Informatik-Propädeutikum

Dozentin: Dr. Claudia Ermel

Betreuer: Sepp Hartung, André Nichterlein, Clemens Hoffmann

Sekretariat: Christlinde Thielcke (TEL 509b)

TU Berlin
Institut für Softwaretechnik und Theoretische Informatik
Prof. Niedermeier
Fachgruppe Algorithmik und Komplexitätstheorie
http://www.akt.tu-berlin.de

Wintersemester 2013/2014

Gliederung

Zufall

Zufällige Ereignisse geschehen objektiv ohne (erkennbare) Ursache.

Z.B tritt dies auf bei Radioaktivität, d.h. der Zeitpunkt des Zerfalls des nächsten radioaktiven Atoms aus einer Stoffmenge ist nicht vorhersagbar.

Der Begriff des Zufalls (inklusive seiner Existenz) wird heftig in verschiedenen Gebieten wie Philosophie, Physik, Psychologie, Soziologie, etc. diskutiert (z.T. sehr kontrovers, auch aufgrund verschiedener Weltanschauungen).

Nachfolgend gehen wir von der Existenz zufälliger Ereignisse (Basismodell Münzwurf) aus und interpretieren Zufall in erster Linie als Hilfsmittel für diverse Informatikanwendungen.

Was ist Zufall?

Zufallsexperimente: Ein (gedanklicher oder tatsächlicher) Versuch, der unter gleichen Bedingungen beliebig oft wiederholbar ist, dessen Ausgang sich aber nicht exakt vorhersagen lässt, d.h. bei Wiederholung des Versuchs jedesmal ein anderes Ergebnis möglich.
Beispiele: Münzwurf, Anzahl der eine Kreuzung zwischen 12 und 13 Uhr befahrenden Fahrzeuge, Roulette, etc.

Kolmogorov-Komplexität: Eine Zeichenkette ist genau dann zufällig, wenn es kein Computerprogramm gibt, das diese Zeichenkette erzeugt und das selbst **kürzer** ist als die Zeichenkette.

Zufall in der (bzw. für die) Informatik

In dieser Vorlesung schon gesehen:

- **Simulated Annealing:** Mit einer gewissen Wahrscheinlichkeit eine schlechtere Lösung akzeptieren.
  $\rightsquigarrow$ Vermeidet das Hängenbleiben in lokalen Optima.
- **Genetische Algorithmen:** Zufällige Mutationen existierender Lösungen.

Lastverteilung (Load balancing): Zufällige Verteilung von Berechnungen oder Anfragen auf mehrere parallel arbeitende Systeme.
Bsp.: Zufällige Zuweisung von HTTP-Requests an einen von mehreren zur Verfügung stehenden Webservern.

Wichtige Gebiete, wo die Ressource Zufall genutzt wird:

- Effiziente (und einfache) Algorithmen
- Simulationsmethoden wie Monte Carlo
- Heuristiken
- Protokollentwurf
  - Internet
  - Kommunikationsnetze
  - Kryptologie

Pseudozufallszahlengeneratoren

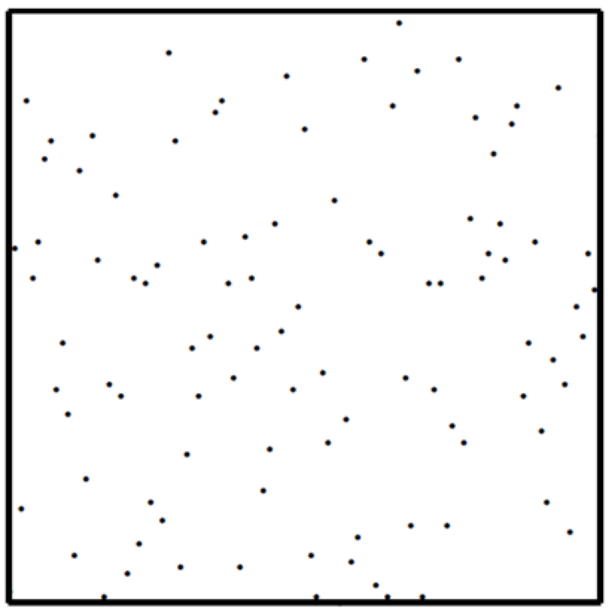
Ein Pseudozufallszahlengenerator ist ein Verfahren, das eine Folge von (vermeintlich) zufälligen bzw. zufällig erscheinenden Zahlen erzeugt. Man beginnt in der Regel mit einem vorgegebenen Startwert.

Beispiele:

- Linearer Kongruenzgenerator (z.B. in Java):
  Startwert $\boldsymbol{X_0}$
  $\boldsymbol{X_n} = (a\boldsymbol{X_{n-1}} + b) \bmod m$
- Multiply With Carry (MWC):
  r Startwerte $\boldsymbol{X_0}, \ldots \boldsymbol{X_{r-1}}$, initialer Übertrag $\boldsymbol{C_{r-1}}$
  $\boldsymbol{X_n} = (a\boldsymbol{X_{n-r}} + \boldsymbol{C_{n-1}}) \bmod m$, $\boldsymbol{C_n} = \lfloor \frac{a\boldsymbol{X_{n-r}} + \boldsymbol{C_{n-1}}}{m} \rfloor, n \geq r$
- Weitere Beispiele wie „Mersenne Twister“ (z.B. in Python, PHP, MATLAB, C++, ..) und „Linear additive feedback RNG“

Beispiel: Linearer Kongruenzgenerator

$\boldsymbol{X_n} = (a\boldsymbol{X_{n-1}} + b) \bmod m$ mit
$X_0 = 58854338,$
$a = 397204094,$
$b = 0,$
$m = 2^{31} - 1$



Waagerechte Achse: Zähler X_i von 0 bis 100,
Senkrechte Achse: Größe des Folgenglieds
von 0 bis $2^{31} - 1$

1292048469	319941267	173739233	1992841820
345565651	2011011872	31344917	592918912
1827933824	1691830787	857231706	1416540893
1184833417	145217588	589958351	1776690121
1330128247	558009026	1479515830	1197548384
1627901332	929586843	19840670	1268974074
1682548197	760357405	666131673	1642023821
787305132	1314353697	167412640	1377012759
963849348	971229179	247170576	1250747100
703109068	1791051358	1978610456	1746992541
177131972	1844679385	1328403386	1811091691
1586500120	1175539757	74957396	753264023
468643347	821920620	1269873360	963348259
1698955999	139484430	30476960	1327705603
1266305157	1337811914	1808105128	640050202
37935526	1185470453	2111728842	380228478
808553600	934194915	824017077	881361640
1492263703	414709486	298916786	1883338449
771128019	558671080	1935988732	798347213
120356246	1378842534	37149011	272238278
1190345324	1006355270	1161592162	1079789655
220609946	1918105148	791775291	979447727
1160648370	779600833	1170336930	1271974642
375813045	1089009771	280197098	1144249742
1236647368	1729816359	650188387	1714906064

100 Glieder der Kongruenzfolge

Unfaire Münzen

Basismodell des Zufalls: „Fairer Münzwurf“, d. h. Münzwurf liefert mit Wahrscheinlichkeit $\frac{1}{2}$ *Kopf* und mit Wahrscheinlichkeit $\frac{1}{2}$ *Zahl*.

Hat man Zugriff auf eine faire Münze, so kann man damit andere Wahrscheinlichkeitsverteilungen erzeugen...
Aber was machen falls Münze unfair, d. h. $Pr[\mathit{Kopf}] \neq Pr[\mathit{Zahl}]$?

Man kann faire Münzen simulieren $\rightsquigarrow$ **Satz von von Neumann**

John von Neumann, 1903–1957

- Mathematiker.
- Bedeutende Beiträge auf den Gebieten der Logik, Funktionalanalysis, Quantenmechanik, Numerik und Spieltheorie.
- von Neumannsches Rechnerarchitekturprinzip noch heute von Bedeutung $\rightsquigarrow$ gehört zu den Vätern der Informatik.

Satz von von Neumann

Theorem

Eine faire Münze kann durch eine deterministische Turingmaschine (det. Algorithmus) mit Zugriff auf eine Folge „unfairer Münzwürfe“ ($Pr[Kopf] = \rho \neq \frac{1}{2}$) in erwarteter Laufzeit $O(\frac{1}{\rho(1-\rho)})$ simuliert werden.

Beweis (Idee): Mache jeweils zwei Münzwürfe mit „unfairer“ Münze (Wurf 1 und 2), bis zwei unterschiedliche Werte (also *Kopf* und *Zahl*) auftauchen. Falls Wurf 1 *Kopf* zeigt, so gib *Kopf* aus, andernfalls gib *Zahl* aus.

Obiger Algorithmus simuliert fairen Münzwurf:
W'keit für (Wurf 1 = *Kopf* & Wurf 2 = *Zahl*) ist $\rho \cdot (1 - \rho)$.
Analog: W'keit für (Wurf 1=*Zahl* & Wurf 2=*Kopf*) ist $(1 - \rho) \cdot \rho$.
Also sind beide Wahrscheinlichkeiten gleich.

Beachte, dass man ρ für obigen Algorithmus noch nicht mal kennen muss....!

Zufall in Internetprotokollen

Protokolle bilden das Rückgrat des Internets zur Versendung und dem Empfang von Datenpaketen.

Modellierung wichtiger Eigenschaften als Zufallsvariablen:

- Paketverlustraten,
- Paketlatenz,
- Transferraten.

Zufallsbasierte Entscheidungen in Algorithmen:

- Aufwachzeiten in drahtlosen Sensornetzwerken.
- Auswahl von Paketen zum „Wegwerfen“ bei Pufferüberlauf in Routern.

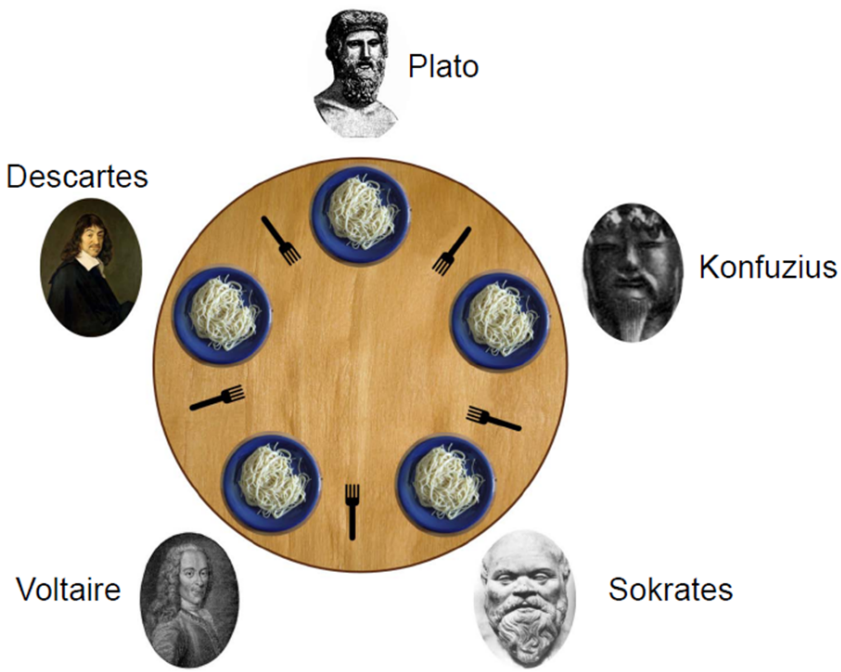
Zufall in verteilten Systemen: Speisende Philosophen

Auflösung von Verklemmungen:

Szenario: n Philosophen im Kreis mit je einer Gabel dazwischen. Um zu essen braucht ein Philosoph beide Gabeln links und rechts von ihm.

Problem: Finde Protokoll welches

- **deadlock-frei** ist: falls jemand hungrig ist, dann isst auch irgendwann jemand.
- **lockout-frei** ist: ein Hungriger bekommt irgendwann zu essen.



Speisende Philosophen: Deterministische Protokolle

Deterministische Protokolle benötigen gemeinsamen Speicher oder Kommunikation mittels Nachrichtenübermittlung.

Beispiel: Nummeriere Philosophen durch und lasse die geraden bzw. ungeraden abwechselnd essen.

Eigenschaften von Protokollen:

- **Voll verteilt**: es gibt keinen zentralen Speicher oder zentralen Prozess auf den alle zugreifen können.
- **Symmetrisch**: alle Prozesse führen das selbe Programm aus, Prozessoren kennen nicht ihre Identität (z. B. Nummer)

Lehmann/Rabin 1981: Kein deterministisches Protokoll kann voll verteilt und symmetrisch sein.

Speisende Philosophen: Randomisiertes Protokoll

Voll verteilt, symmetrisch und deadlockfrei (noch nicht lockout-frei):

Protokoll für jeden Philosophen:
Wiederhole für immer:

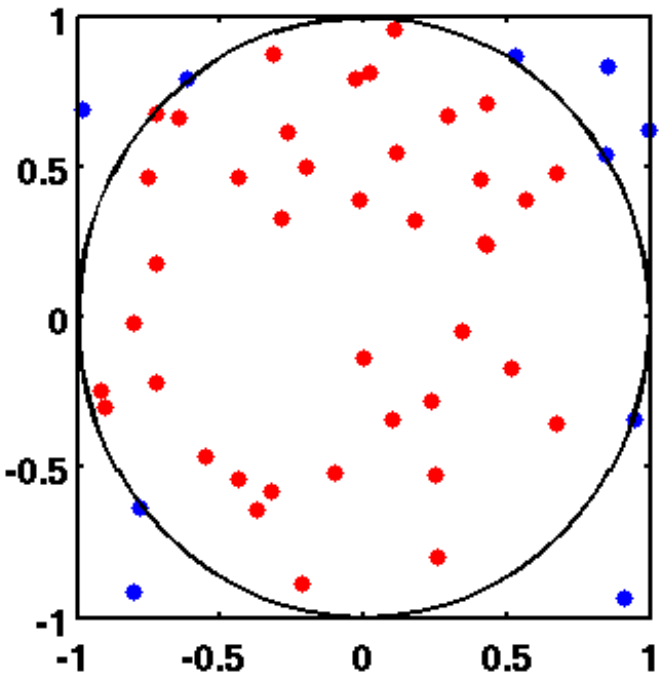
1. Denke bis hungrig...
2. Entscheide per Münzwurf, ob zuerst die linke oder die rechte Gabel genommen wird.
3. Warte, bis gewählte Gabel frei ist und nimm sie dann.
4. Falls andere Gabel nicht frei, so lege Gabel zurück und gehe zu (2). Andernfalls nimm auch die andere Gabel.
5. Kritischer Bereich: Iss!
6. Lege beide Gabeln wieder hin.

Nicht lockout-frei: Gefräßiger Philosoph kann Nachbarn immer die Gabel wegnehmen...

Monte Carlo-Simulation

Wikipedia: Die Monte Carlo-Simulation ist ein Verfahren aus der Stochastik, bei dem sehr häufig durchgeführte Zufallsexperimente die Basis darstellen.

Z. B. Monte Carlo-Verfahren zur Bestimmung der Kreiszahl π:
Wähle **zufällig** Punkte aus dem Intervall $[-1, 1] \times [-1, 1]$.



$$Pr\,[\text{Im Kreis}] = \frac{\text{Kreisfläche}}{\text{Quadratfläche}} = \frac{r^2 \cdot \pi}{(2 \cdot r)^2} = \frac{\pi}{4} = \frac{\text{Treffer in Kreisfläche}}{\text{Anzahl der Punkte}}$$

$\rightsquigarrow$ Gesetz der großen Zahlen:

$$\pi \approx 4 \cdot \frac{\text{Treffer in Kreisfläche}}{\text{Anzahl der Punkte}}$$

Zufall für „Fingerabdrücke“: Matrixmultiplikation I

Verifikation einer Matrixmultiplikation
Eingabe: $n \times n$ Matrizen A, B, C.
Frage: $A \cdot B = C$?

Algorithmus von Freivalds (1977):

1. Wähle zufälligen gleichverteilten Vektor $r \in \{0,1\}^n$
2. $y := A \cdot (B \cdot r)$
3. $z := C \cdot r$
4. **if** $y = z$ **then output** „Ja“ **else output** „Nein“

Beobachtung: Falls $A \cdot B = C$ dann antwortet obiger Algorithmus “Ja”.
Was ist, falls $A \cdot B \neq C$?

Zufall für „Fingerabdrücke“: Matrixmultiplikation II

Mitteilung: Seien A, B, C $n \times n$ Matrizen mit $A \cdot B \neq C$. Dann gilt für eine zufällig per Gleichverteilung gewählte Zahl $r \in \{0,1\}^n$, dass $\Pr[A \cdot B \cdot r = C \cdot r] \leq 1/2$.

Bemerkungen:

- $O(n^2)$ Laufzeit statt $O(n^3)$ durch Ausmultiplizieren gemäß Schulmethode.
- Wiederholt man den Algorithmus k-mal, dann sinkt die Fehlerwahrscheinlichkeit auf $\leq 2^{-k}$.
- Sogenannter **Monte Carlo-Algorithmus** mit „einseitigem Fehler“.

Zufall für „Fingerabdrücke“: Gleichheit von Polynomen

Eingabe: Zwei Polynome $p_1(x_1, x_2, \ldots, x_n)$ und $p_2(x_1, x_2, \ldots, x_n)$.
Frage: Sind die Polynome gleich, d. h. $p_1(x) = p_2(x)$ für alle $x \in \mathbb{R}^n$?
Äquivalente Frage: $p_1(x) - p_2(x) = p(x) = 0$ für alle $x \in \mathbb{R}^n$?

Algorithmus (analog zu Matrixmultiplikation):

1. Für eine endliche Teilmenge $S \subseteq \mathbb{R}$, wähle zufällig & gleichverteilt $x \in S^n$.
2. Falls $p(x) = 0$, so gib „Ja“ aus und andernfalls „Nein“.

Klar: Falls p das Nullpolynom ist, dann gibt der Algorithmus „Ja“ aus.
Was ist, falls p nicht das Nullpolynom ist? Dazu:

Lemma von Schwartz-Zippel: Sei p ein Polynom vom Grad $d \geq 0$ und sei $S \subseteq \mathbb{R}$ eine endliche Menge. Dann gilt für eine zufällige gleichverteilte Zahl $x \in S^n$:

$$Pr[p(x) = 0] \leq \frac{d}{|S|}$$

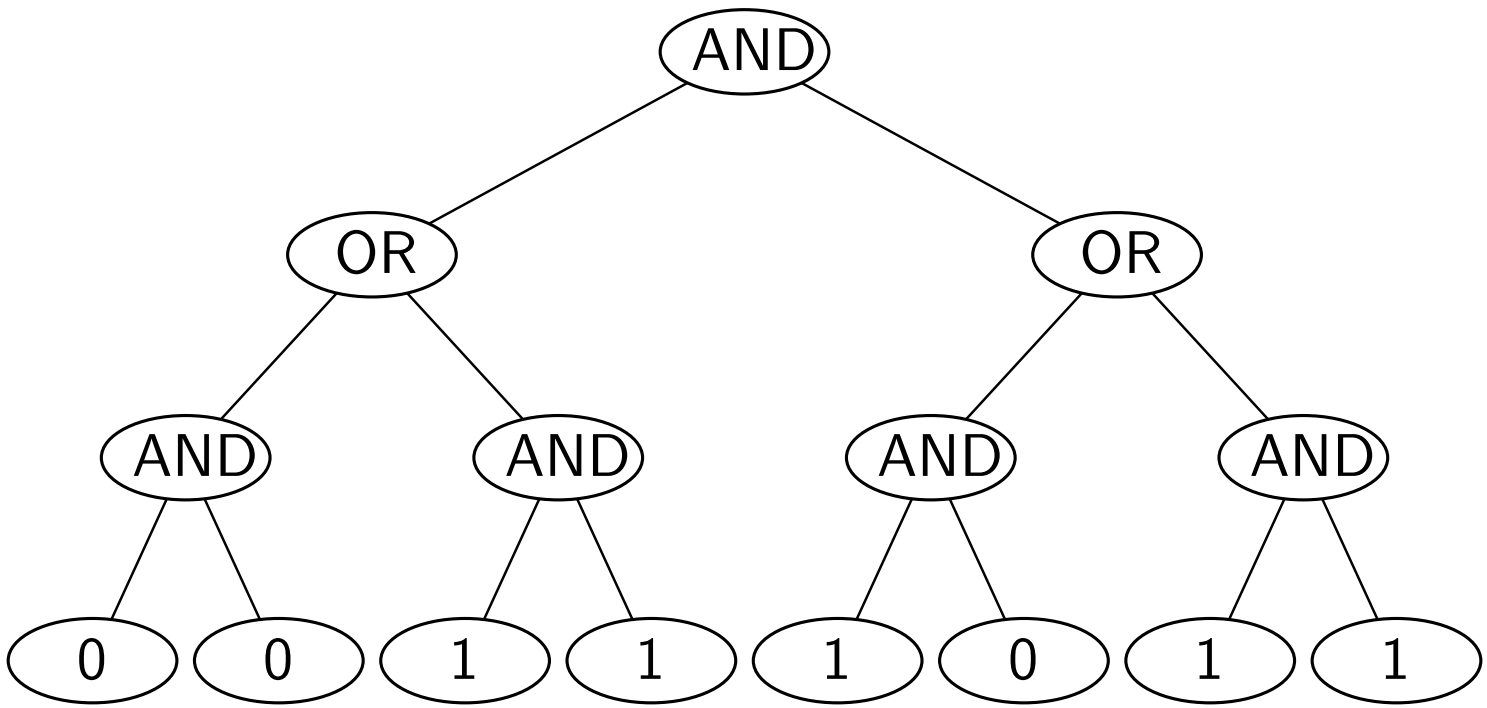
Anwendungsbeispiele: String-Matching (Textsuche), 2D-Bildkompression

Formelauswertung

Eingabe: Logische Formel als in Binärbaum (Spezialfall!) T_k der Höhe $2k$ mit

- Wurzel und interne Knoten auf „geradzahliger Schicht“ sind mit dem logischen AND bezeichnet.
- Alle anderen internen Knoten sind mit dem logischen OR bezeichnet.
- Blätter haben Wert 1 oder 0

Aufgabe: Berechne den Wert der Wurzel.



Randomisierter Algorithmus zur Baumauswertung

Frage: Wieviele Blätter anschauen um Wert der Wurzel zu berechnen?
Klar: Deterministischer Alg. muss im schlimmsten Fall alle $n = 2^{2k-1}$ Blätter anschauen.

Randomisierter Algorithmus zur Auswertung von Formelbäumen:

1 **if** v ist AND **then**
2 Wähle zufällig Kind u von v & evaluiere rekursiv
3 **if** u hat Wert 1 **then** evaluiere anderes Kind von v
4 **if** u hat Wert 0 **then** v hat Wert 0
5 **if** v ist OR **then** ... ▷ analog zu AND, mit 0 und 1 ausgetauscht

Mitteilung: Die erwartete Anzahl von zu evaluierenden Blättern ist 3^k.
Zentrale Idee: Falls Spielbaum Wert 0 hat, dann ist mindestens eines der beiden Kinder der Wurzel 0. Mit W'keit $\frac{1}{2}$ wird dieses Kind in Schritt 2) ausgewählt und deshalb das zweite Kind gar nicht mehr betrachtet.
Mitteilung: 3^k kann nicht verbessert werden.

Zufall für die Wettervorhersage

Mittels **Markov-Modellen** können nicht bzw. schwer vorhersagbare Prozesse (wie z.B. der Wetterverlauf) simuliert / analysiert werden.

А. А. Марков (1886).

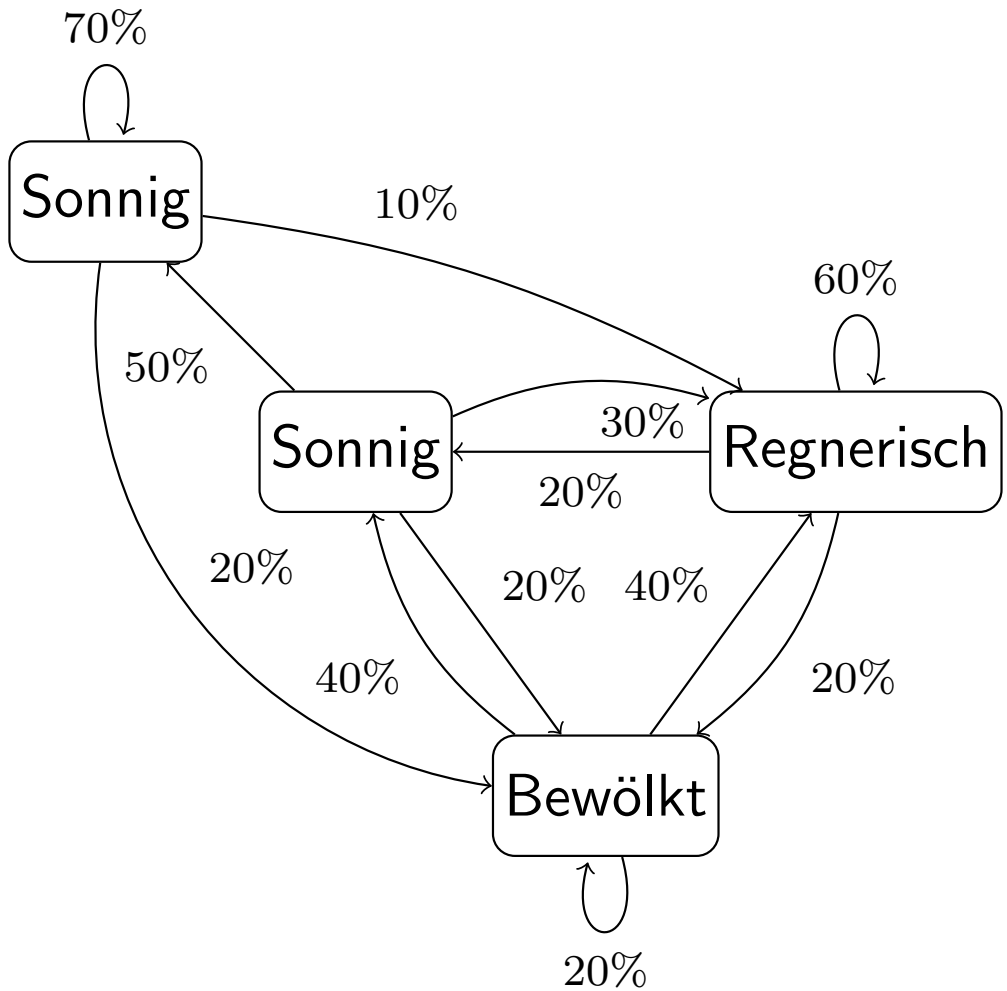
Andrei Andrejewitsch Markov, 1856-1922, Mathematiker

Zufall für die Wettervorhersage – Markov-Modelle

Zustandsgetrieben: Ein Zustand des Modells bildet Zustand in der Welt ab.

Übergangswahrscheinlichkeiten (Erfahrungswerte): Wie wahrscheinlich ist ein Übergang von einem Zustand in einen anderen?

Bsp.: Wahrscheinlichkeit p dass auf einen bewölkten Tag zwei sonnige Tage folgen: $p = 0,4 \cdot 0,5 = 0,2 = 20\%$.



Interaktion und Zufall: Zero Knowledge-Beweise I

Problem: Alice möchte Bob überzeugen ein bestimmtes Wissen oder „Geheimnis“ zu kennen ohne Bob das Geheimnis preiszugeben.

Beispiel: Alice kennt eine Lösungsformel für Polynome 3. Grades

1. Alice bittet Bob ihm ein Polynom 3. Grades zu nennen.
2. Alice löst das Polynom und teilt Bob die Nullstellen mit.
3. Bob überprüft die Nullstellen.

Nennt Alice hinreichend oft die korrekten Nullstellen für verschiedene Polynome, so ist Bob wohl überzeugt dass Alice die Lösungsformel bzw. das „Geheimnis“ kennt...

Bob hat aber während des gesamten Protokolls nichts gelernt
$\rightsquigarrow$ Zero Knowledge!

Hinweis: Beachte Parallele zu NP-vollständigen Problemen: Schritt 2) Beweise finden und Schritt 3) Beweise verifizieren. Schritt 3) ist effizient durchführbar.

Zero Knowledge-Beweise II

Alice möchte Bob überzeugen dass zwei ähnlich aussehende Bilder nicht identisch sind. Wie kann Alice Bob überzeugen, ohne die Unterschiede zu verraten?

1. Alice wendet sich ab.
2. Bob vertauscht mit Wahrscheinlichkeit $1/2$ die Bilder.
3. Alice schaut sich die Bilder an und sagt Bob ob sie vertauscht wurden.

Die Wahrscheinlichkeit, dass Alice im Schritt 3) zufällig die richtige Lösung rät, ist $\frac{1}{2}$. Wiederholt man das Protokoll k-mal und Alice antwortet immer richtig, dann ist die W'keit dafür 2^{-k}.

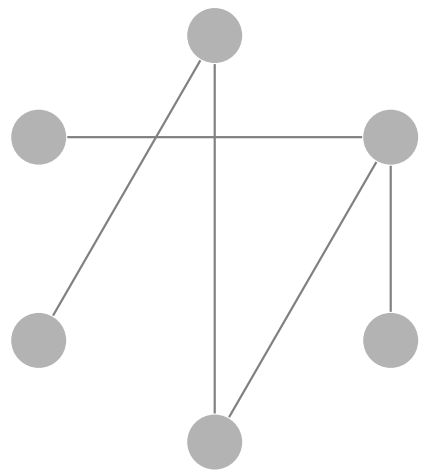
Zufallsgraphen

Zufallsgraphen spielen eine wichtige Rolle in einer Vielzahl von Gebieten.
Frage: Wie erzeugt man zufällige Graphen?

Einfacher Ansatz zum Erzeugen eines Graphen mit n Knoten: Jede (der $\binom{n}{2}$ möglichen) Kanten ist mit einer Wahrscheinlichkeit p vorhanden.
$\rightsquigarrow$ zwei Parameter: n und p.

Beispiel: $n = 6, p = \frac{1}{3}$

Erwartete Anzahl Kanten: $\binom{6}{2} \cdot \frac{1}{3} = 5$

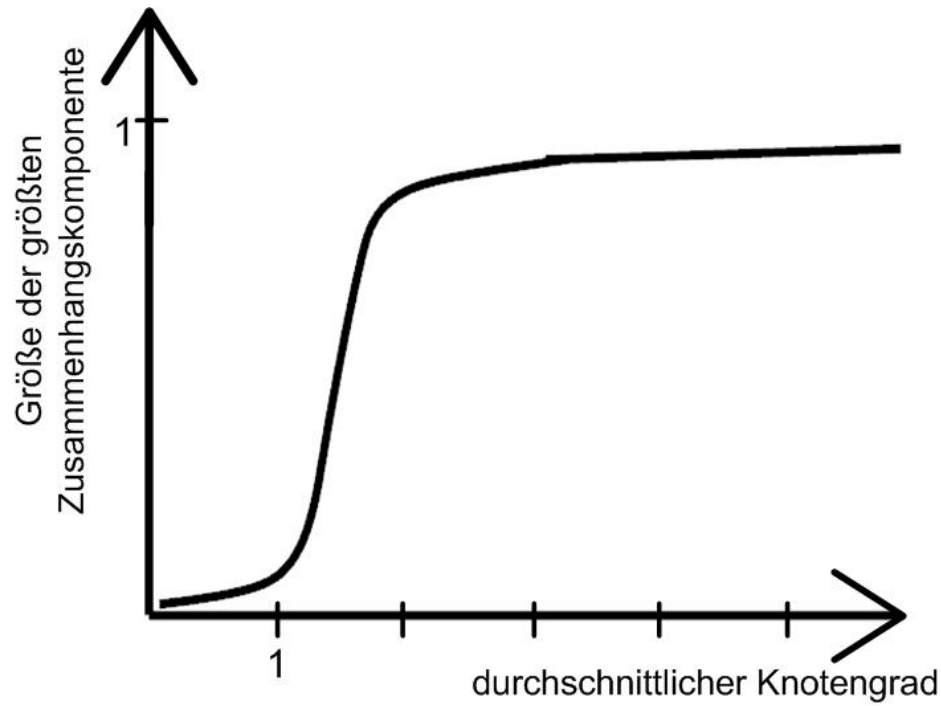


$\rightsquigarrow$ Haben solche zufällig erstellten Graphen bestimmte Eigenschaften?

Zufallsgraphen - Eigenschaften

Haben solche zufällig erstellten Graphen bestimmte Eigenschaften? Ja!

- Knotengradverteilung entspricht einer **Binomialverteilung**, nicht der Gleichverteilung! $\rightsquigarrow$ Wenige Knoten mit hohem bzw. niedrigem Knotengrad.
- Größe der größten Zusammenhangskomponente steigt sprunghaft!



- Viele kleine und eine große Zusammenhangskomponente.

Zitate zum Zufall

„Das, wobei unsere Berechnungen versagen, nennen wir Zufall.“
„Gott würfelt nicht.“
Albert Einstein

„Anyone who considers arithmetical methods of producing random digits is, of course, in a state of sin.“
John von Neumann (1951)

„Die zwei größten Tyrannen der Erde: der Zufall und die Zeit.“
Johann Gottfried Herder

„Das Glück gehört denen, die sich selbst genügen. Denn alle äußeren Quellen des Glücks und Genusses sind ihrer Natur nach höchst unsicher, misslich, vergänglich und dem Zufall unterworfen.“
Arthur Schopenhauer

$\rightsquigarrow$ **Fazit:** Viel Skepsis dem Zufall gegenüber, aber für die Informatik ist der Zufall in erster Linie eine wichtige zu nutzende *Ressource*...