1

Approximation von Folgen durch berechenbare Folgen -Eine neue Variante der Chaitin-Kolmogorov-Komplexität

Tobias Gärtner und Günter Hotz
Fakultät für Mathematik und Informatik
Universität des Saarlandes

März 2002

[1]Anschrift: D-66041 Saarbrücken
Emailadresse: hotz@cs.uni-sb.de

Abstract

Wir betrachten Approximationen unendlicher Folgen durch berechenbare unendliche Folgen minimaler Programmkomplexität. Dieser Zugang zur Charakterisierung zufälliger Folgen hat den Vorteil, dass Einbrüche der Programmkomplexität, wie sie im Falle der Approximation dieser Folgen durch ihre Präfixe gegebener Länge auftreten, nicht vorkommen. Wir erhalten so Klassen zufälliger Folgen, deren Relationen zu den Martin-Löf [M.-L. 66] und Schnorr [Schn. 71] zufälligen Folgen sowie den auf dem Konzept der *monotonen Programmkomplexität* [Schn. 73], [Lev. 73] beruhenden Charakterisierung zufälliger Folgen geklärt wird. Unser Ansatz wird geleitet von der Vorstellung der natürlichen Fortsetzung endlicher Folgen w durch unendliche berechenbare Folgen $w\ldots$. Wir erweitern auch die Menge der zulässigen Berechnungen, indem wir konvergierende unendliche Berechnungen [Ho.99] in Betracht ziehen.

1 Einleitung

Kolmogorov [Kol. 65] und Chaitin [Chai. 66] versuchten unendliche Folgen x über einem endlichen Alphabet X hinsichtlich *Zufälligkeit* zu charakterisieren, indem sie die Folge der Präfixe $x(i)$ der Länge i dieser Folgen x betrachteten und deren Beschreibung durch Paare (p, i), die bei Eingabe in eine universellen Maschine F, die Ausgabe $F(p, i) = x(i)$ erzeugen. Die Länge $|p|$ eines kürzesten Programmes mit dieser Eigenschaft definierte Kolmogorov als Programmkomplexität $K_F(x(i))$. Nach seiner Vorstellung sollte es zu jeder zufälligen Folge x und gegebener universellen Maschinen F eine Konstante c geben, so dass $|i - K_F(x(i))| < c$ für alle hinreichend grossen i gelten sollte. Diese Eigenschaft unendlicher Folgen sollte darüber hinaus die Zufälligkeit von Folgen charakterisieren. Die Abhängigkeit der Definition von F ließ sich leicht durch die Beobachtung eliminieren, dass es zu jeder Maschine einen Kompiler zur Übersetzung der Programme anderer Maschinen in Programme von F gibt. Wie Martin-Löf [M.-L. 66]zeigte, war diese Forderung zu streng. Er konnte sie aber durch eine Variante ersetzen, die der Intention Kolmogorovs genügte. Martin-Löf hatte beobachtet, dass man immer wieder Einbrüche der Komplexität von der Größenordnung $\log(i)$ hinnehemen müsse und schwächte diese Forderung ab, indem er nur verlangte, dass diese Ungleichung unendlich oft erfüllt sein müsse. Wir fragen nicht nach einem kürzesten Programm, das $x(i)$ erzeugt, sondern nach einem kürzesten Programm, das eine unendliche Folge y erzeugt, die $x(i) = y(i)$ erfüllt. Hierzu lassen wir auch konvergierende unendlich lange Berechnungen zu [Ho.99] [Ho.94]. Der Einfachheit halber beschränken wir uns auf den Fall binärer Folgen. Die Verallgemeinerungen zu grösseren Alphabeten und nicht gleichverteilten Wahrscheinlichkeiten sind nahe liegend. Schnorr hatte in seiner Habilitationsschrift [Schn. 69] den Ansatz von Martin.Löf verschärft, indem er nur totale berechenbare Funktionen in die Diskussion mit einbezog. Er konnte auf dieser Basis die Äquivalenz von vier verschieden Zugängen zu einer Fassung erstmals nachweisen. Er mußte aber auch stets wiederkehrende Einbrüche der Komplexität in Kauf nehmen und konnte die Idee der Hierarchien von Zufälligkeiten (bei ihm pseudorandom-Folgen) nicht auf der Basis der Programmkomplexität, sondern nur unter der Heranziehung des Konzeptes der Unmöglichkeit eines gewinnbringenden Spieles gegen eine Zufallsfolge bei beschränktem Kredit realisieeren.
Von diesen Komplexitätseinbrüchen konnten sich Schnorr und Levin unabhängig voneinader befreien, indem sie eine neue Variante der Komplexität einführten, indem sie eine Art Konvergenzbedingung für die approximierenden Programmfolgen forderten, nämlich zulässig sollten nur solche Programmfolgen sein, in den das vorhergehende Element Präfix des nachfolgenden

ist. Hierdurch wurde die Theorie insbesondere hinsichtlich der resourcenbedingten Hierarchien sehr vereinfacht. Diese Annnahme hat einen gewissen Charme, indem sie den Computer eine stetige Abbildung der Programmfolgen auf die zu approximierenden Folgen berechnen lässt. Sie erscheint uns allerdings auch als stark durch beweistechnische Gründe motiviert zu sein. Die an sich widerspenstigen Hierarchiesätze werden nun trivial. Unser Ansatz, nicht auf die Beschreibung von Worten abzuheben, sondern zur Approximation unendliche Folgen zu verwenden, ergibt die oben geforderte Monotonie von selbst . Einen technisch einfacheren Zugang zur obigen Theorie erhält man, indem man nach Vorbild von Chaitin davon ausgeht, dass die Programme einen präfixfreien Kode bilden. Damit werden die kürzesten Programme, die berechenbare unendliche Folgen beschreiben, die ein vorgegebenes Wort als Präfix besitzen, i.a. wesentlich länger als dieser Präfix, was unbefriedigend ist, da die Maschine doch in der Lage sein sollte, zur Erzeugung eines vorgegebenen Wortes mit einem Programm auszukommen, das nicht länger ist als dieses Wort plus einer festen Konstante. Das motieviert die zweite Version einer Maschinenklasse, die wir diskutieren. Die zulässigen Programme werden in diesem Fall durch einen Präfix der Länge 1 in Printbefehle und eine präfixfreie Sprache aufgespalten. Wir erhalten dadurch als obere Programmlänge zur Approximation eines Wortes der Länge n stets ein Programm der Länge $n+1$. Wenn in unserer Theorie auch die minimalen Programmlängen $\Delta(x(n))$ bei der wachsenden Approximation einer vorgegebenen Folge $x \in X^\infty$ bis zur Stelle n stets monoton wachsen, so müssen wir doch auch in Kauf nehmen, daß die Differenz $n - \Delta(x(n))$ oszlliert.

Wir verfolgen in unserer Arbeit also zwei verschiedene Ansätze zur Beschreibung der Komplexität von Folgen, erweitern den Maschinentyp, indem wir unendlich lange dauernde konvergente Berechnungen zulassen und definieren auf dieser Basis zufällige Folgen. Wir vergleichen diesen Begriff mit den Schnorr- und Martin-Löf zufällgen Folgen. Schränken wir die Möglichkeiten unserer Maschine ein, indem wir fordern, dass sich der Schreibkopf auf dem Ausgabeband nur nach rechts hin bewegen darf, dann erhalten wir die Martin-Löfzufälligen Folgen. Setzen wir noch schärfer voraus, dass die Berechnungen stets konvergieren ergibt sich die Charakterisierung der Schnorrzufälligen Folgen. Wir geben hier keinen expliziten Beweis für die Invarianz der in unseremSinne zufälligen Folgen unter blinden Auswahlen, die durch unsere analytische Maschine erzeugt werden sondern reduzieren diese Aussage ebenso wie Aussagen über die Gültigkeit des starken Gesetzes der großen Zahlen auf entsprechende Sätze von Schnorr. Danach geben wir allen Fassungen eine einfache abstrakte Version, wodurch einerseits der Beweis des Hierarchiesatzes übersichtlich wird und was andererseits deutlich macht, wie empfindlich dieser Satz gegenüber leichten Veränderungen der zugrunde liegenden Programmiersprache ist. Der Hierarchiesatz umfaßt die Abhänigkeit des Konzeptes der zufälligen Folgen von Einschränkungen hinsichtlich der zur Berechnung zur Verfügug stehenden Ressourcen an Zeit und Speicherplatz.

2 Grundlegende Definitionen

Sei N die Menge der natürlichen Zahlen und $X := \{0,1\} \subset N$, X^* die Menge der endlichen Folgen und X^∞ die Menge der nach rechts hin unendlichen Folgen über X. Mit xy bezeichnen wir die Konkatenation der Folgen x und y; hierin ist x eine endliche Folge, y darf auch eine unendliche Folge sein. Sind u, x und y Folgen und gilt $x = uy$, dann heißt u Präfix von x. Wir schreiben in diesem Fall $u \prec x$.

Wir betrachten Maschinen mit einem Eingabeband, einem Rechenspeicher und einem Ausgabeband. Auf dem Eingabeband findet die Maschine ihr Programm, das sie einliest und ausführt. Die Maschinen werden mit leerem Rechenspeicher gestartet. Die Resultate schreiben sie auf das Ausgabeband, wobei zugelassen wird, dass frühere Ausgaben überschrieben werden. Alle Maschinen rechnen unendlich lange. Eine Berechnung wird genau dann akzeptiert, wenn die Beschriftung des Ausgabebandes gegen eine feste Folge konvergiert.

Unter diesen Maschinen gibt es in dem Sinn universelle Maschinen, dass sie in der Lage sind, die Berechnungen jeder Maschine zu simulieren. M sei im folgenden stets eine in diesem Sinne universelle Maschine und $Prog$ sei die Menge der für M zulässigen Programme, die alle binär kodiert seien. Weiter setzen wir voraus, dass die für M zulässigen Programme einen präfixfreien Kode bilden. Das realisieren wir auf folgende Weise. Wir setzen $A := \{0,1\}^{32}$ und wählen $Prog \subset (0^{32}(A - \{0^{32}, 1^{32}, (01)^{16}\})1^{32})^*$. Hierin spielt 0^{32} die Rolle einer öffnenden und 1^{32} die Rolle der zugehörigen schließenden Klammer. $(01)^{16}$ ist unser $print - Befehl$. Das macht $Prog$ zu einem präfixfreien Kode. Die Idee ist mittels des Progammes $0^{32}(01)^{16}w1^{32}$ auf dem Eingabeband als Ausgabe $w0^\infty$ für $w \in A^*$ zu erzeugen. Das geht nicht ohne weiteres, da dieses Programm i.a. nicht in $Prog$ liegt, so dass die Maschine das Ende des Programes nicht mit Sicherheit erkennen kann. Wir erreichen unser Ziel, indem wir auch für w eine präfixfreie Kodierung

$$\gamma : A \to S^*$$

mit $S := A - \{0^{32}, 1^{32}, (01)^{16}\}$ angeben, die unsere Maschine entschlüsseln kann. Wir können γ so wählen, dass

$$H(A) \leq \log(m) * E(\gamma) < H(A) + \log(m)$$

gilt. Hierin ist $m := \sharp S$ die Anzahl der Elemente von S, und

$$H(A) := -\sum_{a \in A} \mu(a) * \log(\mu(a))$$

die zu der Quelle (A, μ) gehörige Shannon'sche Entropie [Ho.97] und μ das durch die Gleichverteilung auf $\{0, 1\}$ induzierte Maß. $E(\gamma)$ bezeichnet die mittlere Kodelänge. In unserem Fall erhalten wir also für die mittlere Kodelänge $E(\gamma, n) := \sum_{w \in A^n} \mu(w) * |w|$

$$\frac{n}{1 + \frac{1}{32} * \log(1 - \frac{3}{2^{32}})} \leq E(\gamma, n) < \frac{n}{1 + \frac{1}{32} * \log(1 - \frac{3}{2^{32}})} + 1$$

. Damit können wir die mittlere Länge $E(printw)$ des Printprogrammes, das auf dem Ausgabeband $w0^\infty$ erzeugt, durch

$$E(printw) < \frac{n}{1 - \delta} + 97 \quad mit \quad \delta := -\frac{1}{32} * \log(1 - \frac{3}{2^{32}})$$

nach oben abschätzen. Betrachten wir den Beweis zu der angegebenen oberen Schranke [Ho.97] Seite 29, dann erhalten wir auch die Abschätzung

$$|printw| < \frac{|w|}{1 - \delta} + 97$$

2.1 Programmkomplexität

Die Menge der berechenbaren Folgen $x \in X^\infty$ liegt dicht in X^∞. Wir interessieren uns für den Zusammenhang zwischen der Güte der Approximation von beliebigen durch berechenbare Folgen und das damit verbundene Wachstum der zu den berechenbaren Folgen gehörigen minimalen Programmgrößen. In diesem Zusammenhang stehen die folgenden Definitionen.

Definition 1 *Wir betrachten zu vorgegebem Wort $w \in X^*$ die Programme p mit $w \prec M(p)$ und definieren*

$$\Delta_M(w) := \min\{|p| : p \in Prog, w \prec M(p)\}$$

Ist $x \in X^\infty$ berechenbar, dann definieren wir

$$\Delta_M(x) := \min\{|p| : p \in Prog, M(p) = x\}$$

Aufgrund der Struktur von M gilt also für $w \in A^n$

$$\sum_{w \in A^n} \Delta_M(w) * \mu(w) < \frac{|w|}{1-\delta} + 97$$

und damit für beliebige $w \in X^r$

$$\sum_{w \in X^r} \Delta_M(w) * \mu(w) < \frac{|w|}{1-\delta} + 128$$

da wir nicht in A^* liegende Folgen über X um eventuell 31 Stellen zu in A^* liegenden Folgen ergänzen müssen. Wir erhalten darüber hinaus auch die individuelle Abschätzung

$$\Delta_M(w) < \frac{|w|}{1-\delta} + 128$$

Wir betrachten eine zweite, allgemeinere Klasse von Möglichkeiten, Folgen $x \in X^\infty$ durch berechenbare Folgen zu approximieren. Hierzu verwenden wir eine Distanz $d(x, y)$.

Definition 2 *Für $x, y \in X^\infty$ definieren wir*

$$|x| := \sum_{i=1}^{\infty} |x_i| * 2^i$$

für $x = x_1, x_2, x_3, \ldots$ und setzen

$$d(x, y) := |x - y|,$$

worin die Differenz $x - y$ komponentenweise zu nehmen ist.

Wir definieren diese Distanz auch für endliche Folgen und zwischen endlichen und unendlichen Folgen, indem wir die Summation nur bis zur Länge der kürzeren Folge führen. Man kann diese Distanzen auch als Abstände zwischen den von den Folgen x, y erzeugten Zylindermengen $[x], [y]$ auffassen, oder als Distanz zwschen den durch diese Folgen repräsentierten reellen Zahlen.

Definition 3 *Wir definieren zu gegebenem $\varepsilon > 0$ und der universellen Maschine M*

$$\Delta_{M,\varepsilon}(x) := \min\{|p| : d(M(p), x) < \epsilon\}$$

Wir werden diese Definition später motivieren.

3 Eigenschaften der Komplexitätsmaße

Um die im Zusammenhang mit der Kolmogorov Komplexität auftretenden unvermeidlichen Einbrüche in der Folge der Komplexitäten der Präfixe uendlicher Folgen zu vermeiden, haben Schnorr und Levin die Auswahl der zulässigen Programme durch eine Monotonieforderung eingeschränkt. Wir zeigen zunächst, dass die hier eingeführten Δ- Komplexitäten im ersteren Fall ohne weitere Annahmen und im zweiten Fall unter der Voraussetzung monoton fallender ϵ-Folgen zu jeder unendlichen Folge x mit i momoton wachsende Folgen $\Delta_M(x(i))$ bezw. $\Delta_{M,\varepsilon_i}(x(i))$ erzeugen.

Lemma 1 *Für $x, y \in X^*$ und $x \prec y$ gilt*

$$\Delta_M(x) \leq \Delta_M(y)$$

Beweis: Seien $x, y \in X^*$ und p_1 für x und p_2 für y jeweils kürzeste Programme mit $x \prec M(p_l)$ und $y \prec M(p_2)$. Nun ist $x \prec y \prec M(p_2)$. Also ist $|p_1| \leq |p_2|$, was zu zeigen war.

Lemma 2 *Für $x, y \in X^*$ und $x \prec y$ gilt für jede monoton fallende Nullfolge $\epsilon = \varepsilon_1, \varepsilon_2, \varepsilon_3, \ldots$*

$$\Delta_{M,\epsilon_{|x|}}(x) \leq \Delta_{M,\epsilon_{|y|}}(y)$$

Beweis: Seien p_1 und p_2 zu $x, \varepsilon_{|x|}$ bezw. $y, \varepsilon_{|y|}$ gehörigen Programme minimaler Länge. Wegen $x \prec y$ und $\varepsilon_{|y|} \leq \varepsilon_{|x|}$ würde aus $|p_2| < |p_1|$ die Relation $d(x, M(p_2)) \leq \varepsilon_{|y|} \leq \varepsilon_{|x|}$ folgen, was der Minimalitätsbedingung in der Auswahl von p_2 widerspricht.
Hieraus ergibt sich unmittelbar der folgende

Satz 1 (Monotoniesatz) *Ist $x \in X^\infty$ und ϵ eine monoton fallende Nullfolge, dann gilt $\Delta_M(x(i)) \leq \Delta_M(x(i+1)$ und $\Delta_{M,\varepsilon_i}(x(i)) \leq \Delta_{M,\varepsilon_{i+1}}(x(i+1))$.*

3.1 Δ-Komplexität

Wir werden in diesem Abschnitt eine Klasse von unendlichen Folgen als *zufällig* definieren und sie in Beziehung setzen zu den *Martin-Löf* und den *Schnorr-zufälligen* Folgen.

Definition 4 *Sei für $c \in \Re$*

$$K_M^{(\delta)}(c) := \{x \in X^\infty : \forall_{n \in N} n - (1-\delta) * \Delta_M(x(n)) < c\}$$

und

$$K_M^{(\delta)} := \cup_{c \in N} K_M^{(\delta)}(c)$$

K_{Δ_M} *heißt die Menge der* Δ_M-zufälligen Folgen

Wir wollen zeigen, dass es zufällige Folgen gibt, und darüber hinaus, dass die Menge dieser zufälligen Folgen im Sinne der Gleichverteilung das Maß 1 hat. Sei Ω die Menge der durch M berechenbaren unendlichen Folgen und $\kappa : \Omega \to Prog$ eine Abbildung mit den Eigenschaften (1) und (2):

$$(1) \quad M(p) = x \qquad (2) \quad |\kappa(x)| = \Delta_M(x)$$

Wir definieren weiter Abbildungen $\kappa_n : X^n \to Prog$ für $n \in N$, die mit κ in folgender Weise verträglich sind: Für $w \in X^n$, $x \in \Omega$ und $w \prec x$ gilt

$$\Delta_M(w) = \Delta_M(x) \Longrightarrow \kappa_n(w) = \kappa(x)$$

Offensichtlich sind κ und κ_n injektiv. Da $Prog$ präfixfrei ist, können wir κ_n als präfixfreien Kode ansehen und die oben angegebenen Shannon'schen Ungleichungen für gedächtnislose Quellen und deren Folgerungen anwenden. Wir erhalten so die Abschätzung

$$n \leq \sum_{w \in X^n} \frac{\Delta_M(w) * (1-\delta)}{2^n}$$

Wir definieren für $h : N \to N$

$$\Lambda_M(h(n), n) := \{w \in X^n : n - h(n) > \Delta_M(w) * (1-\delta)\}$$

Sei μ das durch die Gleichverteilung über X induzierte Wahrscheinlichkeitsmaß.

Lemma 3

$$\mu(\Lambda_{M(h(n),n)} * X^\infty) < \frac{128 * (1-\delta)}{h(n) + 128 * (1-\delta)}$$

Beweis: Wir zerlegen die Summation über $w \in X^n$ in zwei Summen, nämlich über $C_1 := \Lambda_M(h(n), n) * X^\infty$ und $C_2 := (X^n - \Lambda_M(h(n), n)) * X^\infty$ und setzen $\mu_1(n) := \mu(C_1)$ und $\mu_2(n) := \mu(C_2)$. Damit erhalten wir

$$n < \sum_{C_1} \frac{\Delta(w) * (1-\delta)}{2^n} + \sum_{C_2} \frac{n + 128 * (1-\delta)}{2^n}$$

$$< \sum_{C_1} \frac{(n - h(n)) * (1-\delta)}{2^n} + \sum_{C_2} \frac{(n + 128) * (1-\delta)}{2^n}$$

$$= \mu_1(n) * (n - h(n)) + \mu_2(n) * (n + 128 * (1-\delta)),$$

woraus sich wegen $\mu_1(n) + \mu_2(n) = 1$

$$\mu_1(n) < \frac{128 * (1 - \delta)}{h(n) + 128 * (1 - \delta)}$$

ergibt, was zu zeigen war. Für

$$\Lambda_M(h) := \{x \in X^\infty : \forall_n (n - h(n) > \Delta_M(x(n)) * (1 - \delta))\}$$

gilt

$$\Lambda_M(h) = \cap_n \Lambda_M(h, n) * X^\infty$$

und für unbeschränktes h gilt also

$$\mu(\Lambda_M(h)) = 0.$$

Mit dieser Bezeichnung gilt

Lemma 4 *Ist $x \in X^\infty$ und gilt für jede unbeschränkte Abbildung $h : N \to N$*

$$x \notin \Lambda_M(h),$$

dann gilt

$$\exists_{c \in N} \forall_{n \in N} (n - c < \Delta_M(x(n)) * (1 - \delta)).$$

Wir setzen erinnern an Definition 4 und vermuten also den

Satz 2 *Die Menge der Δ_M-*zufälligen *Folgen hat das Maß 1, d.h. es gilt*

$$\mu(K_M^{(\delta)}) = 1$$

Beweis: Wir schreiben anstelle von $K_M^{(0)}(c)$ abkürzend $K_M(c)$ und verwenden $K_M^{(0)} \subset K_M^{(\delta)}$. Es genügt dann zu zeigen, dass für jedes c

$$\mu(K_M(c)) > 1 - 2^{-c}$$

gilt. Sei

$$L_n := \{w \in X^n : \exists_{m \leq n} (m - \Delta(w(m))) > c\}$$

und $\tilde{L_n} \subseteq L_n$ sei die maximale präfixfreie Teilmenge von L_n. Da unsere Sprache $Prog$ präfixfrei ist, gilt

$$1 \geq \sum_{w \in \tilde{L_n}} \frac{1}{2^{\Delta_M(w)}} > \sum_{w \in \tilde{L_n}} \frac{1}{2^{n-c}} = 2^c * \sum_{w \in \tilde{L_n}} \frac{1}{2^n} = 2^c * \mu(L_n * X^\infty).$$

Hieraus folgt

$$\mu(L_n * X^\infty) < 2^{-c},$$

woraus die Behauptung folgt.
Die in den obigen Konstruktionen auftretende Konstante δ, ist von beweistechnischer Natur. Sie kommt herein durch den Wunsch, Resultate der Kodierungstheorie verwenden zu können. Ihre Größe ergab sich aus der Definition des Alphabetes A. Wählen nun anstelle von 32 in $\{0,1\}^{32}$ eine andere Potenz k, dann nimmt die von uns benutze Ungleichung aus der Kodierungstheorie die Form

$$\frac{n}{\alpha} \leq \sum_{w \in X^n} \frac{\Delta(w)}{2^n} < \frac{n}{\alpha} + \beta$$

an. α und β sind hierin positive Konstanten, die sich aus dem Exponenten k ergeben. Hieraus erhält man dann für $\mu_1 := \mu(\Lambda_M(h,n))$ nach kurzer Rechnung die Abschätzung

$$\mu_1 < \frac{\alpha * \beta}{h(n) + \alpha * \beta}$$

Ist $h(n)$ eine monotone Funktion, dann gilt das auch für

$$\tilde{h}(n) := \frac{h(n) + \alpha * \beta}{\alpha * \beta}$$

und wir erhalten nun

$$\mu_1 < \frac{1}{\tilde{h}(n)}.$$

Hieraus folgt, dass die spezielle Wahl des Exponenten in der Definition unserer Programmiersprache für die Definition von Λ_M und K_M keine Rolle spielt. Wir können also darauf verzichten die Abhängigkeit von der speziellen Maschine zum Ausdruck zu bringen.

Definition 5

$$K_\Delta := K_{M,k},$$

worin M eine universelle Maschine ist, und die oben definierte Programmiersprache Prog verwendet wird.

Wir fassen unser Resultat in dem folgenden Satz zusammen.

Satz 3 *Für alle universellen Maschinen M und Programmiersprachen vom Typ Prog ist*

$$K_\Delta = K_M$$

von M unabhängig und es gilt

$$\mu(K_\Delta) = 1$$

Wir haben uns damit natürlich nicht von der speziellen Voraussetzung befreit, dass $Prog$ eine präfixfreie Sprache ist. Wir waren auch nicht konsequent in der Verwendung unendlicher Berechnungen, indem wir nämlich voraussetzen, dass M nur einen endlichen Abschnitt des Eingabebandes liest. Indem wir uns davon befreien, erübrigt sich auch die Voraussetung der Präfixfreiheit der Programmiersprache, wodurch die ganze Theorie völlig elementar ohne einen Bezug auf die Kodierungstheorie begründet werden kann. Das ist der Gegenstand des folgenden Abschnittes.

3.2 Δ^* - Komplexität

Das Maschinenmodell bleibt das selbe. Allerdings wandert der Lesekopf der Maschine mit jedem Rechenschritt um eine Position weiter nach rechts. Das Eingabeband trägt stets nur endlich viele Einsen. Alle Felder, die keine 1 tragen sind mit 0 beschrieben. Die Menge dieser Eingabebänder bezeichnen wir als *rational*. Als Programmlänge bezeichnen wir de Länge des längsten Präfixes des Eingabandes, der mit 1 endet, falls überhaupt eine 1 auf dem Eingabeband steht. Als Länge der Eingabe 0^∞ definieren wir 1. Die Menge der Programme bildet jetzt keinen präfixfreien Kode mehr und wir müssen sagen, wie die Maschine das Ende des Programmes bestimmt. Die Maschine weiß natürlich niemals, wann sie die letzte 1 gefunden hat. Sie geht immer davon aus, dass die zuletzt gefundene 1 auch die letzte ist. Wenn der Lesekopf auf seiner Wanderschaft eine neue 1 findet, dann startet die Maschine nun unter der Annahme,das Programm zu kennen, die Berechnung von neuem. Da nur endlich viele 1 auf dem Band stehen, hat die Maschine nach endlich vielen Schritten das richtige Programm in Arbeit.
Wir führen wieder einen speziellen *Printbefehl* ein. Steht auf dem Eingabeband die rationale Folge

$$ox \in X^*0^\infty,$$

dann gibt die Maschine stets x aus. Steht $1x \in X^\infty$ auf dem Eingabeband, dann interpretiert die Maschine $1x$ als Programm einer univesellen Maschine. Wir verwenden den Printbefehl, um die Programmlängen und die mittlere Programmlänge nach oben abzuschätzen. Zur Erzeugung der Ausgabe $w0^\infty$ genügt offensichtlich die Eingabe $0w0^\infty$, das heißt ein Programm der Länge $\leq |w| + 1$. Endet w mit einer 1, hat der Printbefehl genau die Länge $|w|$. Gilt $w = v10^k$, dann erzeugt bereits $v10^\infty$ die Ausgabe $w0^\infty$. Die mittlere Länge der Printbefehle minimaler Länge zu Erzeugung aller Wörter aus X^n ergibt sich somit als

$$\frac{1}{2^n} + \sum_{i=0}^{n-1} \frac{i+2}{2^{n-i}} = n + \frac{1}{2^n}.$$

Man beachte, dass wir dem speziellen Printbefehl 0^∞ zur Erzeugung von $0^n 0^\infty$ die Länge 1 zugewiesen haben. Nun schätzen wir die mittlere Programmlänge nach unten ab. Da zu zwei verschiedenen Wörtern $w, v \in X^n$ stets verschiedene Programme gehören, genügt es die mittlere Länge der 2^n kürzesten mit 1 endenden Wörter in X^* abzuschätzen um eine untere Schranke zu erhalten. Man erhält $n - 1 + \frac{1}{2^n}$. Somit gilt also der

Satz 4 *Die mittlere Größe der Programmkomplexität $\Delta^*(w)$ unserer Maschine erfüllt die folgende Ungleichungen*

$$n - 1 + \frac{1}{2^n} \leq \sum_{w \in X} \frac{\Delta^*(w)}{2^n} \leq n + \frac{1}{2^n}.$$

Nun definieren wir etwas anders als vorher im Falle der Δ-Komplexität

Definition 6 *Sei H die Menge der monoton wachsenden und unbeschränkten berechenbaen Abbilungen $h : N \to N$ und $\tilde{H}$ die Menge der monoton und unbeschränkt wachsenden Abbildungen $\tilde{h} : N \to N$, die langsamer wachsen als jedes $h \in H$ oder die beschränkt sind. Wir definieren*

$$K_\Delta^* := \{x \in X^\infty : g(n) := n + 1 - \Delta^*(x(n)) \in \tilde{H}\}$$

und erhalten den

Satz 5 *Die Menge K_Δ^* der Δ^*-zufälligen Folgen hat das Maß*

$$\mu(K_\Delta^*) = 1.$$

Beweis: Sei für $h : N \to N$

$$\Lambda_M(h) := \{x \in X^\infty : \forall_n (n - h(n) \geq \Delta_M^*(x(n))\}$$

und für $n \in N$

$$\Lambda_M^{(n)}(h) := \{x \in X^\infty : (n - h(n) \geq \Delta_M^*(x(n)))\}.$$

Weiter sei

$$L_n(h) := \{w \in X^n : (n - \Delta^*(w) \geq h(n))\}.$$

Offensichtlich gilt

$$L_n(h) * X^\infty = \Lambda_M^{(n)}(h)$$

und

$$\cap_{n=1}^\infty \Lambda_M^{(n)}(h) = \Lambda_M(h).$$

Wir zeigen zunächst das

Lemma 5 *Ist h monoton und unbeschränkt, dann gilt $\mu(\Lambda_M(h)) = 0$.*

Beweis: Dazu genügt es zu zeigen, dass

$$\mu(\cap_{i=c}^{n} L_i(c) * X^\infty) = 0.$$

gilt. Hierzu verwenden wir die Abschätzung der mittleren Programlänge, die wir oben hergeleitet haben. Indem wir die darin auftretende Summe in zwei Teilsummen aufspalten erhalten wir

$$n - 1 < \sum_{n-h(n) \geq \Delta_M^*(w)} \frac{\Delta_M^*(w)}{2^n} + \sum_{n-h(n) < \Delta_M^*(w)} \frac{\Delta_M^*(w)}{2^n} < n + 2$$

und hieraus

$$n - 1 < (n - h(n)) * \sum_{n-h(n) \geq \Delta_M^*(w)} \frac{1}{2^n} + (n+2) * \sum_{n-h(n) < \Delta_M^*(w)} \frac{1}{2^n}.$$

Wir setzen

$$\mu_1 := \mu(\Lambda_M^{(n)}(h))$$

und erhalten

$$n - 1 < (n - h(n)) * \mu_1 + (n+1) * (1 - \mu_1) = -h(n) * \mu_1 + n + 1 - 2 * \mu_1$$

oder

$$\mu_1 < \frac{2}{h(n) + 2}$$

woraus schließlich die Behauptung unseres Lemmas

$$\mu(\Lambda_M^*(h)) = 0$$

folgt.

Sei nun H, wie oben die Menge der totalen monotonen und unbeschränkten berechenbaren Abbildungen $h : N \to N$ und

$$\Lambda(H) := \cup_{h \in H} \Lambda(h).$$

Da H abzählbar ist gilt

$$\mu(\Lambda(H)) = 0.$$

Nun ist offfensichtlich K_Δ^* das Komplement von $\Lambda(H)$, woraus sich die Behauptung von $Satz 5$ ergibt.

Wir haben gezeigt, dass die von uns definierte Programmkomplexität monoton wächst. Es stellt sich aber die Frage, in welcher Weise sie wächst.

Man kann zeigen, das in der Tat immer wieder Verzögerungen im Wachstum auftreten und dass diese unbeschränkt sind. Hat man einen Printbefehl auch in den Programmen, die mit dem Präfix 1 beginnen, zur Verfügung, dann können diese Verzögerugen nicht beliebig groß werden, da das Wachstum des Programmes doch so groß ist, dass es immer wieder bis auf die Konstante c an n herankommt. Wir verfolgen diese Frage hier nicht weiter sondern betrachten eine weitere Variante der Δ^*-Komplexität.

Satz 6 *Bilden die mit dem Präfix 1 beginnenden Programme unserer Maschine einen präfixfreien Kode, dann gilt für*

$$K_M^* := \{x \in X^\infty : \exists_{c \in N} \forall_{n \in N} (n - \Delta_M^* < c)\}$$

$$\mu(K_M^*) = 1.$$

Es stellt sich nun die Frage, mit welchem Recht wir die in K_Δ liegenden Folgen als *zufällig* bezeichnen. Eine Motivation dafür liefert der allgemeine Sprachgebrauch: Es ist üblich, eine Folge, in der wir keine Gesetzmäßigkeiten erkennen, als zufällig zu bezeichnen. Das allein reicht aber nicht aus. Es mag ja Gesetzmäßigkeiten geben, und solche gibt es auch, die sich in der Δ-Komplexität nicht widerspiegeln. Schnorr hat gezeigt, dass die in mehren Hinsichten ausgezeichnete Klasse der von ihm als *zufällig* definierten Folgen alle im Sinne der Theorie der Berechenbarkeit entscheidbaren fast-überall Gesetze der Wahrscheinlichkeitstheorie erfüllt. Indem wir zeigen, dass unsere Δ-zufälligen Folgen auch im Sinne Schnorrs zufällig sind, motivieren wir unsere Definition hinreichend.

3.3 Beziehung zu Martin-Löf- und Schnorr- zufälligen Folgen

Seien $K_{M.L}$ die Menge der in Sinne Martin-Löfs zufälligen und K_{Sch} die im Sinne Schnorrs zuälligen Folgen. Bekantlich gilt [Schn. 77] $K_{M.L} \subset K_{Sch}$. Wir zeigen, dass die Menge der Δ-*zufälligen* Folgen echt in $K_{M.L}$ enthalten ist. Für subberechenbare Martingale $V : X^* \to \Re$ mit $V(\epsilon) < 1$ gibt es definitionsgemäß primitivrekursive Funktionen

$$g : N \times X^* \to R \quad mit \quad g(i, w) \leq g(i + 1, w)$$

und

$$\lim_{i \to \infty} g(i, w) = V(w)$$

Nun definieren wir ein Prädikat P durch

$$P(w) = 1 \iff V(w) \leq 1$$

und

$$z \in X^\infty \quad mit \quad z(0) := \epsilon, \quad z(i+1) := z(i) * \min\{\beta : P(z(i) * \beta) = 1\}$$

Wegen $V(\epsilon) < 1$ gibt es solche Folgen. Es gibt, wie Martin-Löf gezeigt hat, subberechenbare Martingale, die universelle Nullmengen definieren. Definiert V eine universelle Nullmenge, dann ist z eine im Sinne der Definition von M.-L. zufällige Folge.

Lemma 6 *Die oben definierte Folge $z \in X^\infty$ ist M-berechenbar, das heißt es gilt $\Delta(z) < \infty$*

Beweis:Zum Beweis definieren wir die Prädikate P_i durch

$$P_i(w) = 1 \Longleftrightarrow g(i, w) \leq 1$$

und $z^{(i)} := z_1^{(i)} z_2^{(i)} ... z_i^{(i)}$ induktiv durch

$$z^{(i)}(0) = \epsilon, z^{(i)}(l+1) := z^{(i)}(l) * \min\{\beta : P_i(z^{(i)}(l) * \beta) = 1\}$$

. die Wörter $z^{(i)}(i)$ sind berechenbar. Indem wir diese Wörter für $i = 1, 2, 3, ...$ auf das Ausgabeband schreiben und zwar so, dass $z^{(i+1)}(i+1)$ das Wort $z^{(i)}(i)$ überschreibt, konvergiert die Ausgabe gegen die Folge z, da mit wachsendem i die Anzahl der Stellen der $z^{(i)}(i)$, die mit $z(i)$ übereinstimmen, monoton zunimmt. Da die Menge der berechenbaren Funktionen echt in der Menge der analytisch berechenbaren enthalten ist, ist jede rekusive Nullmenge auch eine Nullmenge in unserem Sinne, woraus die Behauptung des Satzes folgt.
Damit haben wir auch gezeigt, dass die Folgen $x \in K_\Delta$ die von zufälligen Folgen erwarteten Gesetze der großen Zahlen und die Invarianzeigenschaften gegenüber der "blinden" Auswahl von Teilfolgen erfüllen.

4 Hierarchien von Zufälligkeiten

Die Motivation für die von Schnorr in Saarbrücken 1967 auf meine Anregung hin begonnene Untersuchung unseres Fragenkreises war die Untersuchung des Einflusses von Beschränkungen in den Ressourcen an Rechenzeit und Speicherplatz auf die Klassifikation von unendlichen Folgen hinsichtlich ihrer Programmkomplexität. Das Hauptresultat Schnorrs waren neben seinen Invarianzsätzen hisichtlich der Transformation der Kollektive bei *blinder* Auswahl von Teilfolgen vier Fassungen des Begriffes der Zufälligkeit, nämlich auf Basis des Brouwerschen Ansatzes für *konstruktive Nullmengen*, eine Fassung des von Mises'schen *Kollektives*, dem ausgeschlossenen *gewinnbringenden Spieles* gegen eine Zufallsfolge bei beschränktem Kredit nach Ville [Vi39] und dem Konzept der *Chaitin-Kolmogorov'schen Programmkomplexität*. Er konnte zeigen, dass seine Fassungen dieser vier Konzepte zu äquivalenten Definitionen der zufälligkeit von Folgen führt, was die erste überzeugenden Rechfertigung jedes dieser Ansätze darstellte. Verfeinerungen dieses Ansatzes durch die Einbringung des Gesichtspunktes der Ressourcen konnte er auf Basis des ausgeschlossenen Spielsystemes, nicht aber auf Basis der Programmkomplexität durchführen. Später [Schn. 73] schränkte er die zulssigen Approximationen der unendlichen Folgen durch berechenbare endliche Folgen stark ein, indem er eine Monotonieforderung stellte, die auch zu einer befriedigenden Erfassung des Ressourcen Gesichtspunktes führte. Diese Forderung erscheint mir allerdings nicht so sehr natrlich, sondern mehr gerechtfertigt durch die sich ergebende einfache Theorie. Gestützt wird dieser Zugang allerdings durch einen äquivlenten und unabhngig von Levine [Lev. 73] gewählten Ansatz. Unsere Definition der Komplexität liefert die erwüschte Monotonie auf natürliche Weise und ergibt eine einfachere Theorie dadurch, dass weniger Resultate aus der Theorie der Berechenbarkeit erforderlich sind. Natürlich müssen wir uns an einigen Stellen den Vorwuf geringerer Konstruktivität gefallen lassen, aber praktikabel ist keiner der untersuchten Ansätze, so dass wir zunächst der Einfachheit der Theorie den Vorzug geben. In diesem Abschnitt wollen wir die Konsequenzen der Beschränkungen von Resoucen untersuchen, was wir als den interessantesten Teil der Theorie ansehen.

4.1 Ein abstraktes Modell

Wir führen zunächst ein abstraktes Modell ein, das aus einer Folge von Abbildungen und einer Bewertung dieser Abbildungen besteht, die es uns erlaubt ohne einen Bezug auf das Konzept der Berechenbarkeit einen Hierarchiesatz zu beweisen, der in Spezialisierungen den oben skizzierten Hierarchisatz über zufällige Folgen ergibt.

Definition 7 *Sei $\Psi = (z_1, z_2, z_3, ...)$ eine Abzählung von unendlichen Folgen $z : N \to X$. Ψ heißt eine Komplexitätsklasse, mit dem Maß $|z_i| := \lceil \log(i) \rceil$, wenn (1) und (2) gelten.*

$$X^* * \Psi = \Psi \quad und \quad (X^*)^{-1} * \Psi = \Psi \tag{1}$$

$$\text{Zu jedem} \quad w \quad \text{gibt es ein} \quad i < 2^{|w|+1} \quad \text{mit} \quad w \prec z_i \tag{2}$$

(wir multiplizieren nach der Regel $\alpha^{-1} * \alpha = 1$ und $\alpha^{-1} * \beta = 0$ für $\alpha \neq \beta$).

Wir definieren weiter

$$\Delta_\Psi(w) := \min\{|z_i| : w \prec z_i\}$$

Wir ordnen nun jedem $w \in X^*$ eine Folge $x_w \in X^\infty$ zu. Das geschieht durch die folgende Konstruktion:

Definition 8 *Aus der Definition der Komplexitätsklasse folgt, dass es einen Index i gibt, so dass $w \prec z_i$ gilt. Wir setzen*

$$i_0 := \min\{i \in N : w \prec z_i\}$$

Wir definieren induktiv

$$i_{k+1} := \min\{i \in N : z_i(|w|+k) = z_{i_k}(|w|+k), z_i(|w|+k+1) \neq z_{i_k}(|w|+k+1)\}$$

und setzen $x_w(|w|+k) := z_{i_k}(|w|+k)$ für $k \in N$ und $\bar{\Psi} := \{x_w : w \in X^*\}$.

Lemma 7 *Für alle $w \in X^*$ gilt*

$$x_w \notin \Psi$$

Beweis: Liegt x_w in Ψ, dann gibt es einen Index i, so dass $z_i = x_w$ gilt, was aufgrund der Konstruktion nicht mglich ist.

Unser Ziel besteht aber in einem wesentlich schärferen Ergebnis: Wir möchten zeigen, dass die durch diese Diagonalisierungen konstruierten Folgen mittels Folgen aus Ψ i.a. nicht effizient approximiert werden können, d.h. Ψ-*zufällig* sind, wie wir das nennen wollen. Wir beharren dabei auf der Reihenfolge der Abzählung, was für den Beweis wesentlich ist. Auf diesen Punkt werden wir später zurckkommen. Aufgrund der Konstruktion sind die zur Definition von x_w verwendeten Folgen z_{i_k} die optimalen Approximationen von $x_w(|w|+k)$, so dass wir nur das Wachstum von i_k abschätzen müssen. Hierzu zählen wir die Anzahl der Folgen z_i ab, die wir bei der Konstruktion der x_w für $|w| \leq n$ verbrauchen. Man erhält hier für

$$2 * n + 2 * (n-1) + 2^2 * (n-2) + ... + 2^{n-1} * (n-(n-1)) = 2^{n+1} - 2.$$

Nun stehen uns, wie aus der Definition der Komplexitatsklasse folgt, gerade 2^{n+1} Elemente $z \in \Psi$ mit $|z| \leq n+1$ zur Verfügung um alle Wörter aus X^n darzustellen. Unsere Abschätzung zeigt, dass wir alle diese Folgen in der Konstruktion der Diagonalfogen verwenden, dass beim Übergang von n zu $n+1$ auch stets 2^n neue Folgen gebraucht werden, so dass i_k also im Mittel in Abhänigigkeit von k exponentiell wächst; d. h. es gilt $|z_{i_k}| \approx k$ für unendlich viele k. Wir fassen das Resultt in dem folgenden Satz zusammen.

Satz 7 (Hierarchiesatz) *Für unendlich viele $w \in X^*$ gibt es ein $c \in N$ so dass für unendlich viele n $n - \Delta_\Psi(x_w(n)) < c$ gilt.*

Wir kommen zurück zu der obigen Andeutung hinsichtlich der Reihenfolge einer Aufzählung oder in anderen Worten zu der zugrunde liegenden Programmiersprache: Wählen wir nicht $\lceil \log(i_k) \rceil$ als Komplexitätsmaß, sondern $\lceil \log(k) \rceil$ als Maß für $|z_{i_k}|$, dann erhalten wir natürlich ein stark abweichendes Resultat über die Komplexität der optimalen Approximationen der Folgen aus $\bar{\Psi}$. Betrachten wir als Beispiel für Ψ die zu einer Zeitkomplexität $O(T(n))$ gehörige Klasse von turingberechenbaren Funktionen. Bekanntlich kann man ein Programm für die Turingmaschine schreiben, das alle in dieser Klasse liegenden Funktionen aufzählt und daraus durch Diagonalisierung eine Funktion defnieren, die nicht in dieser Klasse wohl aber in der zu $O(T(n) * \log(T(n)))$ gehörigen Klasse liegt. Eine leichte Abwandlung dieser Konstruktion haben wir zur Definition von $\bar{\Psi}$ verwendet. Die Aufzählung aller Funkionen der Komplexitätsklasse zu $O(T(n))$ kann man nun auch so abwandeln dass sie bei Eingabe von k die Folge z_{i_k} liefert. Damit haben wir eine neue Maschine oder Programmiersprache definiert, die k als Programm für z_{i_k} interpretiert. Bezüglich dieser Sprache erhalten wir also ein sehr effiziente Approximation. Allerdings wächst die mit der Abschätzung der Laufzeit durch $const * T(n)$ für konkrete Algorithmen verbundene Konstante $const$ dabei sehr rasch an. Es zeigt sich hier, dass die zugrund liegende Programmiersprache in dem Falle beschränkter Komplexitäten eine ganz andere Rolle spielt als in dem unbeschränkten Fall.

5 Ausblick

Diese Arbeit hat einen vorläufigen Charakter. In einer umfassenderen Folgearbeit werden wir Invarianzeigenschaften und das stochastische Verhalten der definierten Hierarchien zufälliger Folgen auch uter Einbeziehung des in Definition 2 beschriebenen Distanzbegriffes näher untersuchen. Die hier mitgeteilten Resultate gehen im Wesentlichen auf den zweiten Autor zurück. Hinsichtlich der angekündigten Erweiterungen verweisen wir auf die Dissertation von Tobias Gärtner.

References

[Kol. 65] A. N. Kolmogorov *Drei Zugänge zur Definition des Begriffs Informationsgehalt(russisch)*, Probl. Peredaci Inform. 1 (1965), 3-11.

[Chai. 66] G.J. Chaitin *On the length of programs for computing finite binary sequences*, J. Assoc. comp. Machin. 13 (1969), 547-569.

[M.-L. 66] P. Martin-Löf *The definition of random sequences*,Information and Control, vol. 8(1966), pp. 602-619.

[Schn. 71] C.-P. Schnorr *Zufälligkeit und Wahrscheinlichkeit, eine algorithmische Begründung der Wahrscheinlichkeitstheorie*, Lecture Notes in Mathematics, vol. 218, Springer Verlag, 1971

[Schn. 73] —-*Process complexity and effectiv random tests*, Journal of Computer and System Sciences, vol.7(1973), pp. 376-388.

[Schn. 69] ——*Eine Bemerkung zum Begriff der zufälligen Folge*, Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete, vol. 14 (1969), pp. 27-35.

[Schn. 69] —*Eine neue Charakterisierung der Zufälligkeit von Folgen* Habilitationsschrift zur Erlangung der venia legendi im Fach Matematik an der MNF der Universitä des Saarlandes (1969) 114 Seiten

[Schn. 77] —-*A survey of the theeory of random sequences*, Butts and Hintikka (eds), Basic Problems in Methodology and Linguistics, 193-211. 1977 by D. Reidel Publishing Company, Dordrecht-Holland

[Lev. 73] L. A. Levin *On the notion of a random sequence* Soviet Mathematics, Doklady, vol. 14(1973),pp. 1413-1416.

[Ho.97] G. Hotz *Algorithmische Informationstheoie*, Teubner-Texte zur Informatik, B.G. Teubner,(1997)

[Ho.99] T. Chadzelek und G. Hotz *Analytic machines* Theoretical Computerscience 219(1999),151-167

[Ho.94] G. Hotz *Über Berechenbarkeit fraktaler Strukturen (Kapitel 8)* Abhandlungen der Akademie der Wissenschaften und der Literatur, Mathematisch-Naturwissenschaftliche Klasse (1994), Nr.1. Fritz Steiner Verlag

[Vi39] J. Ville *Etude critique de la notion de collectiv* Paris: Gauthier-Villars 1939