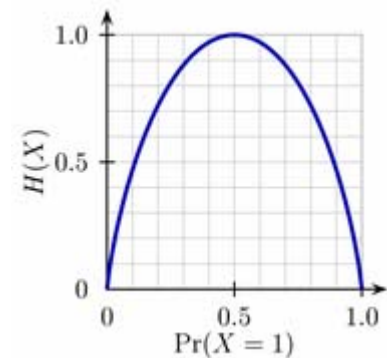


Information und Kommunikation

Hartmut Klauck
Universität Frankfurt
SS 07
4.6.



Eine Bemerkung

- Die Begriffswahl „zufällig“ für nichtkomprimierbare Sequenzen kann genauer begründet werden:
 - Martin-Löf Zufälligkeit
 - Definition über berechenbare statistische Tests
 - Äquivalent zu Nichtkomprimierbarkeit

K-Komplexität

- Wir unterscheiden noch von der C -Komplexität die *präfixfreie* Kolmogorov Komplexität
- **Definition 14.1**
 - $K(x)$ ist wie $C(x)$ definiert, außer, dass die erlaubten Programme einen präfixfreien Code bilden müssen

K-Komplexität

- $K(x,y)$: Komplexität des Paares x,y mit Trennsymbol
- Jetzt gilt:
 - $K(x,y) \leq K(x) + K(y) + O(1)$
- Tatsächlich gilt dies nicht für $C()$:
 - Es gibt x,y mit $C(x,y) \geq C(x) + C(y) + \log n - O(1)$
 - Es gibt $(n+1)2^n$ Paare x,y mit $|x| + |y| = n$.
Eines dieser Paare muss Komplexität $n + \log n$ haben.

Präfixkodierung

- Wir können den Unterschied zwischen $C()$ und $K()$ leicht beschränken
- Zu gegebenem $x = x_1, \dots, x_n$ sei
 $t(x) = x_1, 0, x_2, 0, \dots, x_{n-1}, 0, x_n, 1$
- Wir kodieren x als $t(|x|)x$
- Klar: der Code ist präfixfrei
- Die Länge steigt additiv um $2 \log |x|$

Berechenbarkeit

- $C(x)$ gibt also an, wie stark x komprimiert werden kann, durch eine *beliebige* Methode
- Können wir $C(x)$ bestimmen?
- **Theorem 14.2**
 - $C(x)$ ist nicht berechenbar

Beweis 14.2

- Angenommen es gibt eine TM M , die auf Eingabe x Ausgabe $C(x)$ hat
- Wir beschreiben eine Prozedur, die einen beliebig komplexen String erzeugt
 - Eingabe n
 - Durchlaufe alle i von 1 bis n
 - Durchlaufe alle Strings x der Länge i und berechne $C(x)$
 - Wenn $C(x) \geq n$, gebe x aus

Beweis 14.2

- Klar: Das Programm wird entweder nichts ausgeben, oder einen String x mit $C(x) \geq n$ ausgeben
- Da es Strings der Länge n mit Komplexität n gibt, gibt das Programm einen solchen aus
- Das Programm hat aber konstante Länge
- Wir erhalten ein weiteres Programm ohne Eingabe, in dem wir n „fest verdrahten“ (mit $2 \log n$ Bits), und wir erhalten $n \leq C(x) \leq 2 \log n + O(1)$, Widerspruch
- Das bedeutet, dass die Prozedur zur Berechnung von $C(x)$ nicht existiert.

Gödels Theorem

- Gödels Theorem besagt, dass jedes (korrekte) endliche axiomatische System für die Aussagen der Arithmetik unvollständig ist
- Das bedeutet, dass es für jedes endliche Axiomensystem wahre Aussagen der Arithmetik gibt, die nicht beweisbar sind

Gödels Theorem

- **Theorem 14.3**
 - Aussagen der Form „ x ist zufällig“ sind nicht beweisbar (in einem endlichen axiomatischen System).
- **Beweis:**
 - Angenommen mindestens eine Aussage „ x ist zufällig“ ist beweisbar
 - Sei F die Menge der Axiome
 - $C(F) = C = O(1)$
 - $|x| \gg C$, sonst ist x nicht nichtkomprimierbar
 - Wir durchsuchen also alle Beweise nach einem für „ x ist zufällig“ für ein x mit $|x| \gg C$, geben das erste solche x aus
 - Klar: $C(x) \leq C + O(1)$, denn x ist durch F definiert, aber $C(x) \gg C$, wenn die Aussage korrekt ist. D.h. entweder ist das System fehlerhaft, oder es gibt einen solchen Beweis nicht.
 - Das bedeutet: Es gibt wahre Aussagen, die nicht beweisbar sind
 - Arithmetik ist stark genug, um Aussagen wie „ x ist zufällig“ auszudrücken

Symmetrie

- Wir wissen
$$I(X:Y) = H(X) - H(X|Y)$$
$$= H(X) + H(Y) - H(XY)$$
$$= H(Y) + H(Y|X) = I(Y:X)$$
- **Theorem 14.4**
 - Es gilt: $C(X) - C(X|Y) = C(Y) - C(Y|X) \pm O(\log C(XY))$
- Wir zeigen folgende Kettenregel:
 - **Lemma 14.5**
 - $C(X,Y) = C(X) + C(Y|X) \pm O(\log C(X,Y))$
 - Dies impliziert offensichtlich 14.4

Beweis 14.5

- Wir wollen
 $C(X,Y) = C(X) + C(Y|X) \pm O(\log C(X,Y))$
zeigen
- „ \leq “:
 - Gebe X aus, verwende dazu binär kodiert die Länge des Programms für X
 - Gebe Y aus, dazu darf ein Programm verwendet werden, dass X als Eingabe erhält

Beweis 14.5

- „ \geq “: $C(x,y) \geq C(x)+C(y|x)-O(\log C(x,y))$
- Angenommen
 $C(x,y) < C(x)+C(y|x)-k \log C(x,y)$ für beliebig
grosses k
- Sei $A=\{(u,z): C(u,z) \leq C(x,y)\}$
 - Gegeben $C(x,y)$ ist A rekursiv aufzählbar
- $A_x=\{z: C(x,z) \leq C(x,y)\}$
 - Gegeben x und $C(x,y)$ ist A_x rekursiv aufzählbar

Beweis 14.5

- y kann leicht beschrieben werden, indem man den Index von y in der Aufzählung von A_x angibt
- Damit gilt
$$C(y|x) \leq \log |A_x| + 2 \log C(x,y) + O(1)$$
- Daher folgt: $|A_x| > 2^d$,
wobei $d = C(x,y) - C(x) + k \log C(x,y) - O(1)$
 - Verwende $C(x,y) < C(x) + C(y|x) - k \log C(x,y)$

Beweis 14.5

- $|A_x| > 2^d$, wobei $d = C(x,y) - C(x) + k \log C(x,y) - O(1)$
- Wir erhalten eine kurze Beschreibung von x :
 - Gegeben $C(x,y)$ und d , können Kandidaten u für x rekursiv aufgezählt werden:
 - Kandidaten sind u mit $2^d < |A_u|$, wobei $A_u = \{z: C(u,z) \leq C(x,y)\}$
 - Sei U die Menge all dieser Kandidaten. Es gilt $x \in U$.
 - $\{(u,z): u \in U \text{ und } z \in A_u\} \subseteq A$
 - $|A| \leq 2^{C(x,y)+O(1)}$
 - Damit folgt $|U| < |A|/2^d \leq 2^{C(x,y)+O(1)}/2^d$
 - Also kann x rekonstruiert werden, wenn wir $C(x,y)$, d , und den Index von x in U kennen
 - $C(x) < 2 \log C(x,y) + 2 \log d + C(x,y) - d + O(1)$
 - $C(x) < 4 \log C(x,y) + C(x) - k \log C(x,y) + O(1) < C(x)$, Widerspruch!

Information

- Damit ist ein Begriff von Information $C(x) - C(x|y)$ wohldefiniert