig sind. Das war eine offene Frage in meiner Diplomarbeit. Hier wurden iterierte Berechnungen des Grenzwertes von Folgen benutzt. Dieser Ansatz liefert auch eine Charakterisierung der Funktionen in der C-Hierarchie durch prädikatenlogische Ausdrücke. Wir haben hauptsächlich Funktionen $f :\subseteq \mathbb{B} \longrightarrow \mathbb{N}$ oder $f :\subseteq \mathbb{B} \longrightarrow \{0, 1\}$ behandelt. Analoge Resultate für Funktionen $f :\subseteq \mathbb{B} \longrightarrow \mathbb{B}$ wären erstrebenswert, sind aber noch nicht bewiesen worden.

Somit stimmen die verschiedenen Stufen der C-Hierarchie und die Definition von Funktionen durch iterierte Berechnung von Grenzwerten überein. Das ist ein Zeichen, dass die Definition der Stufen der C-Hierarchie in gewissem Sinne natürlich ist und dass sie benutzt werden können, um unstetige Funktionen miteinander zu vergleichen.

Da die Zeit nicht ausreichte, war ich nicht in der Lage, diese Begriffe mit anderen Definitionen von Graden der Unstetigkeit zu vergleichen. So ist es eine offene Frage, welche der anderen möglichen Definitionen von Stufen von Unstetigkeit ganz oder teilweise mit den Begriffen in dieser Arbeit zusammenfallen.

A Abbildungen

Abbildungsverzeichnis

Index

C Literatur

Literatur

[BiBr85] Errett Bishop, Douglas Bridges. *Constructive Analysis.* Springer, Berlin Heidelberg New York Tokyo 1985.

[BCSS97] Leonore Blum, Felipe Cucker, Michael Shub, Steve Smale. *Complexity and Real Computation.* Springer, New York 1997.

[Bra99] Vasco Brattka. *Recursive and Computable Operations over Topological Structures.* Dissertation, Informatik Berichte Nr. 255, Fernuniversität Hagen 1999.

[Bra03] Vasco Brattka. *Effective Borel measurability and reducibility of functions.* in: CCA 2003 International Conference on Computability and Complexity in Analysis, Informatik Berichte Nr. 302, Fernuniversität Hagen 2003.

[BrSe79a] I. N. Bronstein, K. A. Semendjajew. *Taschenbuch der Mathematik.* Verlag Harri Deutsch, Thun Frankfurt/Main 1979.

[BrSe79b] I. N. Bronstein, K. A. Semendjajew. *Taschenbuch der Mathematik. Ergänzende Kapitel.* Verlag Harri Deutsch, Thun Frankfurt/Main 1979.

[Dav58] Martin Davis. *Computability and Unsolvability.* Dover Publications, New York 1958.

[Harr83] Michael A. Harrison. *Formal Languages.* Kurs 1820, Fernuniversität Hagen 1996.

[Herm78] Hans Hermes. *Aufzählbarkeit, Entscheidbarkeit, Berechenbarkeit.* Springer, Berlin Heidelberg New York 1978.

[Herr81] Horst Herrlich, Walter Tholen, D. Franke. *Einführung in die Topologie.* Kurs 1251, Fernuniversität Hagen 1981.

[Herr98] Horst Herrlich, Walter Tholen. *Topologie I.* Kurs 1354, Fernuniversität Hagen 1998.

[Hert93a] Peter Hertling. *Topologische Komplexitätsgrade von Funktionen mit endlichem Bild.* Informatik Berichte Nr. 152, Fernuniversität Hagen 1993.

[Hert93b] Peter Hertling. *Stetige Reduzierbarkeit auf* Σ^ω *von Funktionen mit zweielementigem Bild und von zweistetigen Funktionen mit diskretem Bild.* Informatik Berichte Nr. 153, Fernuniversität Hagen 1993.

[Hert96] Peter Hertling. *Unstetigkeitsgrade von Funktionen in der effektiven Analysis.* Dissertation, Informatik Berichte Nr. 208, Fernuniversität Hagen 1996.

[Ko91] Ker-I Ko. *Complexity Theory of Real Functions.* Birkhäuser, Boston Basel Berlin 1991.

[Myl92] Uwe Mylatz. *Vergleich unstetiger Funktionen in der Analysis.* Diplomarbeit, Fernuniversität Hagen 1992.

[Pour89] Marian B. Pour-El. *Computability in Analysis and Physics.* Springer, Berlin Heidelberg New York 1989.

[St89] Thorsten von Stein. *Vergleich nicht konstruktiv lösbarer Probleme in der Analysis.* Diplomarbeit, Fernuniversität Hagen 1989.

[Su82] Ivan Hal Sudborough. *Complexity of Turing Machine Computations.* Kurs 1684, Fernuniversität Hagen 1982.

[Wei87] Klaus Weihrauch. *Computability.* Springer, Berlin Heidelberg 1987.

[WeiHn88] Klaus Weihrauch, Bernd Heinemann. *Logik für Informatiker.* Kurs 1825, Fernuniversität Hagen 1988.

[Wei91] Klaus Weihrauch. *The Lowest Wadge Degrees of Subsets of the Cantor Space.* Informatik Berichte Nr. 107, Fernuniversität Hagen 1991.

[Wei92a] Klaus Weihrauch. *The TTE-Interpretation of Three Hierarchies of Omniscience Principles.* Informatik Berichte Nr. 130, Fernuniversität Hagen 1992.

[Wei92b] Klaus Weihrauch. *The Degrees of Discontinuity of some Translators Between Representations of the Real Numbers.* Informatik Berichte Nr. 129, Fernuniversität Hagen 1992.

[WeiHr94] Klaus Weihrauch, Peter Hertling. *On the Topological Classification of Degeneracies.* Informatik Berichte Nr. 154, Fernuniversität Hagen 1994.

[Wei94] Klaus Weihrauch. *Typ 2 Berechenbarkeit (Grundlagen der Effektiven Analysis).* Kurs 1681, Fernuniversität Hagen 1994.

[Wei95] Klaus Weihrauch (Editor). *Computability and Complexity in Analysis.* Informatik Berichte Nr. 190, Fernuniversität Hagen 1995.

[Wei98] Klaus Weihrauch (Editor). *Computability and Complexity in Analysis.* Informatik Berichte Nr. 235, Fernuniversität Hagen 1998.

[WeiHr99a] Klaus Weihrauch, Peter Hertling. *Einführung in die Berechenbare Analysis.* Kurs 1681, Fernuniversität Hagen 1999.

[WeiHr99b] Klaus Weihrauch, Peter Hertling. *Ausgewälte Kapitel aus der Berechenbaren Analysis.* Kurs 1838, Fernuniversität Hagen 1999.

[Wei00] Klaus Weihrauch. *Computable Analysis.* Springer, Berlin Heidelberg 2000.

D Lebenslauf

Geboren: 1.5.1953 in Lüneburg.

Grundschule 23.4.1960 - März 1964 in der Heiligengeistschule in Lüneburg.

Gymnasium März 1964 - 12.5.1972 im Johanneum Lüneburg im mathematisch-naturwissenschaftlichen Zweig. Abitur am 12.5.1972 mit kleinem Latinum.

Studium an der Pädagogischen Hochschule Lüneburg im WS 72/73.

Zivildienst im Krankenhaus Veerßen 5.3.1973 - 30.6.1974.

Fortsetzung des Studiums an der Pädagogischen Hochschule WS 74/75 - SS 77.

Studium an der Universität Hamburg mit Hauptfach Philosophie und den Nebenfächern Mathematik und Indologie WS 77/78 - SS 79.

Beschäftigung als Programmierer im Softwarehaus Alpha-EDV-Beratung in Bissendorf 1.12.1979 - 31.3.1981.

Fortsetzung des Studiums an der Universität Hamburg SS 81 - SS 82. Erwerb von Programmierkenntnissen in verschiedenen Sprachen.

Während des Studiums Beschäftigung als Busfahrer bei der Firma Ernst Röhlsberger im Stadtverkehr in Lüneburg (Dez 77 - Nov 79 und Apr 81 - Okt 85).

Ausbildung zum Finanzbuchhalter an der Volkshochschule Lüneburg Februar 1982 - April 1985 mit erfolgreichem Abschluss. Der Kurs beinhaltete Buchführung, Bilanzierung, Kosten- und Leistungsrechnung, Betriebliches Steuerrecht, Recht und Finanzen und ein Buchführungspraktikum.

Studium an der Fernuniversität Hagen mit Hauptfach Informatik und Nebenfach Mathematik vom WS 82/83 an. Vordiplom am 6.9.1985. Im Hauptstudium Vertiefungsfach Praktische Informatik mit Schwerpunkt Datenbanken und Programmierung. Thema der Diplomarbeit „Vergleich unstetiger Funktionen aus der Analysis" im Gebiet Theoretische Informatik. Diplom am 11.6.92 mit der Gesamtnote „Mit Auszeichnung".

Seit 1992 beschäftigt als wissenschaftlicher Mitarbeiter an der Universität Lüneburg in der Lehre (Grundkurse, Datenbanksysteme, Programmierung und 3D-Animation) und als EDV-Administrator in der Universitätsbibliothek.

Ebenfalls seit 1992 Beschäftigung als Mentor für die Fernuniversität Hagen am Fernstudienzentrum der Universität Lüneburg. Betreuung von Kursen im Grundstudium (Technische, Praktische und Theoretische Informatik) und im Hauptstudium (Praktische und vor allem Theoretische Informatik).