Kolmogorov-Komplexität

ProSeminar zur Theoretischen Informatik im WS 2014/15 bei Herrn Prof. Dr. Alt
vorgetragen am 3. Februar 2015 von Anna Borghardt

1 Motivation

Ziel: Wir möchten ein Maß für den Informationsgehalt eines Objektes finden.

Beispiel. Welcher der Strings ist komplexer? In welchem String steckt mehr Information?

$$A = 01010101010101010101010101010101$$
$$B = 11100101101000111010100001110100$$

Vermutlich hat String B mehr Information, weil String A bloß eine 16fache Wiederholung des Musters 01 ist. A kann zum Beispiel durch $(16, '01')$ beschrieben werden, aber für B gibt es keine offensichtliche kürzere Beschreibung.

Idee: Je kürzer ein Objekt beschrieben werden kann, desto weniger Informationen kann es enthalten. Die Länge der kürzesten Beschreibung eines Objektes liefert ein Maß für dessen Informationsgehalt.

2 Ein Maß für den Informationsgehalt eines Strings

Um im Sinne der Idee ein Maß für den Informationsgehalt eines Objektes zu finden, legen wir zunächst fest, dass wir nur Objekte betrachten, die Binärstrings sind und formalisieren dann den Begriff „Beschreibung“.

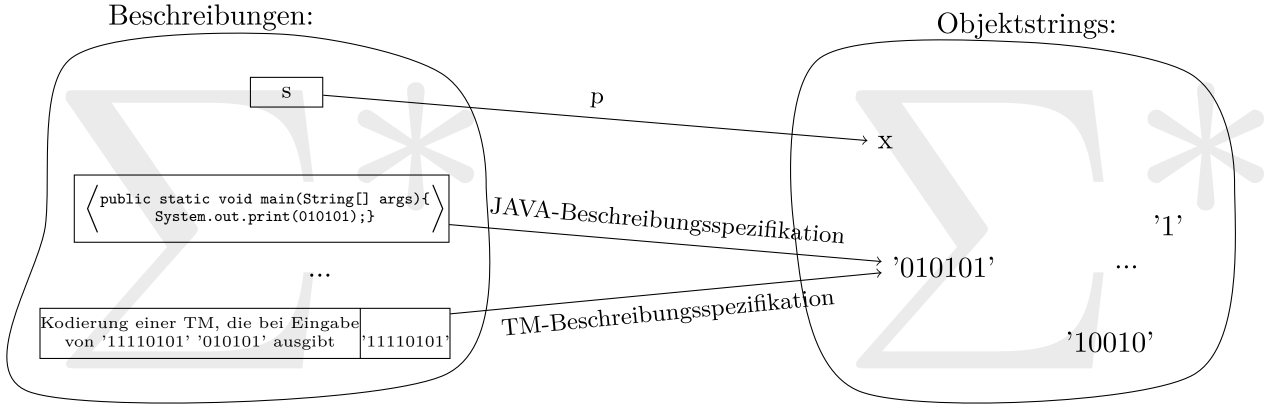
2.1 Alphabet

Zur Vereinfachung beschränken wir uns auf das Alphabet $\Sigma = \{0, 1\}$. Das heißt, dass sowohl die Objekte, die wir beschreiben wollen, als auch die Beschreibungen Binärstrings sind.

2.2 Beschreibung

Definition. Sei $p : D \to \Sigma^*$ mit $D \subseteq \Sigma^*$ eine beliebige berechenbare Funktion. Diese berechenbare Funktion wird *Beschreibungsspezifikation* genannt. Eine *p-Beschreibung* eines Strings $x \in \Sigma^*$ ist ein Binärstring $s \in D$ mit $p(s) = x$.

Diese Skizze (vgl. [V]) veranschaulicht die Definition:



Definition. Die *p-Kolmogorov-Komplexität* eines Strings x ist die Länge seiner kürzesten p-Beschreibung, d.h. $K_p(x) := |d_p(x)|$. Dabei ist $d_p(x)$ die kürzeste p-Beschreibung von x, also die p-Beschreibung mit der geringsten Zeichenanzahl und bei gleicher Zeichenanzahl die lexikographisch Erste.

Wir haben nun gemäß unserer Idee ein Maß für den Informationsgehalt eines Strings gefunden – die p-Kolmogorov-Komplexität. Das Maß ist allerdings abhängig von der Beschreibungsspezifikation p. Damit wir eine allgemeine Kolmogorov-Komplexität definieren können, fixieren wir die folgende universelle Beschreibung durch eine Turingmaschine.

2.3 Beschreibung durch Turingmaschine und allgemeine Kolmogorov-Komplexität

Definition. Die *TM-Beschreibung* eines Strings $x \in \Sigma^*$ ist die eindeutige Verkettung $\langle M \rangle w$ der Kodierung einer Turingmaschine M und einer Eingabe $w \in \Sigma^*$, wobei die TM M bei der Eingabe von w den String x auf das Band schreibt. Dabei sei $\langle M \rangle$ die Gödelnummer der TM M mit verdoppelten Bits ('0'/'1'→'00'/'11') und anschließender Markierung '01'.

Beispiel. 0110101001 sei die Gödelnummer einer TM, die bei der Eingabe $w = 01$ einen String x ausgibt. Eine TM-Beschreibung von x ist:

$$\underbrace{00111100110011000011\overbrace{01}^{\text{Markierung}}}_{\langle M \rangle}\underbrace{01}_{w}$$

Lemma. *Jeder String hat eine Beschreibung.*

Beweis. Sei $x \in \Sigma^*$ ein beliebiger String und sei M die TM, die ihre Eingabe sofort ausgibt. Dann ist $\langle M \rangle x$ eine (TM-)Beschreibung von x. □

Definition. Die *Kolmogorov-Komplexität* $K(x)$ eines Strings $x \in \Sigma^*$ ist die Länge der kürzesten TM-Beschreibung von x.

3 Das Invarianztheorem

Wir werden nun zeigen, dass unsere allgemeine Definition der Kolmogorov-Komplexität K gerechtfertigt ist, weil die kürzeste p-Beschreibung nicht wesentlich prägnanter sein kann als die kürzeste TM-Beschreibung.

Satz (Invarianztheorem)**.** *Für alle Beschreibungsspezifikationen p existiert eine Konstante $c > 0$, sodass für alle Strings $x \in \Sigma^*$ gilt:*

$$K(x) \leq K_p(x) + c.$$

Beweis. Sei $x \in \Sigma^*$ ein beliebiger String und sei p eine beliebige Beschreibungsspezifikation.
Betrachte folgende TM U.

```
U mit Eingabe s der Form einer p-Beschreibung eines Strings x:
    1. Gib p(s)=x aus.
```

$\langle U \rangle d_\mathrm{p}(x)$ ist eine TM-Beschreibung von x, denn die TM U mit Eingabe $d_p(x)$ gibt $p(d_p(x)) = x$ aus. Die kürzeste TM-Beschreibung ist höchstens so lang wie diese TM-Beschreibung, also gilt

$$K(x) \leq \underbrace{|\langle U \rangle|}_{=:c} + |d_\mathrm{p}(x)| = c + K_\mathrm{p}(x),$$

wobei die Konstante c die Länge der Kodierung der TM U ist, also unabhängig vom String x ist. □

Das Invarianztheorem lässt sich einfach auf andere universelle Beschreibungsspezifikationen übertragen. Dabei bedeutet „universelle Beschreibungsspezifikation“, dass die Beschreibungsspezifikation einen Algorithmus mit seiner Eingabe simuliert, wie zum Beispiel eine Beschreibungsspezifikation für JAVA-Programme oder Python-Programme mit ihren Eingaben. Für alle Strings x existiert eine Konstante c, sodass $K_{Java}(x) \leq K_{Python}(x) + c$ und $K_{Python}(x) \leq K_{Java}(x) + c$ gilt. Damit folgt $|K_{Java}(x) - K_{Python}(x)| \leq c$. Es ist also egal, ob ein String mit einem JAVA-Programm, mit einem Python-Programm oder einer Turingmaschine beschrieben wird, denn die entsprechenden Kolmogorov-Komplexitäten unterscheiden sich höchstens um eine Konstante.

4 Eigenschaften der Kolmogorov-Komplexität

4.1 Abschätzungen der Kolmogorov-Komplexität

Satz. *Die Kolmogorov-Komplexität eines Strings ist höchstens um eine feste Konstante größer als die Länge des Strings selbst, d.h.*

$$\exists c > 0 \, \forall x \in \Sigma^* : \, K(x) \leq |x| + c.$$

Beweis. Sei $x \in \Sigma^*$ ein beliebiger String und sei M eine TM, die ihre Eingabe sofort ausgibt. Dann ist $\langle M \rangle x$ eine TM-Beschreibung von x. Die Kolmogorov-Komplexität ist höchstens so groß wie die Länge dieser Beschreibung. Damit folgt $K(x) \leq \underbrace{|\langle M \rangle|}_{=:c} + |x|$. □

Satz. *Der Informationgehalt von xx ist nicht wesentlich größer als der von x, d.h. es gibt eine Konstante $c > 0$, sodass für jeden String $x \in \Sigma^*$ $K(xx) \leq K(x) + c$ gilt.*

Beweis. Sei $x \in \Sigma^*$ ein beliebiger String. Betrachte folgende Turingmaschine M.

```
M bekommt eine Eingabe der Form ⟨N⟩w, wobei N eine TM mit Eingabe w ist.
    1. Führe N mit Eingabe w aus und ermittle so die Ausgabe s.
    2. Gib ss aus.
```

$\langle M \rangle d_{TM}(x)$ ist eine TM-Beschreibung von xx, wobei $d_{TM}(x)$ die kürzeste TM-Beschreibung ist.
Die kürzeste TM-Beschreibung von xx ist höchstens so lang wie die gerade konstruierte Beschreibung.
Damit folgt $K(xx) \leq \underbrace{|\langle M \rangle|}_{=:c} + |d_{TM}(x)| = c + K(x)$. □

Für String A aus unserem Anfangsbeispiel bedeutet diese Tatsache, dass String A tatsächlich nicht viel mehr Information enthält als der String '01':

$$\text{K}(01010101010101010101010101010101) \leq \text{K}(0101010101010101) + c \leq ... \leq \text{K}(01) + \underbrace{4}_{\log_2(16 \text{ Wiederholungen})} c$$

4.2 Nichtberechenbarkeit der Kolmogorov-Komplexität

Schön wäre es die Kolmogorov-Komplexitäten der Beispielstrings A und B zu vergleichen, aber leider ist die Kolmogorov-Komplexitäten nicht berechnenbar. Um dies zu zeigen, benötigen wir folgendes Lemma über nicht komprimierbare Strings.

Definition. Ein String $x \in \Sigma^*$ heißt *nicht komprimierbar*, falls x keine kürzere TM-Beschreibung hat als $|x|$, d.h. $K(x) \geq |x|$.

Lemma. *Es gibt nicht-komprimierbare Strings beliebiger Länge.*

Beweis. Sei $n \in \mathbb{N}$. Die Anzahl an Binärstrings der Länge n ist 2^n. Die Anzahl von Beschreibungen mit kleinerer Länge als n lässt sich wie folgt abschätzen:

$$\left|\substack{\text{Beschreibungen mit} \\ \text{Länge} < \text{n}}\right| \leq \sum_{i=0}^{n-1} \left|\substack{\text{Strings der} \\ \text{Länge i}}\right| = \sum_{i=0}^{n-1} 2^i \underbrace{=}_{\text{geom. Reihe}} 2^n - 1$$

Es gibt also mehr Strings der Länge n als Beschreibungen mit kürzerer Länge. Deshalb ist mindestens ein String der Länge n nicht-komprimierbar. □

Satz. *Die Kolmogorov-Komplexität ist nicht berechenbar.*

Beweis (durch Widerspruch). Es wird angenommen, dass die Kolmogorov-Komplexität berechenbar ist. Betrachte folgende TM M.

```
M mit beliebiger Eingabe:
  1. Verwerfe die Eingabe.
  2. for(len = 1; ; len++){
  3.     - Berechne lexikographisch geordnete Liste aller Strings der Länge len.
  4.     - Überprüfe nacheinander, ob die Kolmogorov-Komplexität der Listenelemente
           größer ist als eine hart kodierte Zahl z.
  5.       Falls ja, gib das Listenelement aus und halte an.
     }
```

Die TM M gibt einen String x aus, der eine größere Kolmogorov-Komplexität hat als z, d.h. $K(x) > z$. Eine TM-Beschreibung von x ist $\langle M\rangle\varepsilon$ mit dem leeren Wort ε. Die Länge dieser TM-Beschreibung ist $|\langle M\rangle|$. Wähle die Zahl z größer als diese TM-Beschreibung von x lang ist, d.h. $z > |\langle M\rangle|$. Dann folgt, dass die Kolmogorov-Komplexität von x größer ist als die TM-Beschreibung von x lang ist, d.h. $K(x) > |\langle M\rangle|$. Dies ist ein Widerspruch, weil die Kolmogorov-Komplexität die Länge der kürzesten TM-Beschreibung ist. Also war die Annahme falsch, d.h. die Kolmogorov-Komplexität ist nicht berechenbar.

Einige nicht sofort klare Stellen der TM M sollten noch betrachtet werden:
Das Überprüfen, ob die Kolmogorov-Komplexität größer ist als die Zahl z in Zeile 4 ist möglich, weil die Kolmogorov-Komplexität nach Annahme berechenbar ist.
Die for-Schleife (Zeile 2) terminiert, weil es nach obigem Lemma nicht komprimierbare Strings jeder Länge gibt. Also gibt es spätestens im $z+1$-ten Durchlauf der Schleife einen String x mit $K(x) \geq |x| = z+1$, also $K(x) > z$.
Die Turingmaschine kann eleganter konstruiert werden, indem sie die Länge ihrer eigenen Kodierung berechnet statt die Zahl z fest zu kodieren. Das ist aufgrund des Rekursionssatzes von Kleene und dessen Folgerungen möglich.(vgl. [S], Kapitel 6.1) □

Wir haben gezeigt, dass die Kolmogorov-Komplexität eines Strings x nicht größer sein kann als $|x|+c$ für $c > 0$ und nicht berechenbar ist. Man kann die Kolmogorov-Komplexität aber approximieren.

5 Ausblick: Nichtkomprimierbare Strings und Zufälligkeit

Anwendungen hat die Kolmogorov-Komplexität zum Beispiel in der Datenkomprimierung und bei zufälligen Strings. Dabei wird ein zufälliger String so definiert, dass er auf uns wirkt als sei er zufällig entstanden. Das wird deutlich bei der Betrachtung der Strings '1111111111' und '1101110010'. Die Wahrscheinlichkeiten die beiden Strings durch Münzwürfe zu werfen ist gleich, aber der zweite String erscheint uns zufälliger. Ein Versuch diese gefühlte Zufälligkeit zu beschreiben ist zufällige Strings als nicht komprimierbare Strings zu definieren. Eigenschaften nicht komprimierbarer Strings lassen sich dann auf zufällige Strings übertragen.
Man kann zum Beispiel zeigen, dass nicht komprimierbare Binärstrings der Länge n ungefähr gleich viele 1en und 0en haben und die längste Folge von 0en ungefähr $\log n$ lang ist. Diese Anforderungen sollten bei der Konstruktion von Zufallszahlen auch erfüllt werden. Außerdem kann gezeigt werden, dass die Frage, ob ein String komprimierbar ist, nicht entscheidbar ist und dass fast alle Strings nicht komprimierbar sind.

Literatur

[S] Sipser, Michael: Introduction to the Theory of Computation, 3rd Edition, Kapitel 6.4, Cengage Learning, Boston, 2013. ISBN-13: 978-1-133-18779-0.

[V] Vitanyi, Paul: Vorlesungsfolien zur Kolmogorov-Komplexität, Amsterdam, 2004. http://homepages.cwi.nl/~paulv/lectures/lectures_kc/jpg/PV_Lecture_016.jpg (Stand: 31.1.2015)

[G] Gresag, Jörg: Die Grenzen der Berechenbarkeit – Unvollständigkeit und Zufall in der Mathematik, Kapitel 3.4, 2008. http://www.joergresag.privat.t-online.de/mybk3htm/chap34.htm (Stand: 31.1.2015)