Theoretische Informatik Übung 02 — W03 — Rita Nagy

2.4 Kolmogorov-Komplexität

Def: Für jedes Wort $x \in (\Sigma_{bool})^*$ ist die <u>Kolmogorov-Komplexität</u> $K(x)$ des Wortes x das Minimum der binären Längen der Pascal-Programme, die x generieren.

Lemma 2.4: Es existiert eine Konstante d s.d. $\forall x \in (\Sigma_{bool})^*$:

$K(x) \leq |x| + d$

Algo A_x:
```
begin
  write (x);
end
```
Alles ausser x ist konst.

Def: Die <u>Kolmogorov-Komplexität einer natürlichen Zahl</u> n ist

$K(n) = K(Bin(n))$

Aufg 1: Wie lange ist $Bin(n)$ bei einer geg. Zahl $n \geq 1$?

Lösung: $\lceil \log_2(n+1) \rceil$

$= \lfloor \log_2(n) \rfloor + n$

Erklärung: Grösste Zahl n mit $|Bin(n)| = m$ ist $n = 2^m - 1$

$\log((2^m - 1) + 1) = \log(2^m) = m$

Satz 2.2: Sei L eine Sprache über Σ_{bool}.

Sei $\forall n \in \mathbb{N}^+$ z_n das n-te Wort in L bzgl. der kanonischen Ordnung.

Falls $\exists A_L$ Programm, s.d. A_L löst Entscheidungsproblem (Σ_{bool}, L),

dann gilt $\forall n \in \mathbb{N}^+$:

$K(z_n) \leq \lceil \log_2(z_n + 1) \rceil + c$

wobei c eine von n unabhängige Konstante ist.

Bmk: $x :=$ Nachfolger von x in kanonischer Ordnung;

Code funktioniert über $(\Sigma_{bool})^*$, aber nicht über $L \subseteq (\Sigma_{bool})^*$

Aufg 2: Sei $(w_j)_{j=0}^{\infty}$ eine Folge von Wörtern geg. durch $w_j = 0^{j^3}$.

Zz: Es gibt eine Konstante c s.d. $\forall j \in \mathbb{N}$:

$K(w_j) \leq \frac{1}{3} \log |w_j| + c$ (Midterm HS04)

Lösung: Pseudo-Code:

```
begin
  N := j;
  N := N·N·N;
  for i:=1 to N do
    write(0);
end
```

Komplexitätsberechnung:

Alles ausser das j im Pseudo-Code ist konstant.

$K(w_j) \leq K(\mathrm{Bin}(j)) + c' \leq \lceil \log_2(j+1) \rceil + c'$

$|w_j| = |0^{j^3}| = j^3 \Rightarrow j = |w_j|^{\frac{1}{3}}$

$\Rightarrow K(w_j) \leq \lceil \log(|w_j|^{1/3} + 1) \rceil + c' \leq \frac{1}{3} \log(|w_j|) + c$ □

Rezept: 1) Pseudo-Code schreiben

2) $K(w_j)$ in Abhängigkeit von nicht-konst. Variable j angeben

3) j in Abhängigkeit von $|w_j|$ umschreiben

4) $j = \ldots$ in $K(w_j)$ einsetzen

5) umformen

Aufg. 3: Sei $w_n := (01)^{18}(0101)^{2^{3^n}} \in \{0,1\}^* \quad \forall n \in \mathbb{N}^+$.

Finde eine möglichst gute obere Schranke von $K(w_n)$ in Abhängigkeit von $|w_n|$.

Lösung: Pseudo-Code:

```
P_wn: begin
        for i:=1 to 18
          write (01);
        end
        M := n;
        a := 3^M
        b := 2^a
        for i:=1 to b
          write (0101)
        end
      end
```

Komplexitätsberechnung:

Bis auf Kodierung von n ($\lceil \log_2(n+1) \rceil$) hat Maschinencode von P_{w_n} konst. Länge (c).

$\Rightarrow |P_{w_n}| \leq \lceil \log(n+1) \rceil + c \leq \log(n) + c'$

$|w_n| = 36 + 4 \cdot 2^{3^n}$

$\Rightarrow n = \log_3 \log_2 \left(\frac{|w_n| - 36}{4} \right)$

$\Rightarrow K(w_n) \leq \log_2 \log_3 \log_2 \left(\frac{|w_n| - 36}{4} \right) + c'$

$\leq \log_2 \left(\log_3 \log_2 (|w_n| - 36) - \log_3 \log_2 (4) \right) + c'$

$\leq \log_2 \log_3 \log_2 (|w_n|) + c'$

Def: Ein Wort $x \in (\Sigma_{bool})^*$ heisst <u>zufällig</u>, falls $K(x) \geq |x|$.

Eine Zahl n heisst <u>zufällig</u>, falls $K(n) = K(Bin(n)) \geq \lceil \log_2(n+1) \rceil - 1$.

Lemma 2.5: $\forall n \in \mathbb{N}^+ \quad \exists w_n \in (\Sigma_{bool})^n$ s.d.

$$K(w_n) \geq |w_n| = n$$

d.h. Für jede Zahl n existiert ein nichtkomprimierbares Wort der Länge n.

Aufg 4: Zz: Die Kolmogorov Komplexität von mehr als 99% der Wörter in $\{0,1\}^n$ ist grösser als $n-8$ für $n \in \mathbb{N}$.

Lösung: • Für $n \leq 8$ trivial, da die Kolmogorow-Komplexität einer positiven Zahl entspricht.

Sei $n > 8$

• Anzahl Wörter in $\{0,1\}^n$: $|\{0,1\}^n| = \underline{2^n}$.

• Anzahl Programme mit Länge $\leq n-8$: $\sum_{i=1}^{n-8} 2^i = 2^{n-8+1} - 2 = \underline{2^{n-7} - 2}$

$\Rightarrow$ Mit $2^{n-7} - 2$ versch. Programmen können höchstens

$$\frac{2^{n-7}-2}{2^n} = 2^{-7} - 2^{1-n} < 2^{-7} = \frac{1}{128} < 1\%$$

aller Wörter der Länge n dargestellt werden. $\square$

Serie 2

Hinweise:

A1 a) Siehe Aufg 2

b) Korrektheit der geg. Folge beweisen (wie in a)

A2 Siehe Aufg 4

A3 a) Anzahl Wörter dieser Form bestimmen; Satz 2.2