Auswertung endlicher Bitfolgen

Nicolas Windt

21. September 2020

Einführung

Den Anlass zu dieser Abhandlung gibt der 1934 von Karl Popper in „Logik der Forschung" erbrachte Nachweis, dass das Gesetzt der großen Zahlen tautologisch ist. In wenigen Worten gefasst bedeutet es, dass die Existenz von Grenzwerten gemäß dem Gesetzt der großen Zahlen zwar nicht bestritten wird, jedoch braucht es dazu kein Axiom, sie ergibt sich deduktiv aus der Forderung der Nachwirkungsfreiheit und ihren logischen Konsequenzen. Die tautologische Eigenschaft des Grenzwertaxioms wird hier nicht diskutiert und deshalb weder verworfen, noch bestätigt. Sie ist lediglich der Anreiz zu dieser Veröffentlichung.

In dieser Abhandlung wird zuerst ein Auswertungsverfahren auf endlichen Bitfolgen entworfen, der Einschritttest, und das Ergebnis seiner Anwendung auf die Ausgabe eines beliebigen sog. (pseudeo) Zufallsalgorithmus analysiert und formal untersucht. Es wird sich experimentell zeigen, dass die Häufigkeitsverteilung der Läufe und Teilläufe der Ausgaben eines sog. Zufallsgenerators sich einer exponentiell gearteten Verteilung annähert. Motiviert durch dieses experimentelles Ergebnis wird die Struktur der Haüfigkeitsverteilungen der Läufe solcher exponnentiell gearteten Bitgolge formal untersucht und damit die experimentellen Beobachtungen erklärt. Schließlich wird ein Algorithmus entworfen, der als Eingabe eine beliebiege natürliche Zahl und eine beliebig, hinreichend lange Bitfolge bzw. einen Bitstrom annimmt, um eine Bitfolge mit exponentiell gearteten kumulierten Häufigkeitsverteilung ihrer Läufe in der geforderteten Länge ausgibt.

Es gibt keinen Nachweis dafür, dass die Methoden der Forschungslogik die besseren Ergebnisse liefert, oder dass sie überhaupt irgendein Ergebnis liefern können. Ein solcher Nachweis kann es auch nicht geben. Lediglich glauben wir mit der Logik der Forschung, die besseren Ergebnisse zu erzielen. Indes ist die Wissenschaft eine Weltanschauung, die mit allen anderen Weltanschauungen in stetigem Wettbewerb tritt, sich jeden Tag aufs neue bewähren muss, und die es jedem überlassen ist, sie zu vertreten oder auch nicht.

Der Autor, nach Karl Popper

Inhaltsverzeichnis

1 Läufe und Teilläufe

Wir definieren die Menge der endlichen Bitfolgen.

Definition 1. *Menge der endlichen Bitfolgen*

$$\mathcal{B} := \{(b_m)_{0\leq m<M<\infty}, b_m \in \{0,1\}\} \tag{1.1}$$

Wir gehen nun von einer beliebigen endlichen Bitfolge aus $\mathcal{B}$

$$(b_m)_{0<m\leq M,M<\infty} \in \mathcal{B} \tag{1.2}$$

aus. Zum Beispiel

$$1000101010011111000001111111000101010 1 \tag{1.3}$$

Diese Bitfolge können wir auch als Folge von Läufen auffassen. Dabei ist ein Lauf eine Folge von gleichbleibende Bits. In unserem Beispiel sind es

$$1 \cdot 000 \cdot 1 \cdot 0 \cdot 1 \cdot 0 \cdot 1 \cdot 00 \cdot 11111 \cdot 00000 \cdot 1111111 \cdot 000 \cdot 1 \cdot 0 \cdot 1 \cdot 0 \cdot 1 \cdot 0 \cdot 1 \tag{1.4}$$

Mit $\mathcal{L}$ definieren wir die Menge der Läufe.

Definition 2.

$$\mathcal{L} := \{(r_n)_{0<n\leq N<\infty},\ r_n \in \{0,1\}\,,\ a_i = a_j\ \forall i,j\} \tag{1.5}$$

Eine Bitfolge ist dem entsprechend eine Folge von Läufen. Das heißt

$$\forall (b_m)_{0<m\leq M<\infty}, b_m \in \{0,1\},\ \exists!(r_n)_{0<n\leq N<\infty} \in \mathcal{L} \mid (b_m) \equiv (r_n)\ \textit{oder}\ (\overline{b}_m) \equiv (r_n) \tag{1.6}$$

Die Gleichung (**??**) ergibt sich trivialerweise aus der Definition der Bitfolgen (b_m) und der Folgen von Läufen (r_n). Sie besagt nur, dass jede Folge von Läufen (r_n) einer eindeutigen Bitfolge (b_m) bis auf ihr binäres Komplement $(\bar{b}_m)$ entspricht.

Als unmittelbarer Folge der Definition (**??**) der Menge $\mathcal{L}$ kann trivialerweise mittels der Länge der Läufe eine bijektive Abbildung auf $\mathbb{N}$ hergeleitet werden

$$\mathcal{L} \equiv \mathbb{N} \tag{1.7}$$

Es müssen nicht mehr die Elemente aus $\mathcal{L}$ explizit genannt werden, sondern nur Elemente aus $\mathbb{N}$.

Des weiteren führen wir die triviale Überlegung an, dass der eindeutig bestimmte Lauf $N \in \mathcal{L}$ bzw. $N \in \mathbb{N}$ auch Läufe $\mathcal{L} \ni s < N$ mit entsprechender Häufigkeitsverteilung

$$_NS = (_Ns_n)_{0<n\leq N},\ mit\ _Ns_n = N - n + 1 \tag{1.8}$$

beinhaltet. Wir nehmen zur Illustration den Lauf $N = 8$

$$1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ bzw.\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0 \tag{1.9}$$

Dann beinhaltet N:
$_8s_1 = 8 - 1 + 1 = 8$ Läufe 1
$_8s_2 = 8 - 2 + 1 = 7$ Läufe 2
$_8s_3 = 8 - 3 + 1 = 6$ Läufe 3
$_8s_4 = 8 - 4 + 1 = 5$ Läufe 4
$_8s_5 = 8 - 5 + 1 = 4$ Läufe 5
$_8s_6 = 8 - 6 + 1 = 3$ Läufe 6
$_8s_7 = 8 - 7 + 1 = 2$ Läufe 7
$_8s_8 = 8 - 8 + 1 = 1$ Lauf 8

Um den Wortgebrauch zur erleichtern, vereinbaren wir die folgende Bezeichnung

Definition 3. *Der Teillauf*

Die in einem Lauf $N \in \mathcal{L}$ enthaltenen Läufe heißen fortan Teilläufe

2 Ein motivierendes Experiment

Wir greifen auf ein experimentelles Mittel zurück, um eine Bitfolge zu Erzeugen, dessen Struktur nicht kausal erklärt werden sollte, eine sog. Zufallsfolge. Ein solches experimentelles Mittel ist zum Beispiel das Gerät /dev/random einer beliebigen `linux` Installation, das den Ruf als „notorisch guter"Lieferant für solchen Bitfolgen genießt, auch wenn er relativ langsam arbeitet. Darüber hinaus kann seine Arbeitsweise Lückenlos nachgeprüft werden.
Wir sammeln die Ausgabe von /dev/random auf einer oder mehreren `linux`-Instanzen bis auf einen Volumen von ca. $16 \cdot 10^6$ Bits. Auf einer Doppelkernmaschine und in Rahmen einer Büronutzung dauert dies ca. ein Jahr. Dies ist mit einem einfachen Aufruf des Kommandos `dd` auf der Konsole getan. In dem hier vorgestellten Fall wurde diese Arbeit mit einem Skript im Hintergrund erledigt.

Das Ergebnis dieses experimentellen Vorgehens ist eine endliche Bitfolge, dessen Länge M durch das Experiment bedingt ist.

$$(b_m)_{0<m\leq M<\infty},\ b_m \in \{0,1\} \tag{2.1}$$

Wie wir es im Kapitel **??** über Läufe und Teilläufe schon angeführt haben, fassen wir nun die Folge (b_m) aus (**??**) als eine Folge aus $\mathcal{L}$ bzw. $\mathbb{N}$

$$(r_n)_{0<n\leq N},\ r_n \in N,\quad N \text{ geeignet} \tag{2.2}$$

Um die Bitfolge (**??**) auszuwerten, ermitteln wir die kumulierte Verteilung

$$L = (l_p)_{0<p\leq P<\infty} \tag{2.3}$$

der Häufigkeiten l_p der Läufe und der darin enthaltenen Teilläufe der Folge $(r_n)_{0<n\leq N<\infty}$.
An der Stelle motivieren wir diesen Ansatz durch die folgende Überlegung: Aus dem zentralen Grenzwertsatz leiten wir ab, dass für das m.te Bit der Ausgabe $(b_m) \in \mathcal{B}$ eines sog. Zufallsalgorithmus die Gleichung $P(b_{n-1} = b_n) = P(b_{n-1} \neq b_n)$ gilt. Da wir auf den zentralen Grenzwertsatz nicht weiter zurückgreifen wollen, bevorzugen wir es, für jeden Bit B_m aus (b_m), die Häufigkeit der Fälle $(b_m) = (b_{m+i})$ zu betrachten. Dies führt uns zu dem Ansatz der kumulierten Verteilungen: Wir fassen $(b_m) \in \mathcal{B}$ als Folge $(r_n) \in \mathcal{L}$ und ermitteln die Häufigkeiten ihrer Läufe und der darin enthaltene Teilläufe. Es gilt auf triviale Weise, dass der größte Index P der Folge (**??**) durch den längsten Lauf von (r_n) gegeben ist.

$$P = \max\{r_1, \ldots, r_N\} \tag{2.4}$$

Aus der vorangegangenen Überlegungen ergibt sich die folgende Vorschrift, die wir hier mit der Bezeichnung Einschritttest versehen

Algorithmus 1. *Der Einschrittest für Folgen von Läufen*

```
Eingabe: Eine beliebig Folge von Läufe der Länge N, N enldich
lege einen array L der Länge maximal N
für jeden Lauf der Folge
 ermittle die Länge des Laufs (hier l)
 für jede Stelle i=1 bis l
     erhöhe die Häufigkeit an der Stelle i um l-i+1
Gib L aus
```

Man kann das selbe Verfahren auf die der Folge von Läufe $(r_n) \in \mathcal{L}$ zu Grunde liegende Bitfolge (b_m) bzw. $(\overline{b}_m)$ anwenden. Es ergibt sich eine etwas andere Rechenvorschrift. Sie liefert allerdings das selbe Ergebnis wie der Algorithmus **??**.

Algorithmus 2. *Der Einschrittest für Bitfolgen*[1]

```
Eingabe: Eine beliebig lange endliche Folge von Bits der Länge P, P enldich
Lege einen array L der Länge maximal P
Lege einen Zäher z, mit dem Wert 0
Für jeden Bit der Folge
 hat sich den Wert des Bit im Vergleich zur vorherigen Schritt geändert?
 wenn nein dann erhöhe den Wert vom Zähler z um 1
 wenn ja dann stelle den Zähler z zurück auf 1
 Für jede Stelle i=1 bis z in L
    erhöhe die Häufigkeit an der Stelle i um 1
Gib L aus
```

Das Ergebnis der Auswertung der Ausgabe des Gerätes /dev/random ist in der Tabelle **??** protokolliert. Es fällt sofort auf, dass die Verteilung der Läufe und Teilläufe der Ausgabe des Gerätes /dev/random exponentiell fallend verläuft.
Wir stellen diese Ergebnisse in der Graphik **??** graphisch dar. Die in der Graphik **??** und in der Tabelle **??** als theoretischen Werte ausgewiesene Reihe besteht lediglich aus den Potenzen der Funktion 2^{25-i}, und dient der Veranschaulichung eines exponentiell abfallenden Verlaufs. Im Sinne der intersubjektiven Nachprüfung dieses Experimentes wird der Leser ausdrücklich dazu ermutigt, dieses Experiment mit den Mitteln seiner Wahl nachzustellen. Wie im unserem Fall sollte der Ausgang des Experiment einen tendenziell exponentiell abfallenden Verlauf aufweisen.

Diese Beobachtung lässt vermuten, dass die Ausgabe des Gerätes /dev/random oder irgendeines Gerätes, das den Zweck erfüllt, Bitfolgen ohne kausale Erzeugung zu liefern, einen annähernd exponentiell fallenden Verlauf der Verteilung ihrer Läufe und enthaltenen Teilläufe aufweist. Widersprüchlicherweise wäre damit doch eine Struktur des Zufalls erklärt. Wir widmen uns in den nächsten Kapiteln der Untersuchung solchen Bitfolgen ohne Rücksicht auf die Art, wie sie gewonnen werden, zu nehmen.

[1] Der Name „Einschritttest" wurde aus der Tatsache hergeleitet, dass in dieser Auffassung des Algorithmus, welche chronologisch als erste entstanden ist, die Eingabe Schritt für Schritt abgearbeitet wird.

Länge	Häufigkeit	log_2	th. Wert	Länge	Häufigkeit	log_2	th. Wert
1	16170866	23, 95	24	13	4021	11, 97	12
2	8087159	22, 95	23	14	2034	10, 99	11
3	4044093	21, 95	22	15	1011	9, 98	10
4	2021615	20, 95	21	16	504	8, 98	9
5	1010537	19, 95	20	17	251	7, 97	8
6	494406	18, 95	19	18	133	7, 06	7
7	251941	17, 94	18	19	69	6, 11	6
8	25921	16, 94	17	20	33	5, 04	5
9	2953	15, 94	16	21	15	3, 91	4
10	1469	14, 94	15	22	7	2, 81	3
11	5776	13, 95	14				
12	7969	12, 96	13				

Tabelle 2.1: Häufigkeit der Läufe und Teilläufe der Ausgabe von /dev/random einer beliebigen `l`inux Installation. In der zweiten Spalte werden die tatsächlichen Häufigkeiten aufgelistet und in der dritten Ihre Entsprechung auf einer logarithmischen Skala auf Basis 2. Die theoretischen Werte sind jene der Funktion 2^{25-n}

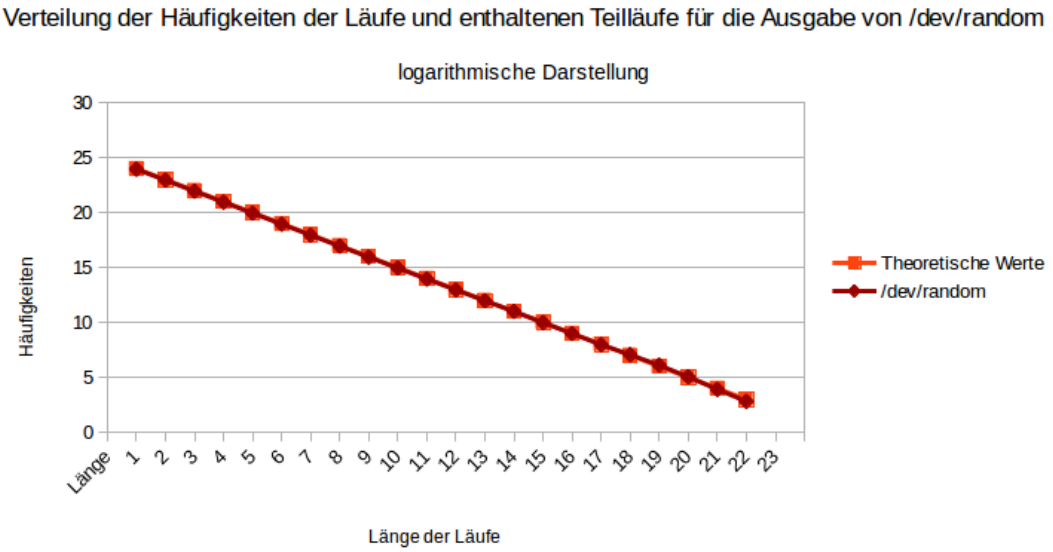

Abbildung 2.1: graphische Darstellung der Ergebnisse aus der Tabelle **??**.

3 Bitfolgen mit exponentiell kumulierten Häufigkeitsverteilungen

Motiviert durch die im Kapitel **??** beobachteten Ergebnisse wollen wir gleich einige Definitionen und Bezeichnungen Einführen. Zunächst fassen wir formal die Verteilung der Häufigkeiten der Läufe und enthaltenen Teilläufe einer Bitfolge auf.

Definition 4. *Sei* $({}_N b_m)$ *eine Bitfolge aus* $\mathcal{B}$ *und* $(r_n)_{0<n\leq N<\infty}$ *ihre entsprechende Folge von Läufe. Wir bezeichnen mit der Folge*

$$ {}_N L := ({}_N l_i)_{0<i\leq N<\infty} \tag{3.1}$$

die Verteilung der Häufigkeiten der Läufe und enthaltenen Teilläufe der Bitfolge $({}_N b_m)$ *und nennen sie die kumulierte Verteilung.*

Unter der exponentiell fallenden, kumulierten Verteilung ${}_N L := (l_i)_{0<i\leq N<\infty}$ einer Bitfolge $({}_N b_m) \in \mathcal{B}$ verstehen wir die

Forderung 1.

$$ \forall\ 0 < n < N < \infty,\ {}_N l_n = 2^{N-n} \tag{3.2}$$

Äquivalent dazu ist die

Forderung 2.

$$ {}_N l_N = 1 \tag{3.3}$$
$$ {}_N l_{n-1} = 2\ {}_N l_n\ ,\ 2 \leq n \leq N \tag{3.4}$$

Wir machen darauf aufmerksam, dass die kumulierte Verteilung einer Bitfolge nicht hinreicht, um diese Folge vollständig zu charakterisieren. Viel mehr entspricht diese Verteilung einer Klasse von Bitfolgen. Daher die folgende Definition

Definition 5. *Wir bezeichnen mit B_N die Menge der Folgen mit längstem Lauf N und mit exponentiell fallender, kumulierter Verteilung gemäß (??) bzw. (??) und (??).*

Die Klassen B_N fassen wir in der folgenden Definition zusammen:

Definition 6. *Menge der Klassen der Bitfolgen mit exponentiell fallenden, kumulierten Verteilung*

$$\mathcal{B}_{\mathcal{Z}} := \{B_N, 0 < N < \infty\} \tag{3.5}$$

Nachdem die grundlegenden Definitionen gestellt wurden, können wir uns der Untersuchung der Elementen aus einem beliebigen B_N widmen. Aus der Definition der exponentiell fallenden, kumulierten Verteilungen einer Bitfolge aus $\mathcal{B}$ ausgehend, kann untersucht werden, ob die Verteilung der Häufigkeiten ihrer Läufe ohne Teilläufe einer Systematik unterliegt. Wir Rechnen also aus jeder Häufigkeit der kumulierten Verteilung ${}_NL$ ihre bereits abgegoltenen Teilläufe ${}_NT = ({}_Nt_n)_{0<\leq N}$ heraus. Das Ergebnis ${}_NT$ nennen wir die Grundverteilung. Zur Veranschaulichung, führen wir ein Beispiel aus B_5 an. In diesem Fall betragen die Werte von ${}_5L = ({}_5l_i)_{0<i\leq 5}$

$${}_5l_i = 2^{5-i} \tag{3.6}$$

Das heißt explizit, dass

- ${}_5l_1 = 16$
- ${}_5l_2 = 8$
- ${}_5l_3 = 4$
- ${}_5l_4 = 2$
- ${}_5l_5 = 1$

Der längste Lauf beträgt 5 und ist kein Teillauf. Wir ziehen also die durch den ${}_5l_5$ abgegoltenen Teilläufe und erhalten. das sind

- 5 Teilläufe 1
- 4 Teilläufe 2
- 3 Teilläufe 3,
- 2 Teilläufe 4
- 1 Lauf 5,

Nun ist der Lauf 5 abgegolten und damit auch beide Läufe 4. Noch geltend zu machen sind die folgende Läufe:

- ${}_5l_1 - 5 = 11$ Läufe 1
- ${}_5l_2 - 4 = 4$ Läufe 2
- ${}_5l_3 - 3 = 1$ Läufe 3
- ${}_5l_4 - 2 = 0$ Läufe 4
- ${}_5l_5 - 1 = 0$ Läufe 5

Wir machen nun den Lauf 3 geltend, da er nun nicht mehr als Teillauf vorhanden ist. Das heißt

- 3 Teilläufe 1
- 2 Teilläufe 2
- 1 Lauf 3,

Nun ist der Lauf 3 abgegolten. Noch geltend zu machen sind die folgende Läufe:

- ${}_5l_1 - 5 - 3 = 8$ Läufe 1
- ${}_5l_2 - 4 - 2 = 2$ Läufe 2
- ${}_5l_3 - 3 - 1 = 0$ Läufe 3

Wir machen nun die Läufe 2 geltend. Das heißt

- 4 Teilläufe 1
- 2 Teilläufe 2

Nun ist der Lauf 3 abgegolten. Noch geltend zu machen sind die folgende Läufe:

- ${}_5l_1 - 5 - 3 - 4 = 4$ Läufe 1

Da die Läufe 1 keine Teilläufe enthalten, werden sie unmittelbar abgegolten. Das Ergebnis stellen wir durch die Folge

$$(4; 2; 1; 0; 1)_{[B_5]} \tag{3.7}$$

dar. Eine Mögliche Realisierung dieser Verteilung ist

$$(0101001100011111) \; mit \; (\underbrace{0\,1\,0\,1}_{4} \; \underbrace{00\,11}_{2} \; \underbrace{000}_{1} \; \underbrace{}_{0} \; \underbrace{11111}_{1}) \in B_5 \tag{3.8}$$

Natürlich ist jede andere Permutation dieser Folge von Läufe gültig. Dabei werden bei Bedarf die Werte der Bits komplementiert. Wir wiederholen beispielhaft diese Berechnung für Bitfolgen aus den Klassen B_1 bis B_8

$$(1)_{[B_1]} \tag{3.9}$$
$$(0 \; ; \; 1)_{[B_2]} \tag{3.10}$$
$$(1 \; ; \; 0 \; ; \; 1)_{[B_3]} \tag{3.11}$$
$$(2 \; ; \; 1 \; ; \; 0 \; ; \; 1)_{[B_4]} \tag{3.12}$$
$$(4 \; ; \; 2 \; ; \; 1 \; ; \; 0 \; ; \; 1)_{[B_5]} \tag{3.13}$$
$$(8 \; ; \; 4 \; ; \; 2 \; ; \; 1 \; ; \; 0 \; ; \; 1)_{[B_6]} \tag{3.14}$$
$$(16 \; ; \; 8 \; ; \; 4 \; ; \; 2 \; ; \; 1 \; ; \; 0 \; ; \; 1)_{[B_7]} \tag{3.15}$$
$$(32 \; ; \; 16 \; ; \; 8 \; ; \; 4 \; ; \; 2 \; ; \; 1 \; ; \; 0 \; ; \; 1)_{[B_8]} \tag{3.16}$$

Wir bemerken sofort, dass mit wachsenden N die Grundverteilung der Elemente aus den B_N eine gewisse Struktur aufweist: Lediglich eine Verschiebung nach rechts der Häufigkeit der Läufe 2 bis N und eine Verdoppelung der Häufigkeit der Läufe 1 sind zu verzeichnen. Wir wollen nun diese experimentellen Ergebnisse formal untermauern und behaupten

Behauptung 1. *Sei* $\mathbb{N} \ni N < \infty$ *beliebig, fest und* $({}_N b_m) \in \mathcal{B}$ *eine Bitfolge mit* $({}_N b_m) \in B_N,\ N < \infty$, N *geeignet. Dann gilt für die Grundverteilung*

$$ {}_N G := ({}_N g_n)_{0<i\leq N} \tag{3.17}$$

der Läufe von $({}_N b_m)$

$$\begin{aligned} {}_N g_n &= 2\ {}_N g_{n+1}\ ,\ \ 2 \leq n < N-2 && (3.18)\\ {}_N g_{N-2} &= 1 && (3.19)\\ {}_N g_{N-1} &= 0 && (3.20)\\ {}_N g_N &= 1 && (3.21) \end{aligned}$$

Aus der Gleichung (**??**) ergibt sich unmittelbar die

Bemerkung 1. *Mit* (**??**) *gilt*

$$ {}_N g_n = 2^{N-2-n},\ \forall n = 1, \cdots, N-2 \tag{3.22}$$

Beweis. Der Beweis der Bemerkung 1 ist trivial. □

Wir beweisen nun die Behauptung **??**.

Beweis. Für die Beweisführung der Behauptung **??** wenden wir die Methode der vollständige Induktion über $1 \leq N < \infty$ an. Seien für beliebige N die Folgen $({}_N b_m) \in \mathcal{B}_{\mathcal{Z}}$ mit längstem Lauf N.

• Induktionsanfang

Für alle $N = 1, \cdots, 5$ ist die Behauptung **??** experimentell durch den Gleichungen (**??**) bis (**??**) bewiesen.

• Induktionsschritt

Sei nun die Behauptung **??** richtig für alle $1, \cdots, N-1$, mit beliebigen $1 < N < \infty$. Wir fassen den Übergang von $N-1$ auf N in 2 Schritte auf:

> Schritt 1: in einem geeigneten Element $({}_{N-1} b_m)$ aus B_{N-1} wird jeder Lauf n wird um einen Bit erweitert.
> Schritt 2: Das Ergebnis $({}_N b_m)$ mit längstem Lauf N dieser Erweiterung wird um so viele Läufe 1 ergänzt, bis die Forderung **??** erfüllt wird.

Wir illustrieren den Schritt1 mit dem Beispiel aus $(0001) \in B_3$: $\quad (0001) \rightarrow (000\underset{\uparrow}{1}\underset{\uparrow}{1})$

Wir Untersuchen nun die damit gewonnene Bitfolge $({}_N b_m)$, Mit dem Schritt 1 wird in einem Element $({}_{N-1} b_m)$ aus B_{N-1} jeder Lauf n zu einem Lauf $n+1$. Damit gilt

$$ {}_N g_n \ = \ {}_{N-1} g_{n-1}\ \forall 1 < n \leq N \tag{3.23}$$

und die Gleichung (**??**) gilt für alle $1 < n \leq N$.
Mit dem Schritt 1 des Induktionsanfangs wird im Element $({}_{N-1}b_m)$ aus B_{N-1} jedem Lauf genau einen Lauf 1 hinzugefügt. Das heißt also, dass genau

$$\sum_{n=1}^{N-1} {}_{N-1}g_n = 1 + \sum_{n=0}^{n=N-3} 2^n \tag{3.24}$$

$$= 2^{N-2} \tag{3.25}$$

Läufe 1 hinzugefügt und nun abgegolten werden müssen. Gefordert werden laut Gleichung (**??**) genau ${}_Nl_1 = 2^{N-1}$ Läufe 1. Somit gilt

$${}_Ng_1 = {}_Nl_1 - \sum_{n=1}^{N-1} {}_{N-1}g_n \tag{3.26}$$

$$= 2^{N-1} - 2^{N-2} \tag{3.27}$$

$$= 2 \cdot 2^{N-2} - 2^{N-2} \tag{3.28}$$

$$= 2^{N-2} \tag{3.29}$$

$$= 2 \cdot 2^{N-3} \tag{3.30}$$

$$= 2 \cdot_N g_2 \tag{3.31}$$

Somit ist die Gleichung (**??**) für alle $1 \leq n \leq N$ bewiesen.

• Induktionsschluss

Da N beliebig war, gilt die Gleichung (**??**) für alle $N < \infty$. □

Bevor wir diese Ergebnisse in Algorithmen umsetzen, widmen wir uns dem folgenden Sachverhalt: Die Abgeltung von Teilläufe, wie sie in dem Verfahren zur Herleitung der Grundverteilung dargestellt wurde, kann auch ermittelt werden, in dem wir die Folge der Häufigkeiten ${}_NT = ({}_Nt_n)_{1 \leq n \leq N}$ der abgegoltenen Teilläufe berechnen. Dies führt zu der folgenden, allgemein geltenden Formel

$${}_Nt_n = \sum_{i=n+1}^{i=N} ({}_Ns_{i-n} \cdot {}_Ng_i) \quad , \forall 0 < n < N-2 \tag{3.32}$$

Wir fassen nun die Anzahl der abzugeltenden Läufe als die Differenz ${}_Nl_n - {}_N g_n$ der kumulierte Verteilungen und der Grundverteilung auf. Damit ergibt sich die Gleichung

$${}_Nt_n = {}_Nl_n - {}_Ng_n \tag{3.33}$$

also den allgemeinen Ausdruck

$${}_Nl_n - {}_Ng_n = \sum_{i=n+1}^{i=N} ({}_Ns_{i-n} \cdot {}_Ng_i) \tag{3.34}$$

Für den speziellen Fall der exponentiell abfallenden kumulierten Verteilungen ${}_Nl_n = (2^{N-n})_{0<n\leq N}$ und jeweils für geeignetes N, müssen wir die vier folgenden Fälle unterscheiden

♦ $n = N$: So vereinbaren wir mit

$$ {}_Nl_N \ - \ {}_Ng_N = 2^{N-N} - 1 = 1 - 1 = 0 \tag{3.35}$$

dass

$$ {}_Nt_N = 0 \tag{3.36}$$

♦ $n = N - 1$: So sind stets

$$\begin{aligned} {}_Nl_{N-1} \ - \ {}_Ng_{N-1} &= 2^{N-N+1} - 0 && (3.37)\\ &= 2 && (3.38)\end{aligned}$$

und

$$ {}_Nt_{N-1} = \underbrace{{}_Ns_{N-1}}_{=2} \cdot \underbrace{{}_Ng_N}_{=1} = 2 \tag{3.39}$$

♦ $n = N - 2$: So sind stets

$$\begin{aligned} {}_Nl_{N-2} \ - \ {}_Ng_{N-2} &= 2^{N-N+2} - 2^{N-2-N+2} && (3.40)\\ &= 4 - 1 && (3.41)\\ &= 3 && (3.42)\end{aligned}$$

und

$$\begin{aligned} {}_Nt_{N-2} &= \underbrace{{}_Ns_{N-1}}_{=2} \cdot \underbrace{{}_Ng_{N-1}}_{=0} + \underbrace{{}_Ns_{N-2}}_{=3} \cdot \underbrace{{}_Ng_N}_{=1} && (3.43)\\ &= 3 && (3.44)\end{aligned}$$

♦ $n < N - 2$ Wir verwenden die Gleichungen

$$\begin{aligned} {}_Nl_n &= 2^{N-n} && (3.45)\\ {}_Ng_n &= 2^{N-2-n} && (3.46)\end{aligned}$$

ein und rechnen gleich nach, dass

$$\begin{aligned} {}_Nt_n &= {}_Nl_n \ - \ {}_Ng_n && (3.47)\\ &= 2^{N-n} \ - \ 2^{N-2-n} && (3.48)\\ &= 3 \cdot 2^{N-1-n} && (3.49)\end{aligned}$$

Aus der Gleichung (**??**) leiten wir nach der Indexverschiebung um $n-1$ ab, dass

$$\begin{aligned} {}_Nt_n &= \sum_{i=n+1}^{i=N} ({}_Ns_{i-n} \cdot {}_Ng_i) && (3.50)\\ &= \sum_{i=n+1-n+1}^{i=N-n+1} ({}_{N-n+1}s_{i-n+n-1} \cdot {}_{N-n+1}g_i) && (3.51)\\ &= \sum_{i=2}^{i=N-n+1} ({}_{N-n+1}s_{i-1} \cdot {}_{N-n+1}g_i) && (3.52)\\ &= {}_{N-n+1}t_1 && (3.53)\\ & && (3.54) \end{aligned}$$

Wir bemerken gleich, dass mit

$${}_Nt_n = 3 \cdot 2^{N-1-n} \tag{3.55}$$

unmittelbar gilt, dass

$${}_{N-n+1}t_1 = 3 \cdot 2^{N-1-n} \tag{3.56}$$

Allgemeiner können wir schreiben, dass

$${}_{N+1}t_1 = 3 \cdot 2^{N-1} \tag{3.57}$$

Also

$${}_Nt_1 = 3 \cdot 2^{N-2} \tag{3.58}$$

Wir behaupten daher

Bemerkung 2. *sei* $N < \infty$

$$2^N = (N+1) + \left(1_{\{N>1\}} \cdot \sum_{n=0}^{n=N-2} (N-1-n) \cdot 2^n\right) \tag{3.59}$$

Beweis. Wir führen den Beweis durch vollständige Induktion durch.

- Induktionsanfang

Für $N = 1$, $N = 2$ und $N = 3$ gilt die Behauptung trivialerweise. Für $N = 4$ gilt

$$2 \cdot 1 + 1 \cdot 2 + 4 = 8 = 2^3 \tag{3.60}$$

und somit gilt auch die Behauptung.

- Induktionschritt

Es gelte die Bemerkung **??** für einen beliebigen $N < \infty$. Das heißt, dass die zu Grunde

liegende Bitfolge die Länge 2^N besitzt. Wie im Induktionsschritt des Beweises der Behauptung **??** erweitern wir nun jeden einzelnen Lauf der zu Grunde liegende Bitfolge um ein Bit. Da die zu Grunde liegende Bitfolge

$$2^{N-1} = 1 + \sum_{n=0}^{N-2} 2^n \tag{3.61}$$

Läufe besitzt, wird sie durch die erweiterung um geanu so viele, also um 2^{N-1} Bits verlängert. Hinzu kommen durch das Hinzufügen der geforderten 2^N Läufe 1 noch soviele Bits dazu. Nach der Erweiterung beträgt die Länge der Bitfolge

$$2^{N-1} + 2^{N-1} + 2^N = 2^{N+1} \tag{3.62}$$

Bit. Damit gilt die Bemerkung **??** für $N + 1$.

- Induktionschluss

Da N beliebig war, gilt die Behauptung für alle $N < \infty$ □

4 Algorithmen zur Erzeugung eines Elementes aus B_N

Motiviert durch die Ergebnisse aus dem vorherigen Kapitel geben wir einen iterativen Algorithmus zur Berechnung eines Klassenvertreters der B_N mit $n < \infty$ an.

Algorithmus 3. *Iterativer Generator für eine Bitfolge aus B_N*

```
  Eingabe: N, die Länge des längsten Laufs.
2 Lege einen leeren Bitvektor B an
  Hänge einen Lauf der Länge N an
4 für alle i von N-2 bis 1
    hänge 2^(N-i-2) Läufe der Länge i an B
```

Listing 4.1: Itereative Erzeugung einer Folge aus B_N für einen beliebigen $N < \infty$

Die Richtigkeit des Algorithmus ergibt sich unmittelbar aus der Behauptung **??**.

Die Beweisführung der Behauptung **??** begründet die Richtigkeit des folgenden rekursiven Algorithmus zur Berechnung eines Klassenvertreters der B_N mit $n < \infty$.

Algorithmus 4. *Rekursive Berechnung einer Bitfolge aus einem B_N*

```
  Eingabe: Ein Element aus B_N mit N>2
2 wiederhole für jeden Lauf
    ermittle die Lange des Laufs und erweitere ihn um 1 Bit
4   falls der Läuf 1 betragen hat füge 2 Läufe 1 hinzu
  Gib das Ergebnis aus
```

Listing 4.2: rekursive Berechung einer Bitfolge aus B_N

Die Fälle $N = 1$ und $N = 2$ können in einer Implementierung des Algorithmus gesondert behandelt werden.

In den bisherigen Ansätzen zum Entwurf von Zufallsgeneratoren mussten die erzeugten Bitfolge stets willkürliche Kriterien erfüllen. Meist handelt es sich um statistische Vorgaben bzw. um die ästhetische Wiedererkennung von Mustern. Von derartigen Kriterien wird hier abgesehen, denn sie sind entweder tautologisch, wenn der in der Einführung angegebenen Beweis von Karl Popper hier anerkannt wird, oder nicht der wissenschaftlichen Methodik entsprechend im Falle der ästhetischen Mustererkennung. Allein die Häufigkeitsverteilung der Läufe einer Bitfolge entscheidet über ihre Zugehörikkeit zu B_N mit $n < \infty$. Dies hat zu Folge, dass z.B.

$$11110010 \tag{4.1}$$

Bereits ein Element aus B_N mit $n < \infty$ ist, mit abfallende Länge ihrer Läufe. Als weitere Illustration können wir jede Bitfolge als binäre Zahl auffassen mit jeder anderen Bitgolge durch die gewohntne Ordnungsrelationen mit einander vergleichen. Wir nehmen die Elemente aus B_5. Das größte Element ist[1]

$$11111\ 0\ 111\ 0\ 11\ 0\ 11\ 0 \tag{4.2}$$

Man erkennt (ästhetisch) sofort, dass die Läufe mit einer Länge größer eins in abfallender Reihe sortiert sind und jeweils von einem Lauf der Länge eins getrennt sind.
Den kleinsten Element aus B_5 ist dagegen

$$00000\ 1\ 000\ 1\ 00\ 1\ 00\ 1 \tag{4.3}$$

Man erkennt (ästhetisch) sofort, dass es sich um das binäre Komplement, des größten Element aus B_5 handelt. Weiter ist

$$1\ 00000\ 1\ 000\ 1\ 00\ 1\ 00 \tag{4.4}$$

Das kleinste Element aus B_5, dass mit einer Eins anfängt und sein binäres Komplement

$$0\ 11111\ 0\ 111\ 0\ 11\ 0\ 11 \tag{4.5}$$

das größte Element, dass mit einer Null anfängt. Weiterhin gibt es in B_5 kein Element zwischen (**??**) und (**??**).

So lässt sich intuitiv eine Gewisse Struktur in den B_N erahnen. Es kommt nun noch nur darauf an, dass diese Wiedererkennbarkeit methodologisch nicht relevant ist, um die B_N als Menge der zufälligen Bitfolgen zu bezeichnen. Viel mehr ist hier angemessen, zu behaupten, dass es den Zufall, wie er in der Wahrscheinlichkeitstheorie begriffen wird, nicht gibt, sondern die Willkür der Wahl eines Elementes aus einem beliebigen B_N, welche nur mit exponentiellem Aufwand zurückverfolgt werden kann. Dieser asymmetrische Aufwand einer Entscheidung in (super) linearer Zeit und eine Rückverfolgung in exponentieller Zeit ist Grundlage der Kryptographie und gilt, solange

$$P \neq NP \tag{4.6}$$

[1] Die Abstände in der Darstellung dienen lediglich der Lesbarkeit und haben keinerlei Bedeutung in Bezug auf den Inhalt

5 Zusammengesetzte Elemente aus verschiedenen B_N

Seien nun $B \in \mathbb{N}$ beliebig und ihre eindeutig bestimmte, binäre Darstellung

$$B = \sum_{n=1}^{n=N} b_n \cdot 2^n \quad \text{mit } b_n \in \{0,1\} \text{ und } N \text{ geeignet} \tag{5.1}$$

Mit der Identität aus der Bemerkung **??** erhalten wir unmittelbar die Gleichung

$$B = \sum_{n=1}^{n=N} b_n \left((n+1) + \left(1_{\{n>1\}} \cdot \sum_{i=0}^{i=n-2} (n-1-i) \cdot 2^i \right) \right) \tag{5.2}$$

Damit können wir auf eindeutiger Weise eine beliebige, natürliche Zahl B mit einer durch dieser Zahl eindeutig bestimmten Zusammensetzung von zu einander fremden Häufigkeitsverteilungsklassen B_N aus $\mathcal{B}_\mathcal{Z}$ identifizieren und erhalten die zusammengesetzte Häufigkeitsverteilung ${}_BH_N$ der Läufe einer Bitfolge mit der vorgegebener Länge B. Wir vereinbaren die folgende Definition

Definition 7. *Jede Bitfoge (b_n), deren Häufigkeitsvertielung ihrer Läufe eine beliebige zusammensetzung von Häufigkeitsverteilungen von zu einander fremden Häufigkeitsverteilungsklassen aus $B_N \in \mathcal{B}_\mathcal{Z}$, nennen wir zufällig.*

Die Häufigkeit einer Lauflänge einer beliebigen Zusammensetzung von zu einander fremden Häufigkeitsverteilungsklassen aus $\mathcal{B}_\mathcal{Z}$

$${}_BH_n = b_n + 1_{\{n<N-1\}} \sum_{i=n+2}^{i=N} b_i \cdot 2^{N-2-i} \tag{5.3}$$

ergibt sich unmittelbar aus der Definition von $\mathcal{B}_\mathcal{Z}$ und somit die Länge einer auf dieser Art gewonnen Bitfolge

$$\begin{aligned} B &= \sum_{n=1}^{n=N} n \cdot H_n & (5.4)\\ &= \sum_{n=1}^{n=N} n \cdot b_n + \sum_{n=1}^{n=N} n \cdot \left(1_{\{n<N-1\}} \cdot \sum_{i=n+2}^{i=N} b_i \cdot 2^{N-2-i} \right) & (5.5) \end{aligned}$$

Diese Identität ist gleichwertig mit der Identität (**??**). Wir illustrieren diesen Sachverhalt mit

$$B = 101010101_2 = 341_{10} \tag{5.6}$$

in der folgenden Tabelle

n	b_n	1	2	3	4	5	6	7	8	9
1	1	1								
2	0	~~∅~~	~~1~~							
3	1	1	0	1						
4	0	~~2~~	~~1~~	~~∅~~	~~1~~					
5	1	4	2	1	0	1				
6	0	~~8~~	~~4~~	~~2~~	~~1~~	~~∅~~	~~1~~			
7	1	16	8	4	2	1	0	1		
8	0	~~32~~	~~16~~	~~8~~	~~4~~	~~2~~	~~1~~	~~∅~~	~~1~~	
9	1	64	32	16	8	4	2	1	0	1
	Σ	86	42	22	10	6	2	2	0	1
	n	1	2	3	4	5	6	7	8	9
	Π	86	84	66	40	30	12	14	0	9
Σ	341									

Wir sind nun im Stande jede beliebige, natürliche Zahl einer endlichen, eindeutig bestimmten Teilmenge $\{B_n\}_{n\in\mathbb{N}} \subset \mathcal{B}_{\mathcal{Z}}$ mit $n_1 \neq n_2\ \forall n_1, n_2$ zuordnen, und damit eine eindeutig bestimmte, zusammengesetzte Häufigkeitsverteilung von Läufen herzuleiten.

Wir untersuchen nun den logarithmischen Wert der Häufigkeiten der Läufen eines solchen Klassensystems. Sei $(b_n)_{n<N<\infty}$ eine Folge von binären Werten mit N beliebig und fest $b_n \in \{0,1\}$ und $b_N \neq 0$. Sei weiterhin n_K die kumukierte Häufigkeit der zu einem n zugehörige Klasse B_n aus $\mathcal{B}_{\mathcal{Z}}$, also

$$({}_nK) = \left(2^n, 2^{n-1}, \cdots, 2, 2^{n-n} = 1\right) \tag{5.7}$$

Sind alle $b_n = 0, n < N$ so ist

$$({}_nK)_{n\leq N} = \left(2^N, 2^{N-1}, \cdots, 2, 2^{N-N} = 1\right) \tag{5.8}$$

und die logarithmischen Werte der zusammengesetzten kumulierten Häufigkeit der zu (b_N) zugehörige Klassen

$$log_2({}_nK)_{n\leq N} = (N, N-1, N-2\cdots, 2, 1, 0) \tag{5.9}$$

Sind dagegen alle $b_n = 1, n < N$ so sind

$$({}_nK)_{n \leq N} = \left(\sum_{i=1}^{i=N} 2^i, \sum_{i=1}^{i=N-1} 2^i, \cdots, \sum_{i=0}^{i=N-N} 2^i = 1 \right) \tag{5.10}$$

und damit

$${}_nK < 2^{n+1} \ \ \forall n \leq N \tag{5.11}$$

Demnach gilt für die logarithmischen Werte der zusammengesetzten kumulierten Häufigkeiten

$$log({}_nK) < n + 1 \ \ \forall n \leq N \tag{5.12}$$

Wechseln also die Werte b_n von 0 auf 1 so wächst der zweier Logarithmus der Kumulierten Häufgikeiten um maximal 1, im Einklang mit den experimentellen Beobachtungen aus dem Kapitel **??**.

6 Literaturhinweise

Da sämtliche Arbeiten um das Themengebiet der Zufallszahlen und -bitfolgen sich auf das tautologische Grenzwertaxiom der Wahrscheinlichkeitsrechnung berufen, und da Tautologien der induktiven Logik zugeordnet werden, welche sich mit den Mitteln der deduktiven Logik sich nicht falsifizieren lässt, gibt es auch keine Literatur, auf die hier verwiesen werden kann.
Wir erinnern an Karl Popper, „Logik der Forschung", Teubner, 1933.