Übungsstunde 6 - A new hope

Montag, 31. Oktober 2022 16:48

Aufgabe: Sei $f: \{0,1\}^* \to \{0,1\}^*$ eine beliebige binäre Funktion zur Kodierung von Binärwörtern. Sei $n \in \mathbb{N}$, zeige, dass für min. die Hälfte aller Wörter $w \in \{0,1\}^{\leq n}$ gilt: $|f(w)| \geq |w|$

Beweis: Sei $A = \{0,1\}^{<n}$, $B = \{0,1\}^{\leq n}$. Es gilt: $|A| = \sum_{i=0}^{n-1} 2^i = 2^n - 1$, $|B| = \sum_{i=0}^{n} 2^i = 2^{n+1} - 1$

Bemerke, dass $|f(w)| < |w|$ für $w \in B \iff f(w) \in A$.

Da f injektiv ist, gilt dass die Anzahl $w \in B$ mit $f(w) \in A$ nicht grösser als $|A|$ sein kann. Definiere $B_{\geq} := B - \underbrace{\{w \in B \mid f(w) \in A\}}_{=: B_{<}} = \{w \in B \mid |f(w)| \geq |w|\}$. Folglich:

$|B_{\geq}| = |B| - |B_{<}| \geq |B| - |A| > |B|/2$ $\square$

Aufgabe:

a) Gebe eine steigende unendliche Folge natürlicher Zahlen an, s.d. bei der Faktorisierung aller dieser Zahlen insgesamt nur endlich viele Primfaktoren vorkommen

b) Sei $(n_i)_{i \geq 1} \subseteq \mathbb{N}$ eine steigende ∞-Folge mit KK $K(n_i) \geq \lceil \log_2(n_i) \rceil$.
Zeige, dass bei der Faktorisierung aller Zahlen n_i insgesamt ∞-viele Primfakt. vorkommen.

Beweis:

a) $(3^n)_{n \geq 1}$

b) Beweis per Widerspruch. Angenommen es kommen nur endlich viele Primzahlen vor.
Sei p_m die grösste vorkommende Primzahl. Dann lässt sich jede Zahl n_i eindeutig als $n_i = p_1^{r_{i,1}} \cdots p_m^{r_{i,m}}$ für $r_{i,1}, \dots, r_{i,m} \in \mathbb{N}$ darstellen. Sei A ein Programm, das für gegebene $r_{i,1}, \dots, r_{i,m}$ die binäre Darstellung von n_i erzeugt. Sei c die bin. Länge des Teils von A, der $\forall i \in \mathbb{N}$ gleich ist (alles außer $r_{i,1}, \dots, r_{i,m}$):

$$K(n_i) \leq c + \sum_{j=1}^{m} \lceil \log_2(r_{i,j} + 1) \rceil \overset{r_{i,j} \leq \log_2 n_i \ \forall j \in \{1,\dots,m\}}{\leq} c + m \cdot \lceil \log_2(\log_2 n_i + 1) \rceil \quad \forall i \in \mathbb{N}_{\geq 1}.$$

Dann gilt aber $\lceil \log_2(n_i) \rceil \leq c + m \lceil \log_2(\log_2(n_i) + 1) \rceil$ $\forall i$ (m, c sind Konstant) $\lightning$

Da die Ungleichung nur für endlich viele Zahlen erfüllt sein kann. $\square$

Aufgabe: Sei $w_n = 0^{2^{2^{3n}}} \in \{0\}^*$. Finde eine möglichst gute obere Schranke für $K(w_n)$ in der Länge $|w_n|$ an.

Lösung: Folgendes Programm generiert w_n:

```
begin
  N := 2^(2^(3*n))
  for i = 1, ..., N begin
    write(0)
  end
end
```

Das Programm ist $\forall n \in \mathbb{N}$ identisch bis auf die binäre Länge von n. Folglich gilt: $K(w_n) \leq \lceil \log_2(n+1) \rceil + c$

Da $|w_n| = 2^{2^{3n}}$, gilt $n = \frac{1}{3} \log_2(\log_2(|w_n|))$ $\square$

Behaupt.: Sei $(p_i)_{i\geq 1}$ alle Primzahlen s.d. $p_i < p_{i+1}$. Dann $\exists n_0 \in \mathbb{N}$ s.d. $\forall n \geq n_0 : K(p_n) < \log_2(p_n) - 2$

Hinweis: Aus dem Primzahlsatz folgt $p_n \in \Theta(n \cdot \ln(n))$

Beweis: Es lässt sich ein einfacher Algorithmus konstruieren, der aus n die n-te Primzahl p_n generiert. Also $K(p_n) \leq \lceil \log_2(n+1) \rceil + e \,\hat{=}\, \log_2(n) + c$.

Hinweis $\Rightarrow$ $p_n \geq d \cdot n \cdot \ln(n)$ für ein $d \in \mathbb{R}$. Dann gilt:

$$\log_2(p_n) - 2 \geq \log_2(d n \ln(n)) - 2 = \log_2(d) + \log_2(n) + \log_2(\ln(n)) - 2$$

Setze nun die Abschätzung ein: $K(p_n) \leq \log_2(n) + c \overset{!}{<} \log_2(d) + \log_2(n) + \log_2(\ln(n)) - 2$

$$\Leftrightarrow \quad c < \log_2(d) + \log_2(\ln(n)) - 2$$

$$\Leftrightarrow \quad c - \log_2(d) + 2 < \log_2(\ln(n))$$

$$\Leftrightarrow \quad \exp(2^{c - \log_2(d) + 2}) < n$$

Somit gilt die Behauptung $\forall n \geq n_0 := \exp(2^{c+2}/d) + 1$ $\square$

Beh.: $L = \{0^n 1^m \mid n, m \in \mathbb{N}_{\geq 0},\ n \leq m \leq 2n\}$ ist nicht regulär.

Beweis: Benutze Lemma 3.3

Sei L regulär. Dann $\exists$ EA $A = (Q, \Sigma, \delta, q_0, F)$ mit $L(A) = L$. Betrachte $(\lambda, 0, \dots, 0^{|Q|})$.

Nach dem Schubfachprinzip existieren $i, j \in \{0, \dots, m\}$ mit $i < j$ s.d. $\delta^*(q_0, 0^i) = \delta^*(q_0, 0^j)$.

Nach Lemma 3.3. gilt $\forall z \in \{0,1\}^* : 0^i z \in L \Leftrightarrow 0^j z \in L$.

Wähle $z = 1^i$. Dann gilt $0^i 1^i \in L$, aber $0^j 1^i \notin L$, da $j > i$ $\lightning$ $\square$