Zufall

"Wahrscheinlichkeit" und "Zufall" werden für verschiedene Konzepte benutzt.

- Klassisch: Man betrachtet alle möglichen Ergebnisse eines Zufallsexperiments und zählt. Beim Würfeln ist die Wahrscheinlichkeit, dass die Augenzahl durch drei teilbar ist 2/6, weil es 2 solche Augenzahlen unter den 6 möglichen gibt.

  Es ist nicht klar, was hier den Zufall ausmacht

- Subjektivistisch: Wahrscheinlichkeiten drücken aus, wieviele Informationen für Vorhersagen man hat. Die Wahrscheinlichkeit, die man einem Ereignis zuordnet, ist die Geldmenge, die man darauf setzen würde, bekäme man 1€ für richtiges Raten.

  Hier ist es egal, ob die Ereignisse echt zufällig eintreten, oder einer Regel folgen, die man bloß nicht kennt.

(für weiteres siehe Stanford Encyclopedia of Philosophy.

Die algorithmische Zufallsdefinition ist ähnlich.

> Eine unendliche Folge (von Bits) ist zufällig, genau dann wenn es keine Turingmaschine gibt, die sie vorhersagen kann.

Das schließt intuitiv zufällige Folgen ein, wie sie etwa durch Werfen einer Münze erzeugt werden aber auch ~~solgen~~ Folgen, die nicht berechenbar sind. Eine solche Folge kann man etwa über das Halteproblem definieren.

Wenn man die Turingmaschinen, die die Folgen vorhersagen sollen einschränkt (z.B. auf polynomielle Zeit) erhält man Zufallsdefinitionen, die für Kryptographieanwendungen interessant sind.

Zufall und Rechnen

Kann man, mit dem Zugang zu einer Quelle von Zufall, mehr Probleme lösen?

Nein, mit genügend Zeit kann eine Turingmaschine alle möglichen Zufallszahlen durchprobieren.

Es ist aber noch ungeklärt, ob man Probleme mit Hilfe des Zufalls schneller lösen kann.

Sortieren von Schrauben und Muttern

Wir haben einen Haufen mit Schrauben und einen Haufen mit Muttern. Zu jeder Schraube passt genau eine Mutter, wir wollen die Paare finden.
Die Schwierigkeit besteht darin, dass Schrauben bzw. Muttern untereinander nicht vergleichbar sind. Man kann lediglich versuchen eine Schraube in eine Mutter zu schrauben und sieht ob sie passt, zu groß oder zu klein ist.

Deterministisch finden wir leicht einen quadratischen Algorithmus.

Mittels Randomisierung können wir aber einen schnelleren Algorithmus angeben, der wie Quicksort funktioniert. Wir wählen zufällig eine Schraube aus und benutzen sie um den Mutternhaufen zu teilen. Mit der passenden Mutter können wir den Schraubenhaufen schreiben.

Für dieses Problem kennt man auch einen deterministischen Algorithmus — er ist aber viel komplizierter und ein bisschen langsamer als die randomisierte Variante.

Datenübertragungen überprüfen

Wir haben eine große Datei heruntergeladen und wollen überprüfen, ob es unterwegs zu Übertragungsfehlern gekommen ist.
Dafür gibt es verschiedene (auch deterministische Verfahren).
Wir könnten zufällig Bits auswählen, aber dann übersehen wir wahrscheinlich einzelne falsche Bits.
Stattdessen definieren wir uns folgendes Polynom:

$$p(x) = \prod (x - b_i) \qquad b_i : i\text{-te bit in der Datei}$$

~~Dann~~ Das macht auch die andere Seite der Übertragung und bekommt ein $q(x)$.
Wir möchten wissen ob $p(x) = q(x)$, ohne alle b_i übertragen zu müssen.
Dazu wählen wir zufällige x aus und übertragen die Funktionswerte an diesen Stellen. Weil zwei verschiedene Polynome nur an begrenzt vielen Stellen übereinstimmen können, finden wir mit ~~konstant~~ großer Wahrscheinlichkeit einen Unterschied wenn wir nur wenige verschiedene x ausprobiert haben.

Man kennt kein deterministisches Verfahren, das zwei solche Polynome auf Gleichheit testet.