Rekursionstheorie

Heike Mildenberger

Skript zu einer
zweistündigen Vorlesung „Rekursionstheorie“
Fassung vom 12.06.2023

KAPITEL 1

Berechenbarkeit

Literatur: Bücher: [**4**], [**6**], [**7**], [**3**], [**10**].

Artikel: Shelah [**9**].

Es sei $\mathbb{N} = \{0, 1, 2, \dots\}$ die Menge der natürlichen Zahlen. Wir stellen uns unter einem Algorithmus eine endliche Folge von endlich langen Anweisungen vor. Eine Funktion $f\colon \mathbb{N} \to \mathbb{N}$ heißt berechenbar, wenn es einen Algorithmus gibt, der für alle $n \in \mathbb{N}$ auf Eingabe von n hin endlich viele Schritte lang rechnet (oder arbeitet) und dann $f(n)$ ausgibt. Dies ist natürlich nur eine vage Beschreibung und noch keine Definition.

Wir beschäftigen uns in diesem Kapitel mit den folgenden Fragen:

1. Welche Funktionen $f\colon \mathbb{N} \to \mathbb{N}$ sind berechenbar?
2. Hängt der Berechenbarkeitsbegriff gar von unserer Präzisierung des Begriffes „Algorithmus" ab?
3. Welche Präzisierungen gibt es?

Wir beginnen mit der dritten Frage. Es gibt die folgenden mathematischen Definitionen für den oben unvollständig beschriebenen Begriff der Berechenbarkeit:

Turing-Berechenbarkeit,

Berechenbarkeit durch Flussdiagramme,

Registermaschinen-Berechenbarkeit,

μ-rekursive Funktionen,

C^{++}-Programme, Prolog-Programme, Lisp-Programme, Basic-Programme, Java-Programme,

Perl-Scripte

und viele mehr. Wir stellen eine Version der Turing-Berechenbarkeit vor, danach werden wir die μ-Rekursivität definieren und den Satz von Turing (1936), dass jede μ-berechenbare Funktion auch Turing-berechenbar ist, beweisen. Über etliche andere Präzisierungen können Sie im Buch von Odifreddi lesen. Die Churchsche These (Alonzo Church, 1936) sagt, dass alle Präzisierungen der Berechenbarkeit äquivalent sind. Aus Zeitgründen betrachten wir nur zwei Präzisierungen und zunächst nur eine Richtung des Beweises, dass die beiden äquivalent sind. Die Technik dieses Beweises, nämlich die Simulierung jedes der endlich vielen Befehlstypen einer Berechnungsart durch eine Kombination von Befehlen in der anderen Berechnungsart, ist repräsentativ für alle effektiven Übersetzungen.

1. Turing-Maschinen

Definition 1.1. Turing 1936, Post 1936. Eine Turing-Maschine M besteht aus

(1) einem endlichen Alphabet $\mathbb{A} = \{s_0, s_1, \ldots, s_n\}$ mit zwei ausgezeichneten Buchstaben: s_0 steht für das Zeichen für die leere Feldinschrift (also ein von der Maschine womöglich noch nicht gelesenes Feld) und $s_1 = 1$ steht für den Zähler. s_0 darf jedoch auch als Helfer für Berechnungen genommen werden. Die beiden ausgezeichneten Zeichen müssen unterschiedlich sein.

(2) einer endlichen Menge $\{q_0, q_1, \ldots, q_n\}$ von Zuständen, worunter es wieder zwei besondere Zustände gibt: q_0, den Anfangszustand, und q_f, den Endzustand. Wieder gilt, dass es mindestens zwei Zustände geben muss.

(3) einer endlichen konsistenten Programmtafel $\{I_0, I_1, \ldots, I_n\}$. Jedes I_j ist von einer der drei folgenden Sorten:

(q_a, s_b, s_c, q_d): falls im Zustand q_a auf dem Arbeitsfeld s_b steht, schreibe stattdessen s_c und gehe in den Zustand q_d.

(q_a, s_b, R, q_d): falls im Zustand q_a auf dem Arbeitsfeld s_b steht, gehe mit dem Lesekopf eins nach rechts und in den Zustand q_d.

(q_a, s_b, L, q_d): selbes für links.

Eine Programmtafel heißt konsistent, wenn sie nicht zwei I_j mit identischen q_a, s_b und unterschiedlichen Fortsetzungen enthält.

Definition 1.2. Wir schreiben A^* für die Menge der Wörter über A, also der endlichen Zeichenreihen des Typs $a_0 \ldots a_{n-1}$ mit $n \in \mathbb{N}$ und $a_i \in A$. Hierbei steht $\square$ für das leere Wort.

A^* kann man mit $A^{<\omega} = \{s \colon \{0, \ldots, n-1\} \to A : n \in \mathbb{N}\}$ identifizieren.

Übung 1.3. Wir halten ein Alphabet A und eine Zustandsmenge Z fest. Wieviele Turing-Maschinen gibt es über A und Z?

Übung 1.4. Überlegen Sie sich, wann zwei Turing-Maschinen isomorph sind. Wieviele Turing-Maschinen gibt es insgesamt modulo Isomorphie?

Definition 1.5. Eine *Konfiguration* einer Turing-Maschine ist ein endlich langes Tupel der Form

$(\sigma_0, \ldots, \sigma_j, q_a, \sigma_{j+1}, \ldots \sigma_z)$.

Hierbei sind die Einträge $\sigma_i \in \mathbb{A}$. Der Lesekopf steht auf σ_j und die Maschine ist im Zustand q_a. Links von σ_0 und rechts von σ_z stehen nur Leerzeichen s_0 auf dem Band, das mindestens nach rechts unendlich lang ist.

Definition 1.6 (Turing 1936, Post 1936). Sie $n \geq 1$. Eine Funktion $f \colon \mathbb{N}^n \to \mathbb{N}$, $f(x_1, \ldots, x_n)$ ist Turing-berechenbar wenn es ein Turing-Maschine gibt, die für alle x_i beginnend in der Konfiguration

$(s_0, x_1, s_0, x_2, s_0, \ldots, s_0, x_n, s_0, q_0)$ nach Befolgen der Befehle in der Turingtafel nach endlich vielen Schritten die Endstellung $(s_0, f(x_1, \ldots, x_n), s_0, q_f)$ erreicht. Hierbei stehen nicht die Zahlen $x_i \in \mathbb{N}$ auf dem Band, sondern es steht auf x_i aufeinanderfolgenden Feldern jeweils der Zähler s_1 auf dem Turingband, denn das Alphabet ist ja nur endlich. Wir notieren die Zahlen x_i also unär.

Übung 1.7. Wieviele Funktionen $f\colon \mathbb{N} \to \mathbb{N}$ gibt es? Folgern Sie: Es gibt Funktionen, die nicht Turing-berechenbar sind.

Lemma 1.8. *Die Funktion* $(s_0, x_1, s_0, q_0) \mapsto (s_0, x_1, s_0, x_1, s_0, q_f)$ *auf* $\mathbb{N}$ *ist Turing-berechenbar.*

Proof. Wir nennen s_1 einfach 1 und arbeiten mit einem weiteren Buchstaben 2.

Wir skizzieren ein Flussdiagramm:

(1) Suche nach der am weitestesten rechts stehende 1, die vom Input geblieben ist, und schreibe eine 2 War sie die letzte?
(2) Falls ja, stelle den Input wieder her und gehe an das rechte Ende des Outputs und eins weiter und stoppe.
(3) Falls nein, gehe zur rechtesten 1 im Output. Gehe eine Zelle nach rechts und drucke eine 1. Gehe zurück über einen Eintrag s_0 zum Rest des Inputs und zu Zeile 1.

Nun wird dies in eine Turing-Maschine mit drei Buchstaben und 12 Zuständen übersetzt:

Wir nennen das einzelne s_0 zwischen dem aktuellen Abwandlung von x_1 und dem schon angefangenen rechts stehenden Kopie von x_1 das "Trennungs-s_0" oder das Trennungszeichen.

Wir nennen das aktuell links von der Trennungs-s_0 stehende Wort "linkes Wort", das hintere "rechtes Wort".

Wir denken uns folgende Zustände aus:

(q_0) Anfang, gehe auf den letzten Buchstaben des linken Worts.
(q_1) Lesekopf steht am Anfang auf dem letzten Buchstaben des aktuellen linken Worts. Suche die am weitestens rechts stehen 1 im linken Wort. Wenn es keine Eins gibt, gehe zu q_{10}, sonst zu q_2.
(q_2) Wissen: noch eine Eins im linken Wort da. Überschreibe die rechteste 1 im linken Wort durch eine 2. Prüfe, ob noch eine weitere 1 da ist, wenn nein q_3, sonst q_4.
(q_4) Wissen: noch eine 1 im linken Wort. Lesekopf auf dem Weg nach rechts, aber vor dem Trennngszeichen. Ab Trennungszeichen: Gehe zu q_5.
(q_5) Wissen noch eine 1 im linken Wort. Lesekopf auf dem Weg nach rechts, nach dem Trennungszeichen. Ende erreicht? Dann q_6.
(q_6) Hänge 1 an das rechte Wort und gehe zurück (nach links) auf das Trennungszeichen und dann auf das eins links stehende Feld und

zu q_1. Dies ist die Hauptschleife im Programm, die nun geschlossen ist.

(q_3) Wissen: Keine 1 mehr im linken Wort. Nun kommen die Endarbeiten. Bewegung nach rechts ans Ende der Zweien, von denen mindestens eine da ist. Wenn Ende (also die Trennungs-s_0) gefunden, dann q_7.

(q_7) Restaurierung von 22222 zu 11111 im linken Wort, von rechts nach links. Wenn fertig, q_8.

(q_8) Auf dem Weg nach rechts, vor dem Trennungszeichen. Danach q_9.

(q_9) Auf dem Weg nach rechts, nach dem Trennungszeichen. Wenn neuerlich eine s_0 gefunden, dann schreibe eine 1 und gehe nach q_{10}.

(q_{10}) Gehe eins nach rechts und nach q_f, zum Herstellen von (s_0, s_0, s_0, q_f) oder von $(s_0, x_1, s_0, x_1, s_0, q_f)$. Der Lesekopf steht am Ende auf der letzten s_0.

(q_f) Ende.

Daher erhalten wir nun folgende **Programmtafel**:

(q_0, s_0, L, q_1)

(q_1, s_0, R, q_9) leeres Wort? Ja.

$(q_1, 2, L, q_1)$ gehe an das linke Ende der Zweien zur nächsten 1 dort.

$(q_1, 1, 2, q_2)$ die rechteste noch verbleibende 1 des linken Worts wird durch 2 ersetzt

$(q_2, 2, L, q_2)$

(q_2, s_0, R, q_3) letzte 1 des linken Worts weg, gehe zu den Endarbeiten, die mit q_3 beginnen.

$(q_2, 1, R, q_4)$ noch eine 1 im linken Wort. Dies ist die wichtigste Fallunterscheidung. Gehe in die Hauptschleife, die mit q_4 beginnt.

$(q_4, 1, R, q_4)$ (Bewegung ans rechte Ende mit Wissen ums Zurückkehren)

$(q_4, 2, R, q_4)$

(q_4, s_0, R, q_5) Ankommen beim Trennungszeichen

$(q_5, 1, R, q_5)$ Weiter auf dem Weg nach rechts, mit Wissen, dass noch eine Eins im linken Wort steht.

$(q_5, s_0, 1, q_6)$ Schreibe statt s_0 eine 1 an das rechte Ende des rechten Worts

$(q_6, 1, L, q_6)$ Nach dem Anhängen der 1 ist der Kopf auf dem Rückweg nach links an einer Stelle rechts vom Trennungszeichen

(q_6, s_0, L, q_1) Nun ist der Lesekopf auf der am weitestesten rechts stehenden 2 des linken Worts. Die zentrale Schleife schließt sich durch Gehen nach q_1.

$(q_3, 2, R, q_3)$ Bewegung nach rechts ans rechte Ende der Zweien

(q_3, s_0, L, q_7) beim Trennungszeichen angekommen

$(q_7, 2, 1, q_7)$ Überschreiben der Zweien durch Einsen, von rechts nach links.

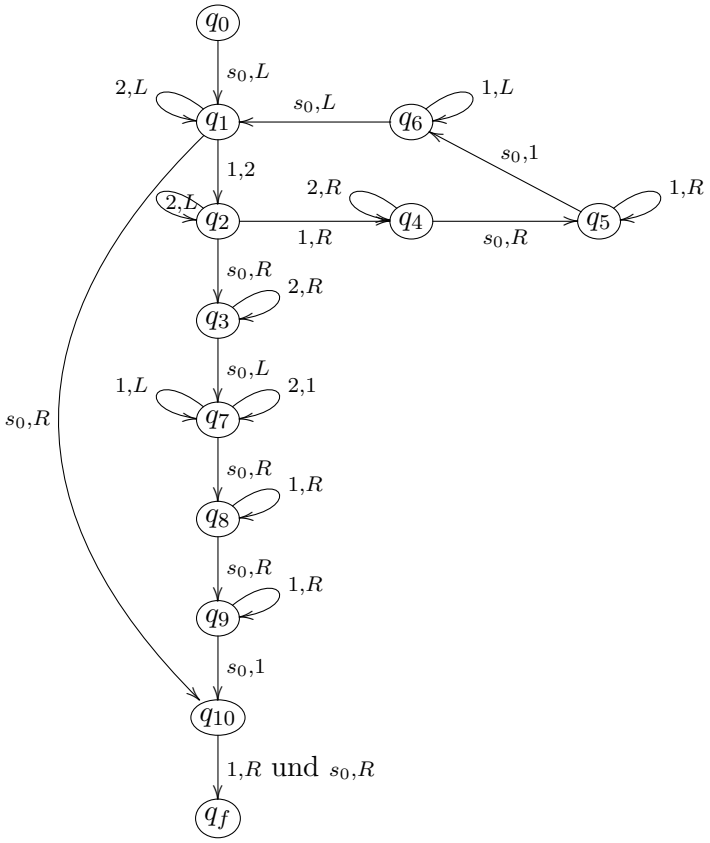
$(q_7, 1, L, q_7)$
(q_7, s_0, R, q_8) Überschreibung fertig

$(q_8, 1, R, q_8)$ auf dem letzten Gang nach rechts vor dem Trennungszeichen
(q_8, s_0, R, q_9) auf dem letzten Gang nach rechts beim Trennungszeichen

$(q_9, 1, R, q_9)$ auf dem letzten Gang nach rechts nach dem Trennungszeichen
$(q_9, s_0, 1, q_{10})$ letzte Eins angehängt

(q_{10}, s_0, R, q_f) Bewegen des Lesekopfs zur normierten Endstellung, auf dem ersten s_0 nach dem kopierten Wort.
$(q_{10}, 1, R, q_f)$

Dank xy-pictures haben wir nun folgende Skizze



□

Auf dem Web gibt es etliche Turingmaschinen-Simulatoren.

2. μ-Rekursion

Nun stellen wir eine weitere Möglichkeit einer mathematischen Definition der Berechenbarkeit vor. Diese wurde im selben Jahr 1936 von Kleene gefunden.

DEFINITION 1.9 (Dedekind 1888). Eine Funktion f wird aus g und h durch *primitive Rekursion* definiert, wenn

$$\begin{aligned} f(\vec{x}, 0) &= g(\vec{x}), \\ f(\vec{x}, y+1) &= h(\vec{x}, y, f(\vec{x}, y)). \end{aligned}$$

DEFINITION 1.10 (Dedekind 1888, Skolem 1923, Gödel 1931).

(a) Die Menge der *primitiv rekursiven Funktionen* ist die kleinste Menge von Funktionen, die
   (1) die *Ausgangsfunktionen* $o(x) = 0$, $s(x) = x+1$, und für alle $n \in \mathbb{N} \setminus \{0\}$ für $i < n$ die *Projektionsfunktionen* $p_i^n(x_0, \ldots, x_{n-1}) = x_i$ enthält und
   (2) unter *Verkettung* abgeschlossen ist, d. h., wenn $g_0, \ldots, g_{m-1}$ und h in der Menge sind, dann ist auch
   $$f(\vec{x}) = h(g_0(\vec{x}), \ldots, h_{m-1}(\vec{x}))$$
   in der Menge und
   (3) unter primitiver Rekursion abgeschlossen ist.

(b) Ein Prädikat $R \subseteq \mathbb{N}$ heißt *primitiv rekursiv*, wenn seine charakteristische Funktion primitiv rekursiv ist.

ÜBUNG 1.11. Warum tritt in h bei der primitiven Rekursion nur der letzte Wert von f auf, dürfte man nicht auch auf frühere Werte zurückgreifen? Was meinen Sie, vergößert diese Möglichkeit die Menge der primitiv rekursiven Funktionen?

ÜBUNG 1.12. Die Addition und die Multiplikation und die Exponentiation auf den natürlichen Zahlen sind zweistellige primitiv rekursive Funktionen.

2.1. Arithmetisierung, Kodierung von Folgen, Komponenten aus Folgen, Längen. Seien p_1, p_2 ... die Primzahlen in aufsteigender Reihung, und sei

$$\begin{aligned} \langle\cdot\rangle \colon \mathbb{N}^{<\omega} &\to \mathbb{N} \\ \langle n_0, \ldots, n_{m-1}\rangle &= \prod_{i=0}^{m-1} p_i^{n_i+1} \\ \langle\emptyset\rangle &= 1 \end{aligned}$$

unsere Injektion aller endlichen Folgen natürlicher Zahlen in die natürlichen Zahlen (Sie können eine andere rekursive Injektion wählen.) Die Dekodierungesfunktionen: *Länge*, *Komponente*, *Folge* (dies ist eine Relation),

Verkettung, auch Konkatenation und *Anfangsstück* (Ralation) sind auch primitiv rekursiv:

$$\begin{aligned}
x \mapsto \lg(x) &= \text{das größte } n \le x, \text{ so dass } p_{n-1}|x,\\
(x,n) \mapsto (x)_n &= (\text{das größte } r \le x, \text{ so dass } r \ge 1 \wedge p_n^r|p) - 1\\
&\quad (\text{—dies ist für } n \ge \lg(x) \text{ nicht definiert—}),\\
\mathrm{Seq}(x) &\Leftrightarrow (\forall n \le x)((x)_n \text{ definiert } \to n < \lg(x)),\\
(x,y) \mapsto x * y &= \prod_{i<\lg(x)} p_i^{(x)_i+1} \prod_{i<\lg(y)} p_{\lg(x)+i}^{(y)_i+1},\\
x \sqsubseteq y &\Leftrightarrow \mathrm{Seq}(x) \wedge \mathrm{Seq}(y) \wedge (\exists u \le y)(\mathrm{Seq}(u) \wedge x * u = y)).
\end{aligned}$$

PROPOSITION 1.13 (Rekursion über den Werteverlauf (Skolem 1923, Péter 1934)). *Die* Geschichtsfunktion zu f *wird definiert durch*

$$\hat{f}(\vec{x}, y) = \langle f(\vec{x}, 0), \ldots, f(\vec{x}, y)\rangle.$$

Es seien g und h primitiv rekursiv und f definiert durch

$$\begin{aligned}
f(\vec{x}, 0) &= g(\vec{x}),\\
f(\vec{x}, y+1) &= h(\vec{x}, y, \hat{f}(\vec{x}, y)).
\end{aligned}$$

Dann ist f primitiv rekursiv.

Beweis: Wir zeigen, dass $\hat{f}$ primitiv rekursiv ist. Da $f(\vec{x}, y) = (\hat{f}(\vec{x}, y))_y$, ist dann auch f primitiv-rekursiv.

$$\begin{aligned}
\hat{f}(\vec{x}, 0) &= \langle f(\vec{x}, 0)\rangle\\
&= \langle g(\vec{x})\rangle,\\
\hat{f}(\vec{x}, y+1) &= \langle f(\vec{x}, 0), \ldots, f(\vec{x}, y+1)\rangle\\
&= \langle f(\vec{x}, 0), \ldots, f(\vec{x}, y)\rangle * \langle f(\vec{x}, y+1)\rangle\\
&= \hat{f}(\vec{x}, y) * \langle h(\vec{x}, y, \hat{f}(\vec{x}, y))\rangle.
\end{aligned}$$

Also greift $\hat{f}(\vec{x}, y+1)$ nur auf den vorigen Wert zurück, und $\hat{f}$ is primitiv rekursiv, weil die Kodierung endlicher Folgen und die Verkettung primitiv rekursiv sind. □

ÜBUNG 1.14. Die Menge der primitiv rekursiven Funktionen enthält genau diejenigen Funktionen, deren Berechnung durch ein Flussdiagramm mit for-Schleifen und ohne Anfrage-Rauten und ohne while-Schleifen skizziert wird.

DEFINITION 1.15 (Kleene 1936). Eine Funktion f wird aus g durch *eingeschränkte μ-Rekursion* definiert, wenn

$$\begin{aligned}
&(\mu 1) \qquad (\forall \vec{x})(\exists y)(g(\vec{x}, y) = 0),\\
&(\mu 2) \qquad f(\vec{x}) = \min\{y \,:\, g(\vec{x}, y) = 0\} = \mu y(g(\vec{x}, y) = 0).
\end{aligned}$$

BEMERKUNG 1.16. Übersehen Sie die nicht leicht überprüfbare Bedingung ($\mu 1$) nicht! Warum diese Bedingung so komplex ist, werden wir später sehen. Die Bedingung ($\mu 1$) heißt Einschränkung.

DEFINITION 1.17 (Kleene 1936). Die Menge der *μ-rekursiven* Funktion ist die kleinste Menge von Funktionen, die

(1) die Ausgangsfunktionen $o(x) = 0$, $S(x) = x + 1$, $p_i^n(\vec{x}) = x_i$ enthält und
(2) unter Verkettung, primitiver Rekursion und unter eingeschränkter μ-Rekursion abgeschlossen ist.

ÜBUNG 1.18. In der Definition der μ-Rekursivität kann man die primitive Rekursion weglassen, denn sie ist aus den anderen Aufbauschritten herstellbar, wenn man die Augangsfunktionen um $+$, Multiplikation und die charakteristische Funktion der $<$-Relation erweitert. Dieses wird im Logikbuch von Ziegler [**12**, Satz 16.1] ausführlich gezeigt.

SATZ 1.19 (Turing 1936). *Jede μ-rekursive Funktion ist Turing-berechenbar.*

Beweis: Dies wird induktiv über den Aufbau der μ-rekursiven Funktionen gezeigt. Wir zeigen mehr: es gibt eine Turing-Maschine, die die Funkion berechnet und den Input stehen lässt, so dass $f(\vec{x})$ nach (d.h. rechts von) einem s_0 und dem Input steht. Damit lässt sich die Induktion leichter durchführen.

Für die einstellige Identität haben wir dies schon oben in Lemma 1.8 gezeigt. Bei der Nullfunktion, der Nachfolgerfunktion und bei den Projektionen verfährt man ähnlich. Nun sind noch die Komposition, die primitive Rekursion und die μ-Rekursion nachzurechnen. Wir zeigen dies exemplarisch für die primitive Rekursion.

Zu den Zeiten der Anfrage habe das Band folgende Form:

$$x_1 s_0 x_2 s_0 \dots s_0 x_n s_0 y s_0 s s_0 x_1 s_0 \dots s_0 x_n s_0 t s_0 f(\vec{x}, t)$$

Sei $f(\vec{x}, 0) = g(\vec{x})$ und$f(\vec{x}, y + 1) = h(\vec{x}, y, f(\vec{x}, y))$, und seien g und h durch die TM M_1 und M_2 berechenbar. Wir führen zwei neue Zähler s und t ein, so dass jederzeit $s + t = y$. Am Anfang ist $s = y$. In jeder „Berechnungsrunde" wird t um 1 erhöht und s um 1 erniedrigt.

Flussdiagramm:

Schreibe rechts des Inputs eine Kopie s von y und eine Kopie der x_i und $t = 0$.

Berechne $f(\vec{x}, 0)$ durch M_1.

$s = 0$?

Falls $s \neq 0$, erhöhe t um 1 und erniedrige s um 1 und berechne durch M_2 mit dem hintersten $\vec{x}$ und $t+1$ als Input den Wert $z = h(\vec{x}, t, f(\vec{x}, t)) = f(\vec{x}, t + 1)$ und schreibe diesen direkt rechts neben das neue $t + 1$.

Gehe zur Frage.

Falls $s = 0$, lösche alles zwischen dem Input und z und schreibe z direkt neben den Input. Lese $z = f(\vec{x}, t)$ ab. Nun ist $t = y$.

Nun muss man wieder mit viel Geduld das Flussdiagramm in eine Turing-Maschine umsetzen. □

Die Umkehrung gilt auch: jede Turing-berechenbar Funktion ist μ-berechenbar. Wir werden später darauf zurückkommen. An unserer dritten Frage könnten wir das ganze Semester (und länger) arbeiten. Doch wir werden nicht nach enzyklopädischer Breite streben, sondern lieber interessante Phänomene und Meilensteine in der Entwicklung der Rekursionstheorie anschauen.

Durch lange Erfahrung wird die nicht-mathematische Antwort auf die zweite Frage gegeben durch:

THESE 1.20 (Church 1936). *Alle Präzisierung des Berechenbarkeitsbegriffs sind äquivalent zur μ-Rekursivität.*

Nun wenden wir uns der ersten Frage zu und schauen einige interessante Beispiele an.

3. Primitiv rekursiv versus μ-rekursiv

Braucht man den μ-Operator überhaupt? Ist die Klasse (well, Menge) der primitiv rekursiven Funktionen echt kleiner als die Klasse der μ-rekursiven Funktionen? Diese Frage wurde um 1950 durch Rosza Péter gelöst. Grzegorczyk führte 1953 eine Klassifizierung der primitiv rekursiven Funktionen ein.

Wir definieren eine Funktion durch doppelte Rekursion:

DEFINITION 1.21 (Die Ackermann-Funktion). Wir definieren $A\colon \mathbb{N} \times \mathbb{N} \to \mathbb{N}$:

$$\begin{aligned} A(0, y) &= y + 1 \\ A(x + 1, 0) &= A(x, 1) \\ A(x + 1, y + 1) &= A(x, A(x + 1, y)). \end{aligned}$$

ÜBUNG 1.22. Die Ackermann-Funktion ist μ-rekursiv.

LEMMA 1.23. *Für jede primitiv-rekursive Funktion f gibt es ein m, so dass $(\forall n) f(n) < A(m, n)$.*

Beweisschritte:
$A(m, n) > n$ für alle n,
$A(m + 1, n) \geq A(m, n + 1)$,
$A(m, n)$ ist monoton in n,
$A(n, m)$ ist monoton in m,
Ausgangsfunktionen $\leq A(2, n)$.
Wenn $f \leq A(m, \cdot)$ und $g \leq A(m', \cdot)$, dann ist $f \circ g \leq A(m + m' + 2, \cdot)$,

Wenn $g \leq A(m, \cdot)$ und $h \leq A(m, \cdot)$, dann gilt für die primitive Rekursion f aus g und h, $f \leq A(m+1, \cdot)$. Den ausführlichen Beweis kann man in [**7**] nachlesen. □

SATZ 1.24 (Rosza Péter 1951). *Die Ackermann-Funktion ist nicht primitiv rekursiv.*

Beweis: Wenn $A(n, n)$ primitiv rekursiv wäre, so gäbe es nach dem obigen Lemma ein m, so dass für alle n $A(n, n) < A(m, n)$. Nun nehmen wir $n = m$ und erhalten einen Widerspruch. □

4. Primitiv-rekursive obere Schranken für die van der Waerden-Zahlen

In diesem Abschnitt präsentieren wir Teile einer Arbeit von Saharon Shelah [**9**].

Wir stellen uns von nun an die natürlichen Zahlen in der *von Neumannschen Form* vor und haben also für jedes n:

$$n = \{0, 1, \ldots, n-1\}.$$

Falls Sie die von Neumannschen natürlichen Zahlen nicht kennen, fassen Sie die Gleichung als Definition auf. Diese Gleichung ist übrigens ein Grund dafür, warum es in vielen Situationen günstiger ist, mit 0 statt mit 1 zu zählen zu beginnen.

DEFINITION 1.25. $P \subseteq \{0, 2, \ldots, N-1\}$ heißt *arithmetische Progression der Länge t*, wenn es $r \geq 0$, $s > 0$, gibt, so dass $P = \{r + s \cdot i \,:\, 0 \leq i < t\}$.

SATZ 1.26 (van der Waerden 1927). *Für alle positiven ganzen Zahlen n und c gibt es eine ganze Zahl N, so dass es für jede Färbung von $\{0, 1, \ldots, N-1\}$ in c Farben eine monochromatische arithmetische Progression der Länge n gibt.*

FRAGE 1.27. *Wie groß sind die kleinsten $N = W(n, c)$, die van der Waerdens Satz bezeugen? Wir halten c fest. Ist $n \mapsto W(n, c)$ eine primitiv-rekursive Funktion?*

ÜBUNG 1.28.

(a) Wenn $n \mapsto W(n, c)$ existiert, dann ist es eine μ-rekursive Funktion.

(b) Wenn f μ-rekursiv ist und $f \leq h$ (punktweise) ist und h primitiv-rekursiv ist, dann ist f primitiv-rekursiv. Hinweis: ein μ-Operator mit einer primitiv-rekursiven Oberschranke des Suchbereichs für $g(\vec{x}, y) = 0$ mit einem primitiv-rekursiven g kann durch primitiv-rekursive Schritte simuliert werden.

(c) Wenn $f \leq g$ und g primitiv-rekursiv ist, dann ist f nicht unbedingt μ-rekursiv. Hinweis: Nehmen Sie für f zum Beispiel die charakteristische Funktion des Halteproblems 3.24 oder auch das Problem K aus 3.27.

Shelahs Ergebnis wird eine etwas stärkere Ramsey-Eigenschaft behandeln. Hierzu brauchen wir die folgende Definition:

DEFINITION 1.29. ${}^k n = \{\eta = (\eta(0), \dots, \eta(k-1)) : \eta(i) \in n\}$. Sei $1 \leq \ell \leq k$. Wir werden ℓ-parametrische Teilmengen von ${}^k n$ definieren. Sei

$$k = A_0 \cup A_1 \cup \dots \cup A_\ell$$

eine Partition in nicht leere A_i für $i < \ell$, A_ℓ darf leer sein. Sei $f\colon A_\ell \to n$ eine Funktion. Wir definieren $\hat{f}\colon {}^\ell n \to {}^k n$ durch

$$\begin{aligned} \hat{f}(y_0, \dots, y_{\ell-1}) &= (x_0, \dots, x_{k-1}), \\ x_i &= f(i) \text{ für } i \in A_\ell \\ x_i &= y_j \text{ für } i \in A_j \text{ und } 0 \leq j < \ell. \end{aligned}$$

Eine *ℓ-parametrische Teilmenge von ${}^k n$* ist die Bildmenge von $\hat{f}$ bei einer Wahl von $A_0, \dots, A_\ell, f$.

Eine einparametrische Teilmenge heißt *Gerade.*

BEISPIEL 1.30. Beispiele mit $\ell = 1$: $A_1 = \emptyset$, $A_0 = k$. Dann ist $\hat{f}(y_0, \dots, y_{\ell-1}) = (y_0, \dots, y_0)$ der Länge k.

$A_1 = \{0, \dots, m-1, m+1, \dots, n-1, n+1, \dots, k-1\}$, $A_0 = \{m, n\}$. Dann ist $\hat{f}(y_0, \dots, y_{\ell-1}) = (f(0), \dots, f(m-1), y_0, f(m+1), \dots, f(n-1), y_0, f(n+1), \dots, f(k-1))$. Eine Gerade ist also eindimensional.

SATZ 1.31 (Hales Jewett 1963). *Für alle n, c gibt es eine Zahl $HJ(n,c)$, so dass für alle $k \geq HJ(n,c)$ gilt: Wenn ${}^k n$ mit c Farben gefärbt wird, dann gibt es eine monochromatische Gerade.*

SATZ 1.32 (Shelah, 1988). *Die Funktion $(n,c) \mapsto HJ(n,c)$ ist primitiv rekursiv.*

SATZ 1.33 (Shelah, 1988). *Die Funktion $(n,c) \mapsto W(n,c)$ ist primitiv rekursiv.*

Beweis: Sei $k = HJ(n,c)$. Wie definieren $\Psi\colon {}^k n \to (n-1)k+1$ durch $\Psi(\eta) = \sum_{i<k} \eta(i)$. Dann bildet Ψ jede Gerade in ${}^k n$ auf eine arithmetische Progression der Länge n ab, daher ist $W(n,c) \leq (n-1)HJ(n,c)+1$. □

Nun zum *Beweis von Satz 1.32.*

DEFINITION 1.34. Wir führen die *Grzegorczyk-Hierarchie* ein.

Für jedes $n \in \mathbb{N}$ definieren wir eine Funktion E_n und eine Funktionenmenge $\mathscr{E}_n$.

$$\begin{aligned} E_0(x,y) &= x + y, \\ E_1(x) &= x^2 + 1 \\ E_{n+2}(0) &= 2 \\ E_{n+2}(x+1) &= E_{n+1}(E_{n+2}(x)). \end{aligned}$$

Wir definieren $\mathscr{E}_0$ als den Abschluss der Nullfunktion, der Nachfolgerfunktion und der Projektionen und unter der Komposition und der *eingeschränkten primitiven Rekursion*. Das heißt, wenn g, h, $j \in \mathscr{C}$ und f durch primitive Rekursion durch g und h definiert ist und zusätzlich gilt

$$f(x_1, \ldots, x_k) \leq j(x_1, \ldots, x_k),$$

dann ist $f \in \mathscr{C}$. $\mathscr{E}^{n+1}$ ist analog zu $\mathscr{E}^0$ definiert, nur dass diesmal E_n zu den Ausgangsfunktionen hinzugenommen wird.

Wir haben zum Beispiel, dass „plus eine Konstante" in $\mathscr{E}_0$, ist, dass $+$ in $\mathscr{E}_1$ ist, dass die Multiplikation in $\mathscr{E}_2$ ist, die Exponentiation in $\mathscr{E}_3$. Dann ist zum Beispiel für $x \geq 2$, $2^{2^x} \leq E_2(x) \leq 3^{3^x}$. und

$$2^{2^{\cdot^{\cdot^{\cdot^2}}}} \leq E_3(x) \leq 3^{3^{3^{\cdot^{\cdot^{\cdot^3}}}}}$$

mit Türmen der Höhe $2x+1$. Die vier wichtigsten Eigenschaften der Grzegorczyk-Hierarchie sind

Satz 1.35.

(1) *Für $n \geq 2$ und $f \in \mathscr{E}^n$ gibt es eine Zahl m, so dass $f(x_1, \ldots, x_k) \leq E_{n-1}^{(m)}(\max(x_1, \ldots, x_k))$. Hierbei ist $E_{n-1}^{(m)}$ die m-fache Iteration von E_{n-1}.*

(2) *Wenn $n \geq 2$ und $f \in \mathscr{E}^n$ einstellig ist, dann ist f ab einem endlichen Argument kleiner als E_n.*

(3) *Wenn g, $h \in \mathscr{E}^n$ und f durch primitive Rekursion aus g und h definiert ist, dann ist $f \in \mathscr{E}^{n+1}$.*

(4) *$\bigcup_n \mathscr{E}^n$ ist die Klasse aller primitiv-rekursiver/n Funktionen.*

Die Beweise dieser Tatsachen können Sie in [**8**] nachlesen. Wir brauchen nur $\mathscr{E}^5$.

$E_{x+1}(x)$ ist ungefähr die Ackermann-Funktion.

Fortsetzung des Beweises von Satz 1.32
Wir werden zeigen, dass $n \mapsto W(n, c) \in \mathscr{E}^5$. Wir erklären die Hauptidee des Beweises. Wir beweisen induktiv über n, dass $HJ(n, c)$ existiert. Im Induktionsschritt dürfen wir nur die Existenz von $HJ(n, c)$ benutzen, nicht die Existenz von $HJ(n, c')$ für größere c', denn doppelte Induktionen können zu Ackermann-artigen Schranken führen. Sei k eine noch unbekannte Zahl. Wir versuchen zu zeigen, dass $k \geq HJ(n+1, c)$. Sei also d eine c-Färbung von ${}^k(n+1)$. Wir versuchen, das Problem auf c-Färbungen von ${}^k n$ zu reduzieren. Sei $\ell = HJ(n, c)$. Ein natürlicher Ansatz ist, eine ℓ-parametrische Teilmenge X von ${}^k(n+1)$ zu finden, so dass zwei Buchstaben nicht unterschieden werden können durch die Färbung d. Genauer partitionieren wir $\{0, 1, \ldots k-1\}$ zuerst in ℓ paarweise disjunkte nicht leere Mengen $A_0, \ldots, A_{\ell-1}$. Jedes A_i wird am Ende einen Freiheitsgrad zu X beitragen, und zwar einen, in dem die Buchstaben $n-1$ und n nicht unterschieden werden durch d. Für jedes $i < \ell$ werden wir eine nicht leere Teilmenge B_i von A_i und eine Funktion

$\eta_i \in {}^{A_i \smallsetminus B_i}(n+1)$ wählen. Dann wird X die ℓ-parametrische Teilmenge derjenigen $\eta \in {}^{k}(n+1)$ sein, so dass

(1η) für $s \in A_i \smallsetminus B_i$ ist $\eta(s) = \eta_i(s)$. Die bedeutet, dass die angestrebte Gerade auf $A_i \smallsetminus B_i$ ein Punkt ist.

(2η) wenn s_1 und s_2 in B_i sind, dann ist $\eta(s_1) = \eta(s_2)$. Die bedeutet, dass die angestrebte Gerade auf B_i eindimensional wird.

Wir sagen, dass $n-1$ *und* n *nicht durch* d *unterschieden werden können,* wenn für alle η und $\tau \in X$ folgendes gilt.

(U1) Wenn es ein $i < \ell$ gibt, so dass
(i) $\eta(s) = \tau(s)$ für alle $s \in k \smallsetminus B_i$ und
(ii) $\eta(s) = n-1$ und $\tau(s) = n$ für $s \in B_i$,
dann ist $d(\eta) = d(\tau)$.

Hierbei ist k immer noch die große Unbekannte, deren Existenz wir zeigen wollen.

Nun sehen wir unter der Annahme von (U1) die Abschätzung $k \geq HJ(n+1, c)$ wie folgt ein: Durch wiederholte Anwendung der Ununterscheidbarkeit (U1) von η und τ sehen wir

(U2) Wenn η und τ in X und für jedes $s < k$ gilt:
(i) ($\eta(s) < n-1$ gdw $\tau(s) < n-1$) und
(ii) (wenn $\eta(s) < n-1$, dann $\eta(s) = \tau(s)$),
dann $d(\eta) = d(\tau)$.

Nun betrachten wir die kanonische Einbettung

$$\pi\colon {}^{\ell}(n+1) \to X \subseteq {}^{k}(n+1),$$
$$(y_0, \dots, y_{\ell-1}) \mapsto (x_0, \dots x_{k-1}), \quad x_i = y_j \text{ für } i \in A_j, j < \ell.$$

Wir setzen $d_1 = d \circ \pi$ und erhalten so eine c-Färbung von ${}^{\ell}(n+1)$. Wenn $d_2 = d_1 \restriction {}^{\ell}n$ ist, dann gibt es nach Induktionsannahme eine d_2-monochromatische Gerade in ${}^{\ell}n$, sei diese $L_0 = \{\tau_0, \dots, \tau_{n-1}\}$. Nun sagt (U2), dass $\pi[L_0] = \{\pi(\tau_0), \dots, \pi(\tau_{n-1})\}$ erweitert werden kann zu einer d-monochromatischen Gerade L in ${}^{k}(n+1)$.

Wir fahren mit Zählargumenten fort:

Die Mengen A_i und B_i und die Funktionen $\eta_i \in {}^{A_i \smallsetminus B_i}(n+1)$ werden auf einfache Weise gewählt. Wir setzen $k = m\ell$ für eine Zahl m an. Die Werte von k und m werden erst nach der Definition 1.36 nach oben abgeschätzt. Wir werden $A_i = [mi, m(i+1))$ wählen und dann geeignete $a_i < b_i < m$ wählen. Danach wählen wir die Mengen $B_i = [mi+a_i, mi+b_i)$. Zum Schluss setzen wir

$$\eta_i(mi+r) = \quad n \quad \text{für } r < a_i,$$
$$\eta_i(mi+r) = \quad n-1 \quad \text{für } b_i \leq r < n$$

setzen. $B_i \neq \emptyset$ ist wichtig, und $B_i = A_i$ ist nicht ausgeschlossen.

Wie erreichen wir die Bedingung (U1)? Wir möchten den i-ten Block von η durch den i-ten Block von τ ersetzen, ohne die Farbe durch d zu ändern. Dies heißt, dass a_i und b_i sich ganz ähnlich verhalten über den anderen a_j, b_j, $j \neq i$.

Technisch ist hierzu folgendes hinreichend:

DEFINITION 1.36. Für $\ell, c \in \mathbb{N}$ sei $f(\ell, c)$ die kleinste natürliche Zahl k mit der folgenden Eigenschaft:

(∗) Wenn $\{d_j : j < \ell\}$ c-Färbungen von ${}^{2\ell-1}k$ sind, dann gibt es zu jedem $j < \ell$ ein Paar $a_j < b_j < k$, so dass für alle $j < \ell$

$$d_j(a_0, b_0, \ldots, a_{j-1}, b_{j-1}, a_j, a_{j+1}, b_{j+1}, \ldots, a_{\ell-1}, b_{\ell-1}) = d_j(a_0, b_0, \ldots, a_{j-1}, b_{j-1}, b_j, a_{j+1}, b_{j+1}, \ldots, a_{\ell-1}, b_{\ell-1}).$$

Es gibt also c^ℓ Farben für $(d_0, \ldots, d_{\ell-1})$. Der Definitionsbereich jedes d_j ist ein $2\ell - 1$-dimenionaler Würfel der Kantenlänge $k = f(\ell, c)$. Dieses soll so groß sein, dass man in jedem d_j an der Stelle $2j$ eine Zahl a_j in b_j tauschen kann, ohne dass d_j dieses sieht.

LEMMA 1.37.

(a) $f(1, c) = c + 1$.

(b) *Für* $\ell \geq 1$ *ist* $f(\ell + 1, c) \leq c^{f(\ell,c)^{2\ell}} + 1$. $f \in \mathscr{E}_4$.

Beweis: (a) ist leicht. (b) Nun sei $\ell \geq 1$, $m = f(\ell, c)$, $k = c^{m^{2\ell}} + 1$ und seien $\{d_j : j < \ell + 1\}$ c-Färbungen von ${}^{2\ell+1}k$. Wir definieren

$$d^1 \colon k \to \{g : g \colon {}^{2\ell}m \to c\}$$

wie folgt: Für jedes $a < k$ sei

$$d^1(a)(a_0, b_0, \ldots, a_{\ell-1}, b_{\ell-1}) = d_\ell(a_0, b_0, \ldots, a_{\ell-1}, b_{\ell-1}, a).$$

Dann gibt es, da k groß ist, $a_\ell < b_\ell < k$, so dass $d^1(a_\ell) = d^1(b_\ell)$. Nun definieren wir die c-Färbungen $\{d^2_j : j < \ell\}$ von ${}^{2\ell-1}m$ durch

$$d^2_j(a_0, b_0, \ldots, a_{j-1}, b_{j-1}, a_j, a_{j+1}, b_{j+1}, \ldots, a_{\ell-1}, b_{\ell-1}) = d_j(a_0, b_0, \ldots, a_{j-1}, b_{j-1}, b_j, a_{j+1}, b_{j+1}, \ldots, a_{\ell-1}, b_{\ell-1}, a_\ell, b_\ell).$$

Nach der Definition von m gibt es $a_j < b_j < m$ für $j < \ell$, so dass (∗) für $\{d^2_j : j < \ell\}$ gilt. Doch nun erfüllen $a_j < b_j$ für $j < \ell + 1$ die Gleichung (∗) auch für $\{d_j : j < \ell + 1\}$.

Wir haben $f(x + 1) = h(f(x), x) = c^{f(x)^{2x}} \in \mathscr{E}_4$, da $h(z, x) = c^{z^{2x}} \in \mathscr{E}_3$, denn h wird seinerseits mit primitiver Rekursion mit einer Hilfsfunktion aus $\mathscr{E}_2$ hergestellt. □

LEMMA 1.38.

(a) $HJ(1, c) = 1$.

(b) *Für* $n \geq 1$

$$HJ(n+1,c) \leq HJ(n,c) \times f(HJ(n,c), c^{(n+1)^{HJ(n,c)}}).$$

Da $f \in \mathscr{E}_4$, *ist* $HJ \in \mathscr{E}_5$.

Beweis: (a) Trivial. (b) Seien $\ell = HJ(n,c)$ und $m = f(\ell, c^{(n+1)^\ell})$ und $k = \ell m$.

Sei d eine c-Färbung von ${}^k(n+1)$. Wir definieren c-Färbungen $\{d^1_j : j < \ell\}$ auf ${}^{2\ell-1}m$ wie folgt. Wenn $j < \ell$, dann ist

$$d^1_j(a_0, b_0, \ldots, a_{j-1}, b_{j-1}, a_j, a_{j+1}, b_{j+1}, \ldots, a_{\ell-1}, b_{\ell-1})$$

eine Funktion mit Definitionsbereich ${}^\ell(n+1)$ und Bildbereich $\subseteq \operatorname{rge}(d)$, so dass jedes $\eta \in {}^\ell(n+1)$ durch d^1_j auf $d(\nu_\eta)$ abgebildet wird mit folgenden Auswahlregeln für ν_η:

(i) Sei $i < \ell$, $i \neq j$, $r < m$, $a_i < b_i$. Dann sei

$$\begin{aligned} \nu_\eta(mi+r) &= n, \text{ falls } r < a_i, \\ &= n-1, \text{ falls } b_i \leq r < m, \\ &= \eta(i), \text{ falls } a_i \leq r < b_i. \end{aligned}$$

(ii) Falls $i < \ell$, $i \neq j$, $r < m$ und $a_i \geq b_i$, dann sei $\nu_\eta(mi+r) = 0$. Dieser Fall wird keinen Einfluß haben.

(iii) Sei $i = j$ und $r < m$. Dann ist

$$\begin{aligned} \nu_\eta(mi+r) &= n, \text{ falls } r < a_i, \\ &= n-1, \text{ falls } a_i \leq r < m. \end{aligned}$$

$|\operatorname{rge}(d^1_j)| \leq c^{(n+1)^\ell}$. Also gibt es $a_j < b_i < m$ für $j < \ell$, die $(*)$ für $\{d^1_j : j < \ell\}$ erfüllen. Für jedes $i < \ell$ sei $A_i = [mi, m(i+1))$ und $B_i = [mi + a_i, mi + b_i)$ und $\eta_i \in {}^{A_i \smallsetminus B_i}(n+1)$ wird durch

$$\begin{aligned} \eta_i(mi+r) &= n, \text{ falls } r < a_i, \\ &= n-1, \text{ falls } b_i \leq r < m. \end{aligned}$$

definiert.

Sei X die ℓ-parametrische Teilmenge von ${}^k(n+1)$, die aus denjenigen η besteht, die die Bedingungen (1η) und (2η) erfüllen. Seien $\eta, \tau \in X$ und gebe es ein $i < \ell$, so dass $\eta(s) = \tau(s)$ für $s \in k \smallsetminus B_i$ und $\eta(s) = n-1$ und $\tau(s) = n$ für $s \in B_i$.

Da wir nach Voraussetzung über die d^1_i wissen, dass

$$\begin{aligned} d^1_j(a_0, b_0, \ldots, a_{j-1}, b_{j-1}, a_j, a_{j+1}, b_{j+1}, \ldots, a_{\ell-1}, b_{\ell-1}) = \\ d^1_j(a_0, b_0, \ldots, a_{j-1}, b_{j-1}, b_j, a_{j+1}, b_{j+1}, \ldots, a_{\ell-1}, b_{\ell-1}). \end{aligned}$$

haben wir $d(\eta) = d(\tau)$. Also haben wir den Wunsch (U1) erfüllt. Wir haben $n \mapsto HJ(n,c)$ ist in $\mathscr{E}^5$. $\square_{1.32}$

Es ist offen, ob $n \mapsto HJ(n,2)$ in $\mathscr{E}_4$ ist.

KAPITEL 2

Kleenes Satz von der Normalform

SATZ 2.1 (Der Satz von der Normalform, Kleene 1936). *Es gibt eine primitiv-rekursive Funktion U und für jedes $n \in \mathbb{N}$ ein primitiv-rekursives Prädikat T_n, so dass es für jede rekursive Funktion f in n Variablen eine Zahl e (den sogenannten Index von f, wir schreiben später φ_e statt f) gibt, so dass*

1. $\forall x_0 \ldots \forall x_{n-1} \exists y T_n(e, x_0, \ldots, x_{n-1}, y)$.
2. $f(x_0, \ldots, x_{n-1}) = U(\mu y T_n(e, x_0, \ldots, x_{n-1}, y))$.

Beweis: Die Idee ist, jeder μ-rekursiven Funktion eine Zahl oder Nummer, Index genannt, zuzuordnen (ähnlich wie die Gödelnummern von Formeln) und auch jeder Berechnung eines Wertes eine Zahl zuzuordnen (ähnlich wie Gödelnummern eines Beweises), so dass für $n \in \omega$ das sogenannte Kleene-Prädikat

$$T_n(e, x_0, \ldots, x_{n-1}, y)$$

folgendes kodieren: y ist die Nummer der Berechnung des Wertes der Funktion mit Index e angesetzt auf den Input $x_0, \ldots, x_{n-1}$. Dann wird $\mu y T_n(e, x_0, \ldots, x_{n-1}, y)$ den kleinsten Code für eine Berechnung ausgeben und U wird den Output aus y extrahieren. Dies alles ist einfacher skizziert als durchgeführt. Wir werden jetzt die einzelnen Schritte durchführen.

1. Schritt. Die Zuordnung der Indizes

Wie setzen folgende Zuordnung fest (Sie können diese variieren):

- $\langle 0 \rangle$ für die Nullfunktion.
- $\langle 1 \rangle$ für die Nachfolgerfunktion.
- $\langle 2, n, i \rangle$ für die Projektion p_i^n, $0 \leq i < n$.
- $\langle 3, b_0, \ldots, b_{m-1}, a \rangle$ für $f(\vec{x}) = g(h_0(\vec{x}), \ldots, h_{m-1}(\vec{x}))$, wenn b_i die Nummer von h_i ist und a die Nummer von g ist.
- $\langle 4, a, b \rangle$ wird $f(\vec{x}, y)$ zugeordnet, wenn f die primitive Rekursion aus g und h ist und a die Nummer von g ist und b die Nummer von h ist.
- $\langle 5, a \rangle$ wird $f(\vec{x}) = \mu y(g(\vec{x}, y) = 0)$ zugeordnet, wenn $\forall \vec{x} \exists y g(\vec{x}, y) = 0$ und a die Nummer von g ist.

Die Nummer oder Zahl, die jeder rekursiven Funktion auf diese Weise zugeordnet wird, heißt der Index e dieser Funktion. Da es viele Arten gibt,

dieselbe Funktion zu definieren, wird jede rekursive Funktion viele verschiedene Indizes haben (wir werden später ein padding Lemma beweisen). Außerdem sind viele Zahlen nicht Indizes einer rekursiven Funktion, entweder, weil sie nicht von der richtigen Form sind, oder, weil ihre Komponenten ihrerseits nicht Indizes einer rekursiven Funktion sind. Wir werden später sehen, dass die Menge der Indizes (totaler Funktionen) nicht rekursiv ist, denn im allgemeinen kann das Kriterium $\forall\vec{x}\exists y g(\vec{x}, y) = 0$ nicht von einem Index von g aus entschieden werden.

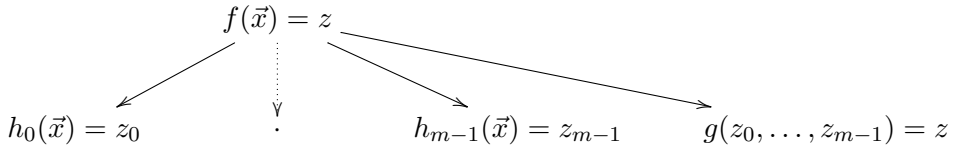
2. Schritt. Wir schreiben die Berechnungen in Bäume.

Jeder Knoten in einem Baum gibt Auskunft über einen Induktionschritt: eine einfachere Funktion, oder dieselbe Funktion mit kleineren Argumenten

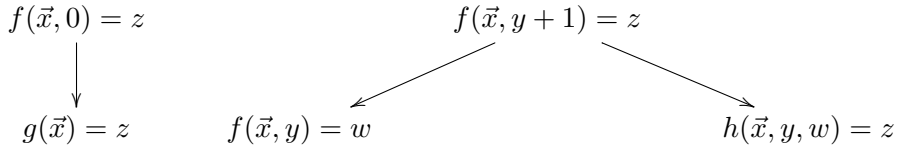
(1) Knoten ohne Vorgänger

$$\begin{aligned} f(x) &= 0 && \text{für die Nullfunktion } f = o, \\ f(x) &= x + 1 && \text{für die Nachfolgerfunktion } f = s, \\ f(x_0, \ldots, x_{n-1}) &= x_i && \text{für die Projektion } f = p_i^n. \end{aligned}$$

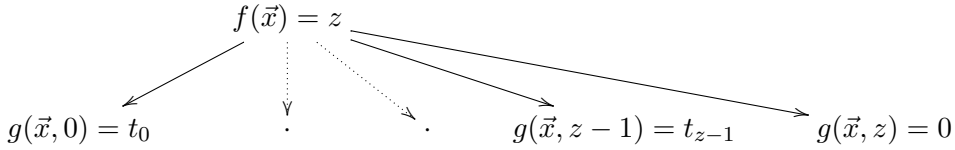
(2) Komposition
Wenn $f(\vec{x}) = g(h_0(\vec{x}, \ldots, h_{m-1}(\vec{x}))$, dann hat der Knoten $f(\vec{x}) = z$ die $m + 1$ Vorgängerknoten



(3) primitive Rekursion
Wenn $f(\vec{x}, y)$ durch primitive Rekursion aus g und h definiert ist, dann gibt es zwei Möglichkeiten für den Knoten $f(\vec{x}, y) = z$



(4) μ-Rekursion
Wenn $f(\vec{x}) = \mu y(g(\vec{x}, y) = 0)$, dann sieht der Baum unterhalb $f(\vec{x})$ wie folgt aus



und die t_i sind $\neq 0$.

3. Schritt. Kodierungen der Berechnungen.

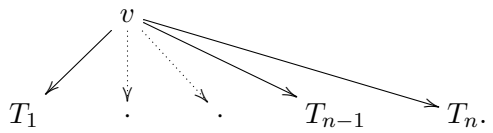
Der dritte Schritt wird induktiv über die Konstruktion des (endlich großen) Berechnungsbaumes durchgeführt. An jedem Knoten stehe ein Ausdruck der Form $f(x_0, \ldots, x_{n-1}) = z$. Wir geben diesem Ausdruck die Zahl

$$\langle e, \langle x_0, \ldots, x_{n-1}\rangle, z\rangle$$

für einen Index e von f. Jeder Knoten wird also durch einen Index, ein Argument und einen Wert repräsentiert, und die Spitze des Baumes hat den Wert $\langle e, \langle x_0, \ldots, x_{n-1}\rangle, z\rangle$, der für $f_e(x_0, \ldots, x_{n-1}) = z$ steht.

Nun ordnen wir den Bäumen Zahlen zu. Jeder Baum besteht aus einem Vertex v und einer endlichen (kann auch 0 sein) Zahl von angeordneten Vorgängern, die jeweils einen echten Teilbaum vertreten. Also ist folgende Zuordnung $\hat{}$ durch Induktion über die Höhe des Baumes definiert:

$$\hat{T} = \langle v, \hat{T_1}, \ldots, \hat{T_n}\rangle \text{ für}$$



Wenn ein Vertex v keine Vorgänger im Baum hat, dann erhält dieser Einpunkt-Baum die Zahl $\langle v\rangle$.

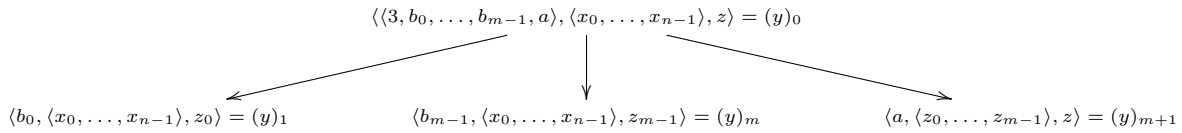
4. Schritt. Wir formulieren ein primitiv rekursives Prädikat $T(y)$, das aussagt, dass y einen Berechnungsbaum kodiert.

Um die Lesbarkeit zu erhöhen, schreiben wir statt geschachtelter Klammern nur Kommata, also zum Beispiel $(y)_{1,0} = ((y)_1)_0$.

(1) Knoten ohne Vorgänger

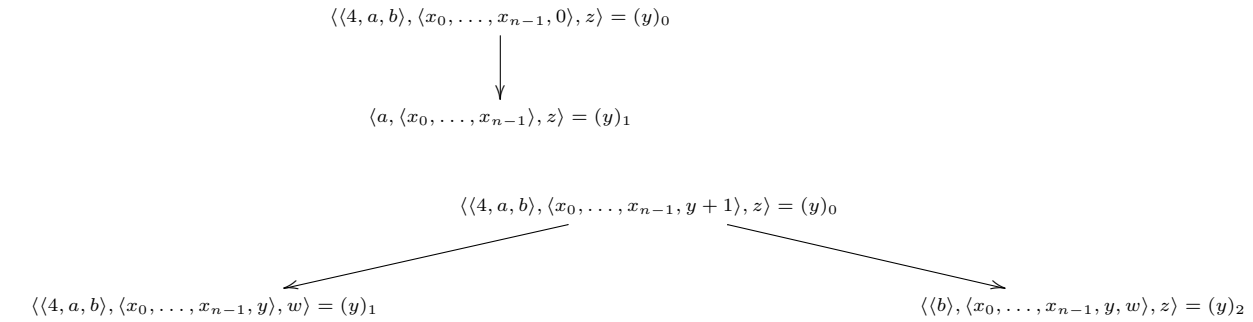
$$\begin{aligned}
\langle\langle 0\rangle, \langle x\rangle, 0\rangle &\quad \text{für die Nullfunktion } f = o,\\
\langle\langle 1\rangle, \langle x\rangle, x+1\rangle &\quad \text{für die Nachfolgerfunktion } f = s,\\
\langle\langle 2, n, i\rangle, \langle x_0, \ldots, x_{n-1}\rangle, x_i\rangle &\quad \text{für die Projektion } f = p_i^n.
\end{aligned}$$

(2) Komposition
Wenn $f(\vec{x}) = g(h_0(\vec{x}), \ldots, h_{m-1}(\vec{x}))$, dann hat der Knoten $f(\vec{x}) = z$ die $m+1$ Vorgängerknoten



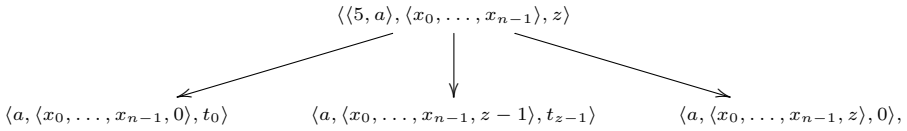
Das Prädikat $B(y)$ beschreibt diese Verzweigung, g.h., $B(y)$ sagt, $y = \langle (y)_0, \ldots y_{m+1}\rangle$ mit geeigneten $b_i, x_j z_k, z$ eine Instanz dieses Baumverzweigungsschemas ist. Hierbei seind a und die b_i die Indices der Funktionen g und h_i für $i < m$.

(3) primitive Rekursion
Wenn $f(\vec{x}, y)$ durch primitive Rekursion aus g (mit Index a) und h (mit Index b) definiert ist, dann gibt es zwei Möglichkeiten für den Knoten $f(\vec{x}, y) = z$



Diese soll durch das Prädikat $C(y)$ beschreiben werden.

(4) μ-Rekursion
Wenn $f(\vec{x}) = \mu y(g(\vec{x}, y) = 0)$, dann sieht der Baum unterhalb $f(\vec{x})$ wie folgt aus



wobei die $t_i \neq 0$ sind.

Das Prädikat $D(y)$ beschreibt dieses Situation.

Nun können wir das primitiv-rekursive Prädikat „y ist ein Berechnungsbaum" formulieren: Die einfachsten Eigenschaften, die auch für Einpunktbäume gelten, stehen in $A(y)$:

$$\begin{aligned} A(y) \quad \Leftrightarrow \quad & \mathrm{Seq}(y) \wedge \mathrm{Seq}((y)_0) \wedge \lg((y)_0) = 3 \\ & \wedge \mathrm{Seq}((y))_{0,0}) \wedge \mathrm{Seq}((y)_{0,1}). \end{aligned}$$

Die Ausgangsfunktionen werden beschrieben durch

$$\begin{aligned} B(y) \quad \Leftrightarrow \quad & \lg(y) = 1 \wedge \\ & \{[(y)_{0,0} = \langle 0\rangle \wedge \lg((y)_{0,1}) = 1 \wedge (y)_{0,2} = 0] \vee \\ & [(y)_{0,0} = \langle 1\rangle \wedge \lg((y)_{0,1}) = 1 \wedge (y)_{0,2} = (y)_{0,1} + 1] \vee \\ & [\lg((y)_{0,0}) = 3 \wedge (y)_{0,0,0} = 2 \wedge (y)_{0,0,1} = \lg((y)_{0,1}) \wedge \\ & 0 \leq (y)_{0,0,2} \wedge (y)_{0,2} = ((y)_{0,1})_{(y)_{0,0,2}}]\}. \end{aligned}$$

Komposition

$$\begin{aligned} C(y) \quad \Leftrightarrow \quad & \lg((y)_{0,0}) \geq 3 \wedge (y)_{0,0,0} = 3 \wedge \lg(y) = \lg((y)_{0,0}) \wedge \\ & (\forall i)_{1 \leq i < \lg(y)-1}[(y)_{i,0,0} = (y)_{0,1,i} \wedge (y)_{i,0,1} = (y)_{0,1,i}] \wedge \\ & (y)_{\lg(y)-1,0,0} = (y)_{0,0,\lg(y)-1} \wedge (y)_{\lg(y)-1,0,2} = (y)_{0,2} \wedge \\ & (y)_{\lg(y)-1,0,1} = \langle (y)_{1,0,2}, \ldots, (y)_{\lg(y)-2,0,2} \rangle. \end{aligned}$$

Primitive Rekursion

$$\begin{aligned} D(y) \quad \Leftrightarrow \quad & \lg((y)_{0,0}) = 3 \wedge (y)_{0,0,0} = 4 \wedge \\ & \{[(y)_{0,1,\lg((y)_{0,1})-1} = 0 \wedge \lg(y) = 2 \wedge (y)_{1,0,0} = (y)_{0,0,1} \wedge \\ & (y)_{1,0,1} * \langle 0 \rangle = (y)_{0,1} \wedge (y)_{1,0,2} = (y)_{0,2}] \vee \\ & [(y)_{0,1,\lg((y)_{0,1})-1} > 0 \wedge \lg(y) = 3 \wedge (y)_{1,0,0} = (y)_{0,0} \wedge \\ & (y)_{1,0,0} = (y)_{0,0} \wedge \lg((y)_{1,0,1}) = \lg((y)_{0,1}) \wedge \\ & (\forall i)_{0 \leq i < \lg(y)_{0,1}-1}(y)_{1,0,1,i} = (y)_{0,1,i} \wedge \\ & (y)_{1,0,1,\lg((y)_{0,1})-1} + 1 = (y)_{0,1,\lg((y)_{0,1})-1} \wedge \\ & (y)_{2,0,0} = \langle (y)_{0,0,2} \rangle \wedge (y)_{2,0,2} = (y)_{0,2} \wedge \\ & (y)_{2,0,1} = \langle (y)_{1,0,1} \rangle * \langle (y)_{1,0,2} \rangle]\}. \end{aligned}$$

μ-Rekursion

$$\begin{aligned} E(y) \quad \Leftrightarrow \quad & \lg((y)_{0,0}) = 2 \wedge (y)_{0,0,0} = 5 \wedge \\ & \lg(y) \geq 2 \wedge (y)_{0,2} = \lg(y) - 2 \wedge \\ & (\forall i)_{1 \leq i < \lg(y)-1}[(y)_{i,0,0} = (y)_{0,0,1} \wedge \\ & (y)_{i,0,1} = (y)_{0,1} * \langle i - 1 \rangle] \wedge \\ & (\forall i)_{1 \leq i < \lg(y)-1}[(y)_{i,0,2} \neq 0] \wedge (y)_{\lg(y)-1,0,2} = 0. \end{aligned}$$

Nun definieren wir das gesuchte primitiv-rekursive Prädikat T:

$$\begin{aligned} T(y) \quad \Leftrightarrow \quad & A(y) \wedge [B(y) \vee C(y) \vee D(y) \vee E(y)] \wedge \\ & [\lg(y) > 0 \rightarrow (\forall i)_{1 \leq i \leq \lg(y)-1} T((y)_i)]. \end{aligned}$$

5. Schritt. Die Definition von T_n und U.

$$\begin{aligned} T_n(e, x_1, \ldots, x_n, y) \quad & \Leftrightarrow \quad T(y) \wedge (y)_{0,0} = e \wedge (y)_{0,1} = \langle x_0, \ldots, x_{n-1} \rangle \\ U(y) \quad & = \quad (y)_{0,2} \end{aligned}$$

Diese sind offensichtlich primitiv-rekursiv.

Nun sei f eine n-stellige μ-rekursive Funktion mit Index e. Da f und alle seine Vorgängerfunktionen beim Aufbau von e total sind, gibt es für jedes Argument $x_0, \ldots, x_{n-1}$ einen Berechnunsgsbaum für $f(x_0, \ldots, x_{n-1})$, der die durch e angegebene Tiefe und Gestalt hat.

Jeder Berechnungsbaum, insbesondere der mit der kleinsten Kodenummer y, gibt den Wert $f(x_1, \ldots, x_n)$ als dritte Komponente an seiner Spitze v. Daher ist

$$f(x_0, \ldots, x_{n-1}) = U(\mu y T_n(e, x_0, \ldots, x_{n-1}, y)).$$

□

KAPITEL 3

Partielle rekursive Funktionen

1. Padding, rekursive Aufzählungen und der s-m-n-Satz

Wir verwenden kleine griechische Buchstaben für partielle Funktionen, also $\varphi\colon \mathbb{N}^k \rightharpoonup \mathbb{N}$. Wir schreiben $\varphi(\vec{x})\downarrow$ gdw $\vec{x} \in \mathrm{dom}(\varphi)$, und $\varphi(\vec{x})\uparrow$ im gegenteiligen Fall. Zwei partielle Funktionen sind gleich, in Zeichen $\varphi \simeq \psi$ gdw $\mathrm{dom}(\varphi) = \mathrm{dom}(\psi)$ und für alle $\vec{x} \in \mathrm{dom}(\varphi)$, $\varphi(\vec{x}) = \psi(\vec{x})$. Analog benutzen wir $\varphi(\vec{x}) \simeq \psi(\vec{x})$ für ein $\vec{x}$. (Dagegen kann für $\varphi(\vec{x}) \simeq z$ immer $\varphi(\vec{x}) = z$ geschrieben werden, da z immer definiert ist.) Die Inklusion $\varphi \subseteq \psi$ ist die Inklusion der Graphen.

Definition 3.1 (Kleene 1936). Die Menge der partiellen rekursiven Funktion ist die kleinste Menge von Funktionen, die

(1) die Ausgangsfunktionen $o(x) = 0$, $S(x) = x + 1$, $p_i^n(\vec{x}) = x_i$ enthält und

(2) unter Verkettung, primitiver Rekursion und unter unbeschränkter μ-Rekursion abgeschlossen ist. Unbeschränkte μ-Rekursion bedeutet: Die Bedingung $\forall\vec{x}\exists y\psi(\vec{x}, y) = 0$ wird fallengelassen und stattdessen setzt man

$$\varphi(\vec{x}) \simeq \mu y[\forall z \leq y\psi(\vec{x}, z)\downarrow \wedge \psi(\vec{x}, y) = 0].$$

(Die heißt nach unseren Konventionen: $\varphi(\vec{x})$ ist nicht definiert, wenn es kein solches y gibt.)

Übung 3.2. Wenn man in der unbeschränkten μ-Rekursion die Bedingung

$$\forall z \leq y\psi(\vec{x}, z)\downarrow$$

fallenlässt, dann erhält man eine echt größere Klasse von Funktionen. Nehmen Sie für $A \subseteq \mathbb{N}$ zum Beispiel das Halteproblem 3.27, das ist eine rekursiv aufzählbare, nicht rekursive Menge. Dann setzen wir

$$\psi(x, y) = 0 \leftrightarrow (y = 0 \wedge x \in A) \vee y = 1.$$

Dann ist ψ partiell rekursiv, aber die Funktion

$$f(x) \simeq \mu y(\psi(x, y) \simeq 0)$$

ist nicht partiell rekursiv, denn wir haben $f(x) = 0$ gdw $x \in A$, und A war ja gerade nicht rekursiv.

SATZ 3.3 (Der Satz von der Normalform für partielle rekursive Funktionen, Kleene 1938). *Es gibt eine primitiv-rekursive Funktion U und für jedes $n \in \mathbb{N}$ ein primitiv-rekursives Prädikat T_n, so dass es genau für jede partielle rekursive Funktion φ in n Variablen einen Index e gibt, so dass*

1. $\forall \vec{x}(\varphi(\vec{x}) \downarrow \leftrightarrow \exists y T_n(e, x_0, \ldots, x_{n-1}, y))$.
2. $\varphi(x_0, \ldots, x_{n-1}) \simeq U(\mu y T_n(e, x_0, \ldots, x_{n-1}, y))$.

Beweis: Der Beweis geht analog zum Beweis des Normalformentheorems für totale rekursive Funktionen. Der Index $\langle 5, a\rangle$ wird dem Aufbauschritt

$$\varphi(\vec{x}) \simeq \mu y[\forall z \leq y \psi(\vec{x}, z) \downarrow \wedge \psi(\vec{x}, y) = 0]$$

zugeordnet, wenn a der Index von ψ ist. Der Berechnungsbaum greift nur auf die Werte $\psi(\vec{x}, z)$ für $z \leq y$ zurück. □

KOROLLAR 3.4. *Die rekursiven Funktionen sind genau die partiellen rekursiven Funktionen, die zufällig total sind.*

Beweis. Natürlich ist jede rekursive Funktion partiell rekursiv und total. Sei nun φ eine partielle rekursive Funktion. Dann gibt es ein e, so dass $\varphi(x_0, \ldots, x_{n-1}) \simeq U(\mu y T_n(e, x_0, \ldots, x_{n-1}, y))$. Wenn φ nun total ist, dann gilt $\forall \vec{x} \varphi(\vec{x}) \downarrow$. Also gibt es nach obiger $\simeq$-Gleichung zu jedem $\vec{x}$ ein y, so dass $T_n(e, x_0, \ldots, x_{n-1}, y)$. Dann ist φ nach dem Normalformentheorem für totale Funktionen rekursiv. □

Das Normalformentheorem für partielle rekursive Funktionen hat nun den Vorteil, dass die Menge der Indizes für partielle rekursive Funktionen rekursiv ist.

ÜBUNG 3.5. Beweisen Sie diese letzte Behauptung.

Die folgende Definition ist von zentraler Bedeutung:

DEFINITION 3.6. Seien $e, n, s \in \mathbb{N}$. Unter s kann man sich stage oder Schrittzahl vorstellen.

1. φ_e^n oder $\{e\}^n$ ist die e-te partielle rekursive n-stellige Funktion:
$$\varphi_e^n(\vec{x}) \simeq \{e\}^n(\vec{x}) \simeq U(\mu y(T_n(e, \vec{x}, y))).$$
2. $\varphi_{e,s}^n$ oder $\{e\}_s^n$ ist die Approximation bis zum Berechnungsbaum der Größe s für die e-te partielle rekursive n-stellige Funktion:
$$\varphi_{e,s}^n(\vec{x}) \simeq \{e\}_s^n(\vec{x}) \simeq U(\mu y((T_n(e, \vec{x}, y)) \wedge y < s)).$$

Wir haben also:

$$\varphi_{e,s}^n(\vec{x}) \simeq \{e\}_s^n(\vec{x}) \simeq \begin{cases} \varphi_e^n(\vec{x}) & \text{wenn } (\exists y < s) T_n(e, \vec{x}, y)) \\ \text{undefiniert} & \text{sonst.} \end{cases}$$

Der folgende symmetrische Satz ist sogar älter als Kleenes Satz von 1936. Posts Arbeit aus den 1920er Jahren wurde von Post selbst unterschätzt (nach

seinen Aussagen hatte er zu wenig Vertrauen in die universelle Bedeutung seines Berechenbarkeitsbegriffs, also in das, was ab Ende der 1930er Jahre dann als Churchsche These bekannt wurde) und nicht zur Veröffentlichung eingereicht.

SATZ 3.7 (Der Aufzählungssatz (Post 1922, Turing 1936, Kleene 1938)). *Die Folge $\langle \varphi_e^n : e \in \mathbb{N} \rangle$ ist eine partielle rekursive Aufzählung aller n-stelligen partiellen rekursiven Funktionen im folgenden Sinne:*

1. *Für jedes e ist φ_e^n eine partielle rekursive Funktion in n Variablen,*
2. *Wenn ψ eine partielle rekursive n-stellige Funktion ist, dann gibt es ein e, so dass $\psi \simeq \varphi_e^n$,*
3. *Es gibt eine partielle rekursive Funktion in $n+1$ Variablen, so dass*

$$\varphi(e, \vec{x}) \simeq \varphi_e^n(\vec{x}).$$

Beweis: Alles folgt aus dem Normalformentheorem. Für den dritten Teil setzen wir

$$\varphi(e, \vec{x}) \simeq U(\mu y T_n(e, \vec{x}, y)).$$

□

PROPOSITION 3.8 (Das Padding Lemma). *Ausgehend von einem Index e einer partiellen rekursiven Funktion kann man auf berechenbare Weise unendlich viele weitere Indizes derselben Funktion herstellen.*

Beweis: Sei $m(e,k)$ der Index von $U(\mu y(T_n(e, \vec{x}, y) \wedge \bigwedge_{i<k} x_0 = x_0))$. Dann berechnen alle $\varphi_{m(e,k)}$ dasselbe φ_e und es ist $m(e,k) \neq m(e,\ell)$ für $\ell \neq k$. □

SATZ 3.9 (Das s-m-n- oder S_n^m-Theorem. Kleene 1938). *Für jedes Paar $(n,m)m$ $n, m \geq 1$, von Stelligkeiten es eine primitiv-rekursive injektive Funktion*

$$S_n^m(e, x_0, \ldots, x_{n-1}),$$

so dass

$$\varphi_{S_n^m(e,x_0,\ldots,x_{n-1})}(y_0, \ldots, y_{m-1}) \simeq \varphi_e(x_0, \ldots, x_{n-1}, y_0, \ldots, y_{m-1}).$$

Beweis: Die gesuchte Funktion S_n^m berechnet den Index der m-stelligen partiellen Funktion

$$(y_0, \ldots, y_{m-1}) \mapsto \varphi_e(C_{x_0}, \ldots, C_{x_{n-1}}, y_0, \ldots, y_{m-1})$$

aus e und x_0, …, x_{n-1}. $z \mapsto C_x(z) = S^{(x)}(0)$ ist die konstante einstellige totale berechenbare Funktion x. Der Aufbau von $S_n^m(e, \vec{x})$ geht induktiv (diese Induktion ist selbst eine primitive Rekursion) über den Aufbau von e. Bei den Anfangsschritten $e = \langle 0 \rangle$, $e = \langle 1 \rangle$ oder $\langle 2, n, i \rangle$ kommen unter

Umständen die x_i hinein. Wir betrachten zum Beispiel $e = \langle 2, n+m, i\rangle$ und $i < n$. Dann ist

$$\varphi_e((x_0, \ldots, x_{n-1}, y_0, \ldots, y_{m-1}) = x_i = S^{(x_i)}(0),$$

und hat den Index

$$S_n^m(e, x_0, \ldots, x_{n-1}) = \langle 3, e', \langle 0\rangle\rangle$$

und e' ist der Index von $S^{(x_i)}$, der sich wiederum aus x_i und dem Index $\langle 1\rangle$ von S und der Regel für Hintereinanderschaltung berechnet. (Der Exponent in Klammern gibt die Zahl der Iterationen an.)

Bei den weiteren Aufbauschritten zieht sich die Berechnung induktiv hoch, zum Beispiel

$$S_n^m(\langle 5, a\rangle, x_0, \ldots, x_{n-1}) = \langle 5, S_n^m(a, x_0, \ldots, x_{n-1})\rangle$$

für die meisten a (solche, die eine kleinste y-Koordinate für eine Nullstelle suchen und berechnen).

Die Hauptarbeit des Einsetzens von Konstanten C_{x_i} für die Variablen x_i geschieht also an den Blättern des Berechnungsbaumes für $\varphi_e(\vec{x}, \vec{y})$.

□

2. Rekursiv aufzählbare Mengen

Definition 3.10 (Post 1922, Kleene 1936). Eine n-stellige Relation heißt rekursiv aufzählbar (r.e.), wenn sie der Definitionsbereich einer n-stelligen partiellen rekursiven Funktion ist. Wir schreiben $W_e^n = \mathrm{dom}(\varphi_e^n)$ und $W_{e,s}^n = \mathrm{dom}(\varphi_{e,s}^n)$.

Satz 3.11 (Das Normalformentheorem für r.e. Relationen. Kleene 1936, Rosser 1936, Mostowski 1947). *Sei P eine n-stellige Relation. P ist r.e. genau dann, wenn es eine $n+1$-stellige rekursive Relation R gibt, so dass*

$$P(\vec{x}) \leftrightarrow \exists y R(\vec{x}, y).$$

Eine andere Formulierung ist: P is r.e. genau dann, wenn es einen Index e von P gibt, so dass

$$P(\vec{x}) \leftrightarrow W_e^n(\vec{x}) \leftrightarrow \exists y R(\vec{x}, y).$$

Beweis: Wenn P r.e. ist, dann gibt es ein e, so dass $P = \mathrm{dom}(\varphi_e)$, also $W_e(\vec{x}) \leftrightarrow \varphi_e(\vec{x}) \downarrow \leftrightarrow \exists y T_n(e, \vec{x}, y)$. Für die umgekehrte Inklusion haben wir: Wenn $P(\vec{x}) \leftrightarrow \exists y R(x, y)$ mit einem rekursiven R ist, dann ist P der Definitionsbereich der partiellen rekursiven Funktion $\varphi(\vec{x}) \simeq \mu y R(\vec{x}, y)$. □

Seien p und q Polynome über $\mathbb{Z}$.

$$D = \{y \in \mathbb{N} \,:\, \exists \vec{x} \in \mathbb{N}^k p(\vec{x}, y) = q(\vec{x}, y)\}.$$

Jedes solche D ist rekursiv aufzählbar, da die Polynome rekursiv sind. Matyasevich [**5**] bewies die erstaunliche Umkehrung: Zu jeder r.e. Menge S gibt es Polynome p, q, so dass die Menge der positiven Lösungen D der

diophantischen Gleichung gleich S ist. Da es rekursiv aufzählbare, nicht rekursive Mengen gibt, löst dieses Ergebnis Hilberts zehntes Problem negativ: Es gibt eine diophantische Menge D, die nicht entscheidbar (= rekursiv) ist.

PROPOSITION 3.12 (Das Graphentheorem). *Sei φ partiell, f total. Dann ist φ eine partielle rekursive Funktion gdw sein Graph r.e. ist. f ist rekursiv gdw sein Graph rekursiv ist.*

Beweis: Der Graph einer Funktion φ ist definiert durch

$$G = \{(\vec{x}, y) \,:\, \varphi(\vec{x}) = y\}.$$

Wenn φ eine partielle rekursive Funktion ist, dann ist $\varphi \simeq \varphi_e$ für einen Index e und daher

$$\begin{aligned} G_\varphi &= \{(\vec{x}, z) \,:\, \varphi_e(\vec{x}) = z\} \\ &= \{(\vec{x}, z) \,:\, U(\mu y T_n(e, \vec{x}, y)) = z\} \\ &= \{(\vec{x}, z) \,:\, \exists y(T_n(e, \vec{x}, y) \wedge (\forall t < y)\neg T_n(e, \vec{x}, t) \wedge U(y) = z)\}. \end{aligned}$$

Also ist G_φ nach dem Aufzählungstheorem 3.7 eine r.e. Menge.

Wenn umgekehrt G_φ r.e. ist, dann gibt es eine rekursive Relation R, so dass

$$(\vec{x}, z) \in G_\varphi \leftrightarrow \exists y R(\vec{x}, z, y).$$

Indem man z und y in eine Zahl $t = \langle z, y\rangle$ kodiert, erhält man

$$\varphi(\vec{x}) \simeq (\mu t R(\vec{x}, (t)_0, (t)_1))_0.$$

Der Algorithmus sucht also die z mit $G(\vec{x}, z)$ aber nicht den z entlang, sondern den t entlang, denn er darf ja nicht bei einem vergeblichen z stecken beiben.

Der Teil des Satzes für die totalen Funktionen ist einfacher: Wenn f rekursiv ist, dann ist die charakteristische Funktion von G_f rekursiv:

$$\chi_{G_f}(\vec{x}, z) = \begin{cases} 0, & \text{wenn } f(\vec{x}) = z, \\ 1, & \text{sonst.} \end{cases}$$

Sei umgekehrt G_f rekursiv. Dann ist $f(\vec{x}) = \mu z G_f(\vec{x}, z)$ und die Voraussetzung, dass f total ist, heißt $\forall \vec{x} \exists y (\vec{x}, y) \in G_f$. Also ist der μ-Rekursionsschritt wohldefiniert. □

DEFINITION 3.13. Unter einer *Uniformisierung einer Relation $R(\vec{x}, y)$* versteht man eine partielle Funktion φ, so dass

$$(\forall \vec{x})(\exists y R(\vec{x}, y) \leftrightarrow (\varphi(\vec{x})\downarrow \wedge R(\vec{x}, \varphi(\vec{x})).$$

Zum Beispiel kann man fragen, ob jede Borel-Relation auf $\mathbb{R}$ eine Borel-Uniformisierung hat (nein) und ob jede koanalytische Relation eine koanalytische Uniformisierung hat (ja, Satz von Kôndo, [**2**]).

PROPOSITION 3.14 (Die Uniformisierungseigenschaft, Kleene 1936). *Wenn P r.e. ist, dann gibt es eine partielle rekursive Funktion φ, so dass*

$$\exists y P(\vec{x}, y) \leftrightarrow (\varphi(\vec{x})\downarrow \wedge P(\vec{x}, \varphi(\vec{x})).$$

Wenn P r.e. ist und $\forall \vec{x} \exists y P(\vec{x}, y)$, dann gibt es eine rekursive Funktion f, so dass

$$\forall \vec{x} P(\vec{x}, f(\vec{x})).$$

Beweis: Wenn P r.e. ist, dann gibt es eine rekursive Relation R, so dass

$$P(\vec{x}, y) \leftrightarrow \exists z R(\vec{x}, y, z).$$

Indem man wieder y und z in eine *Zahl* $t = \langle y, z\rangle$ kodiert, erhält man

$$\varphi(\vec{x}) \simeq (\mu t R(\vec{x}, (t)_0, (t)_1))_0.$$

Der Algorithmus sucht also die y, so dass $P(\vec{x}, y)$ nachgewiesen wird, und dies ist nicht notwendigerweise das kleinste y, da gleichzeitig auch den z entlang gesucht wird. □

SATZ 3.15 (Eine Charakterisierung der r.e. Mengen, Kleene 1936). *Äquivalent sind*

1. *A is r.e.*
2. *A ist das Bild einer partiellen rekursiven Funktion φ.*
3. *$A = \emptyset$ oder A ist das Bild einer rekursiven Funktion f.*

Beweis: Dies geht in einer Runde. $1 \Rightarrow 2$: Sei $A = W_e$. Wir setzen $\varphi(x) \simeq x + 0 \cdot \varphi_e(x)$. Dann ist das Bild von φ gerade $\mathrm{dom}(\varphi_e)$, und φ is partiell rekursiv.

$2 \Rightarrow 3$: Sei A nicht leer und sei $A = \mathrm{rge}(\varphi_e)$. Wir wählen ein $a \in A$. Wir betrachten den Ansatz

$$f(x) = \begin{cases} z, & \text{wenn } \varphi_e(x)\downarrow \wedge \varphi_e(x) = z, \\ a, & \text{sonst.} \end{cases}$$

Doch dieses f ist womöglich nicht rekursiv, da wir $\varphi_e(x)\downarrow$ vielleicht nicht entscheiden können. Daher ändern wir den Aufzählungsmodus und bauen die Schrittzahl als zweites Argument ein. Sei $I\colon \mathbb{N}^2 \to \mathbb{N}$ rekursiv und bijektiv. Dann setzen wir

$$f(I(x,s)) = \begin{cases} z, & \text{wenn } \varphi_{e,s}(x)\downarrow \wedge \varphi_{e,s}(x) = z \\ a & \text{sonst.} \end{cases}$$

Dann ist f total, da I surjektiv ist. Außerdem ist von $z = I(x,s)$ aus x und s berechenbar. Somit ist f rekursiv.

$3 \Rightarrow 1$: Wenn A leer ist, dann ist A das Bild der überall undefinierten Funktion. Es sei nun $A = \mathrm{rge}(f)$, f total und rekursiv. Dann setzen wir $\varphi(x) \simeq \mu z f(z) = x$. Dann ist $\vec{x} \in A \leftrightarrow \exists z f(z) = x$, und $\{(\vec{x}, z) : f(x) = z\}$ ist rekursiv. □

SATZ 3.16 (Eine Charakterisierung der rekursiven Mengen, Kleene 1936). *Äquivalent sind*

1. *A is rekursiv.*
2. *A ist das Bild einer rekursiven monoton wachsenden Funktion f.*

Beweis: Sei A rekursiv und nicht leer und sei $a = \min(A)$. Wir setzen

$$\begin{aligned} f(0) &= a, \\ f(n+1) &= \begin{cases} n+1, & \text{wenn } n+1 \in A, \\ f(n), & \text{sonst.} \end{cases} \end{aligned}$$

Nun von 2 nach 1: Wenn A endlich ist, dann ist A rekursiv. Sei nun A unendlich und das Bild einer monoton wachsenden rekursiven Funktion f.

$$z \in A \leftrightarrow z \in \{f(0), \ldots, f(\mu x(f(x) > z))\},$$

und die rechte Seite ist rekursiv. □

SATZ 3.17 (Post 1943, Kleene 1943, Mostowski 1947). *Jede Menge A is rekursiv genau dann, wenn A und $\bar{A} = \mathbb{N} \setminus A$ beide r.e. sind.*

Beweis: Sei A rekursiv mit charakteristischer Funktion c_A. Dann ist $A = \mathrm{dom}(\varphi)$ und $\bar{A} = \mathrm{dom}(\psi)$ für

$$\begin{aligned} \varphi(x) &\simeq \begin{cases} 1, & \text{wenn } c_A(x) = 1, \\ \uparrow, & \text{sonst.} \end{cases} , \\ \psi(x) &\simeq \begin{cases} 0, & \text{wenn } c_A(x) = 0, \\ \uparrow, & \text{sonst.} \end{cases} \end{aligned}$$

Nun seien für die umgekehrte Richtung A und $\bar{A}$ beide r.e. und nicht leer. Es gibt rekursive Aufzählungen f, g, $A = \mathrm{rge}(f)$, $\bar{A} = \mathrm{rge}(g)$. Dann sind die beiden Aufzählungen disjunkt und sie decken zusammen ganz $\mathbb{N}$ ab. Um $n \in A$ zu entscheiden, lassen wir beide Aufzählungen abwechselnd laufen, bis wir zu einem x gelangen, so dass $n = f(x)$ oder $n = g(x)$. Im ersten Fall ist $n \in A$, im zweiten Fall ist $n \in \bar{A}$. □

SATZ 3.18 (Post 1944). *Jede unendliche r.e. Menge hat eine unendliche rekursive Teilmenge.*

Beweis: Sei A unendlich, und sei $A = \mathrm{rge}(f)$ für eine rekursive Funktion f. Dann definieren wir g durch primitive Rekursion aus f:

$$\begin{aligned} g(0) &= f(0), \\ g(n+1) &= f(\mu y(f(y) > g(n))). \end{aligned}$$

Nach dem Satz über die monoton aufsteigende Aufzählung ist das Bild von g rekursiv. Außerdem ist $\mathrm{rge}(g) \subseteq \mathrm{rge}(f) = A$. □

SATZ 3.19 (Post 1943, Mostowski 1947). *Der Verband der r.e. Mengen.*

1. *Die Menge aller r.e Menge zusammen mit der Relation $\subseteq$ ist ein distributiver Verband mit einem kleinsten und einem größten Element. Die rekursiven Mengen sind genau die Mengen, die ein Komplement haben.*
2. *Die Bilder und die Urbilder r.e. Mengen unter partiellen rekursiven Funktionen sind wieder r.e.*

Beweis: 1. Das kleinste und das größte Element sind $\emptyset$ und $\mathbb{N}$. Die Aussage über die Komplemente folgt aus 3.17. Um zu zeigen, dass mit A und B auch $A \cap B$ und $A \cup B$ rekursiv aufzählbar sind, lassen wir $A = \mathrm{rge}(f)$ und $B = \mathrm{rge}(g)$ abwechselnd ein Element aufzählen. Falls x in $\{g(0), \ldots, g(n)\} \cap \{f(0), \ldots, f(n)\}$ vorkommt, zählen wir x in $A \cap B$ auf, dasselbe für $\cup$.

2. Sei A r.e., und sei φ eine partielle rekursive Funktion. $\varphi(A)$ besteht aus allen Elementen $\varphi(x)$ für $x \in A$. Um $\varphi(A)$ rekursiv aufzuzählen, genügt es, A aufzuzählen und in die Aufzählung eingewoben werden immer mehr Schritte an $\varphi(x)$ zu rechnen.

Ähnlich wird $\varphi^{-1}(A)$ aufgezählt: immer größere Stücke von A werden aufgezählt und immer mehr Schritte wird $\varphi(x)$ gleichzeitig an immer mehr Argumenten x berechnet. Falls ein x gefunden wird, so dass $\varphi(x)\downarrow$ und $\varphi(x) \in A$, dann wird es in $\varphi^{-1}(A)$ aufgenommen. □

SATZ 3.20 (Post 1943, Mostowski 1947). *Die Boolesche Algebra der rekursiven Mengen.*

1. *Die Menge aller rekursiven Menge zusammen mit der Relation $\subseteq$ ist eine Boolesche Algebra.*
2. *Die Urbilder rekursiver Mengen unter rekursiven Funktionen sind wieder rekursiv.*

Beweis: Nur der Teil über die Urbilder ist neu. Da $f^{-1}(\bar{A}) = \overline{f^{-1}(A)}$, sind sowohl $f^{-1}(A)$ als auch $\overline{f^{-1}(A)}$ rekursiv aufzählbar, und damit ist nach 3.17 $f^{-1}(A)$ rekursiv. □

PROPOSITION 3.21 (Die Reduktionseigenschaft, Rosser 1936, Kleene 1950). *Zu je zwei r.e. Mengen A und B gibt es r.e. Teilmengen $A' \subseteq A$, $B' \subseteq B$, so dass*

$$A' \cap B' = \emptyset \text{ und } A' \cup B' = A \cup B.$$

Beweis: Sei $x \in A \leftrightarrow \exists y R(x,y)$ und $x \in B \leftrightarrow \exists y Q(x,y)$ mit rekursiven R, Q. Für die $x \in A \cap B$ wird beschlossen: wenn x in A $\leq$-früher vorkommt als in B, so nehmen wir $x \in A'$, andernfalls in B'. Die Menge A wird also

bei gleichem Zeugen y bevorzugt.

$$\begin{aligned} x \in A' &\leftrightarrow \exists y[R(x,y) \wedge \forall z \leq y) \neg Q(x,z)] \\ x \in B' &\leftrightarrow \exists y[Q(x,y) \wedge \forall z < y) \neg R(x,z)] \end{aligned}$$

□

3. Diagonalisierung

PROPOSITION 3.22 (Eine rekursive Version von Cantors Satz. Kleene 1936, Turing 1936). *Es gibt keine rekursive Funktion, die mindestens einen Index jeder rekursiven 0,1-wertigen Funktion aufzählt.*

Beweis. Sei f rekursiv, so dass $\varphi_{f(x)}$ total und ist für jedes x. Wir zeigen, dass eine Funktion nicht in der Aufzählung vorkommt, nämlich die Funktion

$$g(x) = 1 - \varphi_{f(x)}(x).$$

g hat höchstens 0 und 1 als Werte und g ist partiell rekursiv nach dem Aufzählungstheorem 3.7, und g ist total nach der Voraussetzung über f. Aber $g(x) \neq \varphi_{f(x)}(x)$, also ist $g \neq \varphi_{f(x)}$ für jedes x, also kommt kein Index von g in der Aufzählung durch f vor. □

Wir zeigten gerade, dass

$$\text{Tot} = \{x \,:\, \varphi_x \text{ ist total}\}$$

nicht r.e. ist. Insbesondere ist Tot nicht rekursiv, es gibt also keinen Algorithmus, der entscheidet, ob ein Index ein Index einer totalen Funktion ist.

DEFINITION 3.23.

(1) $K = \{x \,:\, \varphi_x(x)\downarrow\}$ heißt das diagonale Halteproblem.
(2) $K_0 = \{\langle x, e\rangle \,:\, \varphi_e(x)\downarrow\}$ heißt das zweistellige Halteproblem.

SATZ 3.24 (Der Kern aller Unentscheidbarkeitsresultate. Post 1922, Gödel 1931, Kleene 1936). *Es gibt eine nicht rekursive, r.e. Menge. Die Menge*

$$K = \{x \,:\, \varphi_x(x)\downarrow\}$$

ist ein Beispiel.

Beweis: Nach dem Aufzählungssatz 3.7 gibt es eine partielle rekursive Funktion φ, so dass

$$\varphi(x) \simeq \varphi_x(x).$$

Dann ist K r.e, da $K = \text{dom}(\varphi)$. Wir zeigen auf zwei verschiedene Arten, dass K nicht rekursiv ist.

1. Art: Wenn K rekursiv wäre, so wäre auch die Funktion ($\varphi(x) = 0$ wenn $x \in \bar{K}$ und undefiniert sonst) partiell rekursiv. Dann wäre $\varphi \simeq \varphi_e$

für einen Index e. Aber dann ist $\varphi_e(e) \downarrow \leftrightarrow e \notin \bar{K}$, im Widerspruch zur Definition von K.

2. Art: Wenn K rekursiv wäre, dann wäre $\bar{K}$ r.e. Aber dann ist $x \in \bar{K} \leftrightarrow x \notin W_x$, also unterscheidet sich $\bar{K}$ von der x-ten r.e. Menge im Element x, für alle x, und daher ist $\bar{K}$ nicht r.e. □

Definition 3.25 (Kleene 1950, Trakhtenbrot 1953). Zwei disjunkte Mengen A und B heißen *rekursiv trennbar*, wenn es eine rekursive Menge C gibt, so dass $A \subseteq C$ und $B \subseteq \bar{C}$. Im gegenteiligen Falle heißen A und B *nicht rekursiv trennbar* oder „rekursiv untrennbar" (eine direkte Übersetzung von recursively inseparable, meinem Sprachgefühl nach ist dieses unmögliches Deutsch).

Satz 3.26 (Rosser 1936, Kleene 1950, Novikov, Trakhtenbrot 1953). *Es gibt zwei disjunkte nicht rekursiv trennbare r.e. Mengen.*

Beweis: Wir nehmen

$$A = \{x \,:\, \varphi_x(x) = 0\}$$

und

$$B = \{x \,:\, \varphi_x(x) = 1\}.$$

Falls es eine rekursive trennende Menge C gäbe, dann wäre die charakteristische Funktion c_C von C rekursiv und hätte einen Index e. Dann wäre

$$\varphi_e(e) \simeq 0 \to e \in A \to e \in C \to \varphi_e(e) = c_C(e) = 1,$$
$$\varphi_e(e) \simeq 1 \to e \in B \to e \in \bar{C} \to \varphi_e(e) = c_C(e) = 0.$$

□

Satz 3.27 (Die Unentscheidbarkeit des Halteproblems. Turing 1936). *Die Menge*

$$K_0 = \{\langle x, e\rangle \,:\, x \in W_e\}$$

ist r.e. und nicht rekursiv. Die Menge K_0 wird das Halteproblem genannt.

Beweis: K_0 ist r.e. nach dem Aufzählungstheorem. Wenn K_0 rekursiv wäre, so auch K, da

$$x \in K \leftrightarrow \langle x, x\rangle \in K_0.$$

□

Die Unentscheidbarkeit des Halteproblems ist nur die Spitze eines Eisbergs aus unentscheidbaren Mengen.

Definition 3.28. Eine Menge A heißt *Indexmenge*, wenn es eine Menge $\mathscr{A}$ partieller rekursiver Funktionen gibt, so dass A die Indexmenge von $\mathscr{A}$ ist, das heißt:

$$A = \{x \,:\, \varphi_x \in \mathscr{A}\}.$$

Wenn A die Indexmenge von $\mathscr{A}$ ist, dann schreiben wir $A = \theta\mathscr{A}$.

Definition 3.29 (Decker, Rice 1953). Eine Menge partieller rekursiver Funktionen heißt *vollständig rekursiv*, wenn ihre Indexmenge rekursiv ist.

Satz 3.30 (Rice 1953). *Jede Menge partieller rekursiver Funktionen $\mathscr{A}$ ist vollständig rekursiv gdw $\mathscr{A} = \emptyset$ oder $\mathscr{A}$ die Klasse aller partieller/n rekursiver/n Funktionen ist.*

Beweis: Wenn $\mathscr{A} = \emptyset$, dann $\theta\mathscr{A} = \emptyset$, und wenn $\mathscr{A}$ die Menge aller partiellen rekursiven Funktionen ist, dann ist $\theta\mathscr{A} = \mathbb{N}$. (Strenggenommen steht hier statt $\mathbb{N}$ die rekursive Bildmenge der Funktion aus dem ersten Schritt im Beweise der Kleene'schen Normalform 2.1. Doch dies hängt ja von der willkürlich gewählten Kodierung ab.) Nun nehmen wir an, dass $\mathscr{A}$ echt dazwischen liegt. Dann gibt es a und b, so dass $\varphi_a \in \mathscr{A}$ und $\varphi_b \notin \mathscr{A}$. Wenn die nirgendwo definierte Funktion nicht in $\mathscr{A}$ ist, dann setzen wir die rekursive Funktion f an als

$$\varphi_{f(x)} = \begin{cases} \varphi_a, & \text{wenn } x \in K, \\ \text{undefiniert}, & \text{sonst.} \end{cases}$$

Dann ist

$$x \in K \leftrightarrow \varphi_{f(x)} \in \mathscr{A} \leftrightarrow f(x) \in A.$$

Also ist A nicht rekursiv, da K nicht rekursiv ist. Falls die nirgendwo definierte Funktion nicht in $\bar{\mathscr{A}}$ ist, dann nehmen wird φ_b und $\bar{K}$ und zeigen, dass $\bar{A} = \theta(\bar{\mathscr{A}})$ nicht rekursiv ist. □

Rice' Satz ist stark, denn er umfasst zahlreiche Unentscheidbarkeitsresultate. Jede nicht triviale Eigenschaft von rekursiven Funktionen ist unentscheidbar. Zum Beispiel: die Eigenschaft, überall undefiniert zu sein, an einem festen Argument definiert zu sein, total zu sein, einen gewissen Wert im Bild zu haben, surjektiv zu sein, mit einer gegebenen partiellen rekursiven Funktion übereinzustimmen.

4. Der Fixpunktsatz

Über die partiellen rekursiven Funktionen kann man nicht diagonalisieren, denn jede rekursive Diagonalisierung $\varphi_e \mapsto \varphi_{f(e)}$ hat einen Fixpunkt.

Io sentiva osannar di coro in coro
al punto fisso che li tiene alli ubi
(Dante, *Paradiso*, XXVIII)

Ich hörte von Chor zu Chor Hosanna
(hoshianna hebr. errette uns) singen, zu dem Fixpunkt hin, der jeden am Platz hält
(Dante, *Paradiso*, XXVIII)

Satz 3.31 (Der Fixpunktsatz. Kleene 1938). *Zu jeder rekursiven Funktion f gibt es einen Index e, so dass*

$$\varphi_e \simeq \varphi_{f(e)}.$$

Beweis:
Sei b ein Index, so dass

$$\varphi_{\varphi_b(e)} \simeq \varphi_{f(\varphi_e(e))}.$$

Solch ein b beschafft man sich zum Beispiel mit einer Programmierungsbetrachtung. Zu jedem e, x, wird versucht $\varphi_e(e)$ zu berechnen. Falls dies gelingt, kann man (da f total ist) $f(\varphi_e(e))$ berechnen, das wir $\varphi_b(e)$ nennen, und dann versuchen $\varphi_{f(\varphi_e(e))}(x)$ auszurechnen. Das Verfahren hängt nicht von x ab. φ_b ist eine rekursive Funktion in der Variablen e.

Alternative:
Man kann auch das s-n-m-Theorem mit x von dort ist nun e und y von dort ist nun x nehmen. Erinnerung: Es gibt eine primitiv-rekursive injektive Funktion

$$S_n^m(e, x_0, \ldots, x_{n-1}),$$

so dass

$$\varphi_{S_n^m(e,x_0,\ldots,x_{n-1})}(y_0, \ldots, y_{m-1}) \simeq \varphi_e(x_0, \ldots, x_{n-1}, y_0, \ldots, y_{m-1}).$$

Sei nach dem s-m-n-Theorem 3.9 angewandt auf $\varphi_{f(\varphi_e(e))}(x) \simeq \varphi_k(e, x)$ (mit einem geeigneten k) gibt es ein $S_1^{\lg(x)}$, das das Argument e in den Index holt.

Dann nehmen wir $e = b$ und erhalten

$$\varphi_{\varphi_b(b)} \simeq \varphi_{f(\varphi_b(b))}.$$

Also ist $\varphi_b(b)$ ein Fixpunkt von f. □

Satz 3.32 (Der zweite Fixpunktsatz, Kleene 1938). *Zu jeder partiellen rekursiven Funktion ψ gibt es einen Index e, so dass*

$$\varphi_e(x) \simeq \psi(e, x).$$

Wir halten irgendeine rekursive Funktion h fest. Da die Funktion $\psi(h(z), x)$ partiell rekursiv ist, hat sie einen Index a, der von h abhängt. Nach dem s-m-n-Satz ist

$$\psi(h(z), x) \simeq \varphi_{S_1^1(a,z)}(x).$$

Dies gilt insbesondere für $h(z) = S_1^1(z, z)$ und das a für dieses h, also

$$\varphi(S_1^1(z, z), x) \simeq \varphi_{S_1^1(a,z)}(x).$$

Nun setzen wir $e = S_1^1(a, a)$ und erhalten

$$\psi(e, x) \simeq \psi(S_1^1(a, a), x) \simeq \varphi_{S_1^1(a,a)}(x) \simeq \varphi_e(x).$$

□

Satz 3.33 (Kleene 1952). *Die Klasse der partiellen rekursiven Funktionen ist die kleinste Menge von Funktionen, die*

1. *die Ausgangsfunktionen und f_{FU} (s.u) und die Vorgängerfunktion enthält,*
2. *unter Komposition abgeschlossen ist,*

3. unter Definition durch Fallunterscheidung abgeschlossen ist,

4. unter Fixpunkten (von totalen Funktionen!) abgeschlossen ist.

Beweis: Unter Definition durch Fallunterscheidung verstehen wir folgendes Schema:

$$f(\vec{x}) = \begin{cases} g(\vec{x}), & \text{wenn } t(\vec{x}) = 0, \\ h(\vec{x}), & \text{sonst.} \end{cases}$$

aus partiellen rekursiven g, h, und totalem t. Wir setzen diese in die Ausgangsfunktion f_{FU} ein:

$$f_{FU}(x, y, z) = \begin{cases} x, & \text{wenn } z = 0, \\ y, & \text{sonst.} \end{cases}$$

Sei nun $\mathscr{C}$ die kleinste Menge von Funktionen, die die angegebenen Klauseln 1. bis 4. erfüllt. Dann ist $\mathscr{C}$ eine Teilmenge der Menge der partiellen rekursiven Funktionen, denn nach dem Fixpunktsatz gibt es immer einen partiellen rekursiven Fixpunkt.

Um die umgekehrte Inklusion zu beweisen, haben wir zu zeigen , dass die primitive Rekursion und die μ-Rekursion nicht aus $\mathscr{C}$ hinausführen.

Primitive Rekursion. Sei

$$\begin{aligned} \varphi(\vec{x}, 0) &= \psi(\vec{x}), \\ \varphi(\vec{x}, y+1) &= \varrho(\vec{x}, y, f(\vec{x}, y)). \end{aligned}$$

Dann kann φ als Fixpunkt der folgenden Gleichung erhalten werden

$$\varphi(\vec{x}, y) = \begin{cases} \psi(\vec{x}), & \text{wenn } y = 0, \\ \varrho(\vec{x}, y-1, f(\vec{x}, y-1)), & \text{sonst.} \end{cases}$$

und hier benutzen wir nur die Vorgängerfunktion, die Fallunterscheidung und die Komposition.

μ-Rekursion. Sei

$$\varphi(\vec{x}) = \mu y(\psi(\vec{x}, y) = 0 \wedge \forall z < y \psi(\vec{x}, z) \downarrow).$$

Dann kann zunächst $\varrho = \varphi_e$ als Fixpunkt der folgenden Gleichung erhalten werden

$$\varrho(\vec{x}, y) = \begin{cases} y, & \text{wenn } \psi(\vec{x}, y) = 0, \\ \varrho(\vec{x}, y+1), & \text{sonst.} \end{cases}$$

Dann ist $\varphi(\vec{x}) = \varrho(\vec{x}, 0)$, und hier benutzen wir nur die Nullfunktion, die Nachfolgerfunktion, die Fallunterscheidung und die Komposition. □

5. Berechnungen mit Orakeln und die Turinggrade

Definition 3.34 (Relative Berechenbarkeit, Turing, 1939). Sei g eine totale Funktion. Die Menge der *in g-rekursiven Funktionen* ist die kleinste Menge von Funktionen, die alle Ausgangsfunktionen und g enthält und gegen Komposition, primitive Rekursion und eingeschränkte μ-Rekursion abgeschlossen ist. Für eine Menge A ist die Menge der in A rekursiven

Funktionen die Menge der in c_A rekursiven Funktionen. Ein Prädikat ist *rekursiv in* g oder in A, wenn seine charakteristische Funktion dies ist.

Im Buch von Odifreddi [**6**] gibt es eine Sokrates-Stelle über das Orakel von Delphi:

$\chi\alpha\iota\ \delta\eta\ \pi o\tau\varepsilon\ \chi\alpha\iota\ \varepsilon\iota\varsigma\ \Delta\varepsilon\lambda\varphi o\upsilon\varsigma\ \varepsilon\lambda\vartheta\omega\nu$
$\varepsilon\tau o\lambda\mu\eta\sigma\varepsilon\ \tau o\tilde{\upsilon}\tau o\ \mu\alpha\nu\tau\varepsilon\upsilon\sigma\alpha\sigma\vartheta\alpha\iota.$
$\tau\iota\ \pi o\tau\varepsilon\ \lambda\varepsilon\gamma\varepsilon\iota\ o\ \vartheta\varepsilon o\varsigma;$
$o\upsilon\ \gamma\alpha\rho\ \delta\eta\pi o\upsilon\ \psi\varepsilon\upsilon\delta\varepsilon\tau\alpha\iota\gamma\varepsilon$
(Plato, *Socrates' Apology*)

And thus once, gone even to Delphi,
he dared to consult the oracle on his:
what does the god say?
He certainly does not lie.
(Plato, *Socrates' Apology*)

Ein Orakel g ist i.a. nicht rekursiv und es hilft, die Berechnung jeder Funktion, die rekursiv in g ist, an den fraglichen Punkten weiterzuführen, an denen dann g angerufen wird. Das Orakel beantwortet jede Anfrage umsonst. Turing wählte 1939 dieses Wort.

Definition 3.35. Seien f und g Funktionen. Wir sagen f ist Turing-reduzierbar auf g, in Zeichen $f \leq_T g$, wenn f rekursiv in g ist. Wir sagen f und g sind Turing-äquivalent und schreiben $f \equiv_T g$, wenn $f \leq_T g$ und $g \leq_T f$.

Überlegen Sie sich, dass $\leq_T$ transitiv, aber nicht antisymmetrisch ist!

Definition 3.36 (Post 1948). Die Äquivalenzklassen totaler Funktionen unter der Turing-Äquivalenz heißen *Turinggrade*. Die Menge aller Grade heißt $\mathcal{D}$. Die Quasihalbordnung (i.e. transitive Rel.) $\leq_T$ induziert eine Halbordnung (i.e. antisymmetrische transitive Rel.) $\leq$ auf den Graden. Die Struktur $(\mathcal{D}, \leq)$ heißt die Halbordnung der Turinggrade. Die Struktur $(\mathcal{E}, \leq)$ heißt die Halbordnung der r.e. Turinggrade.

KAPITEL 4

Posts Problem, die "finite injury" Methode

DEFINITION 4.1. Undatiert. Sei α eine partielle Funktion. Die Menge der in α partiellen rekursiven Funktion ist die kleinste Menge von Funktionen, die

(1) die Ausgangsfunktionen $o(x) = 0$, $S(x) = x + 1$, $p_i^n(\vec{x}) = x_i$ und α enthält und

(2) unter Verkettung, primitiver Rekursion und unter uneingeschränkter μ-Rekursion abgeschlossen ist. Uneingeschränkte μ-Rekursion bedeutet: Die Bedingung $\forall \vec{x} \exists y \psi(\vec{x}, y) = 0$ wird fallengelassen und stattdessen setzt man

$$\varphi(\vec{x}) \simeq \mu y[\forall z \leq y \psi(\vec{x}, z)\downarrow \wedge \psi(\vec{x}, y) = 0].$$

(Die heißt nach unseren Konventionen: $\varphi(\vec{x})$ ist nicht definiert, wenn es kein solches y gibt.)

Vorsicht: Nicht immer gilt das analoge Resultat für Berechenbarkeit in α statt Berechenbarkeit (Slang: "the result relativizes"), nicht einmal immer für totale α.

DEFINITION 4.2. Sei P die Menge der partiellen Funktionen von $\mathbb{N}$ nach $\mathbb{N}$. Sei T die Menge der totalen Funktionen von $\mathbb{N}$ nach $\mathbb{N}$, für Mengentheoretiker $T = {}^{\mathbb{N}}\mathbb{N}$.

Ein *Funktional* ist eine partielle Funktion $F\colon P^k \times \mathbb{N}^m \to \mathbb{N}$ für $k, m \in \mathbb{N}$. Ein *eingeschränktes Funktional* ist eine partielle Funktion $F\colon T^k \times \mathbb{N}^m \to \mathbb{N}$ für $k, m \in \mathbb{N}$. Ein *totales Funktional* ist eine totale Funktion $F\colon T^k \times \mathbb{N}^m \to \mathbb{N}$ für $k, m \in \mathbb{N}$. Es wird also nicht Totalität auf dem größeren Raum gefordert.

Wir sprechen auch über (eingeschränkte) Relationen auf Zahlen und Funktionen, indem wir uns auf das charakteristische (eingeschränkte) Funktional beziehen.

DEFINITION 4.3. [Kleene 1952] Seien α_i, $1 \leq i \leq m$, partielle Funktionen. Das Funktional $F(\alpha_1, \ldots, \alpha_n, \vec{x})$ ist ein *partielles rekursives Funktional*, wenn es aus

(1) den Ausgangsfunktionen $o(x) = 0$, $S(x) = x + 1$, $p_i^n(\vec{x}) = x_i$ und den α_i, $1 \leq i \leq m$,

(2) durch Verkettung, primitive Rekursion und uneingeschränkte μ-Rekursion herstellbar ist.

Dies bedeutet, dass $\lambda\vec{x}.F(\alpha_1,\ldots,\alpha_n,\vec{x})$ *uniform* partiell rekursiv in den α_i's ist.

PROPOSITION 4.4 (Die Einsetzungseigenschaft. Kleene 1952). *Wenn $F(\alpha,z)$ und $G(\beta,x)$ partielle rekursive Funktionale sind, dann ist such*

$$H(\alpha,x)\simeq G(\lambda z.F(\alpha,z),x)$$

ein partielles rekursives Funktional.

SATZ 4.5 (Der Satz von der Normalform für partielle rekursive Funktionale, Kleene 1952). *Es gibt eine primitiv-rekursive Funktion $\mathcal{U}$ und für jedes $m,n\in\mathbb{N}$ ein primitiv-rekursives Prädikat $\mathcal{T}_{m,n}$, so dass für jedes partielle rekursive Funktinal in m Funktionsvariablen und n Zahlvariabalen ein Index e existiert, so dass:*

1. $\forall\vec{\alpha}\forall\vec{x}(F(\alpha_1,\ldots,\alpha_m,x_1,\ldots,x_n)\downarrow\leftrightarrow$
   $\exists y\mathcal{T}_{m,n}(e,x_1,\ldots,x_n,\alpha_1,\ldots,\alpha_m,y))$.
2. $F(\alpha_1,\ldots,\alpha_m,x_1,\ldots,x_n)\simeq\mathcal{U}(\mu y\mathcal{T}_{m,n}(e,x_1,\ldots,x_n,\alpha_1,\ldots,\alpha_m,y))$.

Beweis: Wir können die Funktionen α_i nicht in den Berechnungen verwenden, da sie womöglich nicht rekursiv sind. Daher benützen wir nur endliche Anfangstücke ihrer Graphen. Wir weisen hier nur auf die Änderungen im Vergleich zum Beweis des Theorems 2.1 hin.

1. Der Index eines partiellen rekursiven Funktionals kann wie gehabt definiert werden, wir fügen nur Klauseln für die Funktionsargumente α_i hinzu, als ob sie Ausgangsfunktionen wären.

2. Wir zeichnen wie gehabt Berechnungsbäume. Einige Knoten tragen nun die Beschriftung $\alpha_i(x)=z$.

3. Den Knoten der Berechnungsbäume werden Zahlen der Art

$$\langle e,\langle x_1,\ldots,x_n\rangle,\langle\bar{\alpha_1}(K),\ldots,\bar{\alpha_m}(K)\rangle,z\rangle$$

zugeordnet, wobei

$$\bar{\alpha}(K)=\prod_{k<K}p_k^{\alpha(k)+1}$$

der Anfang des Wertverlaufs for α ist und, falls $\alpha(k)$ undefiniert ist, der entsprechende Faktor ausfällt.

4. Die Definition von $\mathcal{T}$ bleibt dieselbe außer lokalen Änderungen in den Klauseln des Typs $B(y)$ für die atomaren Bäume: Für die Knoten mit Beschriftung $\alpha_i(x)=z$ muß dann $\bar{\alpha_i}(x+1)(x)=z$ eingesetzt werden.

5. $\mathcal{T}_{m,n}$ ist definiert durch

$$\begin{aligned}\mathcal{T}_{m,n}(e,x_1,\ldots,x_n,\alpha_1,\ldots,\alpha_m,y)\leftrightarrow\mathcal{T}(y)\wedge\\ (y)_{1,1}=e\wedge(y)_{1,2}=\langle x_1,\ldots,x_n\rangle\\ \wedge\lg((y_{1,3})=m\wedge(\forall i)_{1\le i\le m}\bar{\alpha}_i(\lg((y)_{1,3,i}))=(y)_{1,3,i}).\end{aligned}$$

6. Da die Knoten im Berechnungsbaum nun Quadrupel sind, haben wir

$$\mathcal{U}(y)=(y)_{1,4}.$$

□

SATZ 4.6 (Der Satz von der Normalform für eingeschränkte partielle rekursive Funktionale, Kleene 1952). *Es gibt eine primitiv-rekursive Funktion $\mathcal{U}$ und für jedes $m, n \in \mathbb{N}$ ein primitiv-rekursives Prädikat $\mathcal{T}_{m,n}$ von nur numerischen Variablen, so dass für jedes eingeschränkte partielle rekursive Funktional in m Funktionsvariablen (für totale Funktionen!) und n Zahlvariabalen ein Index e existiert, so dass:*

1. $\forall \vec{g} \forall \vec{x}(F(g_1, \ldots, g_m, x_1, \ldots, x_n) \downarrow$
   $\leftrightarrow \exists y \mathcal{T}_{m,n}(e, x_1, \ldots, x_n, \bar{g_1}(y), \ldots, \bar{g_m}(y), y))$.
2. $F(g_1, \ldots, g_m, x_1, \ldots, x_n) \simeq \mathcal{U}(\mu y \mathcal{T}_{m,n}(e, x_1, \ldots, x_n, \bar{g_1}(y), \ldots, \bar{g_m}(y), y))$.

Beweis: Ganz ähnlich wie oben.

DEFINITION 4.7. Seien $e, n, s \in \mathbb{N}$. Unter s kann man sich stage, Stadium, oder Schrittzahl vorstellen. Sei $A \subseteq \mathbb{N}$.

1. $\varphi_e^{A,n}$ oder $\{e\}^n$ ist die e-te in A partielle rekursive n-stellige Funktion:
$$\varphi_e^{A,n}(\vec{x}) \simeq \{e\}^n(\vec{x}) \simeq \mathcal{U}(\mu y(\mathcal{T}_{n,1}(e, \vec{x}, \bar{c_A}(y), y))).$$
2. $\varphi_{e,s}^{A,n}$ oder $\{e\}_s^{A,n}$ ist die Approximation bis zum Berechnungsbaum der Größe s für die e-te in A partielle rekursive n-stellige Funktion:
$$\varphi_{e,s}^{A,n}(\vec{x}) \simeq \{e\}_s^{A,n}(\vec{x}) \simeq \mathcal{U}(\mu y(\mathcal{T}_{n,1}(e, \vec{x}, \bar{c_A}(y), y) \wedge y < s)).$$

Wir haben also:

$$\varphi_{e,s}^{A,n}(\vec{x}) \simeq \{e\}_s^{A,n}(\vec{x}) \simeq \begin{cases} \varphi_e^{A,n}(\vec{x}) & \text{wenn } (\exists y < s)\mathcal{T}_{n,1}(e, \vec{x}, \bar{c_A}(y), y) \\ \text{undefiniert} & \text{sonst.} \end{cases}$$

DEFINITION 4.8. Wir schreiben $W_e^{A,n} = \mathrm{dom}(\varphi_e^{A,n})$ und $W_{e,s}^{A,n} = \mathrm{dom}(\varphi_{e,s}^{A,n})$.

KOROLLAR 4.9. *Kleene 1952 Wenn $F(\alpha, x)$ ein partielles rekursives Funktional ist und $F(\alpha, x) \simeq y$, dann gilt die Kompaktheit: Es gibt eine endliche Funktion $u \subseteq \alpha$, so dass $F(u, x) \simeq y$, und es gilt die Monotonie, $\alpha \subseteq \beta$ impliziert $F(\beta, x) \simeq y$.*

Welche Grade kennen wir?

DEFINITION 4.10. $\mathbf{o}'$, gelesen "$\mathbf{o}$ jump" oder „der Sprung von $\mathbf{o}$", ist der Grad von K (oder von K_0), $\mathbf{o}$ ist der Grad der rekursiven Mengen.

Natürlich ist $\mathbf{o} \neq \mathbf{o}'$.

PROPOSITION 4.11. *A r.e. Dann ist* $\deg(A) \leq \mathbf{o}'$.

Beweis: $A = W_e$ für ein e. Dann ist $x \in A \leftrightarrow \langle x, e \rangle \in K_0$. Der Beweis für K anstelle von K_0 geht am besten mit dem s-m-n-Satz. Führen Sie ihn als Übung. □

$\mathbf{o}'$ ist also maximal unter den Graden rekursiv aufzählbarer Mengen. Gibt es noch weitere Grade?

PROPOSITION 4.12. *$A \leq_T B$ gdw $(\exists x)(\forall y)(y \in A \leftrightarrow y \in W_x^B)$ und $(\exists x)(\forall y)(y \in \bar{A} \leftrightarrow y \in W_x^B)$.*

Beweis: Dies ist die auf B-Berechenbarkeit und B-Aufzählbarkeit relativierte Form von Satz 3.17. □

FRAGE 4.13 (Post 1944. "Post's problem"). *Gibt es eine r.e. Menge A, so dass A nicht rekursiv ist und $K \not\leq_T A$ ist?*

Wir folgen nun Hartley Rogers Juniors Darstellung aus *dem* Klassiker der Rekursionstheorie: "Theory of Recursive Functions and Effective Computablilty" MIT Press 1987, unveränderter Nachdruck der Auflage von 1967, pp. 163 – 166. In Odifreddis Buch nimmt die Beantwortung der Postschen Frage 35 Seiten in Anspruch und ist daher für unsere kurze Vorlesung nicht geeignet, obwohl Odifreddis Darstellung natürlich ausgezeichnet ist. Posts Frage war zwölf Jahre offen, und Odifreddi beschreibt die geschichtliche Entwicklung zahlreicher Versuche bis zur Lösung durch Friedberg und Muchnik 1956, unabhängig voneinander.

Sie können in Odifreddi (pp 250 – 304) finden, warum nur so eine inkonstruktive Methode (finite injury priority method genannt) zum Ziel führen kann.

Beweise des Friedberg-Muchnik-Satzes stehen im genannten Buch von Rogers Seite 163–166, in Soares Recursively Enumerable Sets and Degrees S. 118–120, Oddifreddi, Classical Recursion Theory S. 194–304, S. Barry Cooper, Computablility Theory S. 238–241, [**1**]. Welcher spricht Sie am meisten an?

SATZ 4.14 (Friedberg, Muchnik, 1956). *Es gibt zwei r.e. Mengen A und B, so dass $A \not\leq_T B$ und $B \not\leq_T A$.*

DEFINITION 4.15. Wir kodieren endliche Mengen $\{n_0, \ldots, n_{k-1}\}$ durch Dualentwicklungen $u = \sum_{i<k} 2^{n_i}$. Diese Kodierung hat eine rekursive Umkehrung: Für $u = \sum_{i<k} 2^{n_i}$ setzen wir $D_u = \{n_i \,:\, i < k\}$.

LEMMA 4.16. *Es gibt eine rekursive totale Funktion ρ, so dass für alle endlichen D*

$$W_z^D = \{x \,:\, \exists u, v, w \langle x, u, v, w\rangle \in W_{\rho(z)} \wedge D_u \subseteq D \wedge D_v \subseteq \bar{D}\}$$

Beweis: Wir nützen den Kompaktheit und die Monotonie der Orakelberechnungen und den Normalformensatz für eingeschränkte Funktionale. Genaues können Sie auf Seite 254 in Proposition III.1.4. in Odifreddi nachlesen. □

Wir bezeichnen mit $(W_z^D)_n$ die Menge derjenigen x, die in $\leq n$ Rechenschrittten in $W_{\rho(z)}$ gefunden werden (für genügend großes n, so dass die Prüfungen $D_u \subseteq D$ und $D_v \subseteq \bar{D}$ auch innerhalb dieser Schritte durchgeführt sind und erfolgreich verlaufen sind).

LEMMA 4.17. *Seien die A_i endliche Mengen und $A_{i+1} \supseteq A_i$ und $A = \bigcup A_i$. Dann gilt*

$$\forall a, z(a \in W_z^A \to (\exists m = m(a, z, (A_i)_i))(\forall n \geq m)(a \in (W_z^{A_n})_n)).$$

Beweis: Sei $a \in W_z^A$. Dann werden zu dieser Verifikation nur endlich viele Anfragen an das Orakel gestellt, sagen wir, Anfragen an $A \upharpoonright m' = A \cap \{0, 1, \ldots, m' - 1\}$. Dann gibt es ein $m \geq m'$, so dass für alle $n \geq m$ $A \upharpoonright m' = A_n \upharpoonright m'$. □

Beweis des Satzes 4.14: Wir müssen Anweisungen für die Aufzählungen von A und B geben und sicherstellen, dass es zwei Funktionen f und g gibt, so dass

$$(\forall x)(f(x) \in A \leftrightarrow f(x) \in W_x^B)$$

(also für alle x, $\bar{A} \neq W_x^B$) und

$$(\forall x)(g(x) \in B \leftrightarrow g(x) \in W_x^A).$$

Dann gilt: $\forall x \bar{A} \neq W_x^B$ und daher nicht $A \leq_T B$. Denn letzteres ist nach der nach B relativierten Fassung von Satz 3.17 äquivalent zu

$$\exists x A = W_x^B \wedge \exists y \bar{A} = W_y^B.$$

Ebenso folgt dann $\forall x \bar{B} \neq W_x^A$.

In Odifreddis Buch wird gezeigt, dass solche f und g nicht rekursiv sein können.

Die Konstruktion dieser A und B wird in Schritten vorgenommen. Ein Schritt zerfällt in mehrere Teilschritte, die nicht gezählt werden. Enorm wichtig ist der Zähler $2n + 1$ oder $2n + 2$ für die jeweilige Schrittzahl.

A und B werden als aufsteigende Vereinigungen endlicher Mengen konstruiert. Am Anfang besteht die A-Liste aus $\mathbb{N}$ und die B-Liste aus $\mathbb{N}$.

$$A_n = \{x \,:\, x \text{ erhält ein } + \text{ in der } A\text{-Liste in einem Schritt } \leq n\}.$$

$$B_n = \{x \,:\, x \text{ erhält ein } + \text{ in der } B\text{-Liste in einem Schritt } \leq n\}.$$

Am Schluss setzen wir $A = \bigcup_n A_n$, $B = \bigcup_n B_n$. Die gesetzten $+$-Zeichen werden nie mehr weggenommen. Außerdem gibt es noch $-$-Zeichen und zweimal unendlich viele Marken

$$\boxed{\text{i}}_1,\ i \in \omega,\ \text{für die } A\text{-Liste}$$

$$\boxed{\text{i}}_2,\ i \in \omega,\ \text{für die } B\text{-Liste}.$$

Die Marken sitzen in der Größe ihrer Aufschriften auf den Listen. Man zeigt induktiv über die Schrittzahl, dass dies zu jeder Schrittzahl stimmt. Wenn eine Marke bewegt werden muss, dann müssen sich auch alle Marken auf der selben Liste, die eine höhere Beschriftung tragen, mitbewegen. Im n-ten Schritt sind n Marken gesetzt. Marken dürfen nur zu größeren Elementen derselben Liste verschoben werden. Wir werden zeigen, dass jede Marke nur endlich oft verschoben wird. Minusse dürfen in Plusse verändert werden. Plusse verschwinden nie mehr.

Wenn sich die Marke $\boxed{\text{j}}_1$ nicht mehr bewegt, dann wird sie auf einer Zahl x_j stehen, so dass $x_j \in A \leftrightarrow x_j$ hat ein $+ \leftrightarrow x_j \in W_j^B$.

Wenn wir dies für jedes $\boxed{\text{j}}_1$ arrangieren können, dann haben wir $(\forall j)(\exists x_j)(x_j \in A \leftrightarrow x_j \in W_j^B)$ und daher $A \not\leq_T B$. Umgekehrt arragieren wir für alle Marken mit Index 2: Wenn sich die Marke $\boxed{\text{j}}_2$ nicht mehr bewegt, dann wird sie auf einer Zahl x_j stehen, so dass $x_j \in B \leftrightarrow x_j$ hat ein $+ \leftrightarrow x_j \in W_j^A$.

Definition: Zu jeder Schrittzahl sagen wir eine Zahl ist *frei* in einer Liste, wenn keine größere oder gleiche Zahl ein $+$ oder ein $-$ oder eine Marke $\boxed{\text{j}}_k$ trägt. Eine Zahl heißt *vakant* zu einer Schrittzahl, wenn sie kein Pluszeichen trägt.

Schritt 1: Setze $\boxed{0}_1$ auf die 0 der A-Liste.

Schritt 2: Setze $\boxed{0}_2$ auf die 0 in der B-Liste.

...

Schritt $2n+1$: Setze $\boxed{\text{n}}_1$ auf die kleinste freie Zahl in der A-Liste. Seien (notwendigerweise aufsteigend) $a_0^{(n)}, a_1^{(n)}, \ldots, a_n^{(n)}$ die gegenwärtigen Stellungen der Marken $\boxed{0}_1, \ldots, \boxed{\text{n}}_1$. Berechne n Schritte in der Aufzählungen von $W_0^{B_{2n}}$, $W_1^{B_{2n}}$, ..., $W_n^{B_{2n}}$ also $(W_0^{B_{2n}})_n$, $(W_1^{B_{2n}})_n$, ..., $(W_n^{B_{2n}})_n$ (die alle each vor n liegen). Sei $a_j^{(n)}$ das kleinste Element von $\{a_0^{(n)}, \ldots, a_n^{(n)}\}$, das vakant ist und das in $(W_j^{B_{2n}})_n$ vorkommt. Wenn es kein solches $a_j^{(n)}$ gibt, dann gehe zu Schritt $2n+2$. Wir setzen ein $+$ auf $a_j^{(n)}$ in der A-Liste und ein $-$ auf jede der vakanten Zahlen in der B-Liste, deren zu $\bar{B}_{2n}$-Gehören ausgenutzt wurde zur Berechnung von $a_j^{(n)} \in (W_j^{B_{2n}})_n$. Wenn $j < n$, dann bewege alle Marken in der B-Liste $\boxed{\text{i}}_2$, $j \leq i < n$, zu größeren Zahlen, nämlich zu den ersten $n - j$ freien Zahlen in der B-Liste (also der B-Liste zum Zustand $2n+1$.

Schritt $2n+2$: Setze $\boxed{\text{n}}_2$ auf die kleinste freie Zahl in der B-Liste. Seien (notwendigerweise aufsteigend) $b_0^{(n)}, b_1^{(n)}, \ldots, b_n^{(n)}$ die gegenwärtigen Stellungen der Marken $\boxed{0}_2, \ldots, \boxed{\text{n}}_2$. Berechne n Schritte in der Aufzählungen von $W_0^{A_{2n+1}}$, $W_1^{A_{2n+1}}$, ..., $W_n^{A_{2n+1}}$. Sei $b_j^{(n)}$ das kleinste Element von $\{b_0^{(n)}, \ldots, b_n^{(n)}\}$, das vakant ist und das in $(W_j^{A_{2n+1}})_n$ vorkommt. Wenn es kein solches $b_j^{(n)}$ gibt, dann gehe zu Schritt $2n+3$. Wir setzen ein $+$ auf $b_j^{(n)}$ in der B-Liste und ein $-$ auf jede der vakanten Zahlen in der A-Liste, deren zu $\bar{A}_{2n+1}$-Gehören ausgenutzt wurde zur Berechnung von $b_j^{(n)} \in (W_j^{A_{2n+1}})_n$. Wenn $j < n$, dann bewege alle Marken in der A-Liste $\boxed{\text{i}}_1$, $j < i \leq n$ (beachten Sie den Unterschied zur ungeraden Schrittzahl!), zu größeren Zahlen, nämlich zu den ersten $n - j$ freien Zahlen in der A-Liste.

Induktiv über n zeigt man, dass $\boxed{\text{n}}_1$ und $\boxed{\text{n}}_2$ nur endlich oft bewegt werden: $\boxed{0}_1$ wird niemals bewegt, $\boxed{0}_2$ wird höchstens einmal bewegt, $\boxed{1}_1$

wird höchstens einmal bewegt für jede Stellung von $\boxed{0}_2$, also höchstens zweimal. $\boxed{1}_2$ kann höchstens einmal bei jeder Stellung von $\boxed{0}_1$ und von $\boxed{1}_1$ bewegt werden. Durch Addieren $\sum_{i<k} 2^i = 2^k + 1 - 1$ und nach oben Abschätzen erhält man $\boxed{\text{j}}_2$ wird höchstens $2^{2j} - 1$ Mal bewegt.

Sei $f(x)$ die Schlussstellung von $\boxed{\text{x}}_1$ und sei $g(x)$ die Schlussstellung von $\boxed{\text{x}}_2$. Wir nehmen an, dass $f(x) \in A$. Dann erhält $f(x)$ ein $+$ zu einer Schrittzahl $2n + 1$. Kein Minus, das genau nach Schritt $2n + 1$ auf der B-Liste steht, wird jemals in ein Plus verwandelt in einem Schritt $> 2n + 1$, denn die Marken $\boxed{\text{k}}_2$, die auf der B-Liste vor dem Aenderungspunkt $-$ stehen, hätten $k < x$, da die Marken $\boxed{\text{k'}}_2$, $n > k' \geq x$ ja nach der letzten Anweisung für den Schritt $2n + 1$ ins freie Feld rechts der schützenswerten Minusse geschoben werden, also weiter rechts als Stellung n auf der B-Liste. Wenn aber in einem Schritt nach $2n + 1$ ein $k < x$ mit Marke $\boxed{\text{k}}_2$ vakant wäre und ein Minus zu einem Plus ändern würde, dann würde $\boxed{\text{x}}_1$ wieder zu einer größeren Zahl bewegt und neu bearbeitet, und die Marke $\boxed{\text{x}}_1$ hätte also noch nicht in der Schlussstellung erreicht. Also ist $f(x) \in W_x^B$.

Umgekehrt, wenn $f(x) \in W_x^B$, dann gibt es ein m, so dass $f(x) \in (W_x^{B_{2n}})_n$ für alle $n \geq m$, und, da $f(x)$ die Schlussstellung von $\boxed{\text{x}}_1$ ist, muß $f(x)$ ein $+$ erhalten, und daher ist $f(x) \in A$.

Genauso zeigt man $g(x) \in B \leftrightarrow g(x) \in W_x^A$. Also $A \not\leq_T B$ und $B \not\leq_T A$. □

Nun werdem wir auch Hartley Rogers Jr. untreu und folgen von nun an dem technischsten der mir bekannten Rekursionstheoriebücher: Robert I. Soare, Recursively Enumerable Sets and Degrees, Springer 1989. Im Lichte von Soares Darstellung ist die Beschreibung des Beweises mit den Marken, Plus- und Minuszeichen zu "anthropomorphic". Wir steigen ein auf Seite 122, Theorem 3.1.

DEFINITION 4.18. A heißt *einfach* (simple), wenn A r.e. ist und $\bar{A}$ unendlich ist und für alle e gilt: $W_e \cap A \neq \emptyset$, falls W_e unendlich ist.

ÜBUNG 4.19. Wenn A einfach ist, dann ist A nicht rekursiv.

SATZ 4.20 (Vorstufe zu Sacks' Splitting Satz). *Sei C r.e. und nicht rekursiv. Dann gibt es eine einfache Menge A, so dass $C \not\leq_T A$.*

Beweis: Für jedes $e \in \omega$ sind folgende Forderungen zu erfüllen:

$$N_e \ : \ C \neq \varphi_e^A$$

$$P_e \ : \ W_e \text{ unendlich } \rightarrow W_e \cap A \neq \emptyset.$$

N_e steht für negativ, P_e steht für positiv.

Sei $\langle C_s \ : \ s < \omega \rangle$ eine rekursive Aufzählung von C. Die Konstruktion von A:

DEFINITION 4.21. $u(A,e,x,s) = 1+$die größte Zahl, die in der Berechnung vorkommt, wenn $\varphi_{e,s}^A(x)\downarrow$ und $=0$ sonst.

$$u(A,e,x) = \begin{cases} \min\{u(A,e,x,s) \,:\, \varphi_{e,s}^A(x)\downarrow\}, & \text{falls } \varphi_e^A(x)\downarrow, \\ \text{undefiniert}, & \text{falls } \varphi_e^A(x)\uparrow. \end{cases}$$

u heißt die use function.

Schritt $s=0$. Sei $A_0=\emptyset$.
Schritt $s+1$: Sei A_s schon definiert. Wir definieren für jedes e

$$\text{length } \ell(e,s) = \max\{x \,:\, (\forall y<x)\varphi_{e,s}^{A_s}(y)\downarrow = C_s(y)\};$$

$$\text{restraint } r(e,s) = \max\{u(A_s,e,x,s) \,:\, x\le \ell(e,s)\}.$$

Wir sagen, dass P_i im Stadium $s+1$ bearbeitet werden muß, wenn $i\le s$ und $W_{i,s}\cap A_s=\emptyset$ und

$$(\exists x)(x\in W_{i,s}\wedge x>2i\wedge(\forall e\le i)r(e,s)<x).$$

Wir nehmen in A_{s+1} für jedes i ein kleinstes x auf, so dass x bezeugt, dass P_i bearbeitet werden muß im Zustand $s+1$. Wir sagen, dass P_i im Stadium $s+1$ bearbeitet wird.

BEMERKUNG 4.22. Sei e fest. Wir nehmen an, dass kein P_j für $j<e$ im Stadium $s+1$ bearbeitet wird. Dann gelten:

(1) Für $y<\ell(e,s)$ und $z\in A_{s+1}\setminus A_s$ haben wir $z>u(A_s,e,y,s)$ und
$$\varphi_{e,s+1}^{A_{s+1}}(y)\downarrow = \varphi_{e,s}^{A_{s+1}}(y)\downarrow = \varphi_{e,s}^{A_s}(y) = C_s(y)$$
und $u(A_{s+1},e,y,s+1)=u(A_s,e,y,s)$.

(2) Für $y=\ell(e,s)$ und $z\in A_{s+1}\setminus A_s$ ist $z>u(A_s,e,y,s)$, da in der Definition von $r(e,s)$ $x\le\ell(e,s)$ steht, und daher folgt aus $\varphi_{e,s}^{A_s}(x)\downarrow$, dass
$$\varphi_{e,s}^{A_{s+1}}(y)\downarrow = \varphi_{e,s}^{A_{s+1}}(y)\downarrow = \varphi_{e,s}^{A_s}(y)\neq C_s(y)$$
und $u(A_{s+1},e,y,s+1)=u(A_s,e,y,s)$.

Die negative Forderung N_e kann im Stadium $s+1$ (höchstens) verletzt werden durch ein Element x, wenn $x\le r(e,s)$ und $x\in A_{s+1}\setminus A_s$. Diese Elemente, für alle s zusammen genommen, bilden eine r.e. Menge

$$I_e = \{x \,:\, (\exists s)(x\in A_{s+1}\setminus A_s\wedge x\le r(e,s))\}.$$

$|I_e|\le e$, da jedes N_e höchstens einmal durch jedes P_i, $i<e$, verletzt wird, woraufhin P_i für immer erfüllt wird.

LEMMA 4.23. $(\forall e)(C\neq\varphi_e^A)$.

Beweis: Wir nehmen an, dass $C=\varphi_e^A$. Dann ist $\lim_s \ell(e,s)=\infty$. Wir nehmen ein s', so dass N_e in keinem Stadium nach s' mehr verletzt wird. Dann werden wir, im Gegensatz zu unsrer Voraussetzung, C berechnen:

Um $C(p)$ zu berechnen, nehmen wir das kleinste $s > s'$, so dass $\ell(e,s) > p$. Durch Induktion über $t > s$ folgt

$$(*)(\forall t \geq s)(\ell(e,t) > p \wedge r(e,t) \geq \max\{u(A_s, e, x, s) \,:\, x \leq p\}),$$

und daher is $\varphi_{e,s}^{A_s}(p) = \varphi_e^{A_s}(p) = \varphi_e^A(p) = C(p)$, und daher ist C rekursiv.

Nun beweisen wir $(*)$. Wir nehmen an, dass es für t gilt. Nach der Bemerkung sichern $r(e,t)$ und $s > s'$, dass $A_{t+1} \upharpoonright z = A_t \upharpoonright z$ für alle z, die in $\varphi_{e,t}^{A_t}(x) = y$ verwendet werden, für alle $x \leq p$. Daher ist $\varphi_{e,t+1}^{A_{t+1}}(x) = y$ und daher $\ell(e,t+1) > p$, außer wenn $C_{t+1}(x) \neq C_t(x)$ für ein $x < \ell(e,t)$. Aber wenn $C_t(x) \neq C_s(x)$ für ein $t \geq s$ und minimales $x \leq p$, dann sichert die Definition von $r(e,t)$ und "$\leq \ell(e,t)$" darin, dass die Ungleichung $C_t(x) \neq \varphi_{e,t}^{A_t}(x)$ für immer erhalten bleibt, im Gegensatz zu unsere Annahme $C = \varphi_e^A$. Widerspruch.

Sacks' Strategie wird manchmal beschrieben als eine, die Übereinstimmungen erhält. Aber es ist auch entscheidend, dass mindestes eine Nichtübereinstimmung erhalten wird, wenn dies möglich ist. Die Nichtübereinstimmung wird erhalten, da C r.e. ist und $C_s(x)$ nur von 0 zu 1 geändert werden kann und nicht umgekehrt. □

LEMMA 4.24. *Für jedes e gibt es ein t so dass für alle $s \geq t$ $r(e,s) = r(e,t)$. (Wir schreiben $r(e,t) = \lim_s r(e,s)$ und überlegen Sie sich, dass schließliche Konstanz die einzige Möglichkeit für die Existenz eines endlichen Grenzwertes in ω mit der diskreten Topologie ist.)*

Beweis: Nach dem vorigen Lemma gibt es ein p, so dass $p = \mu x(C(x) \neq \varphi_e^A(x))$. Wir nehmen ein genügend großes s', so dass für alle $s \geq s'$ folgendes gilt:

(1) $(\forall x \leq p)(\varphi_{e,s}^{A_s}(x) \downarrow = \varphi_e^A(x))$,

(2) $(\forall x \leq p)(C_s(x) = C(x))$ und

(3) N_e wird nicht verletzt im Stadium s.

Erster Fall: $(\forall s \geq s')\varphi_{e,s}^{A_s}(p) \uparrow$. Dann $r(e,s) = r(e,s')$ für alle $s \geq s'$.

Zweiter Fall: $\varphi_{e,t}^{A_t}(p) \downarrow$ zu einem Stadium $t \geq s$. Dann ist $\varphi_{e,s}^{A_s}(p) = \varphi_{e,t}^{A_t}(p)$ für alle $s \geq t$, weil $\ell(e,s) \geq p$. Nach der Definition von $r(e,s)$ bleibt die Berechnung $\varphi_{e,t}^{A_t}(p)$ immer erhalten. Daher ist $\varphi_e^A(p) = \varphi_{e,s}^{A_s}(p)$. Aber $C(p) \neq \varphi_e^A(p)$.

Daher ist $(\forall s \geq t)(C_s(p) \neq \varphi_{e,s}^{A_s}(p) \wedge \ell(e,s) = p \wedge r(e,s) = r(e,t))$ und $r(e,t) = \lim_s r(e,s)$. □

LEMMA 4.25. $(\forall e)(W_e$ *unendlich* $\rightarrow W_e \cap A \neq \emptyset)$.

Beweis: Nach dem vorigen Lemma ist $r(i) = \lim_s r(i,s)$ für $i \leq s$ und $R(e) = \max\{r(i) \,:\, i \leq e\}$. Dann impliziert

$$(\exists x)(x \in W_e \wedge x > 2e \wedge x > R(e)),$$

dass $W_e \cap A \neq \emptyset$. Die Bedingung $x > 2e$ impliziert, dass $\bar{A}$ unendlich ist, und daher ist A einfach. □

DEFINITION 4.26. Ein Grad $\mathbf{a} \leq \mathbf{o}'$ heißt *niedrig* oder low, wenn $\mathbf{a}' = \mathbf{o}'$ (das heißt, dass der Sprung von $\mathbf{a}$ den niedrigsten möglichen Grad hat) und *hoch* oder high, wenn $\mathbf{a}' = \mathbf{o}''$ (also $\mathbf{a}'$ den höchsten möglichen Grad annimmt.) Eine Menge A ist niedrig/hoch, wenn ihr Grad $\deg(A)$ niedrig/hoch ist.

SATZ 4.27 (Sacks' Splitting Theorem, Sacks 1963). *Seien B und C r.e. Mengen und sei C nicht rekursiv. Dann gibt es low r.e. Mengen A_0 und A_1, so dass*

(1) $A_0 \cup A_1 = B$ und $A_0 \cap A_1 = \emptyset$.

(2) $C \not\leq_T A_i$ für $i = 0, 1$.

Beweis: Wir nehmen an, dass B unendlich ist, denn sonst ist das Resultat trivial. Diese Annahme beeinträchtigt jedoch nicht die uniforme Berechenbarkeit von A_0, A_1 von B und C aus. Seien $(B_s)_{s\in\omega}$ und $(C_s)_{s\in\omega}$ rekursive Aufzählungen von B und C, so dass $B_0 = \emptyset$ und $|B_{s+1} \setminus B_s| = 1$ für alle s. Wir werden rekursive Aufzählungen $(A_{i,s})_{s\in\omega}$, $i = 0, 1$, für A_i geben, die die einzige positive Forderung

$$x \in B_{s+1} \setminus B_s \to (x \in A_{0,s+1} \vee x \in A_{1,s+1})$$

in die negativen Forderungen für $i = 0, 1$ und alle e

$$N_{e,i} : \quad C \neq \varphi_e^{A_i}$$

erfüllen. Die Niedrigheit von A_i wird automatisch aus den negativen Forderungen folgen.

Stadium $s = 0$: Wir setzen $A_{i,0} = \emptyset$, $i = 0, 1$.

Stadium $s+1$: Für $A_{i,s}$ definieren wir die rekursiven Funktionen $\ell^i(e,s)$ und $r^i(e,s)$ wie im Beweis des vorigen Satzes. Sei $x \in B_{s+1} \setminus B_s$. Wir nehmen das kleinste $\langle e', i'\rangle$, so dass $x \leq r^i(e,s)$ und nehmen dann dieses in $A_{1-i',s+1}$ auf. Wenn es kein solche $\langle e', i'\rangle$ gibt, dann nehmen x in A_0 auf.

Um zu sehen, dass die Konstruktion gelingt, definieren wir die Menge der womöglich verletzten Bedingungen I_e^i wir im Beweis des vorigen Satzes. Durch Induktion über $\langle e, i\rangle$ zeigt man für alle e, i

(1) I_e^i ist endlich,

(2) $C \neq \varphi_e^{A_i}$, und

(3) $r^i(e) = \lim_s r^i(e,s)$ existiert und ist endlich.

Mögen (1), (2) und (3) schon für alle $\langle k,j\rangle < \langle e,i\rangle$ gelten. Nach (3) nehmen wir ein t, so dass $r(k,s) = r(k)$ für alle $\langle k,j\rangle < \langle e,i\rangle$ für alle $s \geq t$. Dann nehmen wir $r \geq r^j(k)$ für alle solchen k. Wir nehmen ein $v > t$, so dass $B \upharpoonright r = B_v \upharpoonright r$. Dann ist $N_{\langle e,i\rangle}$ nach Stadium v nie mehr verletzt und daher gilt (1) für $\langle e,i\rangle$. Dann folgen wie in den ersten beiden Lemmata im Beweis des vorigen Satzes auch (2) und (3).

Um zu sehen, dass alle A_i niedrig sind, definieren wir eine rekursive Funktion g wie folgt.

$$\varphi^X_{g(e)}(y) = \begin{cases} C(0), & \text{wenn } y = 0 \text{ und } \varphi^X_e(e)\downarrow, \\ \text{undefiniert}, & \text{sonst.} \end{cases}$$

Nun haben wir für $i = 0, 1$

$$e \in A'_i \leftrightarrow \varphi^{A_i}_e(e)\downarrow \leftrightarrow \varphi^{A_i}_{g(e)}(0) = C(0) \leftrightarrow \lim_s \ell^i(g(e),s) > 0,$$

also ist A'_i von der syntaktischen Komplexität Δ^0_2, da ℓ^i und f rekursive Funktionen sind und da $\lim_s(g(e),s)$ existiert. Daher ist nach einem Satz von Post $A'_i \leq_T \mathbf{o}'$. □

Übung 4.28. Beachten Sie, dass die Verletzungsmengen I^i_e zwar endlich sind, es jedoch keine rekursive Funktionen $f(e,i)$ als Obergrenze ihrer Mächtigkeit $|I^i_e| \leq f(e,i)$ gibt. Beweisen Sie dies!

Definition 4.29. Der Grad von $A' = \{\langle x,e\rangle : \varphi^A_e(x)\downarrow\}$ (welcher mit dem Grad von $\{x : \varphi^A_x(x)\downarrow\}$ übereinstimmt) der Jump von $A/\equiv_T$.

Satz 4.30 (Das Jump Theorem, siehe z.B [**10**, Theorem III 2.3]). *(1) A' ist r.e. in A.*

(2) $A' \not\leq_T A$.

(3) B ist r.e. in A gdw $B \leq_1 A'$.

(4) $B \leq_T A$ gdw $B' \leq_1 A'$.

(5) A ist r.e. in B gdw A ist r.e. in $\bar{B}$.

Proof. (1) und (2) sind die relativierte Form der für $A = \emptyset$ bekannten Sätze. (3) $B = W^A_e$ heißt $x \in B \leftrightarrow \langle e,x\rangle \in K^A_0$, also $B \leq_1 A'$. (4) Vorwärts: $B \leq_T A$ impliziert: B' ist r.e. in A. Rückwärts: B' ist r.e. in A. B, $\bar{B} \leq_1 B'$ Das Komplement eines Orakels hat die gleiche Berechnungskraft wie das Orakel selbst. □

Definition 4.31. Die Arithmetische Hierarchie. $A \subseteq \mathbb{N}$ heißt $\Sigma_0 = \Pi_0 = \Delta_0$, gdw A rekursiv ist. (Δ_1 über einen primitiv rekursiven Kern.) Dann immer mit unbeschränktem Quantor über die natürlichen Zahlen aufsteigen.

Satz 4.32. *"Post's Theorem" (Siehe z.B* [**10**, Theorem IV 2.2]*)*

(1) B ist Σ_{n+1} gdw B ist r.e. in einer Π_n-Menge.

(2) $\mathbf{o}^{(n)}$ *ist* Σ_n*-vollständig für alle* $n > 0$.

(3) B *ist* Σ_{n+1} *gdw* B *ist r.e. in* $\mathbf{o}^{(n)}$.

(4) B *ist* Δ_{n+1} *gdw* $B \leq_T \mathbf{o}^{(n)}$.

PROOF. (4) B ist Δ_{n+1} gdw $B, \bar{B}\Sigma_{n+1}$ gdw $B, \bar{B}$ r.e. in $\mathbf{o}^{(n)}$, also $B \leq_T \mathbf{o}^n$. □

Nun widmen wir uns einer Hausaufgabe von Blatt 7. Sei $\mathbf{a} \cup \mathbf{b}$ das Supremum der Grade $\mathbf{a}$, $\mathbf{b}$.

SATZ 4.33. *Es gibt eine r.e. Menge* $\{(n, A_n) : n < \omega\}$ *von r.e. Graden* $\mathbf{a}_n = A_n/\equiv_T$, $n \in \omega$, *so dass für alle* $k \in \omega$ *für alle paarweise verschiedenen* $i_0, \ldots, i_{k+1} \in \omega$ *für alle* $\ell \in \{0, \ldots, k+1\}$ *gilt:*

$$\mathbf{a}_{i_\ell} \not\leq_T \mathbf{a}_{i_0} \cup \cdots \cup \mathbf{a}_{i_{\ell-1}} \cup \mathbf{a}_{i_{\ell+1}} \cup \cdots \cup \mathbf{a}_{i_k}$$

PROOF. Wir folgen der Darstellung auf Seite 164/165 in Coopers Buch. Wir teilen rekursiv ω in unendlich viele unendliche paarweise disjunkte Teilmengen B_n, $n \in \omega$. Dies geht durch eine rekursive Bijektion $b\colon \omega \times \omega \to \omega$. $B_n = \{b(n,i) : i \in \omega\}$. Ansatz: $\mathbf{a}_n = A_n/\equiv_T$, $A_n \subseteq B_n$, $\chi_{A_n} \restriction B_n = \bigcup\{\alpha_{n,s} : s \in \omega\}$, für jedes s, $\alpha_{n,s} \subseteq_{\text{end}} \alpha_{n,s+1}$. Also $\text{domain}(\alpha_{n,s}) \subseteq B_n$. Die Variable s steht wie immer für stage.

Wir setzen für $n \in \omega$,

$$P_n = \{(x, Y) : x \in \{0, \ldots, n\}, Y \subseteq \{0, \ldots, n\} \setminus \{x\}\}.$$

Es gilt $2^{n+1} - 1 \leq |P_n| \leq 2^{n+1} \times (n+1)$.

Schritt 0: Wir setzen $\alpha_{k,0} = \emptyset$ für jedes k.

Wir setzen

$$q_n = \sum_{m=0}^{n-1} |P_n|.$$

Sei $f_n\colon [q_n, q_{n+1}) \to P_n$ eine Surjektion, so dass $(f_n)_n$ uniform berechenbar ist.

In den Schritten $s \in [q_n, q_{n+1})$ erledigen wir die Aufgabe $\mathcal{R}(f_n(s), n)$ für die Konstellation $f_n(s) = (x(s), Y(s)) \in P_n$ für das Turingprogramm n, d.h., wir arrangieren für $z_s = \min(B_{x(s)} \setminus \text{domain}(\alpha_{x(s),s}))$

$$\alpha_{x(s),s+1}(z_s) \neq \varphi_n^{\max\{\alpha_{j,s+1} : j \in Y(s)\}}(z_s).$$

(Das Maximum entspricht gerade der charakteristischen Funktion des Turing-joints)

Dies tun wir wie folgt: Sei $z_s = \text{domain}(\alpha_{x(s),s})$.

1. Fall: Es gibt eine Enderweiterung τ von $\max\{\alpha_{j,s} : j \in Y(s)\}$, [1] so dass $\varphi_n^\tau(z_s) = k \in \{0,1\}$. Suche rekursiv das kleinste (in einer lexikographischen Ordnung von $2^{<\omega}$) solche τ, und setze

$$\begin{array}{rcl} \alpha_{x(s),s+1} & = & \alpha_{x(s),s} \frown \langle z_s, 1-k\rangle \\ \alpha_{j,s+1}, \quad \text{so dass} & & \max\{\alpha_{j,s+1} : j \in Y(s)\} = \tau \wedge \\ & & \bigcup_{j\in Y(s)} \mathrm{dom}(\alpha_{j,s+1}) = \mathrm{dom}(\tau) \end{array}$$

2. Fall Es gibt kein solches τ, Wie setzen

$$\alpha_{j,s+1} \quad = \quad \alpha_{x(s),s} \frown 0 \text{ für alle } j \in Y(s) \cup \{x(s)\}$$

Alle anderen $\alpha_{j,s}$, $j \notin Y(s) \cup \{x(s)\}$, werden im Schritt von s auf $s+1$ nicht verlängert.

Auf Seite 164 unten steht, warum dann im Ende alle Forderungen erfüllt sind.

Nun liest man auf Seite 165 in Cooper nach, dass diese Konstruktion tatsächlich eine r.e. Menge von r.e. Mengen $\{(n, A_n) : n \in \omega\}$ gibt. □

[1] Genauer ist gemeint, dass τ auf der Menge $\bigcup\{B_j : j \in Y(s)\}$ eine Enderweiterung von $\max\{\alpha_{j,s} : j \in Y(s)\}$ ist und auf der Menge $\mathrm{dom}(\tau) \setminus \bigcup\{B_j : j \in Y(s)\}$ konstant 0 ist. Die endliche vielen $\alpha_{k,s}$, $k \in \{0,\ldots,n\} \setminus (Y(s) \cup \{x(s)\})$, die in dem stage $s \in [q_n, q_{n+1})$ auch schon etwas gewachsen sein können, sind ungleich 0 höchstens auf ihrem jeweiligen B_k und tragen nichts bei.

KAPITEL 5

Die infinite injury priority Methode

Nun folgen wir Soare, Recursively Enumerable Sets and Degrees, Springer 1987. Kapitel VIII.

Sei $\langle A_s \ : \ s \in \omega \rangle$ eine rekursive Aufzählung einer r.e. Menge A. Wir definieren

$$a_s = \begin{cases} \mu x(x \in A_s \smallsetminus A_{s\dot{-}1}), & \text{wenn } A_s \smallsetminus A_{s\dot{-}1} \neq \emptyset, \\ \max(A_s \cup \{s\}) & \text{sonst.} \end{cases}$$

$$\hat{\varphi}_{e,s}(A_s; x) = \begin{cases} \varphi_{e,s}^{A_s}(x), & \text{wenn dieses def. ist und } u(A_s; e, x, s) < a_s, \\ \text{undefiniert}, & \text{sonst.} \end{cases}$$

$$\hat{u}(A_s; e, x, s) = \begin{cases} u(A_s; e, x, s) & \text{wenn } \hat{\varphi}_{e,s}(A_s; x) \text{ definiert ist,} \\ 0, & \text{sonst.} \end{cases}$$

$$T = \{s \ : \ A_s \upharpoonright a_s = A \upharpoonright a_s\}$$

ist die Menge der wahren Berechnungsschritte von $\langle A_s \ : \ s < \omega \rangle$. T is unendlich, da $\{s : (\forall t > s)(a_t \geq a_s)\}$ unendlich ist, da wiederum $\{a_s : s \in \omega\}$ unendlich ist.

T ist unendlich und $T \equiv_T A$ uniform in der Aufzählung von A. Es gilt

$$\begin{aligned} &(\forall t \in T)(\hat{\varphi}_{e,t}(A_t; x) = y \rightarrow \\ &(\forall s \geq t)(\hat{\varphi}_{e,s}(A_s; x) = \varphi_e(A; x) = y \wedge \hat{u}(A_s; e, x, s) = u(A_t : e, x, t))). \end{aligned} \tag{5.1}$$

Die einfachste Anwendung dieser Methode ist das Dicke-Teilmengen Lemma (thickness lemma).

Hierzu benötigen wir einige Definitionen:

DEFINITION 5.1. Sei $A \subseteq \omega$. Dann definieren wir

$$A^{[y]} = \{\langle x, y \rangle \ : \ \exists z(\langle x, z \rangle \in A \wedge z = y)\}$$

und $A^{[<y]} = \bigcup_{z<y} A^{[z]}$. $A \subseteq B$ heißt eine *dicke Teilmenge* von B wenn $A^{[y]} =^* B^{[y]}$ für alle y. B heißt *stückweise rekursiv*, wenn für alle y, $B^{[y]}$ rekursiv ist.

Das folgende Lemma wurde 1961 von Shoenfield im Fall $C = \mathbf{o}'$ bewiesen. Für allgemeines C verlangt das Lemma Sacks' Erhaltungsmethode zusammen mit der infinite injury Methode, so wie sie zum ersten Mal von Sacks 1963 entwickelt wurden. Wir folgen der Lachlanschen Darstellung von 1973. Auch a_s, $\hat{\varphi}$, $\hat{u}$ und T wurden von Lachlan eingeführt.

SATZ 5.2. *Thickness Lemma. Sei C r.e. und nicht rekursiv und sei B stückweise rekursiv und r.e. Dann gibt es eine dicke Teilmenge A von B, so dass $C \not\leq_T A$.*

Beweis: Wir halten rekursive Funktionen $(B_s)_{s\in\omega}$ und $(C_s)_{s\in\omega}$, die B und C aufzählen, fest. Sei $A_0 = \emptyset$. Zu gegebenem $(A_t)_{t\leq s}$ definieren wir $\hat{\varphi}_{e,s}^{A_s}$ we oben. Wir definieren die anderen Funktionen wie in der Vorstufe zu Sacks' Splitting Satz (Satz 4.20), aber mit $\hat{\varphi}_{e,s}$ anstelle von $\varphi_{e,s}$:

$$\text{length } \hat{\ell}(e,s) = \max\{x \,:\, (\forall y < x)\hat{\varphi}_{e,s}(A_s; y) \downarrow = C_s(y)\};$$

$$\text{restraint } \hat{r}(e,s) = \max\{\hat{u}(A_s, e, x, s) \,:\, x \leq \hat{\ell}(e,s)\}.$$

$$\text{injury set } \hat{I}_{e,s} = \{x \,:\, (\exists v \leq s)(x \in A_{v+1} \smallsetminus A_v \wedge x \leq \hat{r}(e,v))\}.$$

$$\hat{I}_e = \bigcup_s \hat{I}_{e,s}.$$

Nun lesen wir dies alles für $(A_t)_{t\leq s}$.

Um die Forderungen

$$P_e : \quad B^{[e]} =^* A^{[e]} \text{ und } N_e : \quad C \neq \varphi_e^A, \quad e \in \omega,$$

zu erfüllen, nehmen wir x in $A_{s+1}^{[e]}$ auf genau dann, wenn $x \in B_{s+1}^{[e]}$ und $x > \hat{r}(i,s)$ für alle $i \leq e$. Dann erhalten wir $\hat{I}_e \leq_T A^{[<e]}$, da

$$x \in A_s^{[<e]} \to (x \in \hat{I}_e \leftrightarrow x \in \hat{I}_{e,s}).$$

Um $x \in \hat{I}_e$ zu prüfen, prüft man erst $x \in A^{[<e]}$. Da $B^{[<e]} =^* A^{[<e]}$ und ersteres rekursiv ist, geht dies. Wenn nein raus kommt, ist $x \notin \hat{I}_e$. Wenn ja rauskommt, rechnet man $x \in \hat{I}_{e,s}$? aus und entscheidet dann.

Wir halten e fest und nehmen die Induktionsvoraussetzung, dass $C \neq \varphi_e^A$ und $A^{[i]} =^* B^{[i]}$ für alle $i < e$ an. Dann ist $A^{[<e]} =^* B^{[<e]}$ rekursiv und daher ist $\hat{I}_e$ rekursiv. Die folgenden Lemmata sind ähnlich zu denen im Beweis des Satzes 4.20, doch werden nur die wahren Stadien betrachtet.

In den kommenden beiden Lemmata geht es wiederum um die Funktionen $\hat{l}$ usf, die von A_s, $s \in \omega$, aus definiert sind.

LEMMA 5.3. $C \neq \varphi_e^A$.

Beweis: Wir nehmen indirekt an, dass $C = \varphi_e^A$. Dann ist $\lim_s \hat{\ell}(e,s) = \infty$. Wir zeigen nun $C \leq_T \hat{I}_e$, und dies widerspricht der Nichtrekursivität von C, da die $\hat{I}_e$ rekursiv sind. Um $C(p)$ mit dem Orakel $\hat{I}_e$ zu berechnen, finden wir ein s, so dass $\hat{\ell}(e,s) > p$ und

$$(\forall x \leq p)(\forall z)(z \leq u(A_s, e, x, s) \to (z \notin \hat{I}_e \vee z \in A_s))$$

Solch ein s existiert, da $C = \varphi_e^A$. Wie in den Bemerkungen vor Lemma 1 im Beweis zu 4.20 folgt nun induktiv für $t \geq s$

$$(\forall t \geq s)(\hat{\ell}(e,t) > p \wedge \hat{r}(e,t) \geq \max\{\hat{u}(A_s, e, x, s) \,:\, x \leq p\}) \tag{5.2}$$

und daher is $\varphi_{e,s}^{A_s}(p) = \varphi_e^{A_s}(p) = \varphi_e^A(p) = C(p)$ und daher ist C also rekursiv. □

LEMMA 5.4. *Sei T die Menge der wahren Stadien in der Aufzählung $(A_s)_{s\in\omega}$ von A. Dann ist* $\lim_{t\in T} \hat{r}(e,t) < \infty$.

Daher gilt: Falls $C \neq \Phi_i^A$ für alle $i \leq e$, dann ist $\lim_{t\in T} \hat{R}(e,t) < \infty$, *für* $\hat{R}(e,s) = \max\{\hat{r}(i,s) \,:\, i \leq e\}$.

PROOF. Nach dem ersten Lemma wissen wir $C \neq \Phi_e^A$. Wir definieren

$$p = \mu x[C(x) \neq \Phi_e(A;x).$$

Wir nehmen $s' \in T$ so groß, dass für alle $s \geq s'$

$$(\forall x < p)(\hat{\Phi}_{e,s}(A_s;x) = \Phi_e(A;x) \wedge (\forall x \leq p)(C_s(x) = C(x))).$$

1. Fall: $(\forall t \geq s')(t \in T \to \hat{\Phi}_{e,t}(A_t;p)\uparrow)$. Dann ist für $t \in T$, so dass $t \geq s$:

$$\hat{l}(e,t) = p \wedge \hat{r}(e,t) = \max\{\hat{u}(A_{s'},e,x,s') \,:\, x < p\}.$$

2. Fall: $(\forall t \geq s)[(t \in T) \wedge \Phi_{e,t}(A_t,P)\downarrow]$. Dann ist $\Phi_e(A;p) = \hat{\Phi}_{e,s}(A_s;p)$ für alle $s \geq t$ nach Gleichung (5.1). Aber $C(p) \neq \Phi_e(A;p)$. Daher

$$(\forall s \geq t)(\hat{l}(e,t) = p \wedge \hat{r}(e,s) = \hat{r}(e,t) = \max\{\hat{u}(A_s,e,x,s) \,:\, x \leq p\}).$$

□

LEMMA 5.5. $(\forall e)A^{[e]} =^* B^{[e]}$.

Beweis: Jedes $x \in B^{[e]}$, so dass $x > \hat{R}(e) = \lim_{t\in T} \hat{R}(e,t)$, wird in $A^{[e]}$ aufgenommen. □

Nützliche Erläuterungen zur infinite injury priority Methode finden sich auf in Soares Artikel [**11**].

Literaturverzeichnis

[1] S. Barry Cooper. *Computability theory*. Chapman & Hall/CRC, Boca Raton, FL, 2004.

[2] Thomas Jech. *Set Theory*. Addison Wesley, 1978.

[3] Hartley Rogers Jr. *Theorie of Recursive Functions and Effective Computability*. McCraw-Hill, 1967.

[4] Stephen Cole Kleene. *Introduction to Metamathematics*. North Holland, 1952.

[5] Yuri Matyasevich. Diophantine sets. *Usp. Mat. Nauk*, 27:185–222, 1972.

[6] Piergiorgio Oddifreddi. *Classical Recursion Theory*. Elsevier, 1989.

[7] Rosza Péter. *Rekursive Funktionen*. Akadémiai Kiado, 1951.

[8] H. E. Rose. *Subrecursion: Functions and hierarchies*. Oxford Logic Guides, 1984.

[9] Saharon Shelah. Primitive recursive bounds for van der Waerden numbers. *Journal of the American Mathematical Society*, 1:683–697, 1988.

[10] Robert Soare. *Recursively Enumerable Sets and Degrees*. Springer, 1986.

[11] Robert I. Soare. The infinite injury priority method. *J. Symbolic Logic*, 41(2):513–530, 1976.

[12] Martin Ziegler. *Mengenlehre, Vorlesungsskript*. http://home.mathematik.uni-freiburg.de/ziegler. Universität Freiburg, 1996.