11. Universelle Turingmaschinen und der Normalformsatz von Kleene

In diesem Abschnitt beginnen wir mit der Darstellung einiger fundamentaler Ergebnisse der Berechenbarkeitstheorie. Wir zeigen die Existenz universeller (Turing-) Maschinen und damit (nach Churchscher These) die Existenz universeller partiell-berechenbarer Funktionen. Dieses wichtige Ergebnis ist die theoretische Grundlage für den Bau von Universalrechnern. Darüberhinaus zeigen wir, dass die aus einer universellen Turingmaschine gewonnene effektive Aufzählung der partiell rekursiven Funktionen eine Standardaufzählung ist, d.h. – grob gesprochen – dass man aus der Darstellung einer partiell rekursiven Funktion in einem beliebig gegebenen universellen Berechnungssystem eine Turingmaschine zur Berechnung dieser Funktion effektiv gewinnen kann. Der abschließend gezeigte Normalformsatz von Kleene verbindet diese Ergebnisse mit der im letzten Abschnitt erhaltenen rekursiven Darstellung von Turingmaschinen.

Wir definieren zunächst universelle Funktionen, wobei wir uns auf den Fall zahlentheoretischer Funktionen beschränken.

11.1 DEFINITION. Sei F eine Klasse von partiellen Funktionen über $\mathbb{N}$. Eine partielle Funktion $\varphi^{(n+1)}$ ist *n-universell* für F, wenn

(i) $\varphi \in$ F und

(ii) zu jeder n-stelligen partiellen Funktion $\psi^{(n)} \in$ F gibt es eine Zahl $e \in \mathbb{N}$, sodass $\psi(\vec{x}) = \varphi(e, \vec{x})$ für alle $\vec{x} \in \mathbb{N}^n$.

Gilt zusätzlich, dass es zu jedem $\hat{\varphi}^{(n+1)} \in$ F eine total rekursive Funktion h gibt, so dass

(iii) $\hat{\varphi}(e, \vec{x}) = \varphi(h(e), \vec{x})$ für alle $e \in \mathbb{N}$ und $\vec{x} \in \mathbb{N}^n$

gilt, so heißt φ eine *Standardaufzählung* oder *Gödelnummerierung* der n-stelligen partiellen Funktionen in F.

Eine Zahl e wie in (ii) heißt ein *Index von* ψ *bzgl.* φ und wir schreiben $\varphi_e(\vec{x})$ statt $\varphi(e, \vec{x})$, d.h. $\psi = \varphi_e$. Die rekursive Funktion h in (iii) heißt *Übersetzungsfunktion* von $\hat{\varphi}$ nach φ.

In diesem Abschnitt werden wir, wie einleitend schon gesagt, zeigen, dass die Klasse F(REK) n-universelle Funktionen – ja sogar Standardaufzählungen – für jede Stelligkeit n besitzt. Hierzu weisen wir die Existenz universeller Turingmaschinen nach. Nämlich für gegebenes Alphabet Σ und gegebenes $n \in \mathbb{N}$, zeigen wir, dass jede (geeignet normierte) Turingmaschine M zur Berechnung von n-stelligen Funktionen über Σ^* so durch ein Wort $\lceil M \rceil \in \Sigma^*$ dargestellt werden kann, dass man eine Turingmaschine $U = U_{\Sigma,n}$ angeben kann, die bei Eingabe $(\lceil M \rceil, \vec{x})$ mit $\vec{x} \in (\Sigma^*)^n$ die Rechnung der Maschine M bei Eingabe $\vec{x}$ simuliert. Die Resultatsfunktion von U ist also n-universell für die Klasse der partiell Turing-berechenbaren Funktionen über Σ^*. Für das unäre Alphabet erhalten wir dann das entsprechende Ergebnis für die partiell rekursiven Funktionen (unter Verwendung des Äquivalenzsatzes).

Ist e Index einer partiell rekursiven Funktion ψ bzgl. der derart gewonnenen universellen Funktion $\varphi = \varphi_U$, so ist e gerade die Gödelnummer einer Turingmaschine zur Berechnung von ψ. Statt der Turingmaschinen hätten wir ebenso gut von einem anderen universellen Berechnungssystem wie den Registermaschinen oder den Definitionen der partiell rekursiven Funktionen ausgehen können, um eine universelle Funktion für F(REK) zu definieren. Die Tatsache, dass φ_U eine Gödelnummerierung ist, bedeutet dann anschaulich, dass wir jede Maschine (oder Definition) eines dieser Berechnungssysteme in eine äquivalente Turingmaschine effektiv übersetzen können. Dabei müsste das andere System nicht einmal universell sein (von $\hat{\varphi}$ in (iii) oben verlangen wir nicht, dass es universell ist), könnte also auch z.B. das Berechnungssystem der primitiven Registeroperatoren sein oder eine Auflistung $\hat{\varphi}(e, \vec{x}) = e$ der konstanten Funktionen. Grob gesprochen – und natürlich unter Annahme der Church-Turing-These – können wir also jeden Algorithmus eines gegebenen Systems von Algorithmen in eine äquivalente Turingmaschine effektiv überführen. Für (primitive) Registermaschinen oder -operatoren oder für die Definition (primitiv) rekursiver Funktionen haben wir das bereits gezeigt, da wir die Konzepte wechselseitig effektiv simuliert haben. Hier wird diese Beobachtung nun verallgemeinert und formalisiert, da wir nun durch Betrachtung der Gödelnummern die Transformationen als Abbildungen über den natürlichen Zahlen beschreiben und damit als berechenbar im formalen Sinne nachweisen können.

Wir beginnen nun mit der Konstruktion der universellen Turingmaschine. Hierzu geben wir ein Alphabet $\Sigma \supseteq \Sigma_2 = \{0,1\}$ sowie eine natürliche Zahl $n \geq 1$ vor, um die universelle Maschine $U_{\Sigma,n}$ zur Berechnung der partiellen Funktionen von $(\Sigma^*)^n$ nach Σ^* zu bestimmen. (Von der Annahme, dass Σ das binäre Alphabet umfasst werden wir uns später noch lösen.) Indem wir die Varianten des Turingmaschinenmodells aus den Abschitten 5 und 6 aufgreifen, beobachten wir zunächst, dass wir jede Turingmaschine M zur Berechnung von partiellen Funktionen $\varphi : (\Sigma^*)^n \to \Sigma^*$ durch eine äquivalente im folgenden Sinn normierte Turingmaschine ersetzen können.

11.2 DEFINITION. Eine Turingmaschine $M = (B, P)$ zur Berechnung einer n-stelligen partiellen Funktion $\varphi : (\Sigma^*)^n \to \Sigma^*$ heißt *normiert*, falls

$$B = TB(\Sigma, \Sigma, \Sigma \cup \{b, 0, 1\}, n),$$

wobei $0,1$, aber nicht das Blankzeichen b, in Σ vorkommen dürfen, und falls P ein Programm aus bedingten Anweisungen ist mit Zustandsmenge $Z = \{0, 1, \ldots, p\}$ (für geeignetes $p \geq 1$), wobei 0 der Start- und 1 der einzige Stoppzustand ist.

Aus den Beweisen von Satz 6.4 und Lemma 5.1 folgt unmittelbar, dass es im Folgenden genügt, normierte Turingmaschinen zu betrachten:

11.3 LEMMA. *Zu jeder Turingmaschine M zur Berechnung einer n-stelligen partiellen Funktion über Σ^* gibt es eine äquivalente normierte Turingmaschine M'. Weiter gilt*

$$\mathrm{time}_{M'}(\vec{x}) \in O(\mathrm{time}_M(\vec{x}))$$

und

$$\mathrm{space}_{M'}(\vec{x}) \in O(\mathrm{space}_M(\vec{x})).$$

(Die angegebenen Ressourcen-Schranken sind hier zunächst bedeutungslos. Sie werden erst im Abschnitt über die Komplexitätstheorie interessant.)

Als Nächstes kodieren wir normierte Turingmaschinen $M = (B, P)$ durch Binärwörter. Da die Basismaschine hierbei fest vorgegeben ist, genügt es, das Programm zu gödelisieren.

Sei $\Sigma = \{a_1, \ldots, a_m\}$, wobei $a_1 = 0$ und $a_2 = 1$ seien. Weiter sei $a_0 = b$. Wir ordnen dann jedem Buchstaben a_i aus dem Bandalphabet, jedem Zustand z und jedem Bewegungsbefehl ein nichtleeres Unärwort als Kodierung zu:

$$\lceil a_i \rceil := 1^{i+1} \quad \lceil z \rceil := 1^{z+1} \quad \lceil L \rceil := 1 \quad \lceil S \rceil := 1^2 \quad \lceil R \rceil := 1^3.$$

Hiermit kodieren wir eine Instruktion $I = (z, a_i, a_j, B, z')$ aus P durch

$$\lceil I \rceil = \lceil z \rceil 0 \lceil a_i \rceil 0 \lceil a_j \rceil 0 \lceil B \rceil 0 \lceil z' \rceil$$

und ordnen dem Programm $P = \{I_1, \ldots, I_k\}$ die Kodierung

$$\lceil P \rceil = 1000 \lceil I_1 \rceil 00 \lceil I_2 \rceil 00 \ldots \lceil I_k \rceil 000$$

zu. Offensichtlich ist die Abbildung, die einem Programm P seine Kodierung $\lceil P \rceil$ zuordnet, injektiv und berechenbar und umgekehrt können wir von einem Binärwort effektiv feststellen, ob es ein Programm kodiert und, falls ja, welches.

Wir können nun die universelle Turingmaschine $U = U_{\Sigma,n}$ definieren. Dabei genügt es, wegen Satz 6.3, U als 3-Band-Turingmaschine einzuführen.

11.4 LEMMA. *Sei $\Sigma \supseteq \Sigma_2$ ein Alphabet und $n \geq 1$. Es gibt eine 3-Band-Turingmaschine $U = U_{\Sigma,n}$, die bei Eingabe eines kodierten normierten Programmes $\lceil P \rceil$ zusammen mit einer Wortfolge $\vec{x} \in (\Sigma^*)^n$ die zu P gehörende normierte Maschine $M = (B, P)$ simuliert, d.h.*

$$\varphi_U(\lceil P \rceil, \vec{x}) = \varphi_M(\vec{x}).$$

Weiter gilt, dass

$$\begin{aligned} &\mathrm{time}_U(\lceil P \rceil, \vec{x}) \in O(|\lceil P \rceil| \cdot \max(\mathrm{time}_M(\vec{x}), |x_1|, \ldots, |x_n|)) \\ &\mathrm{space}_U(\lceil P \rceil, \vec{x}) \in O(|\lceil P \rceil| + \mathrm{space}_M(\vec{x})). \end{aligned}$$

Die Zeit- und Platzschranken hier werden wiederum erst später interessant werden.

BEWEIS. Wir beschreiben die Arbeitsweise von U nur informell. Bei Eingabe $(\lceil P \rceil, \vec{x})$ verhält sich die Maschine U auf Band 1 wie die zu simulierende Maschine $M = (B, P)$ bei Eingabe $\vec{x}$. Auf Band 2 wird das kodierte Programm $\lceil P \rceil$ von M abgespeichert und Band 3 enthält die Kodierung $\lceil z \rceil$ des aktuellen M-Zustandes z. Hierzu kopiert U in einer *Initialisierungsphase* $\lceil P \rceil$ von Band 1 auf Band 2 (unter gleichzeitigem Löschen) und schreibt den kodierten Startzustand $\lceil 0 \rceil$ von M auf Band 3. In der anschließenden *Simulationsphase* simuliert U die Maschine M Schritt für Schritt. Um einen Schritt von M zu simulieren „liest“ U den Buchstaben a auf dem Arbeitsfeld von Band 1 und sucht durch Mustervergleich das erste Vorkommen der Folge $00 \lceil z \rceil 0 \lceil a \rceil 0$ auf Band 2, wobei $\lceil z \rceil$ die aktuelle Inschrift des 3. Bandes ist. Dort befindet sich dann die Kodierung der auszuführenden M-Instruktion. U simuliert diese, indem sie den angegebenen Druck- und Bewegungsbefehl auf Band 1 ausführt und die Inschrift von Band 3 durch die angegebene Kodierung des Nachfolgezustandes ersetzt. Findet U das gesuchte Muster nicht, so hat M den Stoppzustand erreicht, so dass U ebenfalls die Simulation beendet. In der abschließenden *Ausgabephase* wird noch die Inschrift auf Band 3 durch die Inschrift auf Band 1 ersetzt, damit man dem Ausgabemechanismus der 3-Band-Maschine, die die Ausgabe dem 3. Band entnimmt, gerecht wird.

Die behaupteten Platz- und Zeitschranken lassen sich leicht nachprüfen. Für die Zeit hat man nur zu beachten, dass die Simulation eines M-Schrittes $O(|\lceil P \rceil|)$ Schritte erfordert. $\square$

11.5 SATZ. *(Existenz universeller Turingmaschinen) Zu jedem Alphabet Σ und zu jeder Stelligkeit $n \geq 1$ gibt es eine Turingmaschine $\tilde{U} = \tilde{U}_{\Sigma,n}$ mit folgender Eigenschaft: Zu jeder partiell Turing-berechenbaren Funktion $\psi : (\Sigma^*)^n \to \Sigma^*$ gibt es ein Wort $w \in \Sigma^*$ mit $\psi = (\varphi_{\tilde{U}})_w$, d.h.,*

$$\forall \vec{x} \in (\Sigma^*)^n (\psi(\vec{x}) = \varphi_{\tilde{U}}(w, \vec{x})).$$

BEWEIS. Gilt $\Sigma \supseteq \Sigma_2$, so wählt man $\tilde{U}$ als eine (1-Band-)Turingmaschine, die zu der 3-Band-TM $U = U_{\Sigma,n}$ aus Lemma 11.4 äquivalent ist. Dass es solch ein $\tilde{U}$ gibt, folgt aus dem Bandreduktionssatz (Satz 6.3). Für ψ wie im Satz gegeben, können wir nach Lemma 11.3 eine normierte TM $M = (B, P)$, die ψ berechnet, wählen. Nach Wahl von $\tilde{U}$ und Lemma 11.4 folgt

$$\psi(\vec{x}) = \varphi_U(\lceil P \rceil, \vec{x}) = \varphi_{\tilde{U}}(\lceil P \rceil, \vec{x}).$$

D.h. $w = \lceil P \rceil \in \Sigma_2 \subseteq \Sigma$ ist der gewünschte Index von ψ bezüglich $\varphi_{\tilde{U}}$.

Gilt $\Sigma \supseteq \Sigma_2$ nicht, dann ist o.B.d.A. Σ das unäre Alphabet $\Sigma_1 = \{1\}$. Hier würde es genügen zu zeigen, dass es eine Turingmaschine zur Konvertierung der Unärdarstellung einer Zahl in deren Binärdarstellung gibt. Einen solchen Übersetzer mit 2 Bändern erhält man z.B. durch Formalisierung des folgenden Verfahrens T: Man markiert auf Band 2 einen Nullblock 0^n der Länge der Eingabe 1^n. Dann streicht man Schritt für Schritt eine 1 der Eingabe und inkrementiert gleichzeitig die Binärzahl auf Band 2 (hierzu ersetzt man die am weitesten rechts stehende 0 durch eine 1 und ersetzt die rechts davon stehenden Nullen durch Einsen). Am Ende streicht man eventuell vorhandene führende Nullen. Die Maschine $\tilde{U}_{\Sigma_1,n}$ würde dann wie folgt arbeiten: Zunächst wird mit Hilfe von T die Eingabe $(\mathrm{Un}(m), \vec{x})$ in die Eingabe $(\mathrm{Bin}(m), \vec{x})$ transformiert und dann $\tilde{U}_{\Sigma_2,n}$ ausgeführt.

Tatsächlich gehen wir hier (für spätere Anwendungen) etwas anders vor. Wir ordnen dem Programm P einer normierten Maschine $M = (B, P)$ zur Berechnung von Funktionen über Σ_1^* (d.h. über $\mathbb{N}$) eine Gödelnummer $gn(P) \in \mathbb{N}$ zu und argumentieren dann, dass es eine TM T gibt, die $gn(P)$ in das Codewort $\lceil P \rceil \in \Sigma_2^*$ übersetzt (ohne die restlichen Eingaben zu verändern). Die Maschine $\tilde{U}$ erhält man dann wiederum durch Hintereinanderausführung von T und $U_{\Sigma_2,n}$. Dabei kann man $gn(P)$ z.B. wie folgt definieren. Den Buchstaben a_i des Bandalphabetes, den Zuständen z und den Bewegungen ordnet man die Gödelnummern

$$gn(a_i) = i \;\; gn(z) = z \;\; gn(L) = 1 \;\; gn(S) = 2 \;\; gn(R) = 3$$

zu, einer Instruktion $I = (z, a, a', B, z')$ die Gödelnummer

$$gn(I) = \langle gn(z), gn(a), gn(a'), gn(B), gn(z') \rangle$$

und definiert schließlich für $P = \{I_1, \ldots, I_k\}$

$$gn(P) = \langle gn(I_1), \ldots, gn(I_k) \rangle.$$

Auf die (recht mühselige) explizite Angabe des Übersetzers von $gn(P)$ in $\lceil P \rceil$ verzichten wir hier. □

Für den Fall von Funktionen über natürlichen Zahlen, deren Turing-Berechenbarkeit ja über die Unärdarstellung definiert wurde, erhalten wir mit Hilfe des Äquivalenzsatzes und der in dessen Beweis gefundenen primitiv rekursiven Beschreibung von Turingmaschinen-Rechnungen, folgenden Satz.

11.6 KOROLLAR. *(Normalformsatz von Kleene) Für jede Zahl $n \geq 1$ gibt es eine primitiv rekursive Funktion $U^{(1)}$ und ein primitiv rekursives Prädikat $T^{(n+2)}$, so dass es zu jeder n-stelligen partiell rekursiven Funktion $\psi^{(n)}$ eine Zahl $e \in \mathbb{N}$ gibt mit*

$$\forall \vec{x} \in \mathbb{N}^n (\psi(\vec{x}) = U(\mu y T(e, \vec{x}, y))).$$

In der Tat ist die durch

$$\varphi(e, \vec{x}) = U(\mu y T(e, \vec{x}, y))$$

definierte partielle Funktion $\varphi^{(n+1)}$ eine Gödelnummerierung der n-stelligen partiell rekursiven Funktionen. Darüberhinaus können die Übersetzungsfunktionen nach φ primitiv rekursiv und injektiv gewählt werden.

BEWEIS. Für gegebenes $n \geq 1$ sei $\tilde{U} = \tilde{U}_{\Sigma_1, n}$ die nach Satz 11.5 existierende universelle Turingmaschine zur Berechnung n-stelliger partieller Funktionen über Σ_1^*, also über $\mathbb{N}$. Mit der im Beweis von Lemma 10.2 angegebenen Gödelisierung von Turingmaschinen angewendet auf $\tilde{U}$ erhalten wir dann ein primitiv rekursives Prädikat $\text{RECHNUNG}_{\tilde{U}}^{(n+3)}$, sodass

$$\varphi_{\tilde{U}}(e, \vec{x}) = (\mu y (\text{RECHNUNG}_{\tilde{U}}(e, \vec{x}, (y)_1, (y)_2)))_2$$

für alle $e \in \mathbb{N}$ und $\vec{x} \in \mathbb{N}^n$ gilt. Setzen wir

$$T(e, \vec{x}, y) \Leftrightarrow \text{RECHNUNG}_{\tilde{U}}(e, \vec{x}, (y)_1, (y)_2)$$

und

$$U(y) = (y)_2,$$

so ergibt sich der erste Teil des Normalformsatzes unmittelbar aus der Universalität von $\tilde{U}$ und dem Äquivalenzsatz.

Zu zeigen bleibt die Existenz injektiver, primitiv rekursiver Übersetzungsfunktionen nach φ. Sei also $\hat{\varphi}^{(n+1)} \in \text{F(REK)}$ gegeben. Um ein gewünschtes h mit $\hat{\varphi}_e = \varphi_{h(e)}$ für alle $e \in \mathbb{N}$ zu erhalten, gehen wir wie folgt vor, wobei wir das Argument nur skizzieren.

Sei $M = (B, P)$ eine normierte Turingmaschine zur Berechnung von $\hat{\varphi}$. Für jedes e betrachten wir die normierte TM M_e mit Programm P_e, die bei Eingabe $\vec{x} \in \mathbb{N}^n$ den Funktionswert $\hat{\varphi}(e, \vec{x})$ wie folgt berechnet: M_e ergänzt die Eingabe $\vec{x}$ um den Parameter e und simuliert dann P auf der erweiterten Eingabe $(e, \vec{x})$. Man zeigt dann, dass sich die Gödelnummer eines Programms $\hat{P}_e$, das die Konfiguration $\text{start}(\vec{x})$ in die Konfiguration $\text{start}(e, \vec{x})$ überführt, primitiv rekursiv berechnen lässt, woraus man dann die primitive Rekursivität der Funktion $h(e) = gn(P_e)$ erhält, die offensichtlich auch injektiv ist. Nach Definition der Maschine $\tilde{U} = \tilde{U}_{\Sigma_1,n}$ gilt aber für die oben definierten $U, T \in \mathrm{F(PRIM)}$, dass

$$\hat{\varphi}(e,\vec{x}) = U(\mu y(T(gn(P_e),\vec{x},y))) = \varphi(h(e),\vec{x}),$$

weshalb h die gesuchte Übersetzungsfunktion von $\hat{\varphi}$ nach φ ist. □

Wir werden im Folgenden häufig $\{e\}^{(n)}$ oder – wenn sich die Stelligkeit n aus dem Kontext ergibt – einfach $\{e\}$ für die partiell-rekursive Funktion $\varphi_e^{(n)}$ mit Index e bzgl. der Gödelnummerierung φ aus dem Normalformsatz schreiben. D.h.

$$\{e\}(\vec{x}) = \varphi_e(\vec{x}) = U(\mu y T(e,\vec{x},y))$$

für alle e, $\vec{x}$. Weiter definieren wir

$$\{e\}_s(\vec{x}) = \varphi_{e,s}(\vec{x}) = \begin{cases} U(\mu y \leq s T(e,\vec{x},y)) & \text{falls } \exists y \leq s(T(e,\vec{x},y)) \\ \uparrow & \text{sonst} \end{cases}$$

und wir sagen, dass $\{e\}$ oder φ_e bei Eingabe $\vec{x}$ in s Schritten konvergiert, falls $\{e\}_s(\vec{x})\downarrow$. Hierbei gehen wir im Folgenden stets davon aus, dass T so gewählt ist, dass es zu jedem $(e,\vec{x})$ höchstens ein y mit $T(e,\vec{x},y)$ gibt. (Gilt dies nicht, ersetzen wir T durch

$$T(e,\vec{x},y) \,\&\, \forall z < y(\neg T(e,\vec{x},z)).)$$

Man beachte, dass $\{e\}_s(\vec{x})$ für wachsendes s gegen $\{e\}(\vec{x})$ konvergiert, d.h. für hinreichend großes s den Wert von $\{e\}(\vec{x})$ annimmt. Weiter ist das Verhalten der universellen Funktion $\varphi(e,x) = \{e\}(\vec{x})$ innerhalb einer gegebenen Zeitschranke primitiv rekursiv erkennbar. Insbesondere gilt dies für Definitionsbereich und Graph der beschränkten universellen Funktion.

11.7 LEMMA. *Für $n \geq 1$ gilt*

(i) $\{e\}(\vec{x}) = \lim_s \{e\}_s(\vec{x})$, *d.h.* $\exists s_0 \, \forall s \geq s_0 \, (\{e\}(\vec{x}) = \{e\}_s(\vec{x}))$.

(ii) $D_n = \{(e,\vec{x},s) \in \mathbb{N}^{n+2} : \{e\}_s(\vec{x})\downarrow\}$ *ist primitiv rekursiv.*

(iii) $G_n = \{(e,\vec{x},y,s) \in \mathbb{N}^{n+3} : \{e\}_s(\vec{x})\downarrow = y\}$ *ist primitiv rekursiv.*

BEWEIS. Die Teile (ii) und (iii) ergeben sich unmittelbar aus den Äquivalenzen

$$(e,\vec{x},s) \in D_n \Leftrightarrow \exists y \leq s(T(e,\vec{x},y))$$

und

$$(e,\vec{x},y,s) \in G_n \Leftrightarrow \exists z \leq s(T(e,\vec{x},z) \,\&\, U(z) = y).$$

Ähnlich folgt (i) aus

$$\begin{aligned} \{e\}(x) = y &\Leftrightarrow \exists z(T(e,\vec{x},z) \,\&\, U(z) = y) \\ &\Leftrightarrow \exists s \, \exists z < s(T(e,\vec{x},z) \,\&\, U(z) = y) \\ &\Leftrightarrow \exists s(\{e\}_s(x) = y), \end{aligned}$$

da $\{e\}_s(x)\downarrow= y$ nach Definition impliziert, dass $\{e\}_t(x) = y$ für alle $t \geq s$ gilt. □

Da eine Menge $A \subseteq \mathbb{N}^n$ genau dann rekursiv aufzählbar (r.a.) ist, wenn sie Definitionsbereich einer n-stelligen partiell rekursiven Funktion ist, liefert die Gödelnummerierung φ der partiell rekursiven Funktionen zugleich eine Nummerierung der rekursiv aufzählbaren Mengen. Wir definieren hierzu

$$\begin{aligned} W_e^{(n)} &= Db(\{e\}^{(n)}) \\ W^{(n)} &= \{(e,\vec{x}) : \vec{x} \in W_e^{(n)}\} \\ W_{e,s}^{(n)} &= Db(\{e\}_s^{(n)}). \end{aligned}$$

Ist die Stelligkeit n aus dem Kontext klar, so schreiben wir W statt $W^{(n)}$ etc.

11.8 KOROLLAR. *(Aufzählungssatz) Für jedes $n \geq 1$ ist die $(n+1)$-stellige Menge $W^{(n)}$ eine universelle Menge für die n-stelligen rekursiv aufzählbaren Mengen, d.h. $W^{(n)}$ ist rekursiv aufzählbar und zu jeder rekursiv aufzählbaren Menge $A \subseteq \mathbb{N}^n$ gibt es ein e mit $A = W_e^{(n)} = \{\vec{x} : (e,\vec{x}) \in W^{(n)}\}$.*

BEWEIS. Da $W^{(n)} = \{(e,\vec{x}) : \{e\}(\vec{x})\downarrow\} = Db(\varphi)$, ist W r.a. Ist $A \subseteq \mathbb{N}^n$ r.a., dann ist A der Definitionsbereich einer n-stelligen partiell rekursiven Funktion ψ. Für einen Index e von ψ gilt also, dass $A = Db(\psi) = Db(\{e\}) = W_e^{(n)}$. □

Weiter beobachten wir für späteren Gebrauch:

11.9 LEMMA. *Die Menge*

$$\hat{W}^{(n)} = \{(e,\vec{x},s) : \vec{x} \in W_{e,s}^{(n)}\}$$

ist primitiv rekursiv und es gilt für jedes $e,s \in \mathbb{N}$

$$W_{e,0} \subseteq W_{e,s} \subseteq W_{e,s+1} \subseteq W_e \;\&\; W_e = \bigcup_{s\geq 0} W_{e,s}.$$

BEWEIS. Da $\hat{W}^{(n)} = D_n$ und $W_{e,s} = \{\vec{x} : \{e\}_s(\vec{x})\downarrow\}$, folgt dies unmittelbar aus Lemma 11.7. □

Wir betrachten schließlich noch den Begriff der uniformen Rekursivität und uniformen rekursiven Aufzählbarkeit.

11.10 DEFINITION. Eine Folge $\{\psi_m^{(n)} : m \in \mathbb{N}\}$ von n-stelligen (partiell) rekursiven Funktionen heißt *uniform (partiell) rekursiv*, wenn es eine $(n+1)$-stellige (partiell) rekursive Funktion $\varphi^{(n+1)}$ mit $\varphi_e = \psi_e$ für alle $e \in \mathbb{N}$ gibt. Eine Folge $\{A_m^{(n)} : m \in \mathbb{N}\}$ von rekursiven (r.a.) Teilmengen von $\mathbb{N}^n$ heißt *uniform rekursiv (uniform rekursiv aufzählbar)*, wenn es eine rekursive (r.a.) Menge $B \subseteq \mathbb{N}^{n+1}$ mit $B_m = \{\vec{x} : (m,\vec{x}) \in B\} = A_m$ gibt.

Das Normalformtheorem und der Aufzählungssatz implizieren, dass die Klasse $\mathrm{F}^{(n)}(\mathrm{REK})$ aller n-stelligen partiell rekursiven Funktionen und die Klasse aller n-stelligen r.a. Mengen uniform partiell rekursiv bzw. uniform r.a. sind. Wir werden bald sehen, dass im Gegensatz hierzu, weder die Klasse der n-stelligen total rekursiven Funktionen noch die Klasse der rekursiven Mengen uniform rekursiv, ja letztere noch nicht einmal uniform r.a. ist.