# Fundamente der Informatik

Ablaufmodellierung, Algorithmen und Datenstrukturen

Von
Peter Hubwieser
und
Gerd Aiglstorfer

Errata

Oldenbourg Verlag München Wien

Hashfunktion $h \in \boldsymbol{H}$ können nach wie vor viele Kollisionen auftreten. Im Durchschnitt aller $h \in \boldsymbol{H}$ geschieht dies jedoch nicht mehr so häufig.

**Beispiel:** Die Funktionen in Abbildung 12.3 seien eine Klasse $\boldsymbol{H}$ von Hashfunktionen. Wir benutzen diese zum Einfügen von vier verschiedenen, beliebig gewählten Schlüsseln $k_1$, $k_2$, $k_3$ und $k_4$ der Schlüsselmenge $\boldsymbol{K}$ in eine Hashtabelle und nehmen an, dass die $h_i$ mit $i = 2, 4, 6, 8$ für die vier Schlüssel Adresskollisionen verursachen. Die Hashfunktionen mit ungeradem $i$ produzieren dagegen keine Kollisionen. Zum Einfügen der Schlüssel wählen wir zuerst eine der acht Funktionen (zufällig und gleich verteilt) und ziehen mit Wahrscheinlichkeit $4/8 = 1/2$ ein $h_i$ mit geradem $i$. Somit verursacht das Hashverfahren unter diesen Voraussetzungen mit Wahrscheinlichkeit $1/2$ Adresskollisionen und mit Wahrscheinlichkeit $1 - 4/8 = 1/2$ keine Adresskollisionen. Dies gilt selbst dann, wenn die Schlüssel $k_1$ bis $k_4$ nicht gleich verteilt sind.

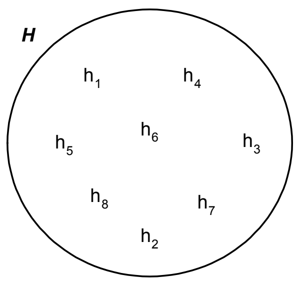


*Abb. 12.3 Eine Klasse von Hashfunktionen*

**Definition:** (*universelles Hashing*) (vgl. Ottmann/Widmayer, 2002, Seite 187) Sei $\boldsymbol{H}$ eine endliche Kollektion von Hashfunktionen, so dass jede Funktion aus $\boldsymbol{H}$ jeden Schlüssel aus $\boldsymbol{K}$ auf eine Hashadresse aus $\{0, \ldots, t-1\}$ abbildet. $\boldsymbol{H}$ heißt *universell*, wenn für je zwei verschiedene Schlüssel $k_1, k_2 \in \boldsymbol{K}$ gilt:

$$\frac{|\{h \in \mathbf{H} : h(k_1) = h(k_2)\}|}{|\mathbf{H}|} \leq \frac{1}{t}.$$

Wir formulieren etwas anders: $\boldsymbol{H}$ ist universell, wenn für jedes Paar von zwei nicht identischen Schlüsseln höchstens der $t$-te Teil aller Funktionen aus $\boldsymbol{H}$ die gleiche Hashadresse berechnen.

Die Wahrscheinlichkeit, dass die Hashadressen für zwei fest gewählte, verschiedene Schlüssel $k_1$ und $k_2$ eines universellen Hashverfahrens kollidieren, beträgt maximal $1/t$.

**Beispiel:** Die Schlüssel 14 und 21 fügen wir in eine Hashtabelle der Länge $t = 7$ ein. Verwenden wir die Hashfunktion $k \bmod 7$, so erhalten wir mit Wahrscheinlichkeit 1 eine Adresskollision. Sei nun $k \bmod 7$ Teil einer universellen Klasse $\boldsymbol{H}$ von Hashfunktionen. Die

Wahrscheinlichkeit einer Kollision der beiden Schlüssel beträgt dann noch höchstens *1/t*. Mit der Wahrscheinlichkeit von mindestens *1 – 1/t* wird eine Hashfunktion gezogen, die die beiden Schlüssel auf unterschiedliche Hashadressen abbildet.

**Beispiel:** (siehe Ottmann/Widmayer, 2002, Seite 189) Sei $|\boldsymbol{K}| = p$ eine Primzahl und $\boldsymbol{K} = \{0, ..., p-1\}$. Für zwei beliebige Zahlen $i \in \{1, ..., p-1\}$ und $j \in \{0, ..., p-1\}$ sei die Funktion $h_{i,j} : \boldsymbol{K} \rightarrow \{0, ..., t-1\}$ wie folgt definiert:

$$h_{i,j}(x) = ((ix + j) \bmod p) \bmod t.$$

Die Klasse $\boldsymbol{H} = \{h_{i,j} \mid 1 \leq i < p \wedge 0 \leq j < p\}$ bildet eine universelle Klasse von Hashfunktionen. Die Autoren obiger Literatur geben dazu auch einen Beweis an.

Wir fassen abschließend zusammen: Universelles Hashing verteilt homogen gewählte Schlüssel so gut wie möglich auf den Adressbereich der Hashtabelle. Der von uns behandelte Fall ist ein Spezialfall des universellen Hashings, es heißt genau genommen *1-universell*. Die Verallgemeinerung wird als *c-universell* ($c \in \boldsymbol{R}$) bezeichnet (Näheres dazu in Mehlhorn, 1988):

$$\frac{|\{h \in \mathbf{H} : h(k_1) = h(k_2)\}|}{|\mathbf{H}|} \leq \frac{c}{t}.$$

Wir haben nun einen Einblick in die Wahl von Hashfunktionen und können uns der Auflösung von Adresskollisionen zuwenden.

## 12.5 Chainingverfahren

Fügen wir in die Hashtabelle *htable* einen Schlüssel *k'* an einer Hashadresse ein, an der sich bereits das Synonym *k* befindet, so ist *k'* ein *Überläufer* und eine Adresskollision liegt vor. Eine Möglichkeit zum Auflösen der Kollision ist die so genannte *Verkettung* der Überläufer (engl. *chaining*).

Beim Verketten werden die Überläufer in dynamisch anpassbaren Datenstrukturen abgelegt, die in geeigneter Weise mit der Hashtabelle verbunden sind. Zu jeder Hashadresse existiert beispielsweise eine lineare Liste, in welche Überläufer eingefügt werden. Die Liste wird mit Hilfe von Zeigern mit der Hashtabelle verknüpft (siehe Abbildung 12.4).