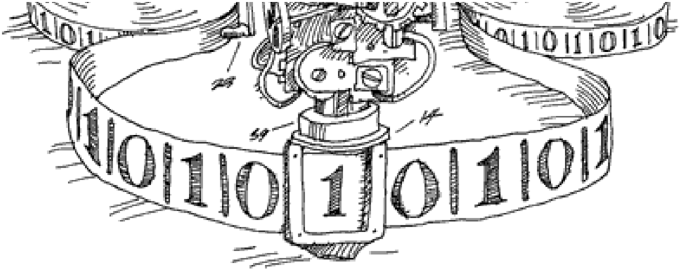
Karlsruher Institut für Technologie

Theoretische Grundlagen der Informatik

Tutorium 11

Institut für Kryptographie und Sicherheit



Wiederholung: Kolmogorov Komplexität

Eine minimale Beschreibung eines Wortes w heißt Kolmogorow-Komplexität $K(w)$

- Also: $\forall d(w) : |d(w)| \geq |K(w)|$
- Die Länge von $K(w)$ ist abhängig von der Struktur von w

Falls $|K(w)| \geq |w|$ heißt das Wort unkomprimierbar.

Wiederholung: Kolmogorov Komplexität

Beispiel:
Geben Sie eine möglichst gute obere Schranke für die Kolmogorow-Komplexität von $x = 0^n$ an!

Wiederholung: Kolmogorov Komplexität

Beispiel:
Geben Sie eine möglichst gute obere Schranke für die Kolmogorow-Komplexität von $x = 0^n$ an!

- n ist die Anzahl an Nullen

Wiederholung: Kolmogorov Komplexität

Beispiel:
Geben Sie eine möglichst gute obere Schranke für die Kolmogorow-Komplexität von $x = 0^n$ an!

- n ist die Anzahl an Nullen
- n lässt sich binär kodieren mit $log_2(n)$ viel Platz

Wiederholung: Kolmogorov Komplexität

Beispiel:
Geben Sie eine möglichst gute obere Schranke für die Kolmogorow-Komplexität von $x = 0^n$ an!

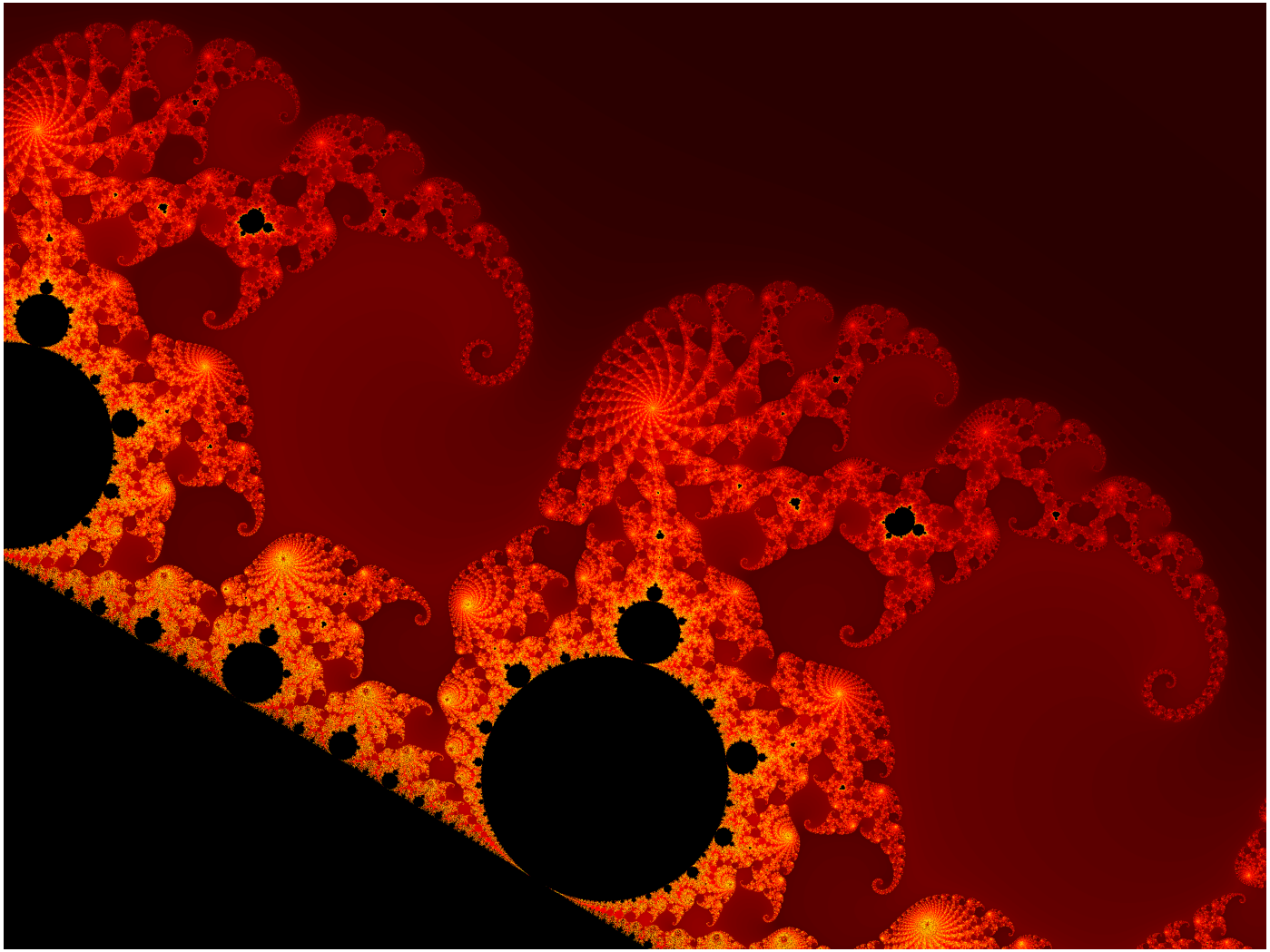
- n ist die Anzahl an Nullen
- n lässt sich binär kodieren mit $log_2(n)$ viel Platz
- $K(x) \leq log_2(n) + c$, wobei c die *konstante* Größe einer Turingmaschine ist, die in Bezug auf das Problem bei *jeder* möglichen Kodierung als Eingabe die Ausgabe x liefert.

Wiederholung: Kolmogorov Komplexität

Beispiel:
Geben Sie eine möglichst gute obere Schranke für die Kolmogorow-Komplexität von $x = 0^n$ an!

- n ist die Anzahl an Nullen
- n lässt sich binär kodieren mit $log_2(n)$ viel Platz
- $K(x) \leq log_2(n) + c$, wobei c die *konstante* Größe einer Turingmaschine ist, die in Bezug auf das Problem bei *jeder* möglichen Kodierung als Eingabe die Ausgabe x liefert.

Vorsicht: Aufpassen, in Abhängigkeit wovon eine obere Schranke angegeben werden soll.



Multiple Choice

Die Kolmogorov-Komplexität einer Zeichenkette $x \in \{0,1\}^n$ ist immer größer als $log_2(n)$

Multiple Choice

Das Hamilton-Kreis Problem ist NP-vollständig

Multiple Choice

In der Klasse NP liegen nicht-entscheidbare Probleme

Multiple Choice

Das Vertex-Cover Problem ist NP-vollständig

Multiple Choice

Semi-entscheidbare Sprachen sind unter Komplementbildung abgeschlossen

Multiple Choice

Nichtdeterministische endliche Automaten sind echt mächtiger als deterministische

Multiple Choice

Zu jeder CH-2-Sprache gibt es eine CH-1-Grammatik

Multiple Choice

Um zu zeigen, dass ein Problem Π NP-vollständig ist, genügt es, ein NP-schweres Problem auf Π zu reduzieren.

Bis zum nächsten Mal!

Kolmogorov Directions

Lizenzen