Übungsstunde 2

Freitag, 30. September 2022 19:45

Heute beschäftigen wir uns weiter mit dem Thema Kolmogorov-Komplexität und geben eine erste Einführung zu endlichen Automaten.

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 $K(y_i) \leq \lceil \log_2(\log_7(\sqrt{|y_i|})) \rceil + c \quad \forall i \geq 1$ 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:

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

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 $Bin(2^L) = 1\underbrace{0 \dots 0}_{L}$. 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 Maschienencodes von Y_i welcher von i abhängt, ist nur die Darstellung von i. Der Maschienencode 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 :)

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.: $\{n^2 \mid n \in \mathbb{N}\}$ 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 $K(n^2) \geq \lceil \log_2(n^2+1) \rceil - 1 \geq 2\log_2(n) - 1 \overset{(I)}{\,}$ für ∞-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 ∞-viele $n \in \mathbb{N}$.

$\iff \log_2(n) \leq c + 2$ "

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

Meine Moneyboy-Empfehlung der Woche: Drip Drop