3. Binäre Zufallsfolgen

Literatur:

Ming Li, Paul Vitányi, An Introduction to Kolmogorov Complexity and Its Applications, 2nd. edn., Springer 1997.

Christian S. Calude, Information and Randomness: An Algorithmic Perspective, 2nd edn., Springer 2002.

A.N. Kolmogorov, V.A. Uspenskii, Algorithms and Randomness, SIAM J. on Theory of Probability and its Applications 32 (1987), 389 - 412.

Experiment:
Folge von Münzwürfen, Kopf $\hat{=}$ 0, Zahl $\hat{=}$ 1
Münze ist fair und wird auch fair geworfen

Wahrscheinlichkeitstheorie $\Rightarrow$
Nach 100 Münzwürfen tritt jede der 2^{100} möglichen Binärzahlen mit derselben Wahrscheinlichkeit, nämlich 2^{-100}, auf.

Frage:
Würden wir 000...0 als zufällige Zahl akzeptieren?

Antwort:
nein, da nicht Anzahl Einsen $\approx$ Anzahl Nullen.

Frage:
Würden wir 101010...10 als zufällige Zahl akzeptieren?

Antwort:
nein, da Struktur zu regelmäßig.

Beobachtung:
- Beide Beispiele lassen Beschreibungen zu, die wesentlich kürzer sind, als die Binärzahl selbst.
- Je schwieriger eine Bitfolge zu beschreiben ist, umso eher akzeptieren wir diese als zufällige Folge.
- Auch würden wir von einer Zufallsfolge erwarten, dass diese jeder "sinnvollen" Majorität angehört.

Ziel:
Formale Charakterisierung von binären Zufallsfolgen.

3.1 Unendliche Zufallsfolgen

Grundlagen aus der Wahrscheinlichkeits und Maßtheorie:

Betrachten wir das Zufallsexperiment "Wurf eines Würfels". Das Ergebnis dieses Zufallsexperimentes ist eine Zahl aus $\Omega = \{1,2,3,4,5,6\}$.

Ω heißt Ergebnisraum des Zufallsexperimentes.

Ein Ereignis kann z.B.

- Werfen eines Sechs,
- Werfen einer Primzahl oder
- Werfen einer Zahl größer als zwei

sein. Jedes dieser Ereignisse läßt sich als Teilmenge von Ω beschreiben, nämlich

$$\{6\}, \{2,3,5\} \text{ bzw. } \{3,4,5,6\}.$$

Jede Teilmenge von Ω, insbesondere auch $\emptyset$ und Ω, kann als Ereignis aufgefasst werden. Dabei sind

$\emptyset$ das unmögliche Ereignis und
Ω das sichere Ereignis.

Die Menge aller Ereignisse bzgl. Ω ist die Potenzmenge 2^Ω von Ω. Sei $A \in 2^\Omega$ ein Ereignis. Dann gilt:

- Je umfangreicher A, umso wahrscheinlicher tritt das Ereignis A ein.

Die Potenzmenge 2^Ω bildet mit den Operationen Durchschnitt, Vereinigung und Komplementbildung einen Boolesche Verband.

Häufig genügt es, anstatt ganz 2^Ω, Mengensysteme $F \subseteq 2^\Omega$, die geeignete Eigenschaften besitzen, zu betrachten.

Ein Mengensystem $F \subseteq 2^\Omega$ heißt <u>Ereignisalgebra über Ω</u>, falls F folgende Eigenschaften besitzt:

(E1) $\emptyset \in F$ und $\Omega \in F$.

(E2) Für alle höchstens abzählbare Indexmengen I gilt

$$A_i \in F \ \forall i \in I \Rightarrow \bigcup_{i \in I} A_i \in F \text{ und } \bigcap_{i \in I} A_i \in F$$

(E3) $A, B \in F \Rightarrow A \setminus B \in F$

Jede Ereignisalgebra ist ein Boolescher Verband.

Übung:
Zeigen Sie, dass eine Ereignisalgebra ein Boolescher Verband ist.

Sei F eine Ereignisalgebra. Eine Funktion $P: F \to \mathbb{R}_0^+$ heißt <u>Wahrscheinlichkeitsmaß auf F</u>, falls folgende Axiome erfüllt sind:

(K1) $P(\Omega) = 1$

(K2) Für höchstens abzählbare Indexmengen I und paarweise disjunkte A_i, $i \in I$ gilt

$$P\left(\bigcup_{i \in I} A_i\right) = \sum_{i \in I} P(A_i)$$

(K1) und (K2) heißen <u>Kolmogorov-Axiome</u>.

Ziel:

Herleitung einer präzisen Charakterisierung der Menge der zufälligen unendlichen Folgen von Nullen und Einsen.

Seien hierzu

Ω die Menge aller unendlichen Folgen von Nullen und Einsen. Für $x \in \{0,1\}^*$ sei

$$\Omega_x := \{ z \in \Omega \mid x \text{ ist Präfix von } z \}.$$

Falls wir ein Element von Ω zufällig wählen, dann erwarten wir, dass dieses Element jedes "sinnvollen" Majorität angehört.

$\rightsquigarrow$

Frage: Was sind sinnvolle Majoritäten?

Zur Beantwortung dieser Frage definieren wir zunächst ein Wahrscheinlichkeitsmaß auf Ω.

Maßtheorie $\Rightarrow$

Ein Wahrscheinlichkeitsmaß auf Ω kann aus den Werten auf den Mengen Ω_x, $x \in \{0,1\}^*$ rekonstruiert werden. Also genügt es, die Werte auf den Mengen Ω_x, $x \in \{0,1\}^*$ zu definieren.

$\rightsquigarrow$

$$P(\Omega_x) := 2^{-|x|}.$$

Eine Menge $X \subset \Omega$ heißt <u>Nullmenge</u>, falls es für jedes $\varepsilon > 0$ eine Folge $x_0, x_1, x_2, \ldots$ von binären Strings gibt mit

i) $X \subset \Omega_{x_0} \cup \Omega_{x_1} \cup \Omega_{x_2} \cup \ldots$ und

ii) $\sum_i 2^{-|x_i|} < \varepsilon$.

$X \subseteq \Omega$ heißt <u>Einsmenge</u>, falls $\bar{X} := \Omega \setminus X$ eine Nullmenge ist.

<u>Idee</u>:
Betrachte Einsmengen als sinnvolle Majoritäten.

$\Rightarrow$
Eine Zufallsfolge darf in keiner Nullmenge liegen.

<u>Beobachtung</u>:

- $\forall x \in \Omega$ ist $\{x\}$ eine Nullmenge.
  $\Rightarrow$
  Die Vereinigung aller Nullmengen in 2^Ω ist gerade Ω.
  $\Rightarrow$
  Es existiert keine Zufallsfolge.

<u>Übung</u>:
Zeigen Sie, dass für jedes $x \in \Omega$ die Menge $\{x\}$ eine Nullmenge ist.

Frage:

Wie rettet man obige sinnvoll aussehende Idee?

Idee (Per Martin-Löf):

Betrachte "konstruktive" Einsmengen als sinnvolle Majoritäten.

$\rightsquigarrow$

Eine Menge $X \subset \Omega$ heißt effektive Nullmenge, falls es einen Algorithmus $\mathcal{O}_X$ gibt, der als Eingabe eine rationale Zahl $\varepsilon > 0$ erhält und eine Menge $\{x_0, x_1, x_2, \dots\}$ von binären Strings aufzählt, so dass

i) $X \subset \Omega_{x_0} \cup \Omega_{x_1} \cup \Omega_{x_2} \cup \dots$ und

ii) $\sum_i 2^{-|x_i|} < \varepsilon$.

Solch ein Algorithmus $\mathcal{O}_X$ heißt Überdeckungsalgorithmus für X.

$X \subseteq \Omega$ heißt effektive Einsmenge, falls $\bar{X}$ eine effektive Nullmenge ist.

Beobachtung:

- Da wir für jedes $\varepsilon > 0$ ein $k \in \mathbb{N}$ mit $2^{-k} < \varepsilon$ wählen können, bleibt die Klasse der effektive Nullmenge dieselbe, wenn wir nur ε der Form 2^{-k}, $k \in \mathbb{N}$ zulassen.

- Da jede Überdeckung einer effektiven Nullmenge X auch für jede Teilmenge von X eine Überdeckung ist, ist jede Teilmenge einer effektiven Nullmenge selbst eine effektive Nullmenge.

Übung:

a) Zeigen Sie, dass eine abzählbare Vereinigung von Nullmengen eine Nullmenge ist.

b) Zeigen Sie, dass die Menge aller Folgen mit Nullen in den geradzahligen Positionen eine Nullmenge bildet

c) Zeigen Sie, dass die Vereinigung zweier effektiven Nullmengen eine effektive Nullmenge ist.

- Wenn wir sinnvolle Majoritäten und effektive Einsmengen identifizieren, dann müsste jede Zufallsfolge im Durchschnitt aller effektiven Einsmengen liegen.

$\Rightarrow$

Frage:
Ist der Durchschnitt aller effektiven Einsmengen nichtleer und selbst eine effektive Einsmenge?

Zur Beantwortung dieser Frage zeigen wir zunächst, dass die Vereinigung aller effektiven Nullmengen eine effektive Nullmenge ist und wenden dann die de Morgan'schen Regeln an.

Satz 3.1
Es existiert eine effektive Nullmenge N, die jede effektive Nullmenge X enthält.

Beweis:

Idee:
Konstruiere einen Überdeckungsalgorithmus $\mathcal{O}_N$, der alle Überdeckungsalgorithmen aufzählt und parallel verschränkt simuliert. Die Ausgabe von $\mathcal{O}_N$ ergibt sich aus den Ausgaben aller simulierten Überdeckungsalgorithmen.
Da $\mathcal{O}_N$ ein Überdeckungsalgorithmus ist, überdeckt er eine Nullmenge N. Da für jede effektive Nullmenge X mindestens ein Überdeckungsalgorithmus aufgezählt und simuliert wird, gilt $X \subseteq N$.

Problem:
Für einen gegebenen Algorithmus $\mathcal{O}$, der eine rationale Zahl $\varepsilon > 0$ als Eingabe erhält und binäre Strings ausgibt, ist es unentscheidbar, ob dieser ein Überdeckungsalgorithmus ist

oder nicht.

$\Rightarrow$

Die Menge der Überdeckungsalgorithmen ist nicht auf offensichtliche Art und Weise aufzählbar

Lösung des Problems:

Zähle die Menge aller Algorithmen auf und modifiziere jeden aufgezählten Algorithmus $\mathcal{A}$ derart, dass für den resultierenden Algorithmus $\mathcal{A}'$ folgendes erfüllt ist:

1) Falls $\mathcal{A}$ ein Überdeckungsalgorithmus für eine effektive Nullmenge ist, dann generieren $\mathcal{A}$ und $\mathcal{A}'$ dieselbe Folge von Strings.

2) Andernfalls ist der resultierende Algorithmus $\mathcal{A}'$ ein Überdeckungsalgorithmus für irgendeine Nullmenge.

Modifikation eines aufgezählten Algorithmus $\mathcal{A}$:

Annahme:

- ε' ist die Eingabe des Algorithmus $\mathcal{A}$
- Bei Eingabe ε' produziert $\mathcal{A}$ die Ausgabe

$x_0, x_1, x_2, \ldots$

Der modifizierte Algorithmus $\mathcal{A}'$ verwirft manche der von $\mathcal{A}$ generierten Strings.

Sei

L_k, $k \geq 0$ die Menge derjenigen Strings aus $x_0, x_1, x_2, \ldots, x_{k-1}$, die von $\mathcal{O}'$ nicht verworfen wurden.

Wenn $\mathcal{O}$ den String x_k aufzählt, dann entscheidet $\mathcal{O}'$, ob der String x_k verworfen wird oder nicht, indem $\mathcal{O}'$ überprüft, ob

$$2^{-|x_k|} + \sum_{x \in L_k} 2^{-|x|} < \varepsilon'.$$

$\mathcal{O}'$ verwirft genau dann x_k, wenn dies nicht der Fall ist.

Beobachtung

i) Falls $\mathcal{O}$ ein Überdeckungsalgorithmus ist, dann verwirft $\mathcal{O}'$ keinen der von $\mathcal{O}$ generierten Strings. Demzufolge ist $\mathcal{O}'$ nach wie vor ein Überdeckungsalgorithmus für dieselbe Nullmenge.

ii) Andernfalls ist $\mathcal{O}'$ "nahezu" ein Überdeckungsalgorithmus.

20. AA.

Konstruktion des Überdeckungsalgorithmus $\mathcal{O}_N$:

$\mathcal{O}_N$ zählt alle Algorithmen auf und modifiziert diese wie oben beschrieben. Dies ist möglich, da das Modifikationsprogramm stets dasselbe ist.

Seien ε die Eingabe von $\mathcal{O}_N$ und $\mathcal{O}_0', \mathcal{O}_1', \mathcal{O}_2', \ldots$

die aufgezählte Folge von modifizierten Algorithmen.

σ_N lässt die Algorithmen $\sigma_0', \sigma_1', \sigma_2', \ldots$ parallel verschränkt laufen, wobei $2^{-(i+1)} \cdot \varepsilon$ die Eingabe des Algorithmus σ_i' ist. Die Ausgabe von σ_N setzt sich aus den Ausgaben der Algorithmen $\sigma_0', \sigma_1', \sigma_2', \ldots$ zusammen. Sei

$$x_0, x_1, x_2, \ldots$$

die von σ_N generierte Folge von binären Strings

Konstruktion $\Rightarrow$

$$\sum_j 2^{-|x_j|} < \sum_{i \geq 0} 2^{-(i+1)} \varepsilon \leq \varepsilon$$

$\Rightarrow$

σ_N ist ein Überdeckungsalgorithmus für eine Nullmenge N.

Da für jede effektive Nullmenge X mindestens ein Überdeckungsalgorithmus σ_i aufgezählt wird, ist jede effektive Nullmenge X in N enthalten. ∎

Korollar 3.1

Der Durchschnitt aller effektiven Einsmengen ist nichtleer und selbst eine effektive Einsmenge.

Beweis:

Bezeichne $\mathcal{E}$ bzw. $\mathcal{N}$ die Mengen der effektiven Eins- bzw. Nullmengen. Dann gilt

$$E := \bigcap_{X \in \mathcal{E}} X \underset{\text{de M.}}{=} \overline{\bigcup_{X \in \mathcal{E}} \overline{X}} = \overline{\bigcup_{X \in \mathcal{N}} X} \underset{\text{Satz 3.1}}{=} \overline{N} \neq \emptyset,$$

da $N \neq \Omega$. Da N eine effektive Nullmenge ist, ist $E = \overline{N}$ eine effektive Einsmenge. ∎

Eine Folge $x \in \Omega$ von Nullen und Einsen ist eine <u>Martin-Löf-Zufallsfolge</u>, falls x im Durchschnitt aller effektiven Einsmengen liegt.

Offensichtlich gilt $E \cap N = \emptyset$ und $E \cup N = \Omega$.

Um für $x \in \Omega$ nachzuweisen, dass x keine Martin-Löf-Zufallsfolge ist, genügt es wegen $E \cap N = \emptyset$ zu zeigen, dass eine effektive Nullmenge existiert, die x enthält.

Bemerkung:

Als erster hatte bereits 1910 Richard von Mises eine formale Charakterisierung von Zufallsfolgen vorgeschlagen. Seine Zielsetzung war, die Wahrscheinlichkeitstheorie auf der Grundlage derartiger Zufallsfolgen aufzubauen.

Eine Folge $x = x_0 x_1 x_2 \ldots \in \Omega$ heißt <u>Mises-zu-fällig</u>, falls gilt:

1, Der Grenzwert der Häufigkeit der Einsen in x ist $1/2$. D.h.,

$$\lim_{n \to \infty} \frac{x_0 + x_1 + \ldots + x_{n-1}}{n} = 1/2.$$

2, Für jede unendliche Teilfolge, die durch eine zulässige Auswahlregel aus der Gesamtfolge konstruiert wird, ist der Grenzwert der Häufigkeit der Einsen $1/2$.

<u>Beispiele zulässiger Auswahlregeln</u>:

- Wähle die Teilfolge, bestehend aus den Folgegliedern mit geraden Indizes.
- Wähle diejenige Folgeglieder, die einer Null folgen. z.B. $00101100\ldots \rightsquigarrow 0110\ldots$

Von Mises gab keine exakte Definition einer zulässigen Auswahlregel an. Dies konnte er auch nicht, da die Algorithmentheorie 1910 noch nicht existierte. Gegen 1940 schlug Alonzo Church folgende Definition für eine zulässige Auswahlregel vor:

Eine <u>zulässige Auswahlregel</u> ist eine totale berechenbare Funktion S, die auf endliche Strings (inklusive ε) definiert ist und die Werte 0 oder 1 annimmt. Wenn wir S auf eine Folge $x_0x_1x_2\ldots$ anwenden, dann wählen wir genau diejenige x_n mit $S(x_0x_1\ldots x_{n-1}) = 1$ aus.

Die ausgewählten Folgeglieder bilden eine endliche oder unendliche Teilfolge, so dass jede zulässige Auswahlregel eine Abbildung

$$\sigma_S : \Omega \longrightarrow \{0,1\}^* \cup \Omega$$

definiert.

Falls $S(y) = 1 \quad \forall\, y \in \{0,1\}^*$, dann ist σ_S gerade die Identitätsfunktion. Somit folgt in der Definition der Mises-Zufälligkeit die erste Regel aus der zweiten.

↝

Eine Folge $x = x_0x_1x_2\ldots \in \Omega$ heißt <u>Mises-Church-zufällig</u>, falls für jede zulässige Auswahlregel S die Folge $\sigma_S(x)$ entweder endlich ist oder den Grenzwert der Häufigkeit der Einsen $1/2$ hat.

<u>Frage:</u>
Welcher Zusammenhang besteht zwischen der Martin-Löf- und der Mises-Church-Zufälligkeit?

Ziel:
Beweis, dass jede Martin-Löf-Zufallsfolge auch Mises-Church-zufällig ist.

Satz 3.2
Falls eine zulässige Auswahlregel auf eine Martin-Löf-Zufallsfolge angewendet wird, dann ergibt dies entweder eine endliche Folge oder eine Martin-Löf-Zufallsfolge.

Beweis:

Seien
- w eine beliebige Martin-Löf-Zufallsfolge
- S eine beliebige zulässige Auswahlregel mit korrespondierender Abbildung $\sigma_S : \Omega \to \{0,1\}^* \cup \Omega$.

2.2.
$\sigma_S(w) \in \{0,1\}^*$ oder $\sigma_S(w)$ ist eine Martin-Löf-Zufallsfolge.

Annahme:
$\sigma_S(w) \in \Omega$ und keine Martin-Löf-Zufallsfolge.

$\Rightarrow$
$\exists$ effektive Nullmenge $X \subset \Omega$ mit $\sigma_S(w) \in X$.

Da jede Teilmenge einer effektiven Nullmenge selbst eine effektive Nullmenge ist, gilt

$\{\sigma_S(w)\}$ ist eine effektive Nullmenge.

Idee:

Beweise, dass

$\{\sigma_S(w)\}$ effektive Nullmenge $\Rightarrow$ $\{w\}$ effektive Nullmenge

Durchführung:

Es genügt zu zeigen, dass $\exists$ Algorithmus σ_1, der als Eingabe eine rationale Zahl $\varepsilon > 0$ erhält und eine Menge $\{x_0, x_1, x_2, \dots\}$ von binären Strings aufzählt, so dass

i) $\{w\} \subseteq \Omega_{x_0} \cup \Omega_{x_1} \cup \Omega_{x_2} \cup \dots$ und

ii) $\sum_i 2^{-|x_i|} < \varepsilon$.

Ziel: Konstruktion eines solchen Algorithmus σ_1

$\{\sigma_S(w)\}$ effektive Nullmenge $\Rightarrow$

$\exists$ Algorithmus σ_1', der eine rationale Zahl $\varepsilon > 0$ als Eingabe erhält und eine Menge $\{x_0', x_1', x_2', \dots\}$ von binären Strings aufzählt, so dass

i) $\{\sigma_S(w)\} \subseteq \Omega_{x_0'} \cup \Omega_{x_1'} \cup \Omega_{x_2'} \cup \dots$ und

ii) $\sum_i 2^{-|x_i'|} < \varepsilon$.

Sei für $i \in \mathbb{N}_0$

$$A_{x_i'} := \{ z \in \Omega \mid x_i' \text{ ist Präfix von } \sigma_S(z) \}$$

Wegen

$$\{\sigma_S(\omega)\} \subseteq \Omega_{x_0'} \cup \Omega_{x_1'} \cup \Omega_{x_2'} \cup \dots$$

existiert ein $i \in \mathbb{N}_0$ mit

$$\omega \in A_{x_i'}.$$

Idee:
Konstruktion einer geeigneten Überdeckung von $A_{x_0'} \cup A_{x_1'} \cup A_{x_2'} \cup \dots$

Sei ε die Eingabe von σ_1. Unter Verwendung von σ' zählt σ_1 zunächst $\{x_0', x_1', x_2', \dots\}$ mit

i) $\{\sigma_S(\omega)\} \subseteq \Omega_{x_0'} \cup \Omega_{x_1'} \cup \Omega_{x_2'} \cup \dots$ und

ii) $\sum_i 2^{-|x_i'|} < \varepsilon$

auf.

Parallel verschränkt erzeugt σ_1 für jedes x_i' in dieser Aufzählung eine Überdeckung $\{y_0, y_1, y_2, \dots\}$ von $A_{x_i'}$ mit der Eigenschaft, dass

$$\sum_j 2^{-|y_j|} \leq 2^{-|x_i'|}.$$

Bevor wir uns überlegen, wie σ_1 diese Überdeckung erzeugt, führen wir den Beweis zu Ende.

Sei $x_0, x_1, x_2, \ldots$ die Aufzählung derjenigen Strings, die $A_{x_0} \cup A_{x_1} \cup A_{x_2} \cup \ldots$ überdecken. Da $w \in A_{x_i}$ für mindestens ein i gilt offensichtlich

$$\{w\} \subset \Omega_{x_0} \cup \Omega_{x_1} \cup \Omega_{x_2} \cup \ldots.$$

Ferner gilt

$$\sum_j 2^{-|x_j|} \leq \sum_i 2^{-|x_i'|} < \varepsilon.$$

Somit müssen wir uns nur noch überlegen, wie wir die gewünschten Überdeckungen der Mengen A_x, $x \in \{0,1\}^+$ erhalten.

Sei hierzu

$$x = ya, \text{ wobei } y \in \{0,1\}^* \text{ und } a \in \{0,1\}.$$

Idee:
Konstruktion einer Menge B_y von Strings, so dass

$$A_x = \bigcup_{z \in B_y} \Omega_{za}$$

und

$$\sum_{z \in B_y} 2^{-|za|} \leq 2^{-|x|}.$$

Als Überdeckung der Menge A_x können die Strings $\{za \mid z \in B_y\}$ aufgezählt werden.

Betrachten wir hierzu für $q \in \{0,1\}^*$ die Menge

$$B_q := \{ z \in \{0,1\}^* \mid S \text{ wählt aus } z \text{ den String } q \text{ sowie das erste Zeichen, das } z \text{ nachfolgt, aus.} \}$$

Dann gilt für $v \in \Sigma$:

$$\begin{aligned} v \in A_x &\Leftrightarrow v \in A_{yq} \\ &\Leftrightarrow yq \text{ ist Präfix von } \tau_S(v). \\ &\Leftrightarrow \text{Es existiert ein Präfix } z \text{ von } v, \text{ so dass } S \text{ aus } z \text{ genau } y \text{ auswählt sowie dasjenige Zeichen, das auf } z \text{ folgt, wobei dieses in } v \text{ ein } a \text{ ist.} \\ &\Leftrightarrow v \in \Sigma_{za} \text{ und } z \in B_y. \end{aligned}$$

Also erreichen wir mit obigen Mengen B_q, $q \in \{0,1\}^*$ unser Ziel.

<u>Induktive Definition der B_q, $q \in \{0,1\}^*$:</u>

- $B_\varepsilon := \{ z \in \{0,1\}^* \mid S(z) = 1 \text{ und } S(z') = 0 \text{ für alle echten Präfixe } z' \text{ von } z \}$.

und für $p \in \{0,1\}^*$ und $b \in \{0,1\}$

- $B_{pb} := \{ z_1 b z_2 \in \{0,1\}^* \mid z_1 \in B_p,\ S(z_1 b z_2) = 1 \text{ und } S(z_1 b z') = 0 \text{ für alle echten Präfixe } z_1 b z',\ z' \in \{0,1\}^* \text{ von } z_2 \}$.

Da S total und berechenbar ist, können die Mengen B_q, $q \in \{0,1\}^*$ parallel verschränkt aufgezählt werden.

Übung:
Zeigen Sie, dass für alle $q \in \{0,1\}^*$ die Menge B_q rekursiv aufzählbar ist.

Gilt $x_i' = ya$ für ein $y \in \{0,1\}^*$ und $a \in \{0,1\}$, dann fügen wir, sobald ein $z \in B_y$ aufgezählt wird, den String za zur Überdeckung von $A_{x_i'}$ hinzu. Die Überdeckung der Menge $A_{x_i'}$ besteht somit aus

$$\{ za \in \{0,1\}^+ \mid z \in B_y \}$$

Somit verbleibt noch

$$\sum_{z \in B_y} 2^{-|za|} \leq 2^{-|x|} \quad \forall\, x = ya \text{ mit } y \in \{0,1\}^*,\ a \in \{0,1\}$$

zu zeigen.

Wir beweisen dies mittels Induktion über $|x|$.

$|x| = 1$:
Dann ist $y = \varepsilon$ und somit

$$A_0 = \bigcup_{z \in B_\varepsilon} \Omega_{z0} \quad \text{und} \quad A_1 = \bigcup_{z \in B_\varepsilon} \Omega_{z1}.$$

Da die Mengen A_0 und A_1 disjunkt sind und dasselbe Maß besitzen, gilt

$$1 = P(\Omega) \geq P(A_0 \cup A_1) = P(A_0) + P(A_1) = 2P(A_0).$$

$$\Rightarrow$$

$$P(A_0) = P(A_1) \leq 1/2 = 2^{-1}.$$

Annahme:
Die Behauptung ist für alle x mit $|x| \leq k$, $k \geq 1$ erfüllt.

$k \rightsquigarrow k+1$:

Betrachte $y \in \{0,1\}^k$ beliebig aber fest. Für

$$A_{y0} = \bigcup_{z \in B_y} \Omega_{z0} \quad \text{bzw} \quad A_{y1} = \bigcup_{z \in B_y} \Omega_{z1}$$

gilt:

i) A_{y0} und A_{y1} sind disjunkt und haben dasselbe Maß.

ii) $A_{y0} \cup A_{y1} \subseteq A_y$

(Es kann Strings z geben, aus denen S genau den endlichen String y und kein weiteres Bit auswählt. Solche Strings sind in A_y, jedoch nicht in $A_{y0} \cup A_{y1}$.

$$\Rightarrow$$

Es gilt nicht notwendigerweise $A_{y0} \cup A_{y1} = A_y$.)

Induktionsannahme $\Rightarrow \quad P(A_y) \leq 2^{-|y|}$

Also gilt:

$$2^{-|y|} \geq P(A_y) \geq P(A_{y0} \cup A_{y1}) = P(A_{y0}) + P(A_{y1})$$
$$= 2 \cdot P(A_{y0}).$$

$\Rightarrow$

$$P(A_{y0}) = P(A_{y1}) \leq 2^{-(|y|+1)},$$

womit der Satz bewiesen ist. ∎

Satz 3.3

Für jede Martin-Löf-Zufallsfolge $w_0 w_1 w_2 \ldots$ gilt

$$\lim_{n \to \infty} \frac{w_0 + w_1 + \ldots + w_{n-1}}{n} = 1/2.$$

Beweis:

Für $n \in \mathbb{N}$ sei $W_n := \sum_{i=0}^{n-1} w_i$.

Es genügt zu zeigen, dass die Menge

$$X := \{ w_0 w_1 w_2 \ldots \in \Omega \mid \lim_{n \to \infty} \frac{W_n}{n} \neq 1/2 \}$$

eine effektive Nullmenge ist.

Übung:

Beweisen Sie

$X := \{ \omega_0 \omega_1 \omega_2 \ldots \in \Omega \mid \lim_{n\to\infty} \frac{\omega_n}{n} \neq \frac{1}{2} \}$ effektive Nullmenge $\Rightarrow \lim_{n\to\infty} \frac{\omega_n}{n} = \frac{1}{2}$.

Es gilt:

$\lim_{n\to\infty} \frac{\omega_n}{n} \neq \frac{1}{2} \iff \exists \delta := \frac{1}{m} > 0, m \in \mathbb{N}$, so dass $\forall n \in \mathbb{N}$ ein $n' \geq n$ existiert mit

$$\left| \frac{\omega_{n'}}{n'} - \frac{1}{2} \right| > \delta.$$

$\iff \frac{\omega_{n'}}{n'} > \frac{1}{2} + \delta$ für unendlich viele n'

oder

$\frac{\omega_{n'}}{n'} < \frac{1}{2} - \delta$ für unendlich viele n'

Für $\delta = \frac{1}{m}$, $m \in \mathbb{N}$ definieren wir

$$N_\delta := \{ \omega \in \Omega \mid \frac{\omega_n}{n} > \frac{1}{2} + \delta \text{ für unendlich viele } n \}$$

und

$$N'_\delta := \{ \omega \in \Omega \mid \frac{\omega_n}{n} < \frac{1}{2} - \delta \text{ für unendlich viele } n \}$$

Definitionen von N_δ und N'_δ $\Rightarrow$

$\omega \in X \iff \exists \delta$ mit $\omega \in N_\delta$ oder $\omega \in N'_\delta$.

Da die Vereinigung von abzählbar vielen effektiven Nullmengen wieder eine effektive Nullmenge ergibt, genügt es zu zeigen, dass für alle $\delta = \frac{1}{m}$, $m \in \mathbb{N}$ die Mengen N_δ und N'_δ effektive Nullmengen sind.

Konstruktion für N_δ:

Definition effektive Nullmenge $\Rightarrow$

Es genügt zu zeigen, dass ein Algorithmus O_1 existiert, der als Eingabe eine rationale Zahl $\varepsilon > 0$ erhält und eine Menge $\{x_0, x_1, x_2, \dots\}$ von binären Strings aufzählt, so dass

i) $N_\delta \subset \Omega_{x_0} \cup \Omega_{x_1} \cup \Omega_{x_2} \cup \dots$ und

ii) $\sum_i 2^{-|x_i|} < \varepsilon$.

Für jedes $w \in N_\delta$ existieren unendlich viele $n \in \mathbb{N}$ mit $\frac{w_n}{n} > \frac{1}{2} + \delta$. D.h., w besitzt unendlich viele Präfixe $w_0 w_1 \dots w_{n-1}$, die mehr als $(\frac{1}{2} + \delta)n$ Einsen enthalten.

$\Rightarrow$

Wir erhalten eine Überdeckung $\{x_0, x_1, x_2, \dots\}$ für N_δ, wenn wir für ein beliebiges $n_0 \in \mathbb{N}$ für alle $n \geq n_0$ sämtliche $x \in \{0,1\}^n$, die mehr als $(1/2 + \delta)n$ Einsen enthalten, aufzählen.

Zu zeigen ist noch, dass n_0 stets so groß gewählt werden kann, so dass für die resultierende Überdeckung $\{x_0, x_1, x_2, \dots\}$ auch

$$\sum_i 2^{-|x_i|} < \varepsilon$$

erfüllt ist.

Bezeichne A_n die Anzahl der Strings der Länge n mit mehr als $(\frac{1}{2}+\delta)n$ Einsen. Dann gilt:

$$\sum_i 2^{-|x_i|} = \sum_{n \geq n_0} A_n 2^{-n}.$$

Aufgrund des Cauchyschen Konvergenzkriteriums für Reihen genügt es zu zeigen, dass die Reihe

$$\sum_{n=0}^{\infty} 2^{-n} \cdot A_n$$

konvergiert.

Hierzu wenden wir Chernoff's Schranke an:

Satz 3.4 (Chernoff-Ungleichung)

Seien $X_1, X_2, \dots, X_n$ unabhängige Zufallsvariablen mit den Werten im Intervall $[0,1]$ und sei $X = X_1 + X_2 + \dots + X_n$ ihre Summe. Sei $\mu = E(X)$ der Erwartungswert für X. Dann gilt für jedes $a > 0$

$$Pr(X \geq \mu + a) \leq e^{\frac{-a^2}{2n}} \quad \text{und}$$

$$Pr(X \leq \mu - a) \leq e^{\frac{-a^2}{2n}}$$

Beweis:

Hier nicht. Siehe z.B.

- Graham, Knuth, Patashnik Mathematik S. 300 oder
- Mitzenmacher, Upfal, Probability and Computing: Randomized Algorithms and Probabilistic Analysis, S. 63 ff.

■

Es gilt

$$Pr\left(\left\{\omega \in \Omega \mid \frac{W_n}{n} > \frac{1}{2} + \frac{1}{m}\right\}\right)$$

$$= Pr\left(W_n \geq \frac{n}{2} + \frac{1}{m} \cdot n\right)$$

$$\underset{\text{Chernoff}}{\leq} e^{-\left(\frac{n}{m}\right)^2/2n} = \underbrace{\left(e^{-\frac{1}{2m^2}}\right)}_{=: q_m}{}^n$$

$$= q_m{}^n$$

Bemerkung:

Gegeben m kann q_m leicht berechnet werden.

Es gilt

$$Pr\left(\left\{ w \in \Sigma \mid \frac{w_n}{n} > \frac{1}{2} + \frac{1}{m} \right\}\right) = 2^{-n} \cdot A_n.$$

Anwendung von Chernoff's Schranke ergibt:

$$\sum_{i=0}^{\infty} 2^{-i} A_i \leq \sum_{i=0}^{\infty} q_m^i = \frac{1}{1 - q_m}$$

$$\Rightarrow$$

$$\sum_{n=0}^{\infty} q_m^n \quad \text{konvergiert.}$$

Wegen $2^{-n} A_n \leq q_m^n \quad \forall\, n \in \mathbb{N}$ folgt aus dem Majorantenkriterium, dass die Reihe

$$\sum_{n=0}^{\infty} 2^{-n} A_n$$

konvergiert.

Somit haben wir die Existenz eines hinreichend großen n_0 bewiesen.

Da n_0 von der Eingabe ε abhängt, reicht die Existenz von n_0 alleine nicht aus. Der Algorithmus OT muss solch ein n_0 berechnen können, bevor die erste Ausgabe erfolgt.

Hierzu überprüft er für wachsende n, ob

$$\frac{1}{1 - q_m} - \sum_{i=0}^{n} q_m^i < \varepsilon$$

Da n_0 existiert, findet OT irgendwann ein n, das obige Ungleichung erfüllt. Dieses n definiert dann n_0.

∎

Aus den Sätzen 3.2 und 3.3 folgt direkt:

Satz 3.5

Jede Martin-Löf-Zufallsfolge ist auch eine Mises-Church-Zufallsfolge.

Übung:

Folgende Auswahlregel ist gemäß unserer Definition nicht zulässig:

- Wähle jedes Folgeglied x_{2n} mit $x_{2n+1} = 0$.

Zeigen Sie, dass die Anwendung dieser Auswahlregel auf eine Martin-Löf-Zufallsfolge dennoch eine Martin-Löf-Zufallsfolge ergibt.

Bemerkung:

Die umgekehrte Richtung von Satz 3.5 gilt nicht. D.h., es gibt Mises-Church-Zufallsfolgen, die keine Martin-Löf-Zufallsfolgen sind. Heute wird die Klasse der Mises-Church-Zufallsfolgen als zu groß angesehen.

- Es gibt Mises-Church-Zufallsfolgen, die das sogenannte "Gesetz des iterierten Logarithmus" nicht erfüllen.

Frage:

Können wir mit Hilfe der Kolmogorov-Komplexität eine weitere Definition für Zufallsfolgen erhalten?

Idee:

Definiere eine unendliche Binärfolge genau dann als Zufallsfolge, wenn alle Präfixe dieser unendlichen Folge maximal große Kolmogorov-Komplexität besitzen.

Problem:

Satz 2.11 besagt, dass jede unendliche Binärfolge unendlich viele Präfixe besitzt, die nicht maximal große Kolmogorov-Komplexität haben

↝

Betrachte anstatt Kolmogorov-Komplexität eine Variante der Kolmogorov-Komplexität, die sogenannte Präfixkomplexität.

Eine berechenbare Funktion $f: \{0,1\}^+ \to \Sigma^+$ heißt Präfixfunktion, falls für jeden String $x \in \{0,1\}^+$ und jeden echten Präfix y von x mindestens einer der Werte $f(x)$ und $f(y)$ undefiniert ist

($\Leftrightarrow$ Für jeden unendlichen String $x \in \Sigma$ existiert höchstens ein endlicher Präfix x' von x, für den $f(x')$ definiert ist.)

Ein Dekompressionsalgorithmus, der eine Präfixfunktion berechnet, heißt Präfixdekompressionsalgorithmus. Ein Präfixdekompressionsalgorithmus U heißt asymptotisch optimal, falls für jeden anderen Präfixdekompressionsalgorithmus D eine Konstante c_D existiert, so dass $\forall x \in \{0,1\}^+$

$$K_U(x) \leq K_D(x) + c_D.$$

Die algorithmische Komplexität $K_U(x)$ (einen String $x \in \{0,1\}^*$) bezüglich eines asymptotisch optimalen Präfixdekompressionsalgorithmus U heißt Präfixkomplexität $KP(x)$ von x.

Ein unendlicher String $x = x_0x_1x_2\ldots \in \Omega$ heißt Kolmogorov-Zufallsfolge, falls eine Konstante c_x existiert, so dass $\forall n \in \mathbb{N}$

$$KP(x_0x_1x_2\ldots x_{n-1}) \geq n - c_x.$$

Ziel:
Beweis der Existenz eines asymptotisch optimalen Präfixdekompressionsalgorithmuses.

Satz 3.6
Es existiert ein asymptotisch optimaler Präfixdekompressionsalgorithmus.

Beweis:

Wir benötigen

- einen universellen Algorithmus U und dessen Fähigkeit, jeden anderen Algorithmus mit Hilfe eines zusätzlichen Eingabe konstanter Länge simulieren zu können.
- Modifikation dieses universellen Algorithmus, so dass er eine Präfixfunktion berechnet.

Erinnerung:

$\bar{p}$ erhalten wir aus einem Binärstring p durch Verdopplung der einzelnen Bits.

Idee:

1) Der universelle Algorithmus U interpretiert eine Eingabe der Form $\bar{p}01y$ als ein Programm p mit Eingabe y.

Bitverdopplung in p $\Rightarrow$

$p_1, p_2 \in \{0,1\}^+, p_1 \neq p_2 \Rightarrow \bar{p_1}01 \neq \bar{p_2}01$.

$\Rightarrow$

Falls eine Eingabe x' für U Präfix einer anderen Eingabe x'' für U ist, dann enthalten x' und x'' dasselbe Programm p und unterscheiden sich lediglich durch die Eingaben für p. Insbesondere ist die Ein-

gabe in x' für p ein Präfix der Eingabe in x'' für p.

2) Wir werden dafür sorgen, dass für die Ausgabe von U bei Eingabe $\bar{p}01y$ folgendes gilt:

i) Falls die durch p berechnete Funktion eine Präfixfunktion ist, dann gilt

$$U(\bar{p}01y) = p(y).$$

D.h., U simuliert korrekt das Programm p bei Eingabe y.

ii) Berechnet p keine Präfixfunktion, dann wird das Programm p derart zu einem Programm p' modifiziert, dass p' eine Präfixfunktion berechnet. U gibt dann $p'(y)$ aus. D.h.,

$$U(\bar{p}01y) = p'(y).$$

Durchführung:

- Parallel verschränkt zählt U alle möglichen Eingaben für p auf und wendet p auf diese Eingaben an.

  ↝

  U zählt eine Folge F von Paaren $\langle y_i, z_i \rangle$ mit $p(y_i) = z_i$ auf.

- U terminiert, sobald ein Paar der Form $\langle y, z \rangle$ aufgezählt wird.

U untersucht dann, ob in F ein Paar $\langle y', z' \rangle$ mit y' ist ein Präfix von y oder y ist ein Präfix von y' existiert.

Falls ja $\rightsquigarrow$ Endlosschleife

Falls nein $\rightsquigarrow$ U gibt den String z aus.

Ziel:
Beweis, dass obige Durchführung das gewünschte leistet.

Betrachte hierzu unendlichen String $x = \bar{p}01z$ beliebig aber fest. Wir unterscheiden zwei Fälle:

1. Fall: p in der Eingabe $\bar{p}01z$ ist ein Präfix des Kompressionsalgorithmus

$\Rightarrow$

U simuliert p bei Eingabe z korrekt.

2. Fall: p in der Eingabe $\bar{p}01z$ ist kein Präfix des Kompressionsalgorithmus.

Betrachte die Menge M aller Präfixe von z.

- Falls in M kein Element existiert, bei dessen Eingabe p terminiert, dann terminiert U auch bei keiner Eingabe $\bar{p}01y$ mit $y \in M$.
- Falls es in M Elemente gibt, bei deren Eingabe p terminiert, dann terminiert

U für genau eine Eingabe $\bar{p}01y$ mit $y \in M$, nämlich dasjenige $y := y_i$, das in obiger Aufzählung als erste Komponente eines Paares $\langle y_i, z_i \rangle$ vorkommt.

Wichtig:

Aufzählung muss unabhängig von dem konkreten z in der Eingabe $\bar{p}01z$ stets in derselben Reihenfolge erfolgen.

$\Rightarrow$

p wurde zu einem Präfixdekompressionsalgorithmus p' modifiziert.

Sei D ein zu U verschiedener Präfixdekompressionsalgorithmus.

z.z. $\exists$ Konstante c_D, so dass $\forall x \in \{0,1\}^+$

$$K_U(x) \leq K_D(x) + c_D.$$

Sei p ein Programm für U, das D simuliert.
Betrachte $x \in \{0,1\}^+$ beliebig aber fest.
Sei y eine Eingabe für D mit

$$D(y) = x \quad \text{und} \quad |y| = K_D(x).$$

$\Rightarrow$

$$U(\bar{p}01y) = p(y) = D(y) = x$$

$\Rightarrow$

$$K_U(x) \leq |\bar{p}01y| = |y| + |\bar{p}011| = K_D(x) + c_D,$$

wobei $c_D := |\bar{p}011|$.

Wir werden nun einige Eigenschaften der Präfixkomplexitätsfunktion herleiten.

Unter Verwendung der Identitätsfunktion haben wir für die Kolmogorov-Komplexitätsfunktion K bewiesen, dass $\forall x \in \{0,1\}^*$

$$K(x) \leq |x| + c_{id},$$

wobei c_{id} eine Konstante ist.

Die Identitätsfunktion ist keine Präfixfunktion

$\Rightarrow$

Analoge Eigenschaft kann nicht mit Hilfe der Identitätsfunktion für K_P bewiesen werden.

Ziel:

Beweis, dass analoge Eigenschaft nicht für die Präfixkomplexitätsfunktion K_P gilt.

Satz 3.7

Es gilt $\sum_{x \in \{0,1\}^*} 2^{-K_P(x)} \leq 1$

Beweis:

Sei U derjenige asymptotisch optimale

Präfix de Kompressionsalgorithmus, der bei der Definition der Präfix Komplexitätsfunktion zugrundegelegt wird.

Für $x \in \{0,1\}^+$ bezeichne p_x eine kürzeste Beschreibung für x bezüglich U.

U berechnet eine Präfixfunktion $\Rightarrow$

$\forall\, x, y \in \{0,1\}^+,\ x \neq y$ gilt:

- p_x ist kein Präfix von p_y und umgekehrt.

$\Rightarrow$

$$(*) \quad \Omega_{p_x} \cap \Omega_{p_y} = \emptyset.$$

$\forall\, x \in \{0,1\}^+$ hat die Menge Ω_{p_x} das Maß

$$(**) \quad 2^{-|p_x|} = 2^{-K_P(x)}.$$

Aus $(*)$ und $(**)$ folgt

$$1 = P(\Omega) \geq P\left(\bigcup_{x \in \{0,1\}^+} \Omega_{p_x}\right) = \sum_{x \in \{0,1\}^+} 2^{-K_P(x)}.$$

■

Korollar 3.2

Es existiert eine Konstante c, so dass

$$K_P(x) \leq |x| + c \quad \forall\, x \in \{0,1\}^+.$$

Beweis:

Annahme:

$\exists$ Konstante c, so dass $K_P(x) \leq |x| + c \quad \forall x \in \{0,1\}^+$.

$\Rightarrow$

$$\sum_{x \in \{0,1\}^+} 2^{-K_P(x)} \geq \sum_{x \in \{0,1\}^+} 2^{-(|x|+c)}$$

$$= 2^{-c} \cdot \sum_{x \in \{0,1\}^+} 2^{-|x|}.$$

Satz 3.7 $\Rightarrow$

$$2^{-c} \cdot \sum_{x \in \{0,1\}^+} 2^{-|x|} \leq 1 \Leftrightarrow \sum_{x \in \{0,1\}^+} 2^{-|x|} \leq 2^c.$$

D.h., $\sum_{x \in \{0,1\}^+} 2^{-|x|}$ ist beschränkt.

Es gilt aber auch

$$\sum_{x \in \{0,1\}^n} 2^{-|x|} = 1 \qquad \forall n \in \mathbb{N}.$$

$\Rightarrow$ $\sum_{x \in \{0,1\}^+} 2^{-|x|}$ ist nicht beschränkt.

Widerspruch!

■

Ziel:

Beweis einer oberen Schranke für die Präfixkomplexitätsfunktion

Satz 3.8

Es existiert eine Konstante c, so dass

$$K_P(x) \leq 2|x| + c \quad \forall x \in \{0,1\}^+.$$

Beweis:

Betrachte den Algorithmus D_0, der aus der Eingabe $\bar{x}01$ die Ausgabe x erzeugt. D.h.,

$$D_0(\bar{x}01) = x.$$

Bei allen anderen Eingaben, die nicht die Form $\bar{x}01$ haben, geht D_0 in eine Endlosschleife.

Beobachtung:

$x, y \in \{0,1\}^+, x \neq y \Rightarrow \bar{x}01$ ist kein Präfix von $\bar{y}01$ und umgekehrt.

$\Rightarrow$

D_0 berechnet in der Tat eine Präfixfunktion, ist also ein Präfixdekompressionsalgorithmus.

Satz 3.6 $\Rightarrow$

$$K_P(x) := K_U(x) \leq K_{D_0}(x) + c_{D_0}$$

$$= 2|x| + 2 + c_{D_0} = 2|x| + c,$$

wobei $c = c_{D_0} + 2$

□

Verbesserung der oberen Schranke:

i) x kann aus der Eingabe $\overline{bin(|x|)}\,01x$ durch Präfix-dekompressionsalgorithmus rekonstruiert werden $\rightsquigarrow$

$$K^P(x) \leq |x| + 2\log|x| + c$$

Diese Methode kann iteriert werden.

ii) $|x|$ kann durch $K(x)$ ersetzt werden.

Übung

a) Beweisen Sie, dass eine Konstante c existiert, so dass $\forall x \in \{0,1\}^+$ gilt:

$$K^P(x) \leq K(x) + \log K(x) + \log\log K(x) + 2\log\log\log K(x) + c$$

b) Was ist die kleinste obere Schranke für $K^P(x)$, die Sie beweisen können?

Ziel:

Beweis, dass die Klassen der Martin-Löf- und der Kolmogorov-Zufallsfolgen gleich sind.

Satz 3.9

Sei $x = x_0x_1x_2\ldots \in \Omega$ eine Martin-Löf-Zufallsfolge. Dann ist x auch eine Kolmogorov-Zufallsfolge.

Beweis: (indirekt)

Annahme: x ist keine Kolmogorov-Zufallsfolge

$\Rightarrow$

$\forall$ Konstanten c existiert ein n_c mit

$$(*) \quad K(x_0 x_1 \dots x_{n_c-1}) < n_c - c.$$

Wir werden beweisen, dass für jedes $x \in \Omega$, das $(*)$ erfüllt, die Menge $\{x\}$ eine effektive Nullmenge ist. Dies impliziert dann, dass x auch keine Martin-Löf-Zufallsfolge ist.

Ziel:

Konstruktion eines Algorithmus O_1, der als Eingabe eine rationale Zahl $\varepsilon > 0$ erhält und eine Menge $\{y_0, y_1, y_2, \dots\}$ von binären Strings aufzählt, so dass

i) $\{x\} \subset \Omega_{y_0} \cup \Omega_{y_1} \cup \Omega_{y_2} \cup \dots$ und

ii) $\sum_i 2^{-|y_i|} < \varepsilon$.

Sei $\varepsilon > 0$ eine beliebige feste Eingabe für O_1. O_1 wählt eine Konstante c groß genug, so dass $2^{-c} < \varepsilon$.

Bezeichne X_c die Menge aller (binären) Strings u mit $K(u) < |u| - c$.

Es ist leicht zu zeigen, dass X_c rekursiv auf-

zählbar ist.

Übung:

a) Zeigen Sie, dass die Präfixkomplexitäts-funktion von oben rekursiv aufzählbar ist.

b) Zeigen Sie, dass die Menge X_c rekursiv auf-zählbar ist.

a) zählt $X_c = \{y_0, y_1, y_2, \dots\}$ auf. Da x einen Präfix u mit $K_P(u) < |u| - c$ besitzt, gilt offensichtlich

$$\{x\} \subset \Omega_{y_0} \cup \Omega_{y_1} \cup \Omega_{y_2} \cup \dots$$

Zu zeigen ist noch

$$\sum_{u \in X_c} 2^{-|u|} < \varepsilon.$$

Wegen $u \in X_c$ gilt $|u| > K_P(u) + c$.

$\Rightarrow$

$$2^{-|u|} < 2^{-(K_P(u)+c)} = 2^{-c} \cdot 2^{-K_P(u)}$$

Also gilt

$$\sum_{u \in X_c} 2^{-|u|} < 2^{-c} \cdot \sum_{u \in X_c} 2^{-K_P(u)}$$

$$\leq_{\text{Satz 3.7}} 2^{-c} \cdot 1$$

$$< \varepsilon$$

Wahl von c

■

Der Beweis, dass jede Kolmogorov-Zufallsfolge auch eine Martin-Löf-Zufallsfolge ist, ist wesentlich schwieriger. Warum ist dies so?

Zur Beantwortung dieser Frage betrachten wir sogenannte Semimaße.

Sei F eine Ereignisalgebra. Eine Funktion $m: F \to \mathbb{R}_0^+$ heißt <u>Semimaß auf F</u>, falls folgende Axiome erfüllt sind:

(S1) $m(\Omega) \leq 1$.

(S2) Für höchstens abzählbare Indexmengen I und paarweise disjunkte Ereignisse A_i, $i \in I$ gilt

$$m\left(\bigcup_{i \in I} A_i\right) \geq \sum_{i \in I} m(A_i).$$

D.h., wir haben in der Definition des Wahrscheinlichkeitmaßes P im ersten Kolmogorov-Axiom $=$ durch $\leq$ und im zweiten Kolmogorov-Axiom $=$ durch $\geq$ ersetzt.

Den Beweis

x Martin-Löf-Z.f. $\Rightarrow$ x Kolmogorov-Z.f.

haben wir indirekt geführt. D.h., wir haben

x $\neg$ Kolmogorov-Z.f. $\Rightarrow$ x $\neg$ Martin-Löf-Z.f.

bewiesen.

$x \neg$ Kolmogorov-Z.f. impliziert, dass für alle Konstanten c ein $n_c \in \mathbb{N}$ existiert, so dass

$$Kp(x_0 x_1 x_2 \dots x_{n_c - 1}) < n_c - c. \qquad (*)$$

Satz 3.7 besagt, dass die Präfix Komplexität die Kraft-Ungleichung erfüllt, so dass wir mit

$$m(\Omega_y) := 2^{-Kp(y)} \quad \forall y \in \{0,1\}^* \qquad (**)$$

ein Semimaß auf Ω definieren können. Satz 3.7 war einfach zu beweisen.

Mit Hilfe von (*) und (**) war es nun leicht einen Überdeckungsalgorithmus für $\{x\}$ zu konstruieren, womit

$$x \neg \text{Martin-Löf-Z.f.}$$

bewiesen ist.

Den Beweis

$$x \text{ Kolmogorov-Z.f.} \Rightarrow x \text{ Martin-Löf-Z.f.}$$

werden wir auch indirekt führen. D.h., wir werden

$$x \neg \text{Martin-Löf-Z.f.} \Rightarrow x \neg \text{Kolmogorov-Z.f.}$$

beweisen.

$x \neg$ Martin-Löf-Z.f. impliziert, dass $\{x\}$ eine effektive Nullmenge ist.

Wir betrachten die Vereinigung N aller effektiven Nullmengen und den im Beweis von Satz 3.1 konstruierten Überdeckungsalgorithmus $\mathcal{O}_N$ für N.

Mit Hilfe von $\mathcal{O}_N$ und der Tatsache, dass gemäß Annahme $\{x\} \subset N$ konstruieren wir ein Semimaß m' auf $\{0,1\}^*$.

Zum Beweis, dass x keine Kolmogorov-Zufallsfolge ist, benötigen wir einen weiteren Zusammenhang zwischen der Präfixkomplexitätsfunktion und Semimaße, nämlich

$$(***) \quad K_P(z) \leq -\log m'(z) + C \qquad \forall z \in \{0,1\}^*,$$

wobei C eine Konstante ist.

Der Beweis von $(***)$ ist wesentlich aufwendiger als der Beweis von Satz 3.7. Wir werden hierzu zunächst eine Theorie der "algorithmischen Wahrscheinlichkeiten" entwickeln. Hat man $(***)$, dann ist der Beweis

x Martin-Löf-Z.f. $\Rightarrow$ x Kolmogorov-Z.f.

kann aufwendiger als der Beweis

x Kolmogorov-Z.f. $\Rightarrow$ x Martin-Löf-Z.f..

Ziel:

Entwicklung einer Theorie der algorithmischen Wahrscheinlichkeiten.

Hierzu benötigen wir zunächst sogenannte probabilistische Maschinen, die wir uns informell definieren werden.

Eine probabilistische Maschine M erhalten wir aus einer nichtprobabilistischen Maschine (z.B. einer Turingmaschine), indem wir diese um die Möglichkeit, Zufallsbits zu erzeugen, erweitern. D.h., sobald M in einen speziellen Zustand (z.B. "random") gerät, erzeugt M zufällig einen Wert aus $\{0,1\}$. Dabei werden 0 und 1 mit gleicher Wahrscheinlichkeit und unabhängig von zuvor generierten Zufallsbits generiert.

Bemerkung:

Die Ausgabe einer deterministischen Maschine ist eine Funktion der Eingabe. Dies ist bei einer probabilistischen Maschine nicht der Fall. Dieselbe Eingabe kann verschiedene Ausgaben erzeugen, wobei jede Ausgabe eine gewisse Wahrscheinlichkeit besitzt.

Zur Entwicklung einer Theorie der algorithmischen Wahrscheinlichkeiten benötigen wir probabilistische Maschinen ohne Eingaben. Sei M solch eine

Maschine.

Frage: Was ist die Wahrscheinlichkeit p, dass M terminiert?

Für $n \in \mathbb{N}$ sei p_n die Wahrscheinlichkeit, dass M in $\leq n$ Schritten anhält.

Beobachtung:
p_n kann wie folgt einfach berechnet werden:

- Simuliere das Verhalten von M für jeden der 2^n möglichen Zufallsstrings der Länge n und zähle, bei wievielen dieser Zufallsstrings M innerhalb n Schritten anhält. Sei t_n die sich ergebende Anzahl. Dann gilt

$$p_n = \frac{t_n}{2^n}.$$

Die Wahrscheinlichkeit p, dass M terminiert, definieren wir durch

$$p := \lim_{n \to \infty} p_n.$$

Da die Folge p_n monoton wachsend und beschränkt ist, existiert dieser Grenzwert.

Eine reelle Zahl p heißt <u>von unten aufzählbar</u>, falls p Grenzwert einer monoton wachsenden, berechenbaren Folge $p_0, p_1, p_2, \ldots$ rationaler Zahlen ist. D.h.,

1. $p = \lim_{n \to \infty} p_n$, wobei $p_0 \leq p_1 \leq p_2 \leq \ldots$ und
2. es existiert ein Algorithmus O_T, der für gegebenes $i \in \mathbb{N}_0$ die rationale Zahl p_i berechnet.

Satz 3.10

a) Sei M eine probabilistische Maschine ohne Eingabe. Dann ist die Wahrscheinlichkeit p der Terminierung von M von unten aufzählbar.

b) Sei $p \in [0,1]$ eine von unten aufzählbare reele Zahl. Dann existiert eine probabilistische Maschine M ohne Eingabe, die mit Wahrscheinlichkeit p terminiert.

Beweis:

Sei M eine probabilistische Maschine ohne Eingabe, die mit Wahrscheinlichkeit p terminiert. Wie p von unten aufgezählt werden kann, haben wir bereits bei der Definition von p beschrieben. Demzufolge ist nur b) noch zu beweisen.

b) Sei $p \in [0,1]$ eine beliebige von unten aufzählbare reele Zahl. Ferner sei $p_0, p_1, p_2, \ldots$ eine monoton wachsende berechenbare Folge von rationalen Zahlen, die gegen p konvergiert.

Ziel:

Konstruktion einer probabilistischen Maschine M ohne Eingabe, die mit Wahrscheinlichkeit p terminiert.

Idee:

- Interpretiere die Menge aller Zufallsfolgen als unendlichen Trie T bezüglich Alphabet $\{0,1\}$, in dem jede linke ausgehende Kante mit 0 und jede rechte ausgehende Kante mit 1 markiert ist.

- Definiere M dergestalt, dass gilt:

  1. Falls M bei der Zufallsfolge $b_0 b_1 b_2 \ldots$ terminiert, dann terminiert M auch bei jeder Zufallsfolge, die links von $b_0 b_1 b_2 \ldots$ im Trie T steht.

  2. Der Anteil der Zufallsfolgen, bei denen M terminiert, konvergiert mit der Länge der Zufallsfolgen gegen p.

Durchführung:

Annahme:

M hat bisher die Zufallsbits $b_0 b_1 b_2 \ldots b_i$ generiert.

$\Rightarrow$

Zufallsstring $b_0 b_1 b_2 \ldots b_i$ korrespondiert in T zu einem Pfad $v_0, v_1, v_2, \ldots, v_{i+1}$, wobei v_0 die Wurzel von T ist.

Die Wahrscheinlichkeit, dass M die Zu-

falls bits $b_0 b_1 \dots b_i$ generiert, beträgt $2^{-(i+1)}$

Berechne $P_e(b_0 b_1 \dots b_i)$ die Wahrscheinlichkeit, dass M einen Zufallsstring links von $b_0 b_1 \dots b_i$ in T generiert.

Um einen String links von $b_0 b_1 \dots b_i$ in T zu generieren, muss auf dem Pfad $v_0, v_1, \dots, v_{i+1}$ in T in einem Knoten v_j, in dem $b_0 b_1 \dots b_i$ die rechte Kante wählt, d.h., $b_j = 1$, die linke Kante genommen werden.

$\Rightarrow$

$$P_e(b_0 b_1 \dots b_i) = \sum_{j : b_j = 1} 2^{-(j+1)}.$$

Berechne $P_g(b_0 b_1 \dots b_i)$ die Wahrscheinlichkeit, dass M den Zufallsstring $b_0 b_1 \dots b_i$ oder einen Zufallsstring links von $b_0 b_1 \dots b_i$ in T generiert. Dann gilt:

$$P_g(b_0 b_1 \dots b_i) = 2^{-(i+1)} + P_e(b_0 b_1 \dots b_i)$$

$$= 2^{-(i+1)} + \sum_{j : b_j = 1} 2^{-(j+1)}$$

Da M immer mit dem leeren Zufallsstring startet, gilt

$$P_g(\varepsilon) = 1.$$

Als nächstes beschreiben wir das Verhalten

von M.

- M enthält den Algorithmus O_1, der für ein gegebenes i den rationalen Wert p_i berechnet.
- M berechnet die rationale Zahl p_0.

Annahme

$b_0 b_1 \dots b_i \in \{0,1\}^*$ ist die bisher von M generierte Zufallsfolge.

Falls $b_0 b_1 \dots b_i = \varepsilon$, dann vereinbaren wir $i = -1$.

Falls $P_g(b_0 b_1 \dots b_i) \leq p_{i+1}$, dann hält M an. Andernfalls generiert M das nächste Zufallsbit b_{i+1} und berechnet die nächste rationale Zahl p_{i+2}.

Falls $P_g(b_0 b_1 \dots b_i) \leq p_{i+1}$, d.h., M hält an, müssen wir uns noch davon überzeugen, dass M auch mit jeder Zufallsfolge, die links von $b_0 b_1 \dots b_i$ in T steht, anhält.

Betrachte hierzu eine beliebige aber feste Zufallsfolge $b_0' b_1' b_2' \dots$, die links von $b_0 b_1 \dots b_i$ in T steht.

Falls M bezüglich eines Präfixes der Länge $\leq i$ von $b_0'b_1' \dots b_i'$ anhält, dann ist nichts mehr zu beweisen.

Annahme:
M hält nicht bzgl. eines Präfixes von $b_0'b_1' \dots b_i'$ an.

Wegen

$$P_g(b_0'b_1' \dots b_i') < P_g(b_0b_1 \dots b_i) \leq p_{i+1}$$

hält M dann mit der generierten Zufallsfolge $b_0'b_1' \dots b_i'$ an.

Wegen $\lim_{n \to \infty} p_n = p$ impliziert obige Konstruktion, dass M mit Wahrscheinlichkeit p anhält. ∎

Interpretation:
Die terminierenden Berechnungen der probabilistischen Maschine M realisieren das Semimaß, bestehend aus der einzelnen von unten aufzählbare Wahrscheinlichkeit p.

Das im Beweis von

× Kolmogorov-Z.f. ⇒ × Martin-Löf-Z.f.

konstruierte Semimaß m' wird abzählbar viele Wahrscheinlichkeiten enthalten. Daher benötigen

wir eine Maschine, deren terminierenden Berechnungen derartige Semimaße realisieren. Hierzu betrachten wir probabilistische Maschinen ohne Eingabe mit Ausgaben, so dass wir terminierenden Berechnungen in Abhängigkeit der korrespondierenden Ausgaben unterscheiden können.

Sei M eine probabilistische Maschine ohne Eingabe, die Zahlen $i \in \mathbb{N}_0$ als Ausgabe produziert.

Ziel:
Verallgemeinerung des Satzes 3.10 auf diese Maschine.

Hierzu benötigen wir zunächst noch einige Bezeichnungen.

Für $i \in \mathbb{N}_0$ und $n \in \mathbb{N}$ bezeichne $p(i,n)$ die Wahrscheinlichkeit, dass M innerhalb von n Schritten mit der Ausgabe i terminiert. $p(i,n)$ kann wie folgt einfach berechnet werden:

- Simuliere das Verhalten von M für jeden der 2^n möglichen Zufallsstrings der Länge n und zähle, bei wievielen dieser Zufallsstrings M innerhalb n Schritten mit der Ausgabe i anhält. Sei $t(i,n)$ die sich ergebende Anzahl. Dann gilt:

$$p(i,n) = \frac{t(i,n)}{2^n}.$$

Die Wahrscheinlichkeit p_i, dass M mit Ausgabe i terminiert, definieren wir durch

$$p_i := \lim_{n \to \infty} p(i,n)$$

Da die Folge $p(i,n)$ monoton wachsend und beschränkt ist, existiert dieser Grenzwert.

Eine Folge $p_0, p_1, p_2, \ldots$ von reellen Zahlen heißt von unten aufzählbar, falls es eine berechenbare totale Funktion $p: \mathbb{N}_0 \times \mathbb{N}_0 \to \mathbb{Q}$ gibt, so dass $\forall i \in \mathbb{N}_0$ gilt:

i) $p(i,0) \leq p(i,1) \leq p(i,2) \leq \ldots$ und

ii) $\lim_{n \to \infty} p(i,n) = p_i$.

Satz 3.11

a) Sei M eine probabilistische Maschine ohne Eingabe, die Zahlen $i \in \mathbb{N}_0$ als Ausgabe produziert. Für $i \in \mathbb{N}_0$ bezeichne p_i die Wahrscheinlichkeit, dass M mit der Ausgabe i terminiert. Dann ist die Folge $p_0, p_1, p_2, \ldots$ von unten aufzählbar und $\sum_i p_i \leq 1$.

b) Seien $p_0, p_1, p_2, \ldots$ eine von unten aufzählbare Folge von nichtnegativen reellen Zahlen

und $\sum_i p_i \leq 1$. Dann existiert eine probabilistische Maschine M ohne Eingabe, die für alle $i \in \mathbb{N}_0$ die Ausgabe i mit Wahrscheinlichkeit p_i produziert.

Beweis:

a) Wie die Folge $p_0, p_1, p_2, \ldots$ von unten aufgezählt werden kann, haben wir bereits bei der Definition von p_i beschrieben.

b) Ziel: Modifikation des Beweises von Satz 3.10 b).

Da die Terminierung von M mit unterschiedlichen Ausgaben als verschiedene Endzustände anzusehen sind, unterteilen wir den einen Bereich des Trias T, der zu Zufallsfolgen, bei denen M anhält, korrespondiert, in Bereiche ein.

Jeder Bereich korrespondiert zu einer Zahl $j \in \mathbb{N}_0$. Zur selben Zahl korrespondieren in der Regel mehrere Bereiche.

Die probabilistische Maschine M gibt genau dann $j \in \mathbb{N}_0$ aus, wenn sie mit einer Zufallszahl, die zu einem zu j korrespondierenden Bereich gehört, anhält.

M enthält einen Algorithmus σ_1, der als

erstes das Paar $(0,0)$ und dann die anderen Paare $(i,j) \in \mathbb{N}_0 \times \mathbb{N}_0$ mit für festes i streng monoton wachsenden zweiten Komponenten aufzählt. Hierzu könnte z.B. das Standardverfahren zur Aufzählung von $\mathbb{N}_0 \times \mathbb{N}_0$ verwendet werden.

Ferner enthält M einen Algorithmus O_2, der für gegebenes (i,j) den rationalen Wert $p(i,j)$ berechnet.

Unter Verwendung von O_1 und von O_2 zählt M die Menge $\{ p(i,j) \mid (i,j) \in \mathbb{N}_0 \times \mathbb{N}_0 \}$ auf.

Sei $\ell \in \mathbb{N}_0$.

- $p(\ell, s(\ell))$ bezeichnet stets das zuletzt aufgezählte Paar mit erster Komponente ℓ.

  Falls kein solches Paar existiert, dann setzen wir $s(\ell) := -1$. und vereinbaren $p(\ell, -1) := 0$.

Die Rolle von p_{i+1} im Beweis von Satz 3.10 b) spielt nun

$$K := \sum_{\ell} p(\ell, s(\ell)).$$

Anpassung der Maschine M:

- M berechnet die erste rationale Zahl $p(0,0)$.

Annahme:

$b_0 b_1 \dots b_i \in \{0,1\}^*$ ist die von M bisher generierte Zufallsfolge und $p(r, s(r))$ die zuletzt aufgezählte rationale Zahl.

$\Rightarrow$

r ist dann die aktuelle Ausgabe.

Ausgaben, die zu zukünftig aufgezählten rationalen Zahlen korrespondieren, heißen zukünftige Ausgaben.

Seien $P_e(b_0 b_1 \dots b_i)$ und $P_g(b_0 b_1 \dots b_i)$ wie im Beweis von Satz 3.10 b) definiert.

M führt in Abhängigkeit, in welcher Relation $P_g(b_0 b_1 \dots b_i)$ und k zueinander stehen, folgende Schritte durch:

(1) Falls $P_g(b_0 b_1 \dots b_i) \leq k$, dann hält M mit der Ausgabe r an.

(2) Falls $P_g(b_0 b_1 \dots b_i) > k$, dann müssen wir für die korrekte Unterteilung von T in Bereiche zwei Fälle unterscheiden:

(2.1) Falls $P_e(b_0 b_1 \dots b_i) < k$, dann korrespondiert ein Teil des unter $b_0 b_1 \dots b_i$ hängenden Teilbaums von T zur aktuellen Ausgabe r, während der andere Teil zu zu-

künftigen Ausgaben korrespondiert.

- M generiert das nächste Zufallsbit b_{i+1} und verfährt bzgl. der erweiterten Folge $b_0 b_1 \dots b_{i+1}$ und dem unveränderten K wie oben beschrieben.

(2.2) Falls $P_e(b_0 b_1 \dots b_i) \geq K$, dann korrespondiert der gesamte unter $b_0 b_1 \dots b_i$ hängende Teilbaum zu zukünftigen Ausgaben.

- M zählt die nächste rationale Zahl $p(r', s')$ auf und aktualisiert K durch

$$K := K + (p(r', s') - p(r', s'-1)).$$

M verfährt bezüglich der unveränderten Zufallsfolge und dem aktualisierten K wie oben beschrieben.

Falls nun M mit der generierten Zufallsfolge $b_0 b_1 \dots b_i$ und der Ausgabe r anhält, dann gilt:

$$K - (p(r, s(r)) - p(r, s(r)-1)) < P_g(b_0 b_1 \dots b_i) \leq K$$

Nur wenn die rechte Ungleichung erfüllt ist, hält M an. Falls die linke Ungleichung nicht er-

fällt ist, denn hätte gemäß Konstruktion M nicht $p(r, s(r))$ aufgezählt und vorher angehalten.

Für alle $i \in \mathbb{N}_0$ summieren sich die Größen der zu i korrespondierenden Bereichen stets zu $p(i, s(i))$ auf.

Wegen $\lim_{n \to \infty} p(i,n) = p_i \quad \forall i \in \mathbb{N}_0$ haben wir somit gezeigt, dass M die Zahl i mit Wahrscheinlichkeit p_i ausgibt. ∎

Sei $p_0, p_1, p_2, \ldots$ eine von unten aufzählbare Folge von nichtnegativen reellen Zahlen mit $\sum_i p_i \leq 1$. Ferner seien S eine beliebige aber feste aufzählbare Menge und $s_0, s_1, s_2, \ldots$ eine beliebige, aber feste Aufzählung der Elemente in S. Eine Abbildung

$$m : S \to [0,1] \text{ mit } m(s_i) = p_i$$

heißt <u>von unten aufzählbares Semimaß auf S</u>.

<u>Bemerkung:</u>
Im Beweis von Satz 3.11 haben wir auf diese Art und Weise ein von unten aufzählbares Semimaß auf $\mathbb{N}_0$ konstruiert. Wir hätten anstatt $i \in \mathbb{N}_0$ auch das i-te Element in einer beliebigen Aufzählung von $\{0,1\}^*$ ausgeben können und somit ein von unten aufzählbares Semimaß auf

$\{0,1\}^*$ erhalten.

Wir werden für das im Beweis von

$\times$ Kolmogorov-Z.f. $\Rightarrow$ $\times$ Martin-Löf-Z.f.

konstruierte Semimaß m' nicht direkt

$$KP(z) \leq -\log m'(z) + C \quad \forall z \in \{0,1\}^*, C \text{ konstant}$$

beweisen. Zunächst zeigen wir, dass ein universelles von unten aufzählbares Semimaß m existiert. Für dieses beweisen wir

$$KP(z) \leq -\log m(z) + c \quad \forall z \in \{0,1\}^*, c \text{ konstant}$$

und verwenden dann dies um die benötigte obere Schranke für die Präfixkomplexitätsfunktion bezüglich m' zu beweisen.

Satz 3.12

Es existiert ein von unten aufzählbares Semimaß m auf $\mathbb{N}_0$ mit der Eigenschaft, dass für jedes von unten aufzählbare Semimaß m' auf $\mathbb{N}_0$ eine Konstante $c_{m'}$ existiert, so dass

$$m'(j) \leq c_{m'} \cdot m(j) \quad \forall j \in \mathbb{N}_0.$$

Beweis:

Oben haben wir uns überlegt, dass wir für jede rekursiv aufzählbare Menge ein Semimaß auf diese Menge konstruieren können. Das Wei-

teren können wir die Menge der von unten aufzählbaren Semimaße aufzählen, indem wir alle probabilistischen Turingmaschinen ohne Eingabe aufzählen.

$\rightsquigarrow$

Idee:

Definiere das Semimaß m als ein Semimaß auf der Menge aller von unten aufzählbaren Semimaße.

Durchführung:

- Sei M eine probabilistische Maschine, die alle probabilistische Maschinen ohne Eingabe $M_0, M_1, M_2, \ldots$ aufzählt.
- Sei $p_0, p_1, p_2, \ldots$ eine von unten aufzählbare Folge von reellen Zahlen mit $p_i > 0 \ \forall i \in \mathbb{N}_0$ und $\sum_i p_i \leq 1$.
- Die Maschine M wählt zufällig mit Wahrscheinlichkeit p_i ein $i \in \mathbb{N}_0$ und simuliert dann die Maschine M_i.

Sei m' ein beliebiges aber festes von unten aufzählbares Semimaß auf $\mathbb{N}_0$.

Satz 3.11 $\Rightarrow$

$\exists$ Maschine M_e, $e \in \mathbb{N}_0$, die das Semimaß m' auf $\mathbb{N}_0$ konstruiert.

Bezeichne m das von M konstruierte Semimaß auf $\mathbb{N}_0$. D.h., $\forall j \in \mathbb{N}_0$ ist $m(j)$ die Wahrscheinlichkeit, dass M mit Ausgabe j anhält. Dann gilt $\forall j \in \mathbb{N}_0$

$$m(j) \geq p_e m'(j) \iff m'(j) \leq p_e^{-1} m(j).$$

Für $c_{m'} := p_e^{-1}$ ergibt sich somit

$$m'(j) \leq c_{m'} m(j) \quad \forall j \in \mathbb{N}_0.$$

Somit haben wir für einen String $x \in \{0,1\}^*$ zwei Maße entwickelt. Das Maß $m(x)$, wobei m ein universelles von unten aufzählbares Semimaß auf $\{0,1\}^*$ ist, misst die Wahrscheinlichkeit, dass x die Ausgabe einer probabilistischen Maschine ohne Eingabe ist. Die Präfixkomplexität $KP(x)$ misst die Schwierigkeit, x mit Hilfe eines Präfixdekompressionsalgorithmus zu rekonstruieren.

Satz 3.13

Sei m ein universelles von unten aufzählbares Semimaß auf $\{0,1\}^*$. Dann existiert eine Konstante c, so dass

$$|-\log m(x) - KP(x)| \leq c \quad \forall x \in \{0,1\}^*.$$

Beweis

Wir zeigen zunächst

$$KP(x) \geq -\log m(x) + c_1 \quad \forall x \in \{0,1\}^*$$

für eine Konstante c_1 und dann

$$KP(x) \leq -\log m(x) + c_2 \quad \forall x \in \{0,1\}^*$$

für eine Konstante c_2.

Dann folgt direkt

$$|-\log m(x) - KP(x)| \leq c \quad \forall x \in \{0,1\}^*,$$

wobei $c := \max\{c_1, c_2\}$.

$KP(x) \geq -\log m(x) + c_1$:

Idee

Konstruktion eines von unten aufzählbaren Semimaßes auf $\{0,1\}^*$ mit Hilfe der Präfix-Komplexitätsfunktion und Anwendung des Satzes 3.12.

Durchführung:

Präfixkomplexitätsfunktion von oben rekursiv aufzählbar

$\Rightarrow$

(*) Abbildung $w: \{0,1\}^* \to [0,1]$ mit $w(x) := 2^{-KP(x)}$ ist von unten aufzählbar.

(**) Satz 3.7 $\Rightarrow \sum_x 2^{-KP(x)} \leq 1.$

(*) und (**) $\Rightarrow$

$\{ 2^{-KP(x)} \mid x \in \{0,1\}^* \}$ kann zur Konstruktion eines von unten aufzählbaren Semimaßes auf $\{0,1\}^*$ verwendet werden

Satz 3.12 $\Rightarrow$

$\exists$ Konstante c_0, so dass $\forall x \in \{0,1\}^*$

$$2^{-KP(x)} \leq c_0 \, m(x)$$

$$\Leftrightarrow \quad KP(x) \geq -\log m(x) - \log c_0$$

$$= -\log m(x) + c_1,$$

wobei $c_1 := -\log c_0$.

<u>$KP(x) \leq -\log m(x) + c_2$:</u>

<u>Idee</u>:

Konstruktion eines Präfixdekompressionsalgorithmus unter Verwendung von m und Anwendung der Kraft-Ungleichung.

<u>Durchführung</u>:

m Semimaß $\Rightarrow$

(***) $\sum_i 2^{\log m(i)} = \sum_i m(i) \leq 1.$

Betrachten wir die Folge

$$\lceil -\log m(0) \rceil, \lceil -\log m(1) \rceil, \lceil -\log m(2) \rceil, \ldots$$

$(***) \Rightarrow$

$$\sum_i 2^{-\lceil -\log m(i) \rceil} \leq 1.$$

Wir haben die Kraft-Ungleichung für endliche Folgen von Kodewortlängen bewiesen. Eine einfache Modifikation des Beweises liefert uns solche auch für unendliche Folgen von Kodewortlängen.

Übung:
Formulieren und beweisen Sie die verallgemeinerte Kraft-Ungleichung für unendliche Folgen von Kodewortlängen.

verallgemeinerte Kraft-Ungleichung $\Rightarrow$

$\exists$ präfixfreier Binärkode ψ für die Kodewortlängen $\lceil -\log m(0) \rceil, \lceil -\log m(1) \rceil, \lceil -\log m(2) \rceil, \ldots$

Kode präfixfrei $\Rightarrow$

Die Abbildung, die das Kodewort $\psi(x)$ auf x abbildet, ist eine Präfixfunktion.

$\rightsquigarrow$

Ziel:
Konstruktion von ψ nebst Entwicklung eines

Algorithmus, der $\psi(x)$ auf x abbildet.

Schwierigkeit:

a) Wir können nicht wie im Beweis der Kraft-Ungleichung annehmen, dass die Kodeworthängen in obiger Folge aufsteigend sortiert sind.

$\Rightarrow$

Kode ψ kann nicht wie im Beweis der Kraft-Ungleichung konstruiert werden.

b) m und somit auch obige Folge kann nicht berechnet werden.

Beobachtung:

- Es ist deshalb erlaubt, dass es für einen Präfix-Dekompressionsalgorithmus mehrere Eingaben geben kann, die dasselbe x rekonstruieren
- m ist ein von unten aufzählbares Semimaß (d.h., $\exists$ berechenbare totale Funktion

$$M: \mathbb{N}_0 \times \mathbb{N}_0 \to \mathbb{Q}, \text{ so dass } \forall i \in \mathbb{N}_0 \text{ gilt:}$$

1) $M(i,0) \leq M(i,1) \leq M(i,2) \leq \ldots$ und

2) $\lim_{t \to \infty} M(i,t) = m(i)$.

- Der im Beweis von Satz 3.11 konstruierte Trie korrespondiert zu einem präfixfreien ~~Binär~~

Binärkode für $\mathbb{N}_0$. Dabei ist der zu einem Blatt, in dem M mit Ausgabe r anhält, korrespondierende Zufallsstring $b_0 b_1 \dots b_i$ das bezüglich dieses Blattes konstruierte Kodewort für r. Die zum Kodewort $b_0 b_1 \dots b_i$ korrespondierende Wahrscheinlichkeit beträgt $2^{-(i+1)}$.

↪

Idee:

Konstruktion eines Präfixdekompressionsalgorithmus, der, gegeben ein Kodewort $b_0 b_1 \dots b_i$ für $r \in \mathbb{N}_0$ den r-ten String x in einer Aufzählung von $\{0,1\}^*$ ausgibt.

Bemerkung:
Dies führt nur zum gewünschten Resultat, wenn es für jeden $r \in \mathbb{N}_0$ ein Kodewort der Länge $\leq -\log m(r) + d$ existiert, wobei d eine Konstante ist, die nicht von r abhängt.

Um diese Eigenschaft herbeizuführen betrachten wir die Konstruktion des Trees T im Beweis von Satz 3.11.

- Denjenigen linken Teil von T, der exakt zu den Zufallsfolgen, bei denen die probabilistische Maschine M terminiert, korrespondiert, haben wir in Bereiche unterteilt.

Bereich B:

- korrespondiert eindeutig zu einem $r \in \mathbb{N}_0$, welches Ausgabe bzgl. jedes Blattes innerhalb B ist.
- Größe $|B|$ des Bereiches B:

$$q := p(r,\ell) - p(r,\ell-1) \text{ für ein } \ell \geq 0,$$

wobei $p(r,-1) := 0$.

Bezeichnungen:

- $t(b)$, $b \in B$ Blatt — Tiefe des Blattes b in T
- $t_B := \min\{t(b) \mid b \in B\}$ minimale Tiefe eines Blattes in B.

Beobachtung:
In einem Bereich B der Größe q existiert stets ein Blatt der Tiefe $\leq -\log q + 2$.
D.h.,

$$t_B \leq -\log q + 2.$$

Bew.:

Konstruktion $\Rightarrow$

- $q = \sum_{b \in B} 2^{-t(b)}$.
- Jeder innere Knoten von B hat maximal ein Blatt als Sohn (ansonsten

wäre v selbst ein Blatt).

$\Rightarrow$

B enthält auf jedem Level maximal zwei Blätter

$\Rightarrow$

$$q \leq 2 \cdot \sum_{t=t_B}^{\infty} 2^{-t}$$

Wegen $\sum_{t=t_B}^{\infty} 2^{-t} \leq 2^{-t_B+1}$ erhalten wir

$$q \leq 2^{-t_B+2}$$

$$\Leftrightarrow t_B \leq -\log q + 2.$$

Bemerkung:
Falls wir zeigen könnten, dass eine Konstante d existiert, so dass für jedes $r \in \mathbb{N}_0$ in T ein zu r korrespondierender Bereich der Größe $\geq d \cdot m(r)$ enthalten ist, dann würde jedes $r \in \mathbb{N}_0$ ein hinreichend kurzes Kodewort besitzen. Jedoch stellt die Abbildung $M: \mathbb{N}_0 \times \mathbb{N}_0 \to \mathbb{Q}$ dies nicht notwendigerweise sicher.

Idee:
Modifiziere M zu einer berechenbaren Abbildung $M': \mathbb{N}_0 \times \mathbb{N}_0 \to \mathbb{Q}$, die obige Eigenschaft sicherstellt.

Durchführung:

Für alle $i \in \mathbb{N}_0$ konstruieren wir eine Teilfolge $M'(i,0), M'(i,1), M'(i,2), \ldots$ der Folge $M(i,0), M(i,1), \ldots$, so dass folgendes erfüllt ist:

1) $M'(i,0) > 0$

2) $M'(i,j) \leq \frac{1}{2} m(i) \Rightarrow M'(i,j+1) \geq 2 M'(i,j)$ für $j \geq 0$ und

3) $\exists \ell_i \geq 0$ mit $M'(i,\ell_i) \geq \frac{1}{2} m(i)$.

Dann gilt:

$$M'(i,\ell_i) - M'(i,\ell_i - 1) \geq \frac{1}{4} m(i).$$

Sei $i \in \mathbb{N}_0$ beliebig, aber fest.

Für die Konstruktion von M' definieren wir

$$j_0 := \min \{ j \geq 0 \mid M(i,j) > 0 \}$$

und für $\ell > 0$

$$j_\ell := \min \{ j > j_{\ell-1} \mid M(i,j) \geq 2 M(i,j_{\ell-1}) \}$$

$\Rightarrow$

Die Teilfolge

$$M'(i,0), M'(i,1), M'(i,2), \ldots$$

mit

$$M'(i,\ell) := M(i,j_\ell) \quad \text{für } \ell \geq 0$$

leistet das Gewünschte.

Beobachtung:

- Für jedes $i \in \mathbb{N}_0$ existiert ein s mit

$$M(i, j_s) \geq \frac{1}{2} m(i).$$

Konstruktion $\Rightarrow$

$M(i, j_\ell)$ ist für $\ell > s+1$ nicht definiert.

$\Rightarrow$

Obige Teilfolge besitzt nur endlich viele Glieder.

Für $i \in \mathbb{N}_0$ bezeichne k_i die zweite Komponente des letzten Gliedes der Teilfolge mit erster Komponente i. Dann definieren wir für $i \in \mathbb{N}_0$

$$m'(i) := M'(i, k_i).$$

$\Rightarrow$

$$\sum_i m'(i) \leq \sum_i m(i) \leq 1.$$

$\Rightarrow$

m' ist ein Semimaß auf $\mathbb{N}_0$.

$\Rightarrow$

m' ist ein Semimaß auf $\{0,1\}^*$.

- Da die von uns konstruierte Abbildung $M' : \mathbb{N}_0 \times \mathbb{N}_0 \to \mathbb{Q}$ nicht total ist, ist gemäß unserer Definition m' nicht not-

wendigerweise von unten aufzählbar.

Idee:
Konstruiere trotzdem den Trie unter Verwendung von M' anstatt von M.

Frage:
Wie gehen wir mit dem Problem um, dass für ein aufgezähltes Paar (i,j) nicht sichergestellt werden kann, dass $M'(i,j)$ berechnet wird?

Lösung:

Anstatt nach Aufzählung eines Paares (i,j) den Wert $M'(i,j)$ zu berechnen, zählen wir stets für alle bisher aufgezählten Paare (r,e) die Werte $M'(r,e)$ parallel und verschränkt auf.

Übung:
Konstruieren Sie den Trie T unter Verwendung von M' anstatt von M. Führen Sie Ihre Konstruktion unter Ausnutzung der Definition von M' und der Berechenbarkeit der totalen Funktion M möglichst geschickt durch.

Der resultierende Trie T' enthält für jedes $r \in \mathbb{N}_0$ einen Bereich der Größe $\geq \frac{1}{4} m(r)$

$\Rightarrow$
T' enthält für jedes $r \in \mathbb{N}_0$ ein Kodewort

der Länge

$$\leq \lceil -\log \tfrac{1}{4} m(\tau) \rceil + 2$$

$$\leq -\log m(\tau) + 5.$$

Frage:
Wie sieht nun der Präfix-Dekompressionsalgorithmus konkret aus?

Idee:
Konstruiere unter Verwendung des gegebenen Kodewortes anstatt der Zufallsfolge den Trie T'.

Durchführung: Übung

Da der konstruierte Präfix-Dekompressionsalgorithmus konstante Länge hat, folgt somit die Behauptung. ∎

Satz 3.14
Sei $x = x_0 x_1 x_2 \ldots \in \Omega$ eine Kolmogorov-Zufallsfolge. Dann ist x auch eine Martin-Löf-Zufallsfolge

Beweis:

Erinnerung:
$x = x_0 x_1 x_2 \ldots$ ist eine Kolmogorov-Zufallsfolge

$\Leftrightarrow \exists$ Konstante d mit
$K(x_0 x_1 \ldots x_{n-1}) \geq n - d \quad \forall n \in \mathbb{N}.$

Annahme:

x ist keine Martin-Löf-Zufallsfolge.

$\Rightarrow$

$\{x\}$ ist eine effektive Nullmenge.

Sei N die Vereinigung aller effektiven Nullmengen und $\mathcal{O}_N$ der im Beweis von Satz 3.1 konstruierte Überdeckungsalgorithmus für N.

Betrachte $c \in \mathbb{N}$ beliebig aber fest. Sei

$$u(c,0),\ u(c,1),\ u(c,2),\ \ldots$$

die vom Überdeckungsalgorithmus $\mathcal{O}_N$ bei der Eingabe $\varepsilon := 2^{-2c}$ generierte Folge von binären Strings.

$\Rightarrow$

$$\forall c \in \mathbb{N}\ \exists i \in \mathbb{N}_0 \text{ mit } \{x\} \subseteq \Omega_{u(c,i)}.$$

Für jedes Paar $(c,i) \in \mathbb{N} \times \mathbb{N}_0$ sei

$$n(c,i) := |u(c,i)| - c.$$

Für gegebenes $c \in \mathbb{N}$ gilt dann

$$\sum_i 2^{-n(c,i)} = \sum_i 2^{-|u(c,i)|+c}$$

$$= 2^c \sum_i 2^{-|u(c,i)|}$$

$$\leq 2^c \cdot 2^{-2c} = 2^{-c}$$

$\Rightarrow$

$$\sum_{c,i} 2^{-n(c,i)} \leq \sum_{c} 2^{-c} = 1$$

Somit ist $m' : \{0,1\}^* \to [0,1]$ mit

$$m'(z) := \sum_{(c,i) : u(c,i) = z} 2^{-n(c,i)}$$

ein Semimaß auf $\{0,1\}^*$.

Übung:
Zeigen Sie, dass das Semimaß m' von unten aufzählbar ist.

Ziel:
Beweis, dass die Folge x keine Kolmogorov-Zufallsfolge ist. D.h., wir zeigen, dass für alle Konstanten $d \in \mathbb{N}$ ein n_d existiert mit

$$K^P(x_0 x_1 \dots x_{n_d - 1}) < n_d - d.$$

Sei hierzu m ein universelles von unten aufzählbares Semimaß auf $\{0,1\}^*$.

Satz 3.12 $\Rightarrow$

$\exists$ Konstante $c_{m'}$, so dass $\forall z \in \{0,1\}^*$

$$m'(z) \leq c_{m'} \, m(z)$$

$\Leftrightarrow$

$$(*) \quad m(z) \geq c_{m'}^{-1} \, m'(z).$$

Satz 3.13 $\Rightarrow$

$\exists$ Konstante $\tilde{c}$, so dass $\forall\, z \in \{0,1\}^*$

$$\begin{aligned} K_P(z) &\leq -\log m(z) + \tilde{c} \\ &\leq -\log c_{m'}^{-1} m'(z) + \tilde{c} \\ &\overset{(*)}{=} -\log m'(z) + C, \end{aligned}$$

wobei $C := \tilde{c} - \log c_{m'}^{-1}$ eine Konstante ist, die nicht von z abhängt.

Also gilt $\forall\, (c,i) \in \mathbb{N} \times \mathbb{N}_0$

$$\begin{aligned} K_P(u(c,i)) &\leq -\log m'(u(c,i)) + C \\ &= -\log\Big(\sum_{(c',i'):\, u(c',i') = u(c,i)} 2^{-n(c',i')} \Big) + C \\ &\leq -\log 2^{-n(c,i)} + C \\ &= n(c,i) + C \\ &= |u(c,i)| - c + C \end{aligned}$$

Betrachte $d \in \mathbb{N}$ beliebig aber fest. Wähle $c \in \mathbb{N}$ mit $c > d + C$. Dann erhalten wir

$$\begin{aligned} K_P(u(c,i)) &\leq |u(c,i)| - c + C \\ &< |u(c,i)| - d. \end{aligned}$$

Also haben wir bewiesen, dass jede Folge, die keine Martin-Löf-Zufallsfolge ist, auch keine Kolmogorov-Zufallsfolge ist. Somit ist jede Kolmogorov-Zufallsfolge auch eine Martin-Löf-Zufallsfolge. ∎

3.2 Endliche Zufallsfolgen

- Einige Bemerkungen zu endlichen Bitfolgen.

⇝

Die korrekte Frage ist nicht

" Ist eine gegebene Folge von Nullen und Einsen der Länge n zufällig? "

Sondern

" Um wieviel ist eine gegebene Folge von Nullen und Einsen der Länge n zufällig? "

gesucht:
Maß für den Grad der Zufälligkeit einer endlichen Folge von Nullen und Einsen.

Idee:
Definiere den Grad der Zufälligkeit eines Strings x in Abhängigkeit einer endlichen Menge M, die x enthält.

⇝

Wir benötigen die bedingte Kolmogorov-Komplexität von x, wenn eine endliche Menge M von endlichen binären Strings mit $x \in M$ bekannt ist.

Beobachtung:
Jede solche Menge M kann durch einen

String $y_M \in \{0,1\}^+$ kodiert werden.

$\Rightarrow$

Diese Situation ist von unserer Definition der bedingten Kolmogorov-Komplexität umfasst.

$K(x|M)$ bezeichnet die bedingte Kolmogorov-Komplexität von x, wenn M bekannt ist.

Frage:
Was ist eine triviale obere Schranke für $K(x|M)$?

Beobachtung:
x kann einfach durch seinen Index in der lexikographischen Ordnung von M spezifiziert werden.

$\Rightarrow$

$(*)$ $\quad K(x|M) \leq \log |M| + c,$

wobei c eine Konstante ist, die nicht von x oder M abhängt.

Für $x \in M$ definieren wir den Defekt $d(x|M)$ der Zufälligkeit von x relativ zu M durch

$$d(x,M) := \log |M| - K(x|M)$$

$d(x,M)$ groß $\Rightarrow$ Existenz einer wesentlich kürzeren als obige Standardbeschreibung.
$\Rightarrow$ x besitzt spezielle Struktur und ist nicht zufällig.