Inhaltsverzeichnis

Vorwort **1**

1 Komplexität Boolescher Funktionen **5**
1.1 Boolesche Funktionen 5
1.2 Schaltkreise und Komplexitätsmaße 10
1.3 Das Vollständigkeitskriterium von Post 15
1.4 Beziehungen zwischen den Maßen 24
1.5 Asymptotische Komplexität – untere und obere Schranken 30
1.6 Minimierung von Schaltkreisen 42
1.7 Verzweigungsprogramme und ihre Komplexität 52
1.8 Schaltkreise versus Turing-Maschinen 56

2 Kolmogorov-Komplexität **65**

Literaturverzeichnis **77**

Kapitel 2

Kolmogorov-Komplexität

In diesem Abschnitt behandeln wir die minimale Komplexität zur Beschreibung eines Wortes.

Dabei sei V ein Alphabet mit mindestens zwei Buchstaben. Die Wörter sollen dann – wenn nicht ausdrücklich anders festgelegt – immer über V gebildet werden. Die in diesem Abschnitt betrachteten Turing-Maschinen sollen stets deterministisch sein.

Definition 2.1 *d-kk1 Die Komplexität $K_A(x)$ eines Wortes $x \in V^+$ bezüglich des Algorithmus A ist die Länge der kürzesten Eingabe $p \in \{0,1\}^+$ mit $A(p) = x$, d.h. in formalisierter Form*

$$K_A(x) = \min\{|p| \mid p \in \{0,1\}^+,\ A(p) = x\}.$$

Falls kein p mit $A(p) = x$ existiert, so setzen wir $K_A(x) = \infty$.

Wenn wir den Algorithmus A_{id} betrachten, der die identische Abbildung von $\{0,1\}^+$ in $\{0,1\}^+$ realisiert, so gilt wegen $A_{id}(x) = x$ und $A_{id}(y) \neq x$ für alle $y \neq x$ offenbar, $K_{A_{id}}(x) = |x|$.

Für ein Wort $a_1a_2 \dots a_m \in \{0,1,2,\dots,k-1\}^*$ definieren wir

$$val_k(a_1a_2 \dots a_m) = a_1k^{m-1} + a_2k^{m-2} + \dots + a_{m-2}k^2 + a_{m-1}k + a_m,$$

d.h. $a_1a_2 \dots a_m$ ist die k-äre Darstellung von $val_k(a_1a_2 \dots a_m)$. Es sei $A_{2,3}$ der Algorithmus, der $\{0,1\}^+$ in $\{0,1,2\}^+$ wie folgt abbildet. Für ein Wort $x_1x_2 \dots x_n \in \{0,1\}^+$ berechnen wir $m = val_2(x_1x_2 \dots x_n)$ Dann setzen wir

$$A_{2,3}(x_1x_2 \dots x_n) = y_1y_2 \dots y_k \in \{0,1,2\}^+,$$

wobei
— für $val_2(x_1x_2 \dots x_n) = 0$ die Beziehungen $y_1 = 0$ und $k = 0$ gelten,
— für $val_2(x_1x_2 \dots x_n) \neq 0$ die Beziehungen $y_1 \neq 0$ und $m = val_3(y_1y_2 \dots y_k)$ erfüllt sind.

Es sei $x = 120 \in \{0,1,2\}^+$ gegeben. Dann ergibt sich $val_3(120) = 15$. Damit gilt für jedes Wort 0^n1111 mit $n \geq 0$, dass es durch $A_{2,3}$ auf 120 abgebildet wird, $val_2(0^n1111) = 15$ ist. Folglich haben wir $K_{A_{2,3}}(x) = |1111| = 4$.

Analog erhalten wir für $y = 1021$ den Wert $val_3(1021) = 34$ und damit $K_{A_{2,3}}(y) = |100010| = 6$ da 100010 das kürzeste Wort z über $\{0,1\}$ mit $val_2(z) = 34$.

Falls das Wort x' mit 0 beginnt, aber von 0 verschieden ist, so gibt es kein Wort $z' \in \{0,1\}^+$, das durch $A_{2,3}$ auf x' abgebildet wird.

Allgemein erhalten wir

$$K_{A_{2,3}}(x) = \begin{cases} 1 & \text{für } x = 0, \\ \lceil \log_2(val_3(x)) \rceil & \text{für } x = ax',\ a \in \{1,2\},\ x' \in \{0,1,2\}^*, \\ \infty & \text{sonst} \end{cases}$$

Offensichtlich hängt die Komplexität $K_A(x)$ bez. A in entscheidendem Maße vom Algorithmus A ab. Wir wollen nun solche A suchen, die bis auf additive Konstanten eine minimale Größe der Komplexität liefern. Formalisiert wird dies durch die folgende Definition.

Definition 2.2 *Ein Algorithmus A_1 ist asymptotisch nicht schlechter als ein Algorithmus A_2, wenn es eine Konstante c_{A_2} so gibt, dass*

$$K_{A_1}(x) \leq K_{A_2}(x) + c_{A_2}$$

für alle $x \in V^*$ gilt.

Wir wollen zeigen, dass es Algorithmen gibt, die asymptotisch besser als jeder andere Algorithmus ist.

Eine Turing-Maschine $U = (X, Z, z_0, Q, \delta)$ (mit der Eingabemenge X, der Menge Z von Zuständen, dem Anfangszustand z_0, der Menge Q von Stoppzuständen und der Überführungsfunktion δ) heißt *universell*, wenn es auf einer Beschreibung einer beliebigen Turing-Maschine M und einer Eingabe w für M den Wert $f_M(w)$ berechnet, wobei f_m die von M induzierte Funktion ist.

Wir bemerken hier nur, dass es universelle Turing-Maschinen gibt. Dazu schreiben wir eine gegebene Turingmaschine

$$M = (\{x_1, x_2, \ldots x_n\}, \{z_0, z_1, \ldots, z_m\}, z_0, \{z_r, z_{r+1}, \ldots, z_m\}, \delta)'$$

ausführlich als

$$\begin{array}{l} *, x_1, x_2, \ldots, x_n; z_0, z_1, \ldots, z_m; z_r, z_{r+1}, \ldots, z_m; \\ (*, z_0, u_1, v_1, R_1), (*, z_1, u_2, v_2, R_2), \ldots, (*, z_{r-1}, u_r, v_r, R_r), \\ (x_1, z_0, u_{r+1}, v_{r+1}, R_{r+1}), (x_1, z_1, u_{r+2}, v_{r+2}, R_{r+2}), \ldots, (x_1, z_{r-1}, u_{2r}, v_{2r}, R_{2r}), \\ \quad \ldots \qquad \ldots \\ (x_n, z_0, u_{nr+1}, v_{nr+1}, R_{nr+1}), \ldots, (x_n, z_{r-1}, u_{n(r+1)}, v_{n(r+1)}, R_{n(r+1)}), \end{array}$$

wobei $*$ das Blanksymbol bezeichnet und für ein Eingabesymbol x und einen Zustand z das Tupel (x, z, u, v, R) durch $\delta'(x, z) = (u, v, R)$ festgelegt ist. Damit wird die Turing-Maschine M durch ein Wort über $B = \{*, x_1, x_2, \ldots, x_n, z_0, z_1, \ldots, z_m, R, L, N, (,), ;\} \cup \{,\}$ beschrieben. Auf diesem Wort kann U immer nachschauen, welche Aktion bei Eingabe von w zu welchem Zeitpunkt auszuführen ist, und diese Aktion dann auf der Eingabe ausführen.

Im Folgenden sprechen wir von universellen Algorithmen, wenn sie durch universelle Turing-Maschinen gegeben sind.

Satz 2.1 *Es sei U ein universeller Algorithmus und $x \in V^*$. Dann gilt*

$$K_U(x) \leq K_A(x) + c_A$$

für jeden Algorithmus A, wobei c_A eine nur von A (und nicht von x) abhängende Konstante ist (d.h. U ist asymptotisch nicht schlechter als jeder andere Algorithmus A).

Beweis. Es seien U ein universeller Algorithmus, A ein beliebiger Algorithmus und s_A die Beschreibung von A, die U neben p erhalten muss um $A(p)$ zu berechnen. Es sei c_a die Länge der Beschreibung s_A. Aus der Eingabe $s_a p$ der Länge $c_a * |p|$ berechnet U den Wert $A(p)$.

Es sei nun p ein Wort kürzester Länge mit $A(p) = x$. Dann gelten $K_A(x) = |p|$ und $U(s_a p) = A(p) = x$. Damit erhalten wir

$$K_U(x) = \min\{|y| \mid U(y) = x\} \leq |s_a p| = |p| + c_A = K_A(x) + c_A.$$

□

Aus Satz 2.1 folgt sofort, dass für zwei universelle Algorithmen U_1 und U_2 eine Konstante c mit

$$|K_{U_1}(x) - K_{U_2}(x)| \leq c$$

existiert. Die Kolmogorov-Komplexitäten bez. der beiden universellen Algorithmen unterscheiden sich also höchstens um eine Konstante. Daher ist es im Folgenden nicht wesentlich, welchen universellen Algorithmus wir wählen.

Definition 2.3 *Es sei U ein (fest gewählter) universeller Algorithmus. Dann definieren wir die Kolmogorov-Komplexität $K(X)$ durch $K(x) = K_U(x)$.*

Entsprechend der Definition ist die Kolmogorov-Komplexität für Wörter über V definiert. Wenn $\#(V) = \{0, 1, 2, \ldots k_1\}$, so können wir eine natürliche Zahl m durch ihre k-äre Darstellung $d_k(m)$, d.h. durch ein Wort über V, angeben. Die Kolmogorov-Komplexität $K(m)$ von m kann daher als Kolmogorov-Komplexität von $d_k(m)$ definiert werden, d.h.

$$K(m) = K(d_k(m)).$$

Die Länge des Wortes $d_k(m)$ bezeichnen wir auch mit $|m|$. Dann gilt

$$|m| = \lceil \log_k(m) \rceil.$$

Wir geben nun einige Eigenschaften der Kolmogorov-Komplexität an.

Lemma 2.2 *Es sei $V = \{0, 1\}$.*

1. *Es gibt eine Konstante c derart, dass für alle $x \in \{0, 1\}^+$ die Beziehung*

$$K(x) \leq |x| + c$$

gilt.

2. *Für jede natürliche Zahl n gibt es eine Konstante c derart, dass*

$$2^{n-c} \leq \#(\{x \mid x \in \{0,1\}^+, K(x) \leq n\}) \leq 2^{n+1}$$

gilt.

3. *Für jede berechenbare Funktion f gibt es eine Konstante c_f derart, dass für alle x, für die $f(x)$ definiert ist,*

$$K(f(x)) \leq k(x) + c_f$$

gilt.

4. *Für $n \in \mathbf{N}$ sei V_n eine endliche Menge mit maximal 2^n Elementen. Ferner sei die Menge $\{(x,n) \mid x \in V_n\}$ rekursiv aufzählbar. Dann existiert eine Konstante c derart, dass dass $K(x) \leq n + c$ für alle $n \in \mathbf{N}$ und alle Elemente $x \in V_n$ gilt.*

Beweis. 1. Da die Kolmogorov-Komplexität auf einem universellen Algorithmus beruht, erhalten wir für den Algorithmus A_{id} aus Lemma 2.1

$$K(X) \leq A_{id}(x) + c_{A_{id}} = |x| + c_{A_{id}}$$

und damit die gewünschte Aussage mit $c = c_{A_{id}}$.

3. Es sei

$$A_n = \{x \mid K(x) \leq n\}.$$

Offenbar kann A_n nicht mehr Wörter enthalten als es Wörter der Länge $\leq n$ über $\{0,1\}$ gibt. Beachten wir noch, dass es 2^i Wörter der Länge i über $\{0,1\}$ gibt, so ergibt sich

$$\#(A_n) \leq 2^0 + 2^1 + 2^2 + \ldots 2^{n-1} + 2^n = 2^{n+1} - 1 < 2^{n+1}.$$

Ferner erhalten wir aus der ersten Aussage, dass für eine gewisse Konstante c alle Wörter der Länge $n-c$ eine Kolmogorov-Komplexität haben, die höchstens $n-c+c = n$ ist. Damit gibt es mindestens 2^{n-c} Wörter, deren Komplexität $\leq n$ ist. Folglich ist $2^{n-c} \leq \#(A_n)$.

4. Es sei U der universelle Algorithmus mit $K_U(x) = K(x)$. Da f berechenbar ist, gibt es eine Beschreibung s_f der Funktion f, dass U auf der Eingabe $s_f x$ den Wert $f(x)$ berechnet, d.h. es gilt $U(s_f x) = f(x)$. Es sei nun p ein kürzestes Wort mit $U(p) = x$ und damit auch $K(x) = |p|$. Wir verwenden nun die Eingabe $s_f p$ für U. Dann gibt es einen Algorithmus A, der bei der Eingabe s_f zuerst analog zu U das Wort p in x überführt, d.h. wir erhalten $s_f x$, und dann erneut analog zu U das Wort $s_f x$ in $f(x)$ überführt. Dann gilt $K_A(f(x)) \leq |s_f| + |p|$ da $s_f p$ nicht ein kürzestes Wort sein muss, dass durch A in $f(x)$ überführt wird. Beachten wir noch Lemma 2.1, so erhalten wir

$$K(f(x)) = K_U(f(x)) \leq K_A(f(x) + c_A \leq |p| + |s_f| + c_A = K(x) + |s_f| + c_A$$

und damit die Aussage mit $c = |s_f| + c_A$.

5. Bei der Aufzählung der Elemente (y,m) sei für gegebenes n und $x \in V_n$ das Paar (x,n) das i-te Element mit zweiter Komponente n. Dann ist $i \leq 2^n - 1$ (wenn wir die Zählung bei 0 beginnen. Damit gibt es $a_j \in \{0,1\}$, $0 \leq j \leq n-1$, mit $\sum_{i=0}^{n-1} a_j 2^j = i$. Wir betrachten nun den Algorithmus A, der bei der Eingabe von $a_{n-1}a_{n-2}\ldots a_0$ (die sowohl n

als auch i als enthält) die Aufzählung der Elemente (y, m) vornimmt und x ausgibt, wenn (x, n) das i-te Paar mit zweiter Komponente n ist. Dann gilt $A(a_{n-1}a_{n-2}\dots a_0) = x$ und somit $K_A(x) \leq n$. Unter Verwendung von Lemma 2.2 erhalten wir

$$K(X) \leq K_A(x) + c_A \leq n + c_A$$

für jedes $x \in V_n$. □

Wir bemerken, dass „fast alle“ Kolmogorov-Komplexitäten von Wörter der Länge n über $\{0, 1\}$ in einem Intervall von $n - c$ bis $n + c$ liegen, wobei c eine passende Konstante ist. Nach der ersten Aussage von Lemma 2.2 erfüllt jedes Wort der Länge n die Relation $K(x) \leq n+c'$ für eine Konstante c'. Es sei nun $c \geq c'$. Dann gilt erst recht $K(x) \leq n+c$ für alle x der Länge n. Die Anzahl der Wörter mit $K(x) < n - c$ ist sicher durch die Anzahl der Wörter über $\{0, 1\}$ mit einer Länge $< n - c$, d.h. durch 2^{n-c} beschränkt. Damit ist der Anteil der Wörter der Länge n mit $n - c \leq K(x) \leq n + c$ bezogen auf alle Wörter der Länge n mindestens

$$\frac{2^n - 2^{n-c}}{2^n} = 1 - 2^{-c}.$$

Für hinreichend großes c liegt der Anteil dann nahe 1.

Lemma 2.3 *Für jede berechenbare Funktion $h : V^+ \to \mathbf{N}$ mit $h(x) \leq K(x)$ für alle $x \in V^+$ gibt es eine Konstante C derart, dass $h(x) \leq C$ für alle $x \in V^*$ gilt.*

Beweis. Ohne Beschränkung der Allgemeinheit sei $V = \{0, 1, 2, \dots, k-1\}$ (für ein k). Es sei h eine Funktion mit den geforderten Eigenschaften. Angenommen, dass keine konstante obere Schranke C für h existiert, dann gibt es zu jeder Zahl m ein Wort x_m mit

$$m < h(x_m) \leq K(x_m).$$

Zur k-ären $d_k(m)$ für ein gegebenes m kann ein Wort x_m mit $m < h(x_m) \leq K(x_m)$ sogar berechnet werden. Dazu zählen wir alle Wörter über V auf und berechnen dann der Reihe nach die Werte $h(x)$, bis ein x mit $h(x) < m$ gefunden wird. Es sei nun $f : V^+ \to V^+$ die Funktion, die $d_k(m)$ das so berechnete Wort x_m zuordnet. Wir erhalten

$$m < k(x_m) = k(f(d_k(m))) \leq K(f(d_k(m))). \tag{2.1}$$

Da f berechenbar ist, gibt es nach der dritten Aussage von Lemma 2.2 eine Konstante c_f mit

$$K(f(d_k(m))) \leq K(d_k(m)) + c_f. \tag{2.2}$$

Nach der ersten Aussage von lemma 2.2 haben wir noch

$$K(d_k(m)) \leq \lceil |d_k(m)| + c = \log_k(m) \rceil + c \tag{2.3}$$

für eine konstante c'. Aus (2.1), (2.2) und (2.3) erhalten wir

$$m < K(f(d_k(m))) \leq \lceil \log_k(m) \rceil + c + c_f < \log_k(m) + c'$$

mit der Konstanten $c' = c + c_f + 1$. Dies ist für große m aber offensichtlich nicht möglich. Folglich ist unsere Annahme falsch und es gibt eine obere Schranke für h. □

Folgerung 2.4 *Die Kolmogorov-Komplexität ist keine berechenbare Funktion.*

Beweis. Wir nehmen an, dass die Kolmogorov-Komplexität berechenbar sei. Da $K(x) \leq K(x)$ für alle x gilt, erhalten wir aus Lemma 2.3 eine konstante obere Schranke C für die Werte von $K(x)$. Dies widerspricht der oben festgestellten Aussage dass es nur endlich viele Wörter gibt, für die die Kolmogorov-Komplexität kleiner als C ist. □

Für die Angabe einer weiteren Eigenschaft der Kolmogorov-Komplexität benötigen wir die folgende Definition.

Definition 2.4 *Eine Funktion $F : V^+ \to \mathbf{N}$ heißt von oben rekursiv aufzählbar, falls es eine totale berechenbare Funktion $k : V^+ \times \mathbf{N} \to \mathbf{N}$ gibt, für die*

$$k(x,0) \geq k(x,1) \geq k(x,2) \geq \ldots \text{ und } F(x) = \lim_{n\to\infty} k(x,n)$$

für alle $x \in V^+$ gelten.

Da die Werte $k(x,m)$ ganzzahlig sind, gibt es offensichtlich für jedes $x \in V^+$ eine natürliche Zahl n_x derart, dass $k(x,n) = F(x)$ für alle $n \geq n_x$ gilt.

Lemma 2.5 *Die Kolmogorov-Komplexität ist von oben rekursiv aufzählbar.*

Beweis. Es sei U der universelle Algorithmus, der zur Definition der Kolmogorov-Komplexität verwendet wird. Wir definieren nun die Funktion k im Wesentlichen dadurch, dass $k(x,n)$ die Länge des kürzesten Wortes $y \in \{0,1\}^+$ ist, das durch höchstens n Schritte in x überführt werden kann. Das „im Wesentlichen“ liegt darin begründet, dass wir für kleine n nicht absichern können, dass x durch U in höchstens n Schritten überhaupt berechnet werden kann. Deshalb nutzen wir für diese Fälle die Schranke aus der ersten Aussage von Lemma 2.2. Daher setzen wir

$$k(x,n) = \min\{|x| + c,\ \min\{|y| \mid U \text{ berechnet } x \text{ bei Eingabe von } y \text{ in} \leq n \text{ Schritten}\}.$$

Wegen der Forderung, dass höchstens n Schritte verwendet werden dürfen folgt sofort $k(x,n-1) \geq k(x,n)$. Da dbei der Definition der Kolmogorov-Komplexität keibe Beschränkung der Schrittzahl vorgenommen wird, gilt auch noch $\lim_{n\to\infty} k(x,n) = K(x)$. Damit erfüllt das so konstruierte k die geforderten Eigenschaften, und K ist als von oben rekursiv aufzählbar nachgewiesen. □

Das folgende Lemma gibt eine Art Umkehrung von Lemma 2.5 an, denn es besagt, dass jede von oben rekursiv aufzählbare Funktion (mit einer gewissen Eigenschaft) bis auf eine additive Konstante die Kolmogorov-Komplexität ist.

Lemma 2.6 *Es seien $K' : \{0,1\}^+ \in \mathbf{N}$ eine von oben rekursiv aufzählbare Funktion und C eine Konstante, für die $\#(\{x \mid K'(x) \leq n\}) \leq C2^n$ für alle n erfüllt ist. Dann gibt es eine Konstante c derart, dass $K(x) \leq K'(x) + c$ für alle $x \in \{0,1\}^+$ gilt.*

Beweis. Da K' von oben rekursiv aufzählbar ist, gibt es eine Funktion k' mit

$$k'(x,0) \geq k'(x,1) \geq k'(x,2) \geq \ldots \text{ und } K'(x) = \lim_{n\to\infty} k'(x,n)$$

für alle $x \in \{0,1\}^+$. Ferner sei

$$W_n = \{x \mid x \in \{0,1\}^+, K'(x) < n\}.$$

Wir zeigen jetzt, dass die Relation $R\{(x,n) \mid x \in W_n\}$ aufzählbar ist. Die Aufzählung kann nämlich wie folgt geschehen: Entsprechend einer Aufzählung der Menge $\{(x,m) \mid x \in \{0,1\}^+, m \in \mathbf{N}\}$[1] werden der Reihe nach die Werte $k'(x,m)$ ermittelt, und (x,n) wird zur Aufzählung von R hinzugefügt, wenn $k'(x,m) < n$ gilt. Wegen $K'(x) \leq k'(x,m)$ für $m \in \mathbf{N}_0$ und $\lim_{m\to\infty} k'(x,m) = K'(x)$ werden genau die Elemente von R aufgezählt.

Wir setzen

$$n' = n + \lceil \log_2(C) \rceil \quad \text{und} \quad V_{n'} = W_n.$$

Dann gilt

$$\#(V_{n'} \leq C2^n \leq 2^{n'}.$$

Aus der vierten Aussage von Lemma 2.2 erhalten wir

$$K(x) \leq n' + c' = n + \lceil \log_2(C) \rceil + c' \tag{2.4}$$

mit einer gewissen Konstanten c' für alle $x \in V_{n'} = W_n$.

Es sei nun $n = K'(x) + 1$. Dann gilt $x \in W_n$. Aus (2.4) ergibt sich daher

$$K(x) \leq n + \lceil \log_2(C) \rceil + c' = K'(x) + 1 + \lceil \log_2(C) \rceil + c' = K'(x) + c$$

mit $c = 1 + \lceil \log_2(C) \rceil + c'$. □

Der folgende Satz besagt, dass die Kolmogorov-Komplexität durch einige ihrer oben angegebenen Eigenschaften charakterisiert wird.

Satz 2.7 *Es sei $K' : \{0,1\}^* \to \mathbf{N}$ eine Funktion mit folgenden Eigenschaften:*

1. *Für jede berechenbare Funktion f gibt es eine Konstante c_f derart, dass*

$$K'(f(x)) \leq K'(x) + c_f$$

   für alle x gilt, für die $f(x)$ definiert ist.

2. *Die Funktion K' ist von oben rekursiv aufzählbar.*

3. *Es existieren Konstanten c und C derart, dass*

$$c2^n \leq \#(\{x \mid x \in \{0,1\}^+,\ K'(x) < n\}) \leq C2^n$$

   für alle $n \in \mathbf{N}$ erfüllt ist.

Dann unterscheidet sich K' von der Kolmogorov-Komplexität höchstens um einen konstanten additiven Term.

[1] Eine solche Aufzählung kann analog zur Aufzählung der positiven rationalen Zahlen, d.h. der Paare (a,b) mit $a,b \in \mathbf{N}$, konstruiert werden.

Beweis. Aus den Eigenschaften 2 und 3 und Lemma 2.6 erhalten wir sofort

$$K(x) \leq K'(x) + \overline{c} \tag{2.5}$$

für eine gewisse Konstante $\overline{c}$.

Da K' nach Eigenschaft 2 von oben rekursiv aufzählbar ist, können wir wie im Beweis von Lemma 2.6 alle Wörter $x \in \{01\}^+$ mit $K'(x) < n$ konstruieren. Wegen Eigenschaft 3 gibt es davon mindestens $c2^n$ Wörter. Wir wählen (die Konstante) d nun minimal mit der Eigenschaft $2^{n-d} \leq c2^n$. Für die natürlichen Zahlen erzeugen wir entsprechend ihrer natürlichen Ordnung der Reihe nach die Mengen S_n von Elementen mit $k'(x) < n$, wobei wir abbrechen, wenn 2^{n-d} Elemente erhalten haben. Dann gilt offensichtlich

$$S_1 \subset S_2 \subset S_3 \subset \dots \ .$$

Für $i \in \mathbf{N}$ setzen wir $T_i = S_{i+1} \setminus S_i$. Nach Konstruktion sind die Mengen T_i, $i \in \mathbf{N}$, paarweise disjunkt. Außerdem gilt

$$\#(T_i) = \#(S_{i+1}) - \#(S_i) = 2^{(i+1)-d} - 2^{i-d} = 2^{i-d}.$$

Da $T_i \subseteq S_{i+1}$ gilt, gilt

$$K'(x) < n+1 \text{ für alle } x \in T_n. \tag{2.6}$$

Wir betrachten nun eine beliebige Funktion f, die die 2^{n-d} Wörter aus T_n eineindeutig auf die 2^{n-d} Wörter über $\{0,1\}$ der Länge $n-d$ abbildet. Wegen der Eigenschaft 1 erhalten wir

$$K'(f(x)) \leq K'(x) + c_f \text{ für alle } x \in T_n. \tag{2.7}$$

Betrachten wir nun $y \in \{0,1\}^+$ mit $|y| = n-d$, so gibt es ein x mit $f(x) = y$. Für ein solches y gilt dann wegen (2.7) und (2.6)

$$K'(y) = K(f(x)) \leq K'(x) + c_f < n + 1 + c_f = |y| + d + 1 + c_f. \tag{2.8}$$

Da $n-d$ alle natürlichen Zahlen durchläuft, gilt (2.8) sogar für alle y.

Es sei U der universelle Algorithmus, auf dem die Kolmogorov-Komplexität beruht. Für $x \in \{0,1\}^+$ sei p_x ein kürzestes Wort aus $\{0,1\}^+$ mit $U(p_x) = x$. Dann gilt $K(x) = |p_x|$. Wegen der Eigenschaft 1 und (2.8) erhalten wir

$$K'(x) = K'(U(p_x)) \leq K'(p_x) + c_U \leq |p_x| + d + 1 + c_f + c_U = K(x) + d + 1 + c_f + c_U = K(x) + c \tag{2.9}$$

mit einer Konstanten $c = d + 1 + c_f + c_U$.

Aus (2.5) und (2.9) folgt die Behauptung. □

Wir wollen nun eine Anwendung der Kolmogorov-Komplexität auf ein zahlentheoretisches Problem geben und dadurch die Brauchbarkeit des Konzepts demonstrieren.

Bereits im Altertum (z.B. in den „Elementen“ Euklids, ca. 300 v.Chr.) war bekannt, dass es unendlich viele Primzahlen gibt. Dagegen war es offen, wie die Primzahlen verteilt sind, oder anders gefragt, wie viele Primzahlen kleiner als eine vorgegebene natürliche Zahl sind. Wir definieren $\pi(n)$ als die Anzahl der Primzahlen p mit $p \leq n$. In den Jahren 1795 bzw. 1798 vermuteten C. F. Gauss (1777–1855) und A.-M. Legendre (1752–1833), dass $\pi(n)$ angenähert durch $n/\ln(n)$ beschrieben wird, wobei ln der natürliche

Logarithmus (zur Basis e) ist. Der russische Mathematiker P. L. Tschebycheff (1821–1894) zeigte 1851, dass

$$0.929 \leq \frac{\pi(n)}{n/\ln(n)} \leq 1,106 \ .$$

Im Jahre 1896 bewiesen der französische und der belgische Mathematiker J. S. Hadamard (1865–1963) und Ch. J. de la Vallee Poussin (1866–1962) die folgende – heute meist Primzahlsatz genannte Aussage.

Satz 2.8 *(Primzahlsatz)*

$$\lim_{n\to\infty} \frac{\pi(n)}{n/\ln(n)} = 1 \ .$$

Alle bekannten Beweise für den Primzahlsatz sind sehr kompliziert und lang und verwenden in der Regel viele Vorkenntnis aus der Zahlentheorie.

Wir wollen jetzt mittels der Kolmogorov-Komplexität einen (einfachen) Beweis für eine nur etwas schwächere Form des Primzahlsatzes geben. Dafür benötigen wir die folgende Aussage.

Lemma 2.9 *Es sei $n_1, n_2, N-3, \ldots$ eine unendliche Folge natürlicher Zahlen mit den Eigenschaften*

$$n_i \leq n_{i+1} \quad \textit{und} \quad K(n_i) \geq \frac{\lceil \log_2(n_i) \rceil}{2},$$

$n \geq 1$. Weiterhin sei q_i, $i \geq 1$, die größte Primzahl, die die Zahl n_i teilt. Dann ist die Menge $Q = \{q_i \mid i \geq 1\}$ unendlich.

Beweis. Wir nehmen an, dass Q endlich ist (und werden einen Widerspruch herleiten). Es seien p die größte Primzahl in Q und $p_1, p_2, \ldots p_m$ die Primzahlen mit $p_i \leq p$. Dann hat jede Zahl n_i eine Primzahldarstellung

$$n_i = p_1^{r_{i,1}} p_2^{r_{i,2}} \ldots p_m^{r_{m,1}}$$

mit gewissen Zahlen $r_{j,i} \in \mathbf{N}_0$. Daher existiert ein Algorithmus A, der für Eingabe $(r_{i,1}, r_{i,2}, \ldots, r_{i,m})$ die Binärdarstellung von n_i berechnet. Dann gibt es nach der ersten Aussage von Lemma 2.2 eine Konstante c mit

$$K(n_i) \leq \lceil \log_2(r_{i,1}) \rceil + \lceil \log_2(r_{i,2}) \rceil + \ldots + \lceil \log_2(r_{i,m}) \rceil + c.$$

Wegen $\log_2(r_{i,j}) \leq \log_2(n_i)$ liefert dies

$$K(n_i) \leq m \cdot \lceil \log_2(\log_2(n_i)) \rceil + c.$$

Aufgrund der Voraussetzung $K(n_i) \geq \frac{\lceil \log_2(n_i) \rceil}{2}$ erhalten wir

$$\frac{\lceil \log_2(n_i) \rceil}{2} \leq K(n_i) \leq m \cdot \lceil \log_2(\log_2(n_i)) \rceil + c$$

für alle $i \geq 1$. Dies ist aber unmöglich, da m und c konstante Werte sind. □

Satz 2.10 *Für unendlich viele $n \in \mathbf{N}$ gilt*

$$\frac{n}{32\log_2(n)\cdot(\log_2(\log_2(n)))^2} \leq \pi(n) \ .$$

Beweis. Mit $bin(n)$ bezeichnen wir die Binärdarstellung von n, d.h. $bin(n) = d_2(n)$. Es sei p_j die j-te Primzahl entsprechend der natürlichen Ordnung der natürlichen Zahlen.

Es seien $n \in \mathbf{N}$ und p_m die größte Primzahl, die n teilt. Dann lässt sich n einfach aus p_m und $\frac{n}{p_m}$ einfach durch Multiplikation gewinnen. Wegen der Existenz einer Aufzählung aller Primzahlen, kann n auch aus den Zahlen m und $\frac{n}{p_m}$ bestimmt werden, indem man aus der Aufzählung und m die Primzahl p_m ermittelt und dann mit $\frac{n}{p_m}$ multipliziert. Als Eingabe ist das Wort $bin(m)bin(\frac{n}{p_m})$ aber nicht verwendbar, da keine Information vorliegt, wo $bib(m)$ aufhört. Deshalb setzen wir für k mit der Binärdarstellung $bin(k) = a_1a_2\ldots a_{\lceil\log_2(k)\rceil}$

$$Bin(m) = a_10a_20\ldots a_{\lceil\log_2(k)\rceil-1}0a_{\lceil\log_2(k)\rceil}1.$$

Wir betrachten nun $Bin(n)bin(\frac{n}{p_m})$. Dieses Wort ermöglicht die Berechnung von n aus m und $\frac{n}{p_m}$, da die erste 1 an einer geraden Position anzeigt, dass die Binärdarstellung von m gerade aus den Buchstaben davor an ungerader Position gebildet wird. Es gilt

$$|Bin(m)bin(\frac{n}{p_m})| = 2\lceil\log_2(m)\rceil + \lceil\log_2(\frac{n}{p_m})\rceil.$$

Wir verkürzen diese Eingabe zur Berechnung von n noch weiter (die Länge der kürzesten Eingabe liefert die Kolmogorov-Komplexität). Dazu betrachten wir

$$w = Bin(\lceil\log_2(m)\rceil)bin(m)bin(\frac{n}{p_m}).$$

Tatsächlich ermöglicht $Bin(\lceil\log_2(m)\rceil)$ die Kenntnis, dass die ersten $\lceil\log_2(m)\rceil$ Positionen nach der ersten 1 an gerader Position gerade $bin(m)$ sind und die restlichen Werte dann $bin(\frac{n}{p_m})$ angeben. Offenbar gilt

$$|w| = 2\cdot\lceil\log_2(\lceil\log_2(m)\rceil)\rceil + \lceil\log_2(m)\rceil + \lceil\log_2(\frac{n}{p_m})\rceil.$$

Dieser Prozess kann beliebig weiter iteriert werden. Wir führen nur noch einen durch. Wir betrachten also

$$v(m,\frac{n}{p_m}) = Bin(\lceil\log_2(\lceil\log_2(m)\rceil)\rceil)bin(\lceil\log_2(m)\rceil)bin(m)bin(\frac{n}{p_m})$$

mit der Länge

$$|v(m,\frac{n}{p_m})| = 2\cdot\lceil\log_2(\lceil\log_2(\lceil\log_2(m)\rceil)\rceil)\rceil + \lceil\log_2(\lceil\log_2(m)\rceil)\rceil + \lceil\log_2(m)\rceil + \lceil\log_2(\frac{n}{p_m})\rceil. \tag{2.10}$$

Wir betrachten nun die Zahlen n mit $2^i \leq n \leq 2^{i+1}-1$. Dies sind 2^i Zahlen. Jede dieser Zahlen hat eine Binärdarstellung der Länge $i+1$. Nach der zweiten Aussage von Lemma 2.2 gibt es höchsten 2^{i-2} Zahlen mit einer Kolmogorov-Komplexität $\leq (i-3)$. Damit gilt für mehr die Hälfte der Zahlen n mit $2^i \leq n \leq 2^{i+1}-1$ die Beziehung $K(n) \geq \lceil\log_2(n)\rceil - 2$.

Da die Länge von $v(m, \frac{n}{p_m})$ den Term $\lceil \log_2(\frac{n}{p_m}) \rceil = \lceil \log_2(n) - \log -2(p_m) \rceil$ enthält und $m < p_m$ gilt, ist es leicht zu sehen, dass auch für mehr als die Hälfte der Zahlen n mit $2^i \leq n \leq 2^{i+1} - 1$ die Beziehung $|v(m, \frac{n}{p_m})| \geq \lceil \log_2(n) \rceil - 1$ gilt.

Somit gibt es für jedes $i \in \mathbf{N}$ eine Zahl n_i mit

$$2^i \leq n_i \leq 2^{i+1} - 1,\ K(n_i) \geq \lceil \log_2(n) \rceil - 2 \text{ und } |v(m, \frac{n_i}{p_m})| \geq \lceil \log_2(n_i) \rceil - 2. \qquad (2.11)$$

Ausführlich lautet die letzte der Beziehungen wegen (2.10)

$$\lceil \log_2(n_i) \rceil - 1 \leq 2 \cdot \lceil \log_2(\lceil \log_2(\lceil \log_2(m) \rceil) \rceil) \rceil + \lceil \log_2(\lceil \log_2(m) \rceil) \rceil + \lceil \log_2(m) \rceil + \lceil \log_2(\frac{n_i}{p_m}) \rceil,$$

woraus

$$\begin{aligned} \lceil \log_2(p_m) \rceil &\leq 2 \cdot \lceil \log_2(\lceil \log_2(\lceil \log_2(m) \rceil) \rceil) \rceil + \lceil \log_2(\lceil \log_2(m) \rceil) \rceil + \lceil \log_2(m) \rceil + 1 \\ &\leq 2 \cdot \log_2(\log_2(\log_2(m))) + \log_2(\log_2(m)) + \log_2(m) + 5 \end{aligned}$$

folgt (wobei die 5 dadurch entsteht, dass wir jedes Mal nicht zur nächsten Zahl übergehen und dadurch eine Verkleinerung um 4 eintreten kann), woraus sich

$$p_m \leq 32 \cdot m \cdot log_2(m) \cdot (\log_2(\log_2(m)))^2 \qquad (2.12)$$

ergibt.

Nach (eq-kk-prim2) erfüllt die Folge $n_1, n_2, \ldots$ die Voraussetzungen des Lemmas 2.9. Damit gilt (2.12) für unendlich viele m. Somit gibt es unter den ersten $32 \cdot m \cdot log_2(m) \cdot (\log_2(\log_2(m)))^2$ natürlichen Zahlen mindestens m Primzahlen, d.h.

$$\pi(32 \cdot m \cdot log_2(m) \cdot (\log_2(\log_2(m)))^2) \geq m.$$

Wir setzen nun $n = 32 \cdot m \cdot log_2(m) \cdot (\log_2(\log_2(m)))^2$. Es ergibt sich

$$\pi(n) \geq m = \frac{n}{32 \cdot log_2(m) \cdot (\log_2(\log_2(m)))^2}.$$

Wegen $n \geq m$ ergibt sich

$$\pi(n) \geq \frac{n}{32 \cdot log_2(n) \cdot (\log_2(\log_2(n)))^2}$$

und damit die gewünschte Aussage. □