Theoretische Informatik: Blatt 2

Abgabe bis 2. Oktober 2015
Assistent: Sascha Krug, CHN D 42

Linus Fessler, Markus Hauptner, Philipp Schimmelfennig

Aufgabe 4

Bemerkung: In dieser Aufgabe verwenden wir im Pascal Code den Exponentiation Operator ^.

(a) Das folgende Programm W_n generiert das Wort w_n:

```
Wn:  begin
       M := n;
       M := 5 × M × M;
       M := 2 ^ (2 ^ M);
       for I = 1 to M do
         write(0);
     end
```

Für $w_n = 0^{2^{2^{5 \cdot n^2}}} \in \{0, 1\}^*$ gilt:

$$
\begin{aligned}
& & |w_n| &= 2^{2^{5 \cdot n^2}} \\
&\Rightarrow & log_2\ |w_n| &= 2^{5 \cdot n^2} \\
&\Rightarrow & log_2\ log_2\ |w_n| &= 5 \cdot n^2 \\
&\Rightarrow & \frac{1}{5} \cdot log_2\ log_2\ |w_n| &= n^2 \\
&\Rightarrow & \sqrt{\frac{1}{5} \cdot log_2\ log_2\ |w_n|} &= n
\end{aligned}
$$

Damit ist die Kolmogorov Komplexität:

$$K(z_n) \leq \lceil log_2(n+1) \rceil + d \leq \lceil log_2\ n \rceil + d' \leq \lceil\ log_2 \left(\sqrt{\frac{1}{5} \cdot log_2\ log_2\ |w_n|} \right) \rceil + d' = \lceil\ \frac{1}{2} \cdot log_2 \left(\frac{1}{5} \cdot log_2\ log_2\ |w_n| \right) \rceil + d'$$

(b) Für die Binärdarstellung einer Zahl der Länge n werden $\lceil log_2(n + 1) \rceil$ Bits benötigt. Daher gilt für $K(y_i) \leq \lceil log_2 log_2 log_2 \sqrt{y_i} \rceil + c$:

$$
\begin{aligned}
& & log_2 log_2 \sqrt{y_i} &= i \\
&\Rightarrow & log_2 \sqrt{y_i} &= 2^i \\
&\Rightarrow & \sqrt{y_i} &= 2^{2^i} \\
&\Rightarrow & y_i &= (2^{2^i})^2 = 2^{2^i} \cdot 2^{2^i} = 2^{2^i + 2^i} = 2^{2 \cdot 2^i} = 2^{2^{i+1}}
\end{aligned}
$$

Binär dargestellt ist $y_i = 10^{2^{i+1}}$, wobei $y_i < y_{i+1}$ nach kanonischer Ordnung trivialerweise erfüllt ist, da sich die Länge von y_i in jedem Schritt verdoppelt.

Das folgende Programm Y_i generiert das Wort y_i:

```
Yi:  begin
       M := i;
       M := 2 ^ (M + 1);
       for K = 1 to M do
         write(10);
     end
```

Aufgabe 5

Wir betrachten Wörter über dem Alphabet $\Sigma_{bool}^* = \{0, 1\}^*$ mit höchstens Länge n: $|w| \leq n$.
Nach der Definiton von Zufälligkeit ist $K(x_n) \geq |w_n| \Leftrightarrow x_n$ ist zufällig.
Also soll $K(x_n) < |w_n|$ sein, damit w_n komprimierbar ist. Jedes Programm, das solch ein Wort darstellt, ist selbst eine Bit-Sequenz in Σ_{bool} mit Länge bis $n-1$. Nehmen wir an, dass jede Bit-Sequenz in $\{\Sigma_{bool}\}^{n-1}$ ein sinnvolles Programm sei, das ein Wort generieren kann, dann ist die maximale Anzahl dieser (nichtleeren) Programme

$$\sum_{k=1}^{n-1} 2^k = 2^n - 2.$$

Die Anzahl aller möglichen Wörter bis Länge n ist:

$$1 + \sum_{k=1}^{n} 2^k = 2^n - 2 + 2^n.$$

Falls ein Programm ein Wort komprimiert, gibt es eine Bijektion zwischen diesem Wort und dem Programm. Ziehen wir von der Anzahl aller möglicher Wörter die Anzahl aller möglicher – diese Wörter komprimierender – Programme ab, bleiben noch

$$1 + \sum_{k=1}^{n} 2^k \; - \sum_{k=1}^{n-1} 2^k = 1 + 2^n - 2 + 2^n - (2^n - 2) = 1 + 2^n$$

Wörter übrig, die nicht von Programmen in $\{\Sigma_{bool}\}^{n-1}$ generiert werden können. Dies sind mehr als die $2^n - 2$ möglichen Programme, also als die Anzahl potentiell komprimierbarer Wörter. Daher ist mehr als die Hälfte der Wörter nicht komprimierbar, also zufällig.

Aufgabe 6

Ein Wort w_n lässt sich durch i und k genau definieren, da $j = 2k - i$. $|w_n| = i + j + k = 2k + k = 3k$. Außerdem lässt sich i abschätzen. $i \in [0, 2k]$

$$\begin{aligned}
K(x_n) &\leq \lceil \log_2 (i+1) \rceil + \lceil \log_2 (k+1) \rceil + c \\
&\leq \log_2 i + \log_2 k + c' \\
&\leq \log_2 2k + \log_2 k + c' = \log_2 (2k^2) + c' \\
&\leq 2 \log_2 k + c'' \\
&\leq 2 \log_2 3k + c'' = 2 \log_2 |w_n| + c''
\end{aligned}$$

Dies zeigt die Aussage.