

# Diagonalisierung und Reduktion

Sebastian Wild und Markus E. Nebel

13. März 2012

## Inhaltsverzeichnis

1	Notationen	1
2	Diagonalisierung	3
2.0	Grundidee . . . . .	3
2.1	Beispielaufgabe . . . . .	3
3	Reduktion	4
3.0	Grundidee . . . . .	4
3.1	„Destruktive“ Reduktion [nicht berechenbare Funktionen] . . . . .	5
3.2	„Destruktive“ Reduktion [nicht rekursiv aufzählbare Mengen] . . . . .	6
	Beispielaufgabe . . . . .	7
	Beispielaufgabe . . . . .	9

---

## 1 Notationen

### • Zur Wiederholung

- $\mathcal{P}$ : die Menge der **P**rimitiv-**r**ekursiven Funktionen
- $\mathcal{T}$ : die Menge der rekursiven [aller **b**erechenbaren, **T**otalen Funktionen]
- $\mathcal{R}$ : die Menge der **p**artiiell-**R**ekursiven [aller berechenbaren, aller  $\mu$ -**R**ekursiven] Funktionen
- $\chi_M$  ist die **charakteristische Funktion** der Menge  $M$ ,  $\pi_M$  die **partiell charakteristische Funktion** der Menge  $M$ . Diese sind wie folgt definiert:

$$\chi_M(x) := \begin{cases} 1 & x \in M \\ 0 & x \notin M \end{cases} \quad \pi_M(x) := \begin{cases} 1 & x \in M \\ \text{undefiniert} & x \notin M \end{cases}$$

- DEF  $f$  ist der **Definitionsbereich** der Funktion  $f$
- Schreibe  $f \in \mathcal{K}_n$  genau dann, wenn  $f$  eine  **$n$ -stellige** Funktion aus der Klasse  $\mathcal{K}$  ist.
- Verwende  $\vec{x} = x_1, \dots, x_n$ , falls die Dimension  $n$  aus dem Zusammenhang klar ist.
- Ist  $\mathcal{G}$  eine Gödelisierung der Funktionenklasse  $\mathcal{K}$ , dann schreiben wir  $F_i^{\mathcal{G}}$  für die Funktion mit Gödelnummer (Index)  $i$ .
- $\mathcal{G}$  ist eine **Standard-Gödelisierung** für  $\mathcal{K}$ , wenn  $\mathcal{G}$  die smn-Eigenschaft erfüllt und für jedes  $n \in \mathbb{N}_0$  eine universelle Funktion  $F^{\mathcal{G}_n}$  der  $n$ -stelligen Funktionen aus  $\mathcal{K}$  in  $\mathcal{K}$  liegt. Wir wissen: Standard-Gödelisierungen existieren für  $\mathcal{K} \in \{\mathcal{P}, \mathcal{R}\}$ .
- Auch  $F^{\mathcal{G}}(i, \bullet)$  bezeichnet die **Funktion mit Index  $i$**  bzgl. der Gödelisierung  $\mathcal{G}$ . Dagegen ist  $F^{\mathcal{G}_3}(i, (1, 2, 6))$  der **Rückgabewert** der Funktion mit Index  $i$  angewendet auf die Eingabe  $(1, 2, 6)$ . Man könnte also sagen

$$F^{\mathcal{G}_3}(i, [1, 2, 6]) = \left( \underbrace{F_i^{\mathcal{G}_3}}_{\text{Funktion}} \right) (1, 2, 6)$$

- $\Phi_n$ : die  $n$ -stellige, **nirgends definierte** Funktion. Schreibe  $\Phi$  statt  $\Phi_1$ .
  - $C_n^k$ : die  $n$ -stellige, **konstante** Funktion mit  $\forall \vec{x} C_n^k(\vec{x}) = k$ . Schreibe  $C^k$  statt  $C_0^k$ .
  - $D$ : die Menge  $\{x \in \mathbb{N}_0 \mid F^{\mathcal{G}_1}(x, x) \text{ ist definiert}\} = \mathcal{T}_{\text{Buch}}^{\mathcal{G}}(2)$   
Wir bezeichnen [wie im Buch]  $d := \chi_D$
  - $\bar{D}$ : die Menge  $\{x \in \mathbb{N}_0 \mid F^{\mathcal{G}_1}(x, x) \text{ ist nicht definiert}\} = \mathbb{N}_0 \setminus D = \mathcal{T}_{\text{Buch}}^{\mathcal{G}}(3)$
  - Wenn  $P \subseteq \mathcal{R}$  eine Menge von berechenbaren *Funktionen* ist, so bezeichnet  $\mathfrak{I}P$  die **Indexmenge** der Funktionenmenge  $P$  [bei uns immer bzgl. der Gödelisierung  $\mathcal{G}$ ], d. h.  $\mathfrak{I}P = \{x \in \mathbb{N}_0 \mid F_x^{\mathcal{G}} \in P\}$ .  
Man beachte, dass in einer Indexmenge immer *alle* Indizes der Funktionen vorkommen.
  - Eine Menge  $M$  der Form  $M = \{i \mid Q(F_i^{\mathcal{G}}, i)\}$  – für ein Prädikat  $Q$ , das von der  $i$ -ten Funktion *und* von  $i$  selbst abhängen darf – nenne ich **Menge von Indizes**.
- Beachte:** Jede *Indexmenge* ist auch eine *Menge von Indizes* [mit  $Q(f, i) := f \in P$ ]; die Umkehrung gilt *nicht*: So ist  $D$  eine *Menge von Indizes*, aber keine *Indexmenge*!
- **Beispiele** setze ich in **blauer Schrift**.

Aufgaben, die die *Anwendung* eines **Schemas** zeigen sollen, enthalten in schwarzer Schrift die **allgemeingültigen Teile** [die also zum Schema gehören] und **Aufgaben-Spezifisches** in blau.

(Eingeklammerte Teile – wie dieser – dienen insbesondere dem Verständnis. Entsprechende Hinweise werden üblicherweise in einer vom Studierenden erarbeiteten Lösung nicht erwartet.)

## 2 Diagonalisierung

### 2.0 Grundidee

Diagonalisierung beschreibt ein Beweisverfahren auf Basis von *reductio ad absurdum*: Wir treffen die Annahme, eine gewisse Art von *Auflistung* existiere für eine Menge von Objekten. Dann konstruieren wir mit Hilfe dieser Auflistung ein Objekt, von dem wir zeigen können,

- dass es sich um ein Objekt aus der betrachteten Menge handelt, und
- dass es in der Auflistung *nicht* vorkommt, weil es sich von jedem Objekt in der Aufzählung an mindestens einer Stelle unterscheidet.

Damit haben wir einen  $\nexists$  *Widerspruch*  $\nexists$  hergeleitet, denn nach Annahme ist unsere Auflistung *vollständig*.

Also muss unsere Annahme falsch gewesen sein und eine Auflistung der Menge der Objekte existiert nicht.

### 2.1 Beispielaufgabe

Behauptung:  $M_1 := \{x \in \mathbb{N}_0 \mid F^{\mathcal{G}_1}(x, x) \text{ ist keine Primzahl}\}$  ist nicht rekursiv. [Und zwar für jede Gödelisierung für  $\mathcal{R}$ !]

Beweis:

(Dass unser Beweis für jede Gödelisierung funktioniert, liegt daran, dass wir nur die Eigenschaften einer beliebigen Gödelisierung verwenden:

**Annahme:**  $M_1$  sei rekursiv.

Definiere

$$f(x) := \begin{cases} 3 & \chi_{M_1}(x) = 1 \\ 4 & \chi_{M_1}(x) = 0 \end{cases}$$

$M_1$  rekursiv  $\leadsto \chi_{M_1} \in \mathcal{T} \leadsto f \in \mathcal{T}$  weil Einsetzen und Fallunterscheidung berechenbar. Da  $F^{\mathcal{G}_1}$  alle einstelligen berechenbaren Funktionen aufzählt, gibt es einen Index  $k$  mit

$$\forall x \ f(x) = F^{\mathcal{G}_1}(k, [x]) \quad (1)$$

Nun gilt aber

$$\boxed{F^{\mathcal{G}_1}(k, [k]) \text{ nicht prim}} \iff k \in M_1 \iff \chi_{M_1}(k) = 1 \iff f(k) = 3 \iff f(k) \text{ prim} \xrightarrow{(1)} \boxed{F^{\mathcal{G}_1}(k, [k]) \text{ prim}}$$

$\nexists$  *Widerspruch*  $\nexists$

Also ist die Annahme falsch und  $M_1$  nicht rekursiv. □

## 3 Reduktion

### 3.0 Grundidee

Die Idee der Reduktion ist, die *Lösbarkeit* eines ‚neuen‘ Problems zu untersuchen, indem man das Problem so umformt, dass sich für ein geeignetes ‚altes‘ Problem folgende Implikation ergibt:

$$\text{altes Problem lösbar} \leadsto \text{neues Problem lösbar}$$

Das hilft uns natürlich nur, wenn vom *alten* Problem schon bekannt, *dass es lösbar ist*. Denn dann kann man daraus auf die Lösbarkeit des neuen Problems schließen.<sup>1</sup>

**Beispiel 1:** Angenommen, Sie wissen, dass ihr Taschenrechner Probleme der Art „Was ergibt  $x \cdot y$ ?“ lösen kann. Nun wollt sie beweisen, dass er auch Probleme der Art „Was ist das Quadrat von  $x$ ?“ lösen kann. Das kann man durch **Reduktion** zeigen:

Wir müssen also das neue Problem **umformulieren**. Dazu betrachten wir eine beliebige Eingabe  $x$  für das **neue** Problem „Was ist das Quadrat von  $x$ ?“. Aus dieser Eingabe des neuen Problems bauen wir uns durch eine **Eingabetransformation** eine [gültige] Eingabe des **alten** Problems. In unserem Fall tut's  $x \mapsto (x, x)$ . [Klar warum? Nein?  $\leadsto$  lesen Sie weiter!]

Wir fragen den Taschenrechner also „Was ergibt  $x \cdot x$ ?“ [Für  $y$  das  $x$  einzusetzen war die ganze Umformung...]. Wir wissen, dass das von unserem Taschenrechner korrekt gelöst wird. Da aber  $x^2 = x \cdot x$ , können wir sein Ergebnis einfach unverändert als Ausgabe des **neuen** Problems zurückgeben [d. h. die Rücktransformation ist sogar noch einfacher:  $z \mapsto z$ ].

Wir konnten also das neue Problem auf ein altes **reduzieren**, indem wir eine **Lösungsstrategie** für das neue Problem unter Verwendung des alten angegeben haben.

Damit haben wir durch Reduktion gezeigt:

$$\text{„Was ergibt } x \cdot y\text{?“ lösbar} \leadsto \text{„Was ist das Quadrat von } x\text{?“ lösbar}$$

Und weil das linke Problem lösbar ist, ist demnach auch das rechte lösbar. □

In diesem Beispiel wurde die Reduktion *konstruktiv* eingesetzt, d. h. wir konnten zeigen, dass etwas *lösbar* ist. Deshalb musste das alte Problem [Multiplikation] *schwieriger/allgemeiner* sein, als das neue [Quadrieren].

---

<sup>1</sup>‚Lösbar‘ kann hier relativ großzügig interpretiert werden; es kann ‚berechenbar‘, ‚abzählbar‘, ‚nummerierbar‘, ‚rekursiv aufzählbar‘ oder auch ‚effizient [=in Polynomialzeit] berechenbar‘ bedeuten.

## 3.1 ‚Destruktive‘ Reduktion [nicht berechenbare Funktionen]

Im Buch werden Reduktionen meistens für das Gegenteil verwenden, also um zu zeigen, dass ein schwieriges Problem *nicht* lösbar ist, weil man von einem ‚leichteren‘ Problem schon weiß, dass es *unlösbar* ist.

Hier führt man den Beweis als *Widerspruchsbeweis*: Man nimmt an, das schwierigere Problem *sei* lösbar und zeigt dann durch Reduktion

$$\text{schwieriges Problem lösbar} \leadsto \text{leichtes Problem lösbar}$$

Da man vom ‚leichten‘ Problem schon weiß, dass es *unlösbar* ist, stellt das einen  $\nexists$  *Widerspruch*  $\nexists$  dar. Also muss die Annahme falsch und das schwierige Problem auch unlösbar gewesen sein.

Beispiel 2: Wir wollen zeigen, dass die *allgemeine Haltefunktion*<sup>2</sup>  $H$  mit

$$H(x, z) = \begin{cases} 1 & F^{\mathcal{G}_1}(x, [z]) \text{ definiert} \\ 0 & F^{\mathcal{G}_1}R(x, [z]) \text{ undefiniert} \end{cases} \quad \stackrel{\text{Buch}}{=} \bigvee_t T(x, z, t)$$

nicht rekursiv ist, indem wir sie auf die *spezielle Haltefunktion*  $d$  mit

$$d(x) = \begin{cases} 1 & F^{\mathcal{G}_1}(x, x) \text{ definiert} \\ 0 & F^{\mathcal{G}_1}(x, x) \text{ undefiniert} \end{cases}$$

reduzieren. Dass  $d \notin \mathcal{R}$  ist aus dem Buch bekannt.

Wir nehmen also an, das *neue* Problem sei lösbar:  $H \in \mathcal{T}_2$ . Nun ist aber  $\forall x d(x) = H(x, x)$ , d. h. wir hätten dann eine Möglichkeit gefunden  $d$  zu berechnen! Damit wäre also das alte Problem zu lösen:

$$H \in \mathcal{T}_2 \leadsto d \in \mathcal{T}_1$$

$$\nexists \text{ Widerspruch } \nexists$$

□

In der Tat haben wir mit  $d \notin \mathcal{R}$  schon die „Mutter aller nicht berechenbaren Funktionen“ gefunden: Das spezielle Halteproblem ist – unter den nicht-berechenbaren Funktionen – gewissermaßen das *leichteste* Problem; und damit der ideale Kandidat für Reduktionen.

In obigem Beispiel war ‚lösbar‘ gleichbedeutend mit ‚berechenbar‘. In diesem Fall versucht man also, aus der Annahme, eine ‚neue‘ Funktion  $f$  sei berechenbar, eine *Berechnungsvorschrift* für  $d$  zu finden.

Dieselbe Vorgehensweise kann man verwenden, um nicht rekursive *Mengen* zu betrachten. Man beweist dann einfach, dass  $d$  sich auf die zugehörige charakteristische Funktion reduzieren lässt.

<sup>2</sup>Um das Beispiel nicht unnötig kompliziert zu machen nimmt diese ‚allgemeine‘ Funktion nur einen Funktionsparameter  $z$  neben dem Index. Der Beweis funktioniert aber genauso, wenn statt  $z$  (die Kodierung) eines beliebigen Eingabetupels  $\vec{z}$  übergeben wird.

## 3.2 ‚Destruktive‘ Reduktion [nicht rekursiv aufzählbare Mengen]

Möchte man von einer Menge  $M$  zeigen, dass sie *nicht rekursiv aufzählbar* ist, so kann man versuchen „die Menge  $\bar{D} = \{x \in \mathbb{N}_0 \mid F^{\mathcal{G}_1}(x, x) \text{ nicht definiert}\}$  auf  $M$  zu reduzieren“, d. h. versuchen zu beweisen

$$M \text{ rekursiv aufzählbar} \leadsto \bar{D} \text{ rekursiv aufzählbar}$$

Ähnlich wie  $d$  für die nicht berechenbaren Funktionen, ist  $\bar{D}$  die ‚leichteste‘ aller nicht rekursiv aufzählbaren Mengen.

Das Schema aus dem Beweis zu Satz 4.24 c) im Buch zeigt eine Möglichkeit, wie man diese Reduktion für eine Menge  $M$  führen kann, *wenn folgende Voraussetzungen erfüllt* sind:

- (1)  $M$  ist eine **Menge von Indizes** [für eine Standard-Gödelisierung],
- (2)  $\mathcal{I}\{\Phi\} \subseteq M$ , d. h.  $M$  beinhaltet **alle** Indizes der **nirgends definierten** Funktion  
**und**
- (3)  $\exists w \in \mathcal{R} \mathcal{I}\{w\} \cap M = \emptyset$ , d. h. **keinen Index** von  $w$  kommt in  $M$  vor.

( Diese Voraussetzungen sind im Allgemeinen unnötig stark, denn das Schema ist auch auf Mengen anwendbar, die nicht direkt die Indizes enthalten, sondern nur eine ‚Codierung‘ davon. Allerdings lassen sich solche Mengen immer auch als Mengen von Indizes umschreiben, ohne dass die Komplexität davon beeinflusst wird (siehe Beispielaufgabe ). )

Eine abstrakte Formulierung des Schemas (ohne konkrete Aufgabe) möchten wir hier auslassen; stattdessen versuche wir an Hand der nächsten Aufgabe nochmal vorzuführen wie es funktioniert.

## Beispielaufgabe

Behauptung:  $M_4 := \{x \in \mathbb{N}_0 \mid F^{\mathcal{G}_1}(x, 3x) \text{ ist undefiniert}\}$  ist nicht rekursiv aufzählbar.

Beweis: Die folgende, eingeklammerte Motivation, *warum* das Schema hier anwendbar ist, gehört nicht zum Beweis. Man sollte sich aber trotzdem [im Kopf] vorher überlegen, ob der Beweis mit dem Schema funktionieren kann.

Dazu müssen wir die Voraussetzungen prüfen, die unser Schema hat:

- $M_4$  ist eine Menge von Indizes mit Prädikat  $Q(f, x) := \text{„}f(3x) \text{ ist undefiniert“}$  ✓
- $M_4$  verlangt von seinen Elementen, dass sie Indizes von Funktionen sind, die an einer bestimmten Stelle nicht definiert sind. Da  $\Phi$  nirgends definiert ist, erfüllen alle Indizes für  $\Phi$  das trivialerweise, also ist  $\mathcal{I}\{\Phi\} \subseteq M$  ✓
- Für  $w$  kommen alle Funktionen in Frage, die an allen Stellen  $3i$  definiert sind, für die  $i$  ein Index für  $w$  ist. Insbesondere gilt das für alle totalen Funktionen; wir könnten also wieder die konstante 0-Funktion wählen. [Wie wir es des Öfteren getan haben.]

Möglich ist aber z. B. auch

$$w(x) := \begin{cases} 0 & 3 \mid x \\ \text{undefiniert} & \text{sonst} \end{cases}$$

Dieses  $w$  werden wir verwenden um zu zeigen, wie wenig sich dadurch am Beweis ändert und dass das Schema auch mit nicht-totalem  $w$  funktioniert.

Nachdem geklärt ist, dass unser Schema anwendbar ist, beginnen wir mit dem eigentlichen Beweis.

**Annahme:**  $M_4$  sei rekursiv aufzählbar.

$$\curvearrowright \exists f \in \mathcal{R} \quad \text{DEF } f = M_4. \quad (2)$$

Sei  $w$  definiert durch

$$w(x) := \begin{cases} 0 & 3 \mid x \\ \text{undefiniert} & \text{sonst} \end{cases}$$

$\curvearrowright w$  ist offensichtlich berechenbar [ $w \in \mathcal{R}_1$ ]. Nach der Argumentation aus der 10. Aufgabe zu Kapitel 4 ist dann das wie folgt definierte  $r$  sogar rekursiv [ $r \in \mathcal{T}_1$ ]

$$F_{r(x)}^{\mathcal{G}} = \begin{cases} w & F^{\mathcal{G}_1}(x, [x]) \text{ definiert} \\ \Phi & F^{\mathcal{G}_1}(x, [x]) \text{ undefiniert} \end{cases} \quad (3)$$

Definiere nun

$$g(x) := f(r(x)) \quad \curvearrowright g \in \mathcal{R} \quad (4)$$

Damit gilt folgende Kette von Äquivalenzen:

$$\begin{aligned}
 \boxed{g(x) \text{ def.}} &\stackrel{(4)}{\iff} f(r(x)) \text{ def.} \stackrel{r \in \mathcal{T}}{\iff} r(x) \in \text{DEF } f \stackrel{(2)}{\iff} r(x) \in M_4 \\
 &\iff F^{\mathcal{G}_1}(r(x), \underbrace{[3 \cdot r(x)]}_{\text{3 teilt das!}}) \text{ undefiniert} \stackrel{(3)}{\iff} F_{r(x)}^{\mathcal{G}} = \Phi \\
 &\stackrel{(3)}{\iff} F^{\mathcal{G}_1}(x, x) \text{ undefiniert} \iff \boxed{x \in \bar{D}}
 \end{aligned}$$

Zusammengefasst haben wir also mit  $g \in \mathcal{R}$  eine berechenbare Funktion gefunden mit  $\text{DEF } g = \bar{D}$  [denn  $g(x)$  ist genau dann definiert, wenn  $x \in \bar{D}$ ].  $\curvearrowright \bar{D}$  ist rekursiv aufzählbar.

⚡ Widerspruch ⚡

□

Man erkennt hoffentlich ganz gut, was man für die Schema-Anwendung tun muss:

- Anwendbarkeit prüfen (siehe Kasten auf Seite 6)
- $w$  geeignet definieren, sodass  $w \in \mathcal{R}$  und die
- Äquivalenz  $r(x) \in M \iff F_{r(x)}^{\mathcal{G}} = \Phi$  gilt.

Die Hülle drum herum kann man dann quasi blind hinschreiben. [Sie sollt trotzdem aufpassen und denken ...]

Schaut man sich den Beweis an, ohne *nur* das Schema im Blick zu haben, so erkennt man das typische Reduktions-Muster: Wir haben gezeigt

$$M_4 \text{ rekursiv aufzählbar} \curvearrowright \bar{D} \text{ rekursiv aufzählbar}$$

und schließen aus dem Wissen, dass  $\bar{D}$  eben *nicht* rekursiv aufzählbar ist, auf einen Widerspruch.



## Beispielaufgabe

Diese Aufgabe weicht in ihrer Formulierung etwas von dem ab, was das vorherige Schema lösen kann. [Spätestens hier müssen Sie dann den Denkkapparat wieder anwerfen! Als Training für selbigen können Sie in dieser Aufgabe mal selbst generische Schema-Teile von Beispiel-Spezifischem trennen.]

Behauptung: Die Mengen  $M_{5,i} := \{x \in \mathbb{N}_0 \mid F^{\mathcal{G}_1}(3x+i, x) \text{ ist undefiniert}\}$  für  $i \in \{0, 1, 2\}$  sind *nicht alle* drei rekursiv aufzählbar.

Beweis: Als ersten Schritt schreiben wir die Mengen  $M_{5,i}$  so um, dass wir *Mengen von Indizes* erhalten:

$$\tilde{M}_{5,i} := \{3x+i \mid x \in \mathbb{N}_0 \wedge F^{\mathcal{G}_1}(3x+i, x) \text{ ist undefiniert}\} = \{3x+i \mid x \in M_{5,i}\}, \quad i \in \{0, 1, 2\}.$$

Diese Mengen  $\tilde{M}_{5,i}$  sind Mengen von Indizes mit  $Q(f, 3x+i) = ,f(x) \text{ undefiniert}'$ .

( Natürlich sind die  $M_{5,i}$  und die  $\tilde{M}_{5,i}$  *nicht gleich!* Es sollte allerdings offensichtlich sein, dass man eine Bijektion zwischen  $M_{5,i}$  und  $\tilde{M}_{5,i}$  finden kann, und damit aus jeder Aufzählungsfunktion für die eine Menge eine für die andere konstruieren kann.  $\hookrightarrow$   
 $\forall i \ (M_{5,i} \text{ rek. aufz.} \leftrightarrow \tilde{M}_{5,i} \text{ rek. aufz.})$   
 Wir können uns also für die Aufzählbarkeit auf die  $\tilde{M}_{5,i}$  beschränken. )

( Bevor wir nun zum Beweis der Behauptung kommen; warum kann man unser Schema nicht direkt auf z. B.  $\tilde{M}_{5,0}$  anwenden?  
 Das Problem ist die Voraussetzung (2). Niemand kann uns garantieren, dass die nirgends definierte Funktion nicht z. B. [unter anderem] den Index 1 hat. [Im Allgemeinen hat jede Funktion mehrere Indizes.] Da sich die 1 nicht als  $3x+0$  mit  $x \in \mathbb{N}_0$  darstellen lässt, wird dieser Index in  $\tilde{M}_{5,0}$  nicht betrachtet [ $F^{\mathcal{G}_1}$  wird nie für Index 1 ausgewertet, selbst wenn  $M_{5,0} = \mathbb{N}_0$  gälte!]

Wir betrachten nun die Vereinigung dieser Mengen:

$$\begin{aligned} \tilde{M}_5 &:= \bigcup_{i \in \{0,1,2\}} \tilde{M}_{5,i} = \{3x+i \mid x \in M_{5,i} \wedge i \in \{0,1,2\}\} \\ &= \left\{ x \mid F^{\mathcal{G}_1}\left(x, \left\lfloor \frac{x}{3} \right\rfloor\right) \text{ undefiniert} \right\} \end{aligned}$$

( Auf  $\tilde{M}_5$  lässt sich unser Schema anwenden: Erstens ist  $\tilde{M}_5$  eine Mengen von Indizes. [Jede Vereinigung von Mengen von Indizes ergibt wieder eine Menge von Indizes.]  
 Da sich jede natürliche Zahl [und damit jeder mögliche Index] als  $3x+i$  mit  $x \in \mathbb{N}_0$  und  $i \in \{0, 1, 2\}$  schreiben lässt, umfasst  $\tilde{M}_5$  *alle* Indizes  $x$  von Funktionen, die an der Stelle  $\left\lfloor \frac{x}{3} \right\rfloor$  undefiniert sind. Letzteres trifft insbesondere auf  $\Phi$  zu, womit die erste Voraussetzung erfüllt ist.  
 Außerdem sind in  $\tilde{M}_5$  keine Indizes von totalen Funktionen enthalten. Also ist auch die dritte Voraussetzung erfüllt und unser Schema lässt sich anwenden: )

**Annahme 1:**  $\tilde{M}_5$  sei rekursiv aufzählbar.  $\leadsto \exists f \in \mathcal{R} \text{ DEF } f = \tilde{M}_5$ . Nach bekannter Argumentation folgt, dass  $r$  – definiert durch

$$F_{r(x)}^{\mathcal{G}} = \begin{cases} C^0 & F^{\mathcal{G}_1}(x, x) \text{ definiert} \\ \Phi & F^{\mathcal{G}_1}(x, x) \text{ undefiniert} \end{cases}$$

– rekursiv ist.  $\leadsto g := f \circ r \in \mathcal{R}$ . Damit gilt nun

$$\begin{aligned} g(x) \text{ def.} &\iff f(r(x)) \text{ def.} \iff r(x) \in \text{DEF } f \iff r(x) \in \tilde{M}_5 \\ &\iff F^{\mathcal{G}_1}(r(x), \lfloor \lfloor r(x)/3 \rfloor \rfloor) \text{ undefiniert} \\ &\iff F_{r(x)}^{\mathcal{G}} = \Phi \iff F^{\mathcal{G}_1}(x, x) \text{ undefiniert} \\ &\iff x \in \bar{D} \end{aligned}$$

$\leadsto \text{DEF } g = \bar{D} \not\vdash \text{Widerspruch} \not\vdash$  zu  $\bar{D}$  nicht rek. aufz.

Wir haben jetzt also gezeigt, dass  $\tilde{M}_5$  nicht rekursiv aufzählbar ist.

Wofür wir das brauchen? Um die nächste Annahme widerlegen zu können:

**Annahme 2:**  $\tilde{M}_{5,i}$  sei für alle  $i \in \{0, 1, 2\}$  rek. aufz.  $\leadsto \pi_{\tilde{M}_{5,0}}, \pi_{\tilde{M}_{5,1}}, \pi_{\tilde{M}_{5,2}} \in \mathcal{R}$ . Dann ist aber auch  $v$  mit

$$v(x) := \begin{cases} 1 & \pi_{\tilde{M}_{5,0}}(x), \pi_{\tilde{M}_{5,1}}(x) \text{ oder } \pi_{\tilde{M}_{5,2}}(x) \text{ definiert} \\ \text{undefiniert} & \text{sonst} \end{cases}$$

berechenbar: Wir führen dazu – nach der Idee des ‚Timesharing‘<sup>3</sup> – die drei Berechnungen abwechselnd aus. So verpassen wir es auf keinen Fall, falls eine der drei terminiert. Und wenn keine terminiert, ist das genau das gewünschte Verhalten für  $v$ .

$\leadsto v \in \mathcal{R}$  und  $\text{DEF } v = \tilde{M}_{5,0} \cup \tilde{M}_{5,1} \cup \tilde{M}_{5,2} = \tilde{M}_5 \not\vdash \text{Widerspruch} \not\vdash$  zu  $\tilde{M}_5$  nicht rek. aufz.  $\square$

---

<sup>3</sup>alias ‚wait-and-see‘



<http://www.springer.com/978-3-8348-1889-8>

Formale Grundlagen der Programmierung

Nebel, M.

2012, VIII, 194 S., Softcover

ISBN: 978-3-8348-1889-8