

Algorithmen und Datenstrukturen (ESE)
Entwurf, Analyse und Umsetzung von
Algorithmen (IEMS)
WS 2013 / 2014

Vorlesung 5, Donnerstag, 21. November 2013
(Wie baut man eine Hash Map, Universelles Hashing)

Junior-Prof. Dr. Olaf Ronneberger
Image Analysis Lab
Institut für Informatik
Universität Freiburg

Blick über die Vorlesung heute

■ Organisatorisches

- Tipp für Windows Benutzer: [cygwin](#)

■ Hash Maps

- Eine mögliche Realisierung von einer Map
- Dabei zentral: **universelles Hashing**
- Beispiele für universelle Klassen von Hashfunktionen
- **Übungsblatt:** Mittels eines Programms nachprüfen, dass eine Klasse von Hashfunktionen universell ist
- **Achtung:** dazu kam in den letzten Prüfungen jedes Mal eine Aufgabe, und wenige haben es ganz richtig gemacht !

Tipp für Windows-Benutzer

■ Cygwin

- Download unter www.cygwin.com
- Dann haben Sie in Ihrer normalen DOS shell auch alle bekannten Unix/Linux Befehle
- Insbesondere: `cut`, `head`, `tail`, `less`, `more`, `sort`, `uniq`, ...

Wie baut man eine Map?

■ Zur Erinnerung

- Ein assoziatives Array ist wie ein normales Array, nur dass die Indizes nicht `0, 1, 2, ...` sind, sondern irgendwas, z.B. Telefonnummern

■ Problem

- Schnell ein Element mit einem bestimmten Schlüssel finden
- Naive Lösung: Paare von Schlüsseln und Werten in einem normalen Feld (Java: `ArrayList`, C++: `vector`) speichern
`Array<KeyValuePair>`
- Bei n Schlüsseln kostet die Suche dann bis zu $\Theta(n)$ Zeit
- Mit einer `Hash Map` geht es im günstigsten Fall in Zeit $\Theta(1)$... und zwar egal wieviele Elemente schon in der Map sind!

HashMap — Grundidee

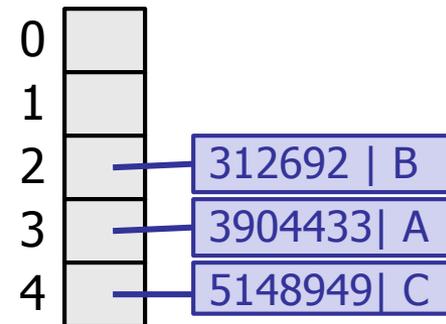
■ Grundidee

- Abbildung der Schlüssel auf die Indizes von einem normalen Feld, mit Hilfe einer sogenannten **Hashfunktion**

■ Ein einfaches Beispiel

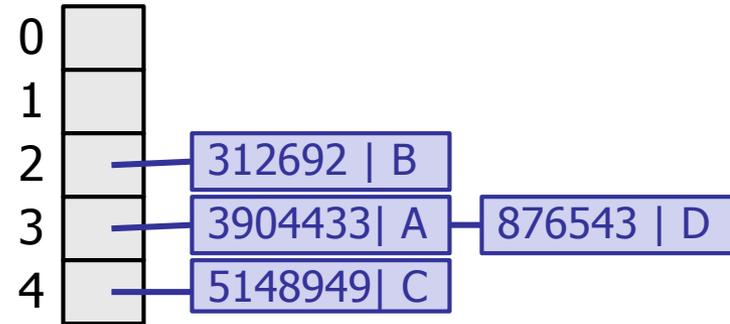
- Schlüsselmenge $\{3904433, 312692, 5148949\}$
- Hashfunktion $h(x) = x \text{ modulo } 5$, also Wertebereich $[0..4]$
- Ein gewöhnliches Feld T der Größe 5 („Hashtabelle“ mit 5 „Buckets“)
- Wir speichern das Element mit Schlüssel x in $T[h(x)]$
 - $\text{insert}(3904433, A)$
 - $h(3904433) \rightarrow 3 ; T[3] = (3904433, A)$
 - $\text{insert}(312692, B)$
 - $h(312692) \rightarrow 2 ; T[2] = (312692, B)$
 - $\text{insert}(5148949, C)$
 - $h(5148949) \rightarrow 4 ; T[4] = (5148949, C)$

Hashtabelle



Hashkollision

Hashtabelle



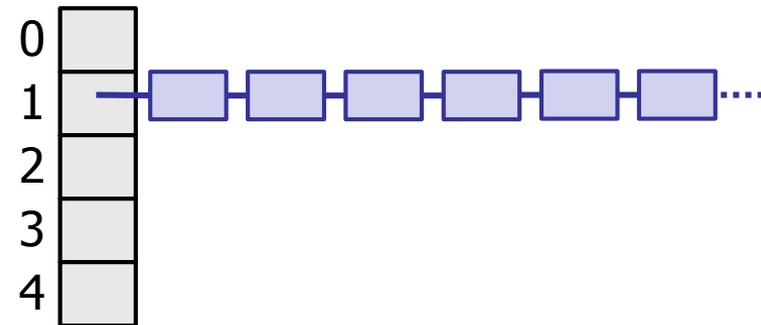
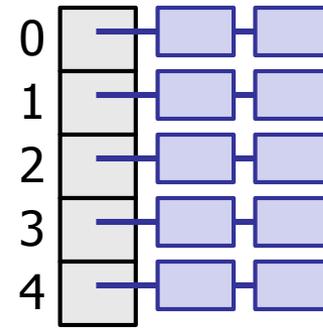
- `search(3904433)`
 - $h(3904433) \rightarrow 3$;
 $T[3] \rightarrow (3904433, A)$
 - `search(123457)`
 - $h(123457) \rightarrow 4$;
 $T[4] \rightarrow 123457$ gibt's nicht
- In unseren Beispiel jetzt Zugriff in $\Theta(1)$ Zeit
- Weitere Elemente einfügen:
- `insert(876543, D)`
 - $h(876543) \rightarrow 3$; **KOLLISION**
- Problem: zwei Schlüssel mit $x \neq y$ aber $h(x) = h(y)$
- Einfachste Lösung: Liste von `KeyValuePair`'s in jedem Bucket, Realisierung z.B. als Array von Arrays
`Array<Array<KeyValuePair>> hashTable;`
- Einfügen dann am Ende der Liste
- $T[3][1] = (876543, D)$

HashMap: Laufzeit

■ Laufzeit für die Schlüsselsuche

- Im **besten** Fall werden die Schlüssel gleichmäßig auf das Feld verteilt
 - Bei n Schlüsseln und einer Hashtabelle der Größe m sind das dann $\approx n/m$ Schlüssel pro Eintrag
- Im **schlechtesten** Fall werden alle Schlüssel auf denselben Eintrag der Hashtabelle abgebildet
 - Dann sind wir wieder bei Zeit $\Theta(n)$

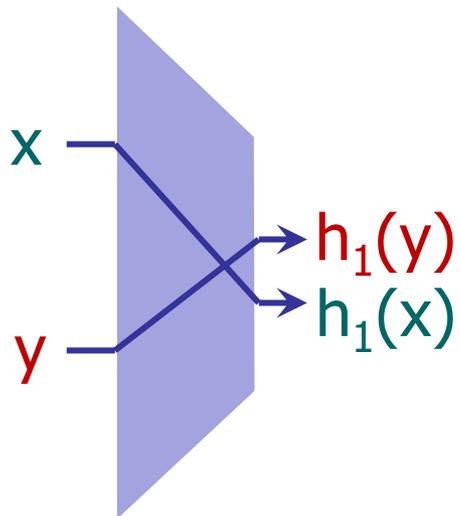
Beispiel: $n=10, m=5$



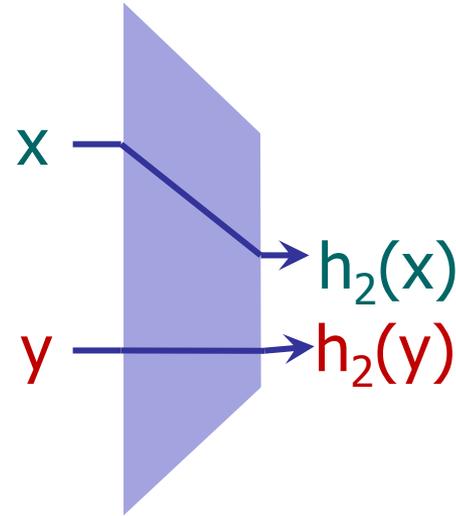
Universelles Hashing: Idee

■ Lösung

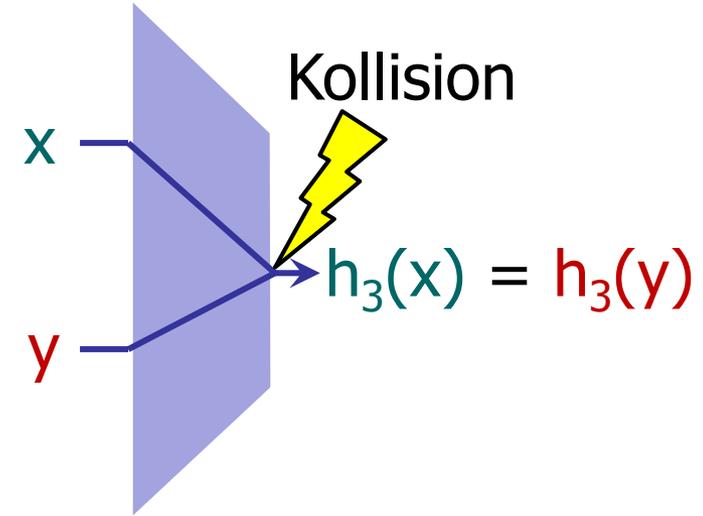
- Wir wählen die Hashfunktion zufällig aus einer geeigneten Menge von Hashfunktionen, so dass die Schlüssel im **Erwartungsfall** gleichmäßig verteilt sind
- Das nennt man **universelles Hashing**



Hashfunktion 1

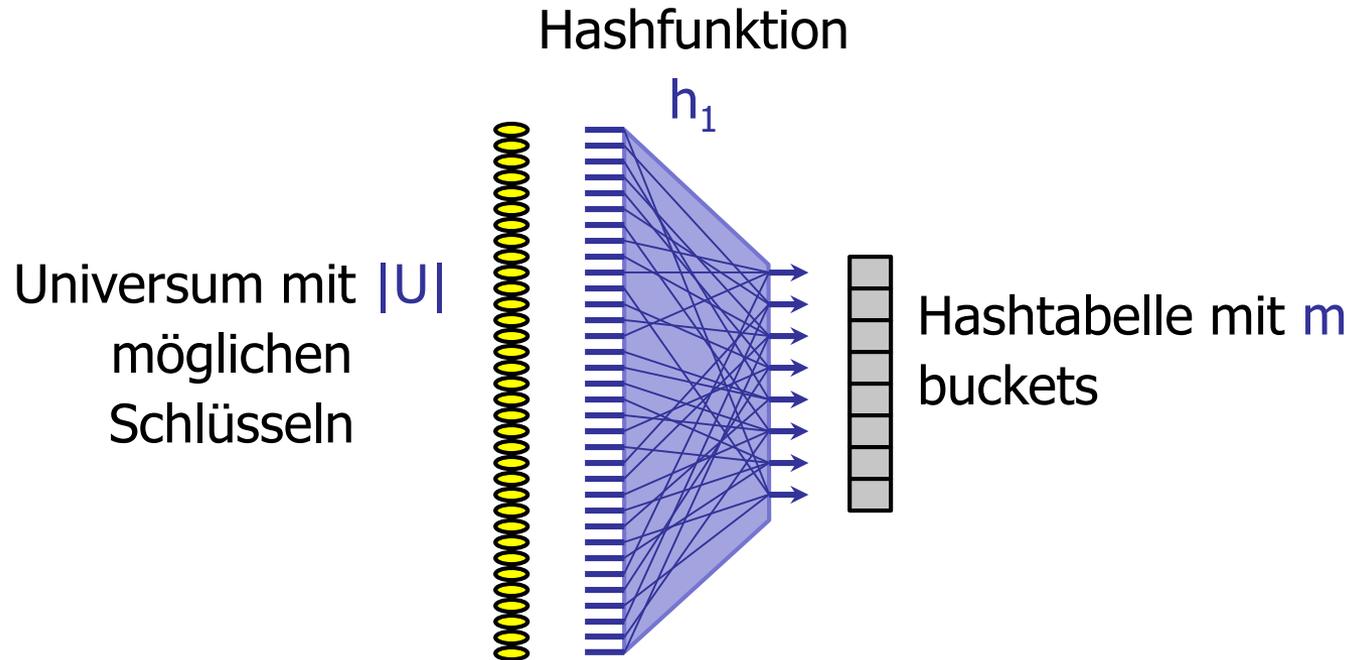


Hashfunktion 2



Hashfunktion 3

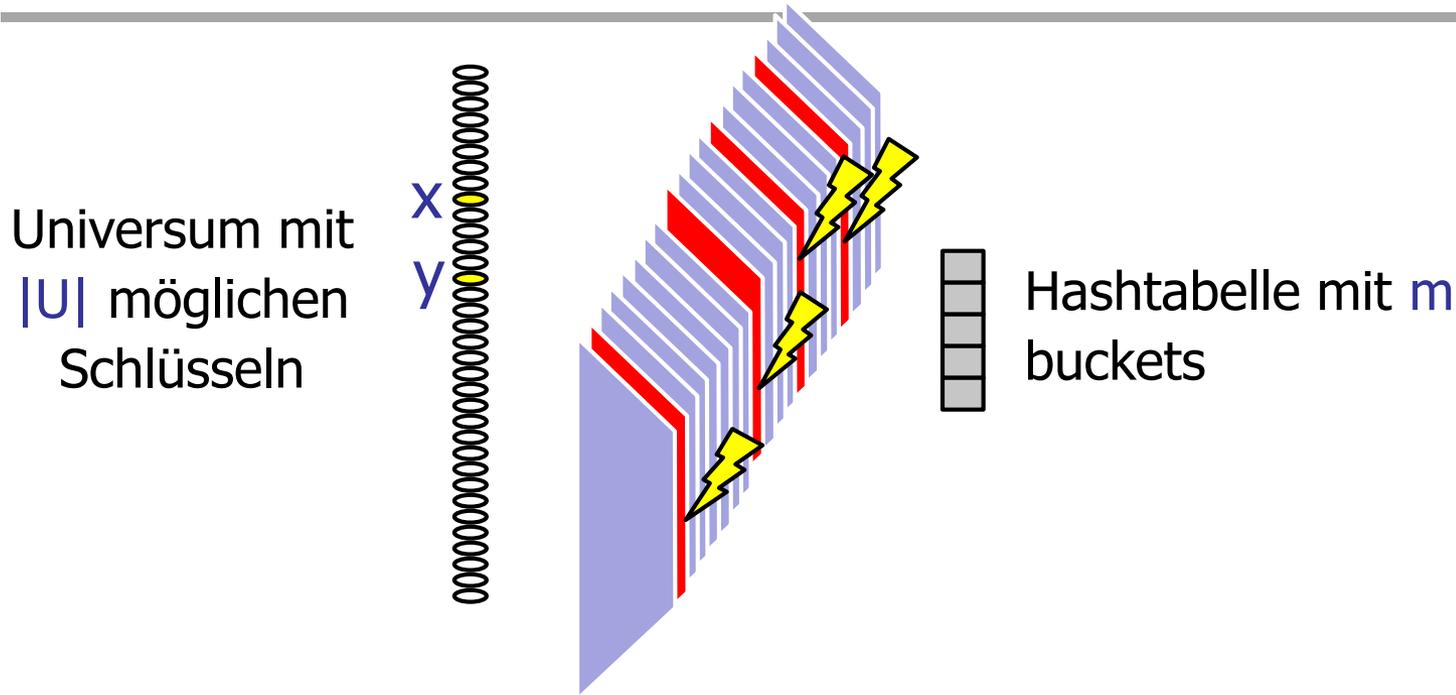
Universelles Hashing: Definition 1/3



■ Definition

- Sei U die Menge der möglichen Schlüssel (Universum)
- Sei m die Größe der Hashtabelle
- Sei $H = \{h_1, h_2, \dots, h_N\}$ eine Menge von Hashfunktionen mit $h_i: U \rightarrow \{0, \dots, m-1\}$

Universelles Hashing: Definition 2/3



H : Menge der Hashfunktionen,

Beispiel: 4 von 20 (=20%) führen zu einer Kollision von x und y

- H ist c -universell wenn für alle $x, y \in U$ mit $x \neq y$ gilt:

Anzahl der Hashfunktionen, die zu einer Kollision führen \rightarrow
$$\frac{|\{h \in H : h(x) = h(y)\}|}{|H|} \leq c \cdot \frac{1}{m}$$

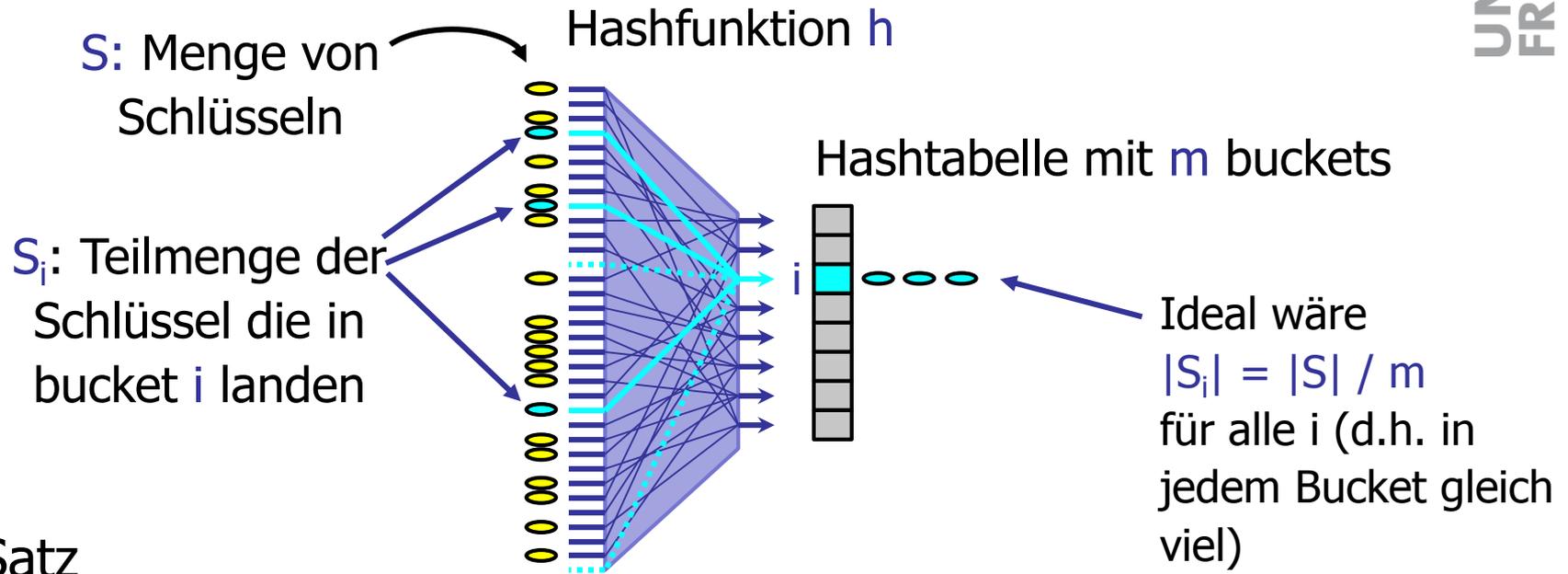
Anzahl aller Hashfunktionen \rightarrow

Universelles Hashing: Definition 3/3

■ Definition:
$$\frac{|\{h \in H : h(x) = h(y)\}|}{|H|} \leq c \cdot \frac{1}{m}$$

- Mit anderen Worten, wenn $h \in H$ zufällig gewählt wird, dann ist $\text{Prob}(h(x) = h(y)) \leq c \cdot 1 / m$
- **Bemerkung:** wenn die Hashfunktion jeden Schlüssel x und y **zufällig** in eine der m buckets der Hashtabelle schmeißt, dann $\text{Prob}(\text{Kollision}) = 1 / m$
- Das wäre dann gerade $c = 1$. Besser geht's nicht

Universelles Hashing: Satz



■ Satz

- Sei H eine c -universelle Klasse von Hashfunktionen
- Sei S eine Menge von Schlüsseln und $h \in H$ zufällig gewählt
- Sei S_i die Menge der Schlüssel x mit $h(x) = i$
- Dann ist der Erwartungswert $E(|S_i|) \leq 1 + c \cdot |S| / m$ für alle i
- Insbesondere: Falls $m = \Omega(|S|)$ gilt $E(|S_i|) = O(1)$

Universelles Hashing: Beweis 1/3

- Für den Beweis brauchen wir Wahrscheinlichkeitsrechnung
- Also zuerst ein kleiner Auffrisch- bzw. Crash- Kurs dazu

■ Wahrscheinlichkeitsraum / Ereignisse

- Wir beschränken uns hier auf den diskreten Fall
- Wahrscheinlichkeitsraum Ω von sog. Elementarereignissen
- Die haben Wahrscheinlichkeiten ... Bedingung $\sum_{e \in \Omega} \Pr(e) = 1$
- Ereignis E = Teilmenge von Ω , Wahrsch. $\Pr(E) = \sum_{e \in E} \Pr(e)$
- Zum Beispiel: zweimal würfeln, dann $\Omega = \{1, \dots, 6\}^2$

e	Pr(e)
1	1/6
2	1/6
3	1/6
4	1/6
5	1/6
6	1/6

e	Pr(e)
(1,1)	1/36
(1,2)	1/36
(1,3)	1/36
...	...
(6,5)	1/36
(6,6)	1/36

- Jedes e aus Ω hat dann Wahrscheinlichkeit $\Pr(e) = 1/36$

- E = beide Augenzahlen sind gerade, dann $\Pr(E) = \frac{9}{36} = \frac{1}{4}$

$(2,2)$ $(4,2)$ $(6,2)$
 $(2,4)$ $(4,4)$ $(6,4)$
 $(2,6)$ $(4,6)$ $(6,6)$

Einschub: Wahrscheinlichkeitsrechnung 2/4

e	Pr(e)	X
(1,1)	1/36	2
(1,2)	1/36	3
(1,3)	1/36	4
...
(2,1)	1/36	3
(2,2)	1/36	4
...
(6,5)	1/36	11
(6,6)	1/36	12



Augenzahl	Pr(X = Augenzahl)
2	1/36
3	2/36
4	3/36
...	...
12	1/36

■ Zufallsvariable

- ... weist einem Ausgang des Zufallsexperiments eine Zahl zu
- Zum Beispiel: $X =$ Summe Augenzahlen bei zweimal Würfeln
- Sowas wie $X = 12$ oder $X \geq 7$ sind dann einfach Ereignisse
- Beispiel 1: $\text{Prob}(X=2) =$
- Beispiel 2: $\text{Prob}(X=4) =$

Einschub: Wahrscheinlichkeitsrechnung 2/4

1 Würfel

Augenzahl	Pr(X=Augenzahl)
1	1/6
2	1/6
3	1/6
4	1/6
5	1/6
6	1/6

2 Würfel

Augenzahl	Pr(X = Augenzahl)
2	1/36
3	2/36
4	3/36
...	...
12	1/36

– **Erwartungswert** ist definiert als $\mathbf{E}(X) = \sum k \cdot \text{Pr}(X = k)$

Intuitiv: gewichtetes Mittel der möglichen Werte von X , wobei die Gewichte die Wahrscheinlichkeiten der entspr. Werte sind

Beispiel: 1x würfeln: $\mathbf{E}(X) = 1 \cdot \frac{1}{6} + 2 \cdot \frac{1}{6} + 3 \cdot \frac{1}{6} + 4 \cdot \frac{1}{6} + 5 \cdot \frac{1}{6} + 6 \cdot \frac{1}{6} = 3.5$

Beispiel: 2x würfeln: $\mathbf{E}(X) = 2 \cdot \frac{1}{36} + 3 \cdot \frac{2}{36} + 4 \cdot \frac{3}{36} + \dots + 12 \cdot \frac{1}{36} = 7$

■ Summe von Erwartungswerten

– Für beliebige (diskrete) Zufallsvariablen X_1, \dots, X_n gilt

$$\mathbf{E}(X_1 + \dots + X_n) = \mathbf{E}(X_1) + \dots + \mathbf{E}(X_n)$$

– Beispiel:

X_1 : Augenzahl von Würfel 1 $\rightarrow \mathbf{E}(X_1) = 3.5$

X_2 : Augenzahl von Würfel 2 $\rightarrow \mathbf{E}(X_2) = 3.5$

$X = X_1 + X_2$: Gesamtaugenzahl $\rightarrow \mathbf{E}(X) = \mathbf{E}(X_1) + \mathbf{E}(X_2) = 3.5 + 3.5 = 7$

Einschub: Wahrscheinlichkeitsrechnung 4/4

- **Korollar:** Bei einem Zufallsexperiment tritt das Ereignis E mit Wahrscheinlichkeit p auf. Sei X die Anzahl der Auftreten von E bei n Ausführungen dieses Experimentes, dann ist $\mathbf{E}(X) = n \cdot p$
- **Beispiel:** $\mathbf{E}(\text{Anzahl Sechser bei 60 mal Würfeln}) = 10$
- **Beweis Korollar:**

$$X_i = \begin{cases} 1, & \text{falls Ereignis eintritt} \\ 0, & \text{sonst} \end{cases}$$

Das nennt man
Indikatorvariable

$$\Rightarrow X = \sum_{i=1}^n X_i$$

$$E(X) = E\left(\sum_{i=1}^n X_i\right) \stackrel{\text{letzte Folie}}{=} \sum_{i=1}^n E(X_i) \stackrel{\text{Def. E-Wert}}{=} \sum_{i=1}^n p = n \cdot p \quad \square$$

$$\begin{aligned} E(X_i) &= 0 \cdot \Pr(X_i = 0) + 1 \cdot \Pr(X_i = 1) \\ &= \Pr(X_i = 1) \\ &= p \end{aligned}$$

Universelles Hashing: Beweis 2/3

- **Gegeben:** Für $x, y \in U$ mit $x \neq y$ und ein zufälliges $h \in H$ gilt: $\text{Prob}(h(x) = h(y)) \leq c \cdot 1/m$
- **Zu beweisen:** $E(|S_i|) \leq 1 + c \cdot |S| / m$ für alle i

$$S_i = \{x \in S : h(x) = i\}$$

$$\text{falls } S_i = \emptyset \Rightarrow |S_i| = 0$$

ansonsten, sei $x \in S_i$ irgendein Schlüssel

Definiere Indikatorfunktion $I_y = \begin{cases} 1, & \text{falls } h(y) = i \\ 0, & \text{sonst} \end{cases}$ für alle $y \in S \setminus \{x\}$

$$\Rightarrow |S_i| = 1 + \sum_{y \in S \setminus \{x\}} I_y$$

$$\Rightarrow E(|S_i|) = E\left(1 + \sum_{y \in S \setminus \{x\}} I_y\right) = 1 + \sum_{y \in S \setminus \{x\}} E(I_y)$$

Universelles Hashing: Beweis 3/3

Nebenrechnung:

$$\begin{aligned} E(I_y) &= \Pr(I_y = 1) \\ &= \Pr(h(y) = i) \\ &= \Pr(h(y) = h(x)) \\ &\leq c \cdot \frac{1}{m} \end{aligned}$$

$$\begin{aligned} \Rightarrow 1 + \sum_{y \in S \setminus \{x\}} E(I_y) &\leq 1 + \sum_{y \in S \setminus \{x\}} c \cdot \frac{1}{m} \\ &= 1 + (|S| - 1) \cdot c \cdot \frac{1}{m} \\ &\leq 1 + |S| \cdot c \cdot \frac{1}{m} = 1 + c \cdot \frac{|S|}{m} \end{aligned}$$

$$E(|S_i|) \leq 1 + c \cdot \frac{|S|}{m} \quad \square$$

Universelles Hashing: Beispiele 1/3

■ Negativbeispiel

- Die Menge aller h mit $h_a(x) = (a \cdot x) \bmod m$, für ein $a \in U$
- Ist nicht c -universell, warum?
- Falls universell: $\forall x, y \ x \neq y : \frac{|\{h \in H : h(x) = h(y)\}|}{|H|} \leq c \cdot \frac{1}{m}$
- Für welches x, y ist der Anteil der Hashfunktionen, die zu einer Kollision führen, größer als c/m ?

$$x = m \quad y = 2m$$

Universelles Hashing: Beispiele 2/3

■ Positivbeispiel 1

- Sei p eine große Primzahl, und zwar $p > m$ und $p \geq |U|$
- Sei H die Menge aller h mit $h_{a,b}(x) = ((a \cdot x + b) \bmod p) \bmod m$
wobei $1 \leq a < p$, und $0 \leq b < p$
- Die ist ≈ 1 -universell, siehe [Exercise 4.11](#) in Mehlhorn/Sanders
- z.B.: $U = \{0, \dots, 99\}$, $p = 101$, $a = 47$, $b = 5$
- Dann ist $h(x) = ((47x + 5) \bmod 101) \bmod m$
- Sehr einfach zu implementieren, etwas kompliziert zu beweisen,
deswegen Übungsaufgabe:
- Empirisch zeigen, dass sie 2-universell ist.

Universelles Hashing: Beispiele 3/3

■ Positivbeispiel 2

- Die Menge aller h mit $h(x) = (a \bullet x) \bmod m$, für ein $a \in U$
 - Schreibe $a = \sum_{i=0..k-1} a_i \cdot m^i$, wobei $k = \text{ceil}(\log_m |U|)$
 - Entsprechend $x = \sum_{i=0..k-1} x_i \cdot m^i$
 - Dann $a \bullet x := \sum_{i=0..k-1} a_i \cdot x_i$
 - Intuitiv: das "Skalarprodukt" der Darstellung zur Basis m
- Die ist **1**-universell, siehe **Theorem 4.4** in Mehlhorn/Sanders
- Beispiel: $U = \{0, \dots, 999\}$, $m=10$, $a=348$, und $x=127$
- Dann ist $a_1=3$, $a_2=4$, $a_3=8$, und $x_1=1$, $x_2=2$, $x_3=7$
- $(a \bullet x) \bmod 10 = (3 \cdot 1 + 4 \cdot 2 + 8 \cdot 7) \bmod 10 = 7$

Literatur / Links

■ Hash Maps

- In Mehlhorn/Sanders:

4 Hash Tables and Associative Arrays

- In Cormen/Leiserson/Rivest

12 Hash Tables

- In Wikipedia

<http://de.wikipedia.org/wiki/Hashtabelle>

http://en.wikipedia.org/wiki/Hash_table

■ Hash Map in Java und in C++

<http://download.oracle.com/javase/1.4.2/docs/api/java/util/HashMap.html>

http://www.sgi.com/tech/stl/hash_map.html

