# Übungsstunde 2

Freitag, 29. September 2023 16:33

## Kolmogorov-Komplexität

**Motivation:** Nachdem wir das Regelwerk für Texte als Informationsgrösse gebildet haben, können wir uns nun nach dem Aufwand einer solchen Darstellung fragen. Wie in der Informatik üblich beschränken wir uns auf das bool'sche Alphabet $\Sigma_{bool} = \{0,1\}$ zur Darstellung von Texten.

Intuitiv wissen wir, dass $(01)^5$ schneller geschrieben werden kann als 0101010101.

Intelligente Leute würden diesen Unterschied so beschreiben: $(01)^5$ ist eine komprimierte Darstellung von

Diese Komprimierung können wir als kleineren Informationsgehalt auffassen. Die KK verallgemeinert diese Idee von "weniger Speicherplatz" auf die binäre Länge von Pascal-Programmen welche die Information als Output erzeugt.

**Theorie:**

**Definition 2.17.** *Für jedes Wort $x \in (\Sigma_{\text{bool}})^*$ ist die* ***Kolmogorov-Komplexität $K(x)$ des Wortes $x$*** *das Minimum der binären Längen der Pascal-Programme, die $x$ generieren.*

↳ i.e. Wähle aus allen Maschienencodes (0110...) von Pascal-Programmen welche x generieren das kürzeste (generiert: Output von Programm ohne Input ist x)

↳ Wenn wir ein Programm schreiben, welches x generiert, so ist dies sicherlich eine obere Schranke für K(x), aber eventuell nicht die beste obere Schranke!

**Aufgabe:** Gibt es eine Konstante $c \in \mathbb{N}$, s.d. $\forall n \in \mathbb{N}-\{0\}: K(x_n) \leq \log_2(\sqrt{n}) + c$ für eine unendliche Folge von paarweise verschiedenen Wörtern $(x_n)_{n \in \mathbb{N}-\{0\}}$ ?

**Beweis:** Aufgaben von diesem Typ lassen sich immer gleich lösen. Entweder man führt einen Widerspruchsbeweis mit einem Kombinatorikargument (ähnlich zu dem in Lemma 2.5) durch oder man entwirft eine solche Folge, schreibt ein Pascal-Programm und schätzt die KK nach oben ab (das sehen wir später). Mit göttlicher Intuition probieren wir nun Variante 1.

Beweis per Widerspruch. Angenommen es existiert eine solche Folge. Sei $n \in \mathbb{N}-\{0\}$ fix und setze

$t := \log_2(\sqrt{n}) + c = \log_2(n^{1/4}) + c = \frac{1}{4}\log_2(n) + c$

Es gibt $\sum_{i=0}^{t} 2^t = 2^{t+1} - 1$ verschiedene Wörter mit binärer Länge höchstens $t$.

$2^{t+1} - 1 = 2^{\frac{1}{4}\log_2(n) + c + 1} - 1 < \infty$. Folglich können höchstens $2^{t+1} - 1$ paarweise verschiedene Wörter durch Programme mit binärer Länge $\leq t$ erzeugt werden.

Der entscheidende Punkt ist die Endlichkeit!

Mathematisch gesehen können wir nun ein $n$ finden/wählen, s.d. $2^{t+1} - 1 < n$, was der Annahme paarweise verschieden widerspricht. Formell: Für ein genügend großes $n$, welches nur von der Konstante $c$ abhängt, gilt: $2^{t-1} - 1 < n$. Also $\exists i,j \in \{1,\dots,n\}$, $i<j$, s.d. $x_i = x_j$ $\lightning$ was unserer Annahme ($x_i \neq x_j$ für $i \neq j$) widerspricht. □

**Frage:** Was ist die binäre Länge einer Zahl $n \in \mathbb{N}$ (wir bezeichnen diese Länge mit $Bin(n)$)?

$Bin(n) = \lceil \log_2(n+1) \rceil$. Woher "+1"? Die "+1" repräsentiert das Vorzeichen in der binären Darstellung.

**Erinnerung:** Zufällig bedeutet im Zusammenhang mit Kolmogorov-Komplexitäten, dass keine komprimierte Darstellung existiert. Woher kommt die "-1" in der Def. 2.19 für Zahlen? Da alle Binärdarstellungen von Zahlen mit einer 1 beginnen, können wir diese vernachlässigen.

**Aufgabe:** Geben Sie eine unendliche Folge von natürlichen Zahlen $\underbrace{y_1 < y_2 < \dots}_{(I)}$ an, s.d. ein $c \in \mathbb{N}$ existiert für das $K(y_i) \leq \lceil \log_2(\log_7(\sqrt{|y_i|})) \rceil + c$ $\forall i \geq 1$ gilt.

**Lösung:** In einem ersten Schritt versuchen wir $|y_i|$ zu bestimmen. Dafür nutzen wir Satz 2.2.: $K(y_i) \leq \lceil \log_2(i+1) \rceil + c$. Ein einfacher Koeffizientenvergleich mit der Aufgabenstellung liefert:

$$\begin{aligned} & i+1 = \log_7(\sqrt{|y_i|}) \\ \Leftrightarrow \quad & 7^{i+1} = \sqrt{|y_i|} \\ \Leftrightarrow \quad & (7^{i+1})^2 = |y_i| \quad (II) \end{aligned}$$

Nun haben wir die Länge für das $i$-te Wort gefunden. Aber wie finden wir nun eine Folge von natürlichen Zahlen, welche (I) und die binäre Länge (II) erfüllen? Wir nutzen hierfür den Standardtrick, dass $(2^L)_{L \geq 1}$ (I) erfüllt und $Bin(2^L) = 10\underbrace{\dots 0}_{L}$. Wir definieren also $y_i := 2^{(7^{i+1})^2} = 2^{7^{2i+2}}$

Im nächsten Schritt konstruieren wir nun eine solche Folge und schätzen ihre KK ab.

Hierfür schreiben wir ein Pascal-Programm.

```
Programm y_i : begin
                 K := 0
                 I := i
                 I := (7^(I+1))^2
                 write(1)
                 for K < I
                     begin
                       write(0)
                       K := K+1
                     end
               end
```

$Y_i$ berechnet zunächst in zwei Schritten den Wert $I = (7^{i+1})^2$. Note: $I = |Y_i|$

Der Teil des Maschinencodes von $Y_i$ welcher von $i$ abhängt, ist nur die Darstellung von $i$. Der Maschinencode des restlichen Programms hat also nur eine konstante Länge, während die binäre Kodierung von $i$ die Länge $\lceil \log_2(i+1) \rceil \leq 2 + \log_2 i$ hat

Damit gilt: $K(Y_i) \leq \lceil \log_2(i+1) \rceil + d \leq \lceil \log_2(\log_7(\sqrt{|Y_i|})) \rceil + c$, wir wählen $c, s.d.\ c \geq d$.

⚠ Falls wir eine möglichst gute Darstellung finden wollen können wir einfach den ersten Schritt weglassen und analog argumentieren ☺

**Beh.:** $\{n^2 \mid n \in \mathbb{N}\}$ enthält nur endlich viele Zahlen, die als zufällig betrachtet werden können.

**Beweis:** Per Widerspruch. Angenommen, es existieren unendlich viele Zahlen in $\{n^2 \mid n \in \mathbb{N}\}$. Nach Def. bedeutet dies, dass $K(n^2) \geq \lceil \log_2(n^2+1) \rceil - 1 \geq 2\log_2(n) - 1$ (I) für $\infty$-viele $n \in \mathbb{N}$ gilt.

Weiter lässt sich die Zahl $n^2$ $\forall n \in \mathbb{N}$ mit einem Programm $C_n$ berechnen, das die binäre Kodierung von $n$ enthält, $n^2$ berechnet und schliesslich ausgibt (der emsige Leser darf auch gerne ein solches Programm konkret schreiben, ein muss ist es jedoch nicht). Nun folgt die übliche Argumentation:

Alle Bestandteile von $C_n$ mit Ausnahme der Kodierung von $n$ haben eine konstante Länge. Also hat $C_n$ die binäre Länge $\lceil \log_2(n+1) \rceil + c \leq \log_2(n) + c + 1$ für eine Konstante $c$. Somit gilt: $K(n^2) \leq \log_2(n) + (c+1)$ (II)

Mit (I) & (II) folgt: $2\log_2(n) - 1 \leq \log_2(n) + (c+1)$ für $\infty$-viele $n \in \mathbb{N}$.

$\Leftrightarrow \quad \log_2(n) \leq c + 2$ "

Aber dies ist unmöglich, da $c$ konstant und $\log_2(n)$ mit $n$ beliebig wächst ↯ □

**Aufgabe:** Finde eine Funktion $f: \mathbb{N} \to \mathbb{N} \setminus \{0\}$, s.d. $\forall n \in \mathbb{N}$ min. ein Anteil von $\frac{15}{16}$ der natürlichen Zahlen kleiner als $f(n)$ eine K-K von min. $3n$ haben.

**Lösung:** Wir definieren $f(n) := 2^{3n+4}$. Die Wahl von $f(n)$ sollte am Ende klar werden ☺

Es gibt $2^{3n+4}$ natürliche Zahlen welche kleiner als $f(n)$ sind. Wir zeigen, dass maximal $\frac{1}{16} = 2^{-4}$, also $2^{-4} \cdot 2^{3n+4} = 2^{3n}$ nat. Zahlen, eine K-K kleiner als $3n$ haben.

Die Anzahl der unterschiedlichen Programme der Länge $< 3n$ ist max.: $\sum_{j=1}^{3n-1} 2^j = 2^{3n} - 2$

Also gibt es maximal $2^{3n} - 2$ Wörter $w$ mit $K(w) < 3n$. Da alle natürlichen Zahlen paarweise verschiedene Binärdarstellungen haben, gibt es höchstens $2^{3n} - 2$ Zahlen mit KK $< 3n$.

□

**Beh.:** Mindestens die Hälfte aller Wörter in $\{0,1\}^{\leq n}$ ist zufällig.

**Beweis:** Wir zeigen mittels eines zählerischen Arguments, dass es mindestens $\frac{1}{2}|\{0,1\}^{\leq n}| = \frac{1}{2}|\sum_{i=0}^{n} 2^i| = \frac{1}{2}\cdot(2^{n+1}-1) = 2^n - \frac{1}{2}$ Wörter in $\{0,1\}^{\leq n}$ gibt s.d. $K(x) \geq |x|$ für $x \in \{0,1\}^{\leq n}$.

In $\{0,1\}^{\leq n}$ gibt es $2^{n+1}-1$ Wörter

In $\{0,1\}^{<n}$ gibt es $2^n - 1$ Wörter.

Also gibt es für $2^{n+1}-1$ Wörter in $\{0,1\}^{\leq n}$ nur $2^n - 1$ Kodierungen der Länge kleiner als $n$. Somit muss für die übrigen $(2^{n+1}-1)-(2^n-1) = 2^n$ Wörter $x$ gelten: $K(x) \geq |x|$ □

**Beh.:**

Wir betrachten die Sprache

$$L = \{1^i 0^j 1^k \mid i,j,k \in \mathbb{N} - \{0\}\}.$$

Sei $x_n$ das kanonisch $n$-te Wort in $L$. Zeigen Sie, dass es eine Konstante $c \in \mathbb{N}$ gibt, so dass für alle $n \in \mathbb{N} - \{0\}$

$$K(x_n) \leq 3 \cdot \log_2(|x_n|) + c$$

gilt.

**Beweis:** Offenbar existiert ein Programm $A_L$ welches für $x \in \Sigma_{bool}^*$ entscheidet ob $x \in L$. Nach Satz 2.2:

$$K(x_n) \leq \lceil \log_2(n+1) \rceil + c \leq \log_2(n) + (c+1) \text{ für } c \in \mathbb{R} \text{ und } c \perp n$$

(unabhängig)

Schätze nun $n$ in Abhängigkeit von $|x_n|$ ab:

Die ersten $n$ Wörter aus $L$ in kanon. Reihenfolge haben die Form $1^i 0^j 1^k$ mit $i+j+k \leq |x_n|$

$\Rightarrow i,j,k \leq |x_n| \Rightarrow n \leq |x_n|^3$. Folglich haben wir:

$$K(x_n) \leq \log_2(n) + \underbrace{c+1}_{=:d} \leq \log_2(|x_n|^3) + d = 3\log_2(|x_n|) + d$$ □

⌜Wieso $n \leq |x_n|^3$? Wir haben $1 \leq i,j,k \leq |x_n| \Rightarrow$ max. $|x_n|^3$ Kombinationen ⌟

**Serien-Bingo:**

**A1** sehr wichtig

**A2** sehr wichtig

**A3** wichtig