UNIVERSITÄT DORTMUND
FACHBEREICH INFORMATIK

Klassen universeller Hashfunktionen mit ganzzahliger Arithmetik

Philipp Wölfel

Philipp Wölfel
Schlickumer Weg 36
D-40699 Erkrath
woelfel@Ls2.cs.uni-dortmund.de

10. November 2000

Inhaltsverzeichnis

KAPITEL 1

Einleitung

Universelles Hashing hat seit der Einführung von Carter und Wegman (1979) eine grundlegende Bedeutung in weiten Teilen der Informatik erlangt. Zu den Anwendungen gehören neben Algorithmen für Standardprobleme wie z.B. Wörterbücher oder das „Closest-Pair-Problem“ auch Lösungsansätze für wichtige Fragen der Kryptographie oder der Komplexitätstheorie.

Die grundlegende Idee des universellen Hashings soll zunächst anhand einer der wichtigsten Anwendungen, dem sogenannten Wörterbuchproblem, erläutert werden.

§ 1. Motivation

Beinahe jedes Computerprogramm, das Informationen verarbeitet, muß sich mit einem zentralen Problem der Informatik auseinandersetzen: Wie können die benötigten Daten so gespeichert werden, daß jederzeit ein effizienter Zugriff darauf möglich ist? Ein einfaches Beispiel stellt die Implementation eines Deutsch-Englischen Wörterbuchs dar. Es sollen zu einer Vielzahl von deutschen Begriffen die englischen Übersetzungen verfügbar sein. Dabei ist es wichtig, daß auf Anfrage nach einem deutschen Stichwort die Übersetzung möglichst schnell gefunden wird.

Für Fragestellungen, die sich mit dem Abspeichern und Wiederauffinden von Schlüsselwörtern beschäftigen, benutzt man den Begriff *Wörterbuchproblem*. Formal läßt sich ein Wörterbuchproblem wie folgt charakterisieren: Gegeben ist eine Menge U - das sogenannte *Universum* -, welche die gültigen Eingaben für das Wörterbuch enthält. Im obigen Beispiel könnte das Universum aus allen möglichen Buchstabenkombinationen bestehen. Die Elemente aus U heißen *Schlüssel*. Man unterscheidet dann zwischen dem *statischen* und dem *dynamischen* Wörterbuch. Beim statischen Wörterbuch können die enthaltenen Daten nach dessen Erzeugung nicht mehr verändert werden. Es gibt daher nur eine Operation, die Einfluß auf die gespeicherten Schlüssel hat: `MakeDictionary(S)` erzeugt für eine Menge $S \subseteq U$ von Schlüsseln ein Wörterbuch. Außerdem gibt es eine Operation `Find(`x`)`, die angibt, ob ein Schlüssel $x \in U$ im Wörterbuch verzeichnet ist, und - falls ja - die dazu gespeicherten Daten findet (z.B. die englische Übersetzung des Stichwortes x). Häufig ist es wünschenswert, daß das Wörterbuch nicht nur einmal für eine feste Schlüsselmenge erzeugt wird, sondern auch später verändert werden kann. Bei einem solchen dynamischen Wörterbuch findet man dann weitere Operationen wie `Delete(x)` oder `Insert(x)` zum Löschen und Einfügen von Schlüsseln $x \in U$. Je nach Anwendung können auch andere Operationen notwendig sein, wie z.B. das Verschmelzen zweier Wörterbücher.

Hashing

Das Wörterbuch stellt - neben der konkreten Anwendung z.B. eines Deutsch-Englischen Wörterbuchs - eine für die Informatik fundamentale Datenstruktur dar. Zahlreiche Algorithmen zur Implementation von Wörterbüchern sind bekannt, und werden in den gängigen Lehrbüchern beschrieben (s. z.B. Cormen, Leiserson und Rivest 1990, S. 196ff.; Motwani und Raghavan 1995, S. 197ff.; Aho, Hopcroft und Ullman 1987, S. 117ff.). *Hashing* ist eine der wichtigsten und schon seit vielen Jahren etablierten Methoden. Die ursprüngliche Idee wird in einem internen IBM-Memorandum von H. P. Luhn aus dem Jahre 1953 beschrieben (vgl. Knuth, 1998, S. 547). Üblicherweise geht man beim Hashing von einem endlichen Universum U aus. Für das Deutsch-Englische Wörterbuch heißt das beispielsweise, daß die Stichwörter nur aus Begriffen mit einer begrenzten Anzahl von Buchstaben bestehen dürfen. Um eine Menge $S \subseteq U$ von Schlüsseln in dem Wörterbuch zu speichern, wird eine *Hashtabelle* T mit r Einträgen angelegt (r steht dabei für „range“, engl. Wertebereich). Man benutzt eine *Hashfunktion* $h : U \to \{1, \ldots, r\}$, um die Elemente aus U auf die Tabellenplätze abzubilden. Dann wird jeder Schlüssel $x \in S$ in der Tabelle an der Position $h(x)$ gespeichert. Der Funktionswert $h(x)$ wird auch als *Hashwert* des Schlüssels bezeichnet.

Da die Größe der Hashtabelle durch den verfügbaren Speicherplatz beschränkt ist, enthält das Universum üblicherweise eine erheblich größere Zahl von Elementen, als Tabelleneinträge zur Verfügung stehen ($|U| \gg r$). Dies kann dazu führen, daß mehrere Schlüssel, die in der Tabelle abgespeichert werden sollen, den gleichen Hashwert haben; man spricht dann von einer *Kollision* dieser Schlüssel.

Im Laufe der Jahre wurden zahlreiche Strategien entwickelt, um mit Kollisionen umzugehen (eine ausführliche Diskussion und Analyse bekannter Methoden findet sich z.B. bei Knuth, 1998). Beim *Hashing mit Verkettung* beispielsweise werden in der Tabelle nicht mehr die Schlüssel selbst, sondern linear verkettete Listen gespeichert (vgl. z.B. Mehlhorn, 1988, S. 112ff.). Um einen Schlüssel x zu speichern, wird er an diejenige Liste angehängt, die sich an der Position $h(x)$ der Tabelle T befindet. D.h. bei einem Wörterbuch mit Schlüsselmenge S enthält die i-te Liste genau diejenigen Elemente aus S, die den Hashwert i haben. Die benötigte Zeit, um ein gespeichertes Element zu finden, ist dann nicht mehr allein durch die Auswertung der Hashfunktion bestimmt, sondern im Wesentlichen durch die Länge der Liste, die durchsucht werden muß.

Es zeigt sich, daß diese Methode bei einer geeigneten Wahl von h zu guten Ergebnissen führt, wenn die Auswahl der zu speichernden Schlüssel einem gewissen Zufall unterliegt. Sei z.B. die Schlüsselmenge S insofern zufällig ausgewählt, daß alle Schlüssel unter h jeden Hashwert unabhängig voneinander mit gleicher Wahrscheinlichkeit annehmen. Man kann leicht beweisen, daß die erwartete Suchzeit für ein beliebiges Element $x \in U$ durch $O(1 + |S|/r)$ beschränkt ist (vgl. Cormen, Leiserson und Rivest, 1990, S. 221ff.). Wählt man also die Tabelle groß genug, d.h. mit mindestens $|S|$ Einträgen, so kann man ein beliebiges Element im Mittel in konstanter Zeit auffinden.

Auch wenn das obige Ergebnis auf den ersten Blick zufriedenstellend erscheint, darf man nicht vergessen, daß es auf einer kritischen Annahme beruht. Normalerweise wird man nämlich keine zufällige Daten in einem Wörterbuch speichern. Deswegen muß man auch den Worst-Case berücksichtigen, also den Fall, in dem die Eingabe so gewählt ist, daß sie die größtmögliche Laufzeit erzeugt. Tatsächlich ist das Worst-Case-Verhalten der meisten Methoden, die auf festen Hashfunktionen beruhen, sehr schlecht. Beim Hashing mit Verkettung z.B. könnten bei einer ungünstig gewählten Schlüsselmenge S alle Schlüssel den gleichen Hashwert besitzen. Die Suche nach einem Element entspricht dann der Suche in einer verketteten Liste mit $|S|$ Elementen und benötigt im schlimmsten Fall lineare Laufzeit.

Randomisiertes Hashing

Damit man sich bei der Analyse des durchschnittlichen Laufzeitverhaltens nicht mehr auf die Zufälligkeit der Eingabe verlassen muß, versucht man, randomisierte Algorithmen zu entwerfen. Das sind Algorithmen, bei denen neben den deterministischen auch zufällige Schritte erlaubt sind (formal werden randomisierte Algorithmen in Rabin, 1963 oder Gill, 1977 eingeführt). Im Falle des Hashings ist die Idee naheliegend, die Hashfunktionen während der Laufzeit zufällig auszuwählen, anstatt sie fest vorzugeben.

Sei U wieder das Universum und r die Größe einer Hashtabelle. Es ist sicher nicht sinnvoll, die Hashfunktion $h : U \to \{1,\ldots,r\}$ komplett zufällig zu generieren, da man dafür $|U|$ Zufallszahlen aus dem Wertebereich $\{1,\ldots,r\}$ erzeugen und abspeichern müßte. Der dafür benötigte Speicherplatzbedarf läge dann - unter der Annahme, daß $|U|$ viel größer als r ist - weit über dem, was die Schlüssel selbst benötigen. Man benutzt daher Klassen (oder Familien) von Hashfunktionen, aus denen dann eine Abbildung zufällig ausgewählt wird. Für zwei endliche Mengen U und R bezeichnen wir eine Klasse, die aus Hashfunktionen $U \to R$ besteht, auch als *Hashklasse* mit *Universum* U und *Wertebereich* R oder kurz als *Hashklasse* $U \to R$. Um deutlich zu machen, daß eine Hashklasse $\mathcal{H}$ das Universum U und den Wertebereich R hat, schreiben wir auch einfach $\mathcal{H} : U \to R$.

Wir betrachten wieder ein Wörterbuch, das auf Hashing mit Verkettung beruht. Jetzt wird aber die Hashfunktion h zufällig aus einer Hashklasse $\mathcal{H}$ mit Universum U und Wertebereich $\{1,\ldots,r\}$ ausgewählt. Sei S eine beliebige Menge von Schlüsseln, h zufällig gemäß der Gleichverteilung aus $\mathcal{H}$ gewählt und jeder Schlüssel $x \in S$ in der $h(x)$-ten Liste der Tabelle T gespeichert. Wieviel Zeit benötigt dann eine Operation `Find(`x`)` für ein beliebiges $x \in U$?

Sei $h(x) = i$. Unter der Annahme, daß die Hashfunktion in konstanter Zeit ausgewertet werden kann, ist wieder der Zeitbedarf im Wesentlichen durch die Länge der i-ten Liste bestimmt. Die Menge derjenigen Elemente aus S, die einen Hashwert k haben, bezeichnen wir als den Korb (engl. „bin“) $B_k(S)$ und ihre Kardinalität mit $b_k(S)$. Also ist $b_k(S)$ gleich der Länge der k-ten Liste.

Seien $x_1,\ldots,x_n$ die Schlüssel aus $S\backslash\{x\}$. Für jedes x_j $(1 \le j \le n)$ definieren wir eine Zufallsvariable $Y_j \in \{0,1\}$, die genau dann den Wert 1 annimmt, wenn x_j und x kollidieren, also wenn $h(x_j) = h(x)$ ist. Somit ist $b_i(S) \le 1 + Y_1 + \cdots + Y_n$, und es folgt mit der Linearität des Erwartungswertes

$$\mathbf{E}\big[b_i(S)\big] \;\le\; 1 + \mathbf{E}\left[\sum_{j=1}^{n} Y_j\right] \;=\; 1 + \sum_{j=1}^{n} \mathbf{E}\left[Y_j\right] \;=\; 1 + \sum_{j=1}^{n} \mathbf{Prob}\left(h(x_j) = h(x)\right). \qquad (1.1)$$

Um eine möglichst geringe erwartetet Laufzeit für die `Find(x)`-Operation zu erreichen, muß also für jedes x_j die Wahrscheinlichkeit einer Kollision mit x minimiert werden. Da wir weder Annahmen über S treffen wollen, noch wissen, nach welchem $x \in U$ gesucht wird, sollte also für alle verschiedene Schlüssel $x_1, x_2 \in U$ die *Kollisionswahrscheinlichkeit* $\mathbf{Prob}\left(h(x_1) = h(x_2)\right)$ möglichst klein sein. Genau diese Idee wurde bei der Definition universeller Hashklassen von Carter und Wegman (1979) verfolgt.

1.1.1 Definition. Eine Hashklasse $\mathcal{H} : U \to R$ heißt *universell*, wenn für beliebige verschiedene Schlüssel $x_1, x_2 \in U$ bei zufälliger Wahl von h aus $\mathcal{H}$

$$\textbf{Prob}\left(h(x_1) = h(x_2)\right) \;\leq\; \frac{1}{|R|}$$

gilt.

Ist $\mathcal{H}$ eine universelle Hashklasse $U \to \{1,\ldots,r\}$, so ergibt sich aus Gleichung (1.1)

$$\mathbf{E}\left[b_i(S)\right] \;\leq\; 1 + \sum_{j=1}^{n} \frac{1}{r} \;=\; 1 + \frac{|S|-1}{r}.$$

Für dynamische Wörterbücher, die `Delete`-, `Insert`- und `Find`-Operationen unterstützen, erhält man also folgendes Ergebnis (vgl. Carter und Wegman, 1979):

1.1.2 Satz. *Sei eine beliebige Sequenz von k Operationen, darunter höchstens s `Insert`-Operationen gegeben, und h zufällig aus einer universellen Hashklasse $U \to \{1,\ldots,r\}$ gewählt. Kann h in konstanter Zeit ausgewertet werden, so ist die erwartete Laufzeit für die Ausführung der k Operationen durch $O\big(k(1+s/r)\big)$ beschränkt.* ∎

Wählt man also r groß genug, d.h. in der Größenordnung von s, dann benötigt so eine Sequenz von Operationen im Mittel höchstens lineare Zeit.

Existenz universeller Hashklassen

Die obige Anwendung ergibt selbstverständlich nur dann Sinn, wenn universelle Hashklassen existieren, deren Funktionen effizient berechnet werden können. Das erste Beispiel einer solchen Klasse ist für Rechenmodelle geeignet, die die effiziente Multiplikation und Division von ganzen Zahlen erlauben. Insbesondere sind die Hashfunktionen auf einer RAM $(+,\cdot,/)$ in konstanter Zeit auswertbar.

Es bezeichne im folgenden $\mathbb{Z}_n$ für ein $n \in \mathbb{N}$ den Restklassenring der ganzen Zahlen modulo n. Sei das Universum $U = \{0,\ldots,u-1\}$ und der Wertebereich $R = \{0,\ldots,r-1\}$. Weiterhin sei $p \geq u$ eine beliebige Primzahl und $f : \mathbb{Z}_p \to R$. Dann ist die *lineare Primzahlklasse* $\mathcal{H}^{lin}_{p,f}$ die Menge der Funktionen $h_{a,b}$ mit $a, b \in \mathbb{Z}_p$, $a \neq 0$, wobei

$$h_{a,b} : U \to R, \quad x \mapsto f\big((ax+b) \bmod p\big).$$

Wir zeigen nun, daß $\mathcal{H}^{lin}_{p,f}$ universell ist, wenn f die Elemente aus $\mathbb{Z}_p$ gleichmäßig über R verteilt. Für ein beliebiges $y \in R$ bezeichnet $f^{-1}(y)$ das *Urbild* von y unter f, d.h.

$$f^{-1}(y) = \big\{x \in \mathbb{Z}_p \,|\, f(x) = y\big\}.$$

1.1.3 Satz (Carter und Wegman, 1979). *Gilt $\left|f^{-1}(y)\right| \leq \lceil p/r \rceil$ für alle $y \in R$, so ist $\mathcal{H}^{lin}_{p,f}$ universell.*

Beweis: Es seien $x_1 \neq x_2$ beliebige Schlüssel, und a und b zufällig aus $\mathbb{Z}_p$ gewählt mit $a \neq 0$. Dann sind $z_1 = (ax_1 + b) \bmod p$ und $z_2 = (ax_2 + b) \bmod p$ zufällige, aber verschiedene Werte aus $\mathbb{Z}_p$. Dies folgt sofort aus der allgemein bekannten Tatsache, daß das lineare Gleichungssystem

$$(ax_1 + b) \equiv z_1 \pmod{p} \quad \text{und} \quad (ax_2 + b) \equiv z_2 \pmod{p}$$

für beliebige $z_1 \neq z_2$ im Körper $\mathbb{Z}_p$ eine eindeutige Lösung für $a \neq 0$, b hat. Somit genügt es zu zeigen, daß für zwei zufällig $z_1 \neq z_2$ aus $\mathbb{Z}_p$ gilt:

$$\textbf{Prob}\left(f(z_1) = f(z_2)\right) \;\leq\; 1/r.$$

Sei $f(z_1) = y$ für ein $y \in R$. Dann gilt $f(z_2) = y$ wegen $\left|f^{-1}(y)\right| \leq \lceil p/r \rceil$ und wegen $z_1 \neq z_2$ für höchstens $\lceil p/r \rceil - 1$ der $p-1$ möglichen Werte, die z_2 annehmen kann. Also ist

$$\textbf{Prob}\left(f(z_1) = f(z_2)\right) \;\leq\; \frac{\lceil p/r \rceil - 1}{p-1} \;\leq\; \frac{(p-1)/r}{p-1} \;=\; 1/r. \qquad \blacksquare$$

Für die Funktion f bieten sich z.B. die Abbildungen $x \mapsto x \bmod r$ oder $x \mapsto x \operatorname{div} \lceil p/r \rceil$ an, wobei „div“ die Division mit anschließendem Abrunden bezeichnet. Beide erfüllen offensichtlich die Voraussetzungen aus Satz 1.1.3. Außerdem lassen sie sich besonders effizient auswerten, wenn r bzw. $\lceil p/r \rceil$ Zweierpotenzen sind:

1.1.4 Bemerkung. Ist $x \geq 0$ dargestellt zur Basis 2 gleich $\langle x_{n-1} \ldots x_0 \rangle$, so ist $x \bmod 2^k = \langle x_{k-1} \ldots x_0 \rangle$ und $x \operatorname{div} 2^k = \langle x_{n-1} \ldots x_k \rangle$. Also läßt sich die Modulooperation mit 2^k durch ein bitweises „Und“ mit $2^k - 1$ und die Division durch eine Rechtsverschiebung um k Bits berechnen (s. Abbildung 1.1).

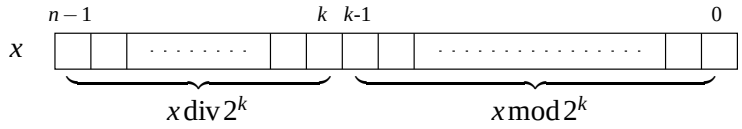

Abbildung 1.1: Division und Modulooperation mit einer Zweierpotenz

Zusammenfassung

Die lineare Primzahlklasse gehört aufgrund ihrer Einfachheit zu den wichtigsten universellen Hashklassen. Die Zeit für die Auswertung einer Hashfunktion ist im Wesentlichen durch eine Addition, eine Multiplikation und zwei Divisionen gegeben. Mit ihrer Hilfe erhält man ein dynamisches Wörterbuch, daß auf einer RAM$(+,\cdot,/)$ alle Operationen in erwarteter konstanter Zeit ausführen kann, sofern die Zahl der Einfügeoperationen den vorher festgelegten Speicherplatz nicht übersteigt. Beschreibungen dieses Wörterbuchs mit der Primzahlklasse oder ähnlichen Klassen finden sich in zahlreichen Lehrbüchern (s. z.B. Mehlhorn, 1988; Motwani und Raghavan, 1995; Cormen, Leiserson und Rivest, 1990). Brassard und Kannan (1988) beschreiben eine Variante des Wörterbuchs, bei der auch beliebig viele `Insert`-Operationen vorkommen dürfen. Die Größe der Hashtabelle wird dann dynamisch vergrößert, so daß immer nur linearer Platz für die Hashtabelle benötigt wird.

In vielen Fällen ist die `Find`-Operation die wichtigste, weil sie am häufigsten benutzt wird. Ein wesentlicher Nachteil des Wörterbuchs mit verketteten Listen besteht darin, daß eine schnelle Ausführung der `Find`-Operation nicht garantiert werden kann. Sucht man nach einem Element, das sich in der längsten Liste befindet, so ist die Suchzeit unter Umständen sogar im Erwartungswert sehr schlecht. Für ein solches Wörterbuch mit $O(r)$ Listen, in dem sich r Schlüssel befinden, garantiert die Universalität der Hashklasse nur eine erwartete Länge von $O(\sqrt{r})$ für die längste Liste. Tatsächlich gibt es universelle Hashklassen, die für bestimmte Schlüsselmengen zu diesen schlechten Ergebnissen führen (vgl. Alon, Dietzfelbinger, Miltersen, Petrank und Tardos, 1997).

Inzwischen existieren Wörterbücher, die auch auf universellen Hashklassen beruhen und asymptotisch bessere Ergebnisse erzielen. Das statische Wörterbuch von Fredman, Komlós und Szemerédi (1984) garantiert bei linearem Speicherplatz eine konstante Zeit für die `Find`-Operation und kann in erwarteter linearer Zeit konstruiert werden. Eine Erweiterung, bei der zusätzlich `Insert`- und `Delete`-Operationen in erwarteter amortisiert konstanter Zeit ausgeführt werden können, wird von Dietzfelbinger, Karlin, Mehlhorn, Meyer auf der Heide, Rohnert und Tarjan (1994) beschrieben. Während dieses dynamische Wörterbuch noch mit universellen Hashklassen auskommt, benötigt das Realzeitwörterbuch von Dietzfelbinger und Meyer auf der Heide (1992) kompliziertere Hashklassen mit noch stärkeren Eigenschaften. Dafür werden die `Insert`- und `Delete`-Operationen mit sehr hoher Wahrscheinlichkeit auch in konstanter Zeit ausgeführt.

Universelles Hashing spielt also eine zentrale Rolle bei der Lösung des Wörterbuchproblems. Die Ansätze reichen von einfachen, aber für viele praktische Fälle effizienten Datenstrukturen, bis hin zu komplizierten Schemata, welche die derzeit asymptotisch besten Algorithmen ausmachen.

§ 2. Ziele

Neben Wörterbüchern haben universelle Hashklassen seit ihrer Einführung von Carter und Wegman (1979) zahlreiche Anwendungen in den unterschiedlichsten Bereichen der Informatik gefunden. Dazu werden teilweise allgemeinere oder auch strengere Definitionen benutzt als die aus Definition 1.1.1 (s. dazu Kapitel 2). Zu den algorithmischen Anwendungen neben Wörterbuchproblemen gehören beispielsweise die Nachrichtenauthentifizierung (s. z.B. Wegman und Carter, 1979; Atici und Stinson, 1996; Shoup, 1996) oder die Bildverarbeitung (s. Lamdan und Wolfson, 1988; Illingworth und Kittler, 1988). Universelle Hashklassen wurden auch in randomisierten Algorithmen für Standardprobleme benutzt, so z.B. von Dietzfelbinger, Hagerup, Katajainen und Penttonen (1997) zum Auffinden von Punktpaaren mit kürzestem Abstand im mehrdimensionalen Raum oder von Matias und Vishkin (1991) sowie von Andersson, Hagerup, Nilsson und Raman (1995) zum Sortieren von ganzen Zahlen. Daneben gibt es wichtige Ergebnisse der Komplexitätstheorie, die auf universellem Hashing beruhen. Dazu zählt z.B. die Simulation von PRAMs (Mehlhorn und Vishkin, 1984), die Einordnung der Komplexitätsklasse *BPP* in der polynomiellen Hierarchie (Sipser, 1983) oder die Wahrscheinlichkeitsamplifikation (Chor und Goldreich, 1989; Impagliazzo und Zuckerman, 1989).

Aufgrund der Vielzahl von Anwendungen sind universelle Hashklassen an sich bereits in den Mittelpunkt zahlreicher Forschungsarbeiten gerückt (s. z.B. Siegel, 1989; Dietzfelbinger, 1996; Goldreich und Wigderson, 1997). Während einerseits die praktischen Anwendungsmöglichkeiten (vor allem Wörterbuchprobleme und Nachrichtenauthentifizierung) den

Bedarf an effizient einsetzbaren Hashklassen wecken, gewinnen andererseits auch theoretische Erkentnisse, z.B. die Beziehungen zu anderen kombinatorischen Objekten (s. Stinson, 1994a, 1996), zunehmend an Bedeutung.

Wenn es darum geht, universelles Hashing zu implementieren, sind mehrere Kriterien für die Auswahl einer Hashklasse zu berücksichtigen. Zunächst ist die Kardinalität der Hashklasse zu nennen. Diese bestimmt die Anzahl der Zufallsbits, die notwendig sind, um eine Hashfunktion zufällig aus der Famile auszuwählen. Da echte Zufallsbits nur schwer zu erzeugen sind, hat diese Größe entscheidenden Einfluß auf die Praktikabilität der Hashklasse.

Weiterhin ist die Wahrscheinlichkeitsverteilung der Hashwerte, die durch die zufällige Auswahl einer Hashfunktion induziert wird, von Bedeutung. Wir werden später den Universalitätsbegriff verallgemeinern, und auch Hashklassen betrachten, bei denen die Kollisionswahrscheinlichkeit durch andere Schranken als $1/r$ begrenzt ist. Die Kollisionswahrscheinlichkeit hat jedoch entscheidenden Einfluß auf die Erfolgswahrscheinlichkeit bzw. die erwartete Laufzeit von randomisierten Algorithmen, und ist daher eine wichtige Kenngröße für die Qualität einer Hashklasse.

Schließlich ist für die Praktikabiliät einer Hashklasse die effiziente Auswertbarkeit ihrer Funktionen von größter Wichtigkeit. Universelles Hashing wird in Wörterbüchern nur dann wirklich zur Anwendung kommen, wenn die Hashfunktionen ähnlich schnell berechnet werden können wie die gängigen Standardhashfunktionen, die ohne Zufall auskommen. So werden beispielsweise häufig für deterministisches Hashing Funktionen benutzt, die nur aus der Modulo-Operation mit einer Primzahl als Operand bestehen. Soll also universelles Hashing als Standardmethode anwendbar sein, so müssen Funktionen eingesetzt werden, die ähnlich schnell auswertbar wie beispielsweise ein Division oder eine Modulo-Operation sind.

Ein Großteil der gängigen Hashklassen kann diesen Anforderungen aber nicht gerecht werden. Viele Konstruktionen basieren auf komplizierter Arithmetik (z.B. Körper- oder Matrizenarithmetik) und erlauben daher nur auf spezieller Hardware eine effiziente Implementierung. Andere, auch für die üblichen Prozessoren effizient implementierbare Hashklassen, haben oft Defizite hinsichtlich ihrer Kardinalität oder der Verteilung der Schlüssel. Das speziell für die Nachrichtenauthentifizierung entwickelte „bucket-hashing“ (Rogaway, 1995) kann z.B. sehr lange Schlüssel äußerst schnell abbilden, garantiert aber nur eine für Wörterbücher unzureichende Kollisionswahrscheinlichkeit. Schließlich entstehen bei manchen bekannten universellen Hashklassen Probleme ganz anderer Art. So ist beispielweise für die lineare Primzahlklasse die Kenntnis einer Primzahl in der Größe des Universums erforderlich. Ist dieses bei der Implementierung des Algorithmus nicht fest vorgegeben, dann muß die Primzahl zur Laufzeit gefunden werden, und die zugrunde liegenden Algorithmen sind nicht mehr notwendigerweise uniform.

Erst neuere Konstruktionen können zu einem großen Teil die oben beschriebenen Anforderungen erfüllen. Die „Ganzzahlklassen“ von Dietzfelbinger, Hagerup, Katajainen und Penttonen (1997) sowie von Dietzfelbinger (1996) beruhen auf Funktionen

$$\{0,\ldots,u-1\} \to \{0,\ldots,r-1\}, \quad x \mapsto \big((ax+b) \bmod (kr)\big) \operatorname{div} k,$$

für ein festes k und zufällige Parameter a und b. Sie kommen also mit ganzzahliger Arithmetik und ohne Kenntnis von Primzahlen aus und haben eine akzeptable Kardinalität. Die Effizienz solcher Funktionen ist besonders hoch, wenn das Universum und der Wertebereich Zweierpotenzen sind (was in der Praxis ohnehin meistens der Fall ist). Dann können die

Modulooperation und die Division durch eine bitweises „Und“ bzw. eine Rechtsverschiebung der Bits ersetzt werden (s. Bemerkung 1.1.4). Damit ist die benötigte Zeit für die Auswertung einer Hashfunktion hauptsächlich durch die Multiplikation bestimmt. Da Multiplikationen normalerweise schneller ausgeführt werden können als Divisionen, sind solche Funktionen erheblich effizienter als beispielsweise die der linearen Primzahlklasse, was bereits auch durch experimentelle Studien belegt wurde (Dietzfelbinger und Hühne, 1996).

Allerdings sind bei den Ganzzahlklassen noch einige wichtige Fragen offen bzw. Verbesserungen möglich. So garantiert beispielsweise die von Dietzfelbinger et al. untersuchte „multiplikative“ Hashklasse nur eine obere Schranke von $2/r$ für die Kollisionswahrscheinlichkeit zweier Schlüssel (bei einem Wertebereich der Kardinalität r). Wünschenswert wäre aber für viele kombinatorische und praktische Anwendungen eine durch $1/r$ (oder kleiner) beschränkte Kollisionswahrscheinlichkeit.

Aufgrund der Bedeutung solcher Ganzzahlklassen ist es das Ziel dieser Diplomarbeit, Hashklassen zu untersuchen, die mit ganzzahliger Arithmetik und ohne Primzahlen auskommen. Dazu werden - nach den Definitionen in Kapitel 2 - zunächst die kombinatorischen Eigenschaften von universellen Hashklassen ausführlich in Kapitel 3 diskutiert. Es werden einerseits bekannte untere Schranken für die Kardinalität von Hashklassen vorgestellt und andererseits neue Konstruktionsmethoden zu Vergrößerung des Universums und des Wertebereichs, bzw. zur Verbesserung der Kollisionswahrscheinlichkeiten erarbeitet. Als Beispiel für die Mächtigkeit dieser Methoden werden wir neue Hashklassen entwickeln, die auf der Faltung von Vektoren beruhen. In Kapitel 4 werden wir schließlich die bekannten Ganzzahlklassen teilweise verallgemeinern und mit neuen Methoden analysieren, sowie die Techniken aus Kapitel 3 anwenden, um neue Ganzzahlklassen mit besseren Eigenschaften zu konstruieren. Unter anderem werden die ersten Hashklassen vorgestellt, die mit ganzzahliger Arithmetik ohne Primzahlen auskommen und eine optimal kleine Kollisionswahrscheinlichkeit für alle Schlüsselpaare garantieren.

KAPITEL 2

Definitionen

Seit der ursprünglichen Definition universeller Hashklassen von Carter und Wegman (1979) (s. Definition 1.1.1) wurde diese für die verschiedensten Anwendungen verallgemeinert, oder es wurden Hashklassen mit stärkeren Eigenschaften definiert. Wir wollen zunächst eine Übersicht über die wichtigsten Definitionen geben.

In diesem und den folgenden Kapiteln wird das Universum immer mit U und der Wertebereich mit R bezeichnet. Die Kardinalität des Universums bzw. des Wertebereichs bezeichnen wir mit den kleinen Buchstaben u bzw. r. Aus formalen Gründen sei immer $u \geq 2$.

§ 1. Der Universalitätsparameter

Ziel der Definition universeller Hashklassen in Kapitel 1 war es, für alle Schlüssel eine Kollisionswahrscheinlichkeit von höchstens $1/r$ zu garantieren. Häufig genügt es jedoch, wenn diese höchstens um einen konstanten Faktor von $1/r$ abweicht. Daher benutzt man oft die folgende, etwas allgemeinere Definition (s. z.B. Mehlhorn, 1982):

2.1.1 Definition. Eine Hashklasse $U \to R$ heißt *c-universell*, wenn für alle verschiedenen $x_1, x_2 \in U$

$$\textbf{Prob}\left(h(x_1) = h(x_2)\right) \;\leq\; \frac{c}{r}$$

gilt. Wir bezeichnen c dann als *Universalitätsparameter*.

Es wird wohl kaum eine Anwendung solcher Hashklassen geben, bei der der Wertebereich größer als das Universum ist. Daher setzen wir immer $|R| \leq |U|$ voraus, wenn wir c-universelle Hashklassen betrachten.

2.1.2 Beispiel. Bei der linearen Primzahlklasse $\mathcal{H}^{lin}_{p,f}$ (s. S. 4) könnte es in einigen Situationen vorteilhaft sein, auf die Addition mit b zu verzichten. Einerseits wird die Berechnung der Funktionen dadurch schneller, andererseits kann die Hälfte der notwendigen Zufallsbits bei Auswahl einer Funktion aus der Hashklasse eingespart werden. Natürlich sollte der Universalitätsparameter dabei nicht zu groß werden.

Sei $U = \{0,\ldots,u-1\}$, $R = \{0,\ldots,r-1\}$, und $p \geq u$ eine Primzahl. Wir definieren die *homogene Primzahlklasse* $\mathcal{H}^{hom}_{p,div}$ als die Menge der Funktionen

$$h_a : U \to R, \quad x \mapsto \big((ax) \bmod p\big) \operatorname{div} \lceil p/r \rceil$$

mit $a \in \mathbb{Z}_p \backslash \{0\}$. Die homogene Primzahlklasse ist 2-universell, wie folgende Überlegungen zeigen:

Sei $k = \lceil p/r \rceil$. Wir betrachten zunächst die Division $x \mapsto x \operatorname{div} k$. Offensichtlich kollidieren beliebige $z_1, z_2 \in \mathbb{Z}_p$ unter dieser Operation höchstens dann, wenn ihr Abstand kleiner als k ist. D.h. $h_a(x_1) = h_a(x_2)$ impliziert

$$\big|(ax_1) \bmod p - (ax_2) \bmod p\big| < k,$$

oder anders ausgedrückt

$$a(x_1 - x_2) \bmod p \;\in\; \{0,\ldots,k-1\} \cup \{p-k+1,\ldots,p-1\}.$$

Da wegen $a \neq 0$ auch $a(x_1 - x_2) \bmod p \neq 0$ ist, kann $a(x_1 - x_2) \bmod p$ also höchstens $2(k-1)$ Werte annehmen, für die x_1 und x_2 unter h_a kollidieren können. Weil $\mathbb{Z}_p$ ein Körper ist, hat $a(x_1 - x_2) \bmod p$ aber für jedes a einen anderen Wert. Also folgt:

$$\textbf{Prob}\left(h(x_1) = h(x_2)\right) \;\leq\; \frac{2(k-1)}{p-1} \;=\; \frac{2(\lceil p/r \rceil - 1)}{p-1} \;\leq\; \frac{2(p-1)/r}{p-1} \;=\; \frac{2}{r}.$$

Die homogenen Funktionen über $\mathbb{Z}_p$ lassen sich auch mit anderen Operationen als der Division verknüpfen. Ersetzt man z.B. die Division durch eine Operation Modulo r, so ist die Hashklasse auch 2-universell. Dies folgt aus einem Beweis in Fredman, Komlós und Szemerédi (1984, Lemma 1), obwohl diese Aussage dort nicht explizit angegeben ist.

§ 2. Optimale Universalität

Die bisher betrachteten Beispiele universeller Hashklassen hatten alle einen Universalitätsparameter von mindestens 1. Ein triviales Beispiel zeigt jedoch, daß dies im Allgemeinen nicht der bestmögliche Wert ist. So ist z.B. die Hashklasse $U \to U$, die nur aus der Identität *id* besteht, 0-universell, da keine zwei verschiedenen Schlüssel unter *id* kollidieren. Natürlich ist dies kein praxisrelevantes Beispiel, aber es wirft die Frage auf, wie klein der Universalitätsparameter werden kann. Eine erste untere Schranke wurde von Carter und Wegman (1979) angegeben, und von Sarwate (1980) zu der des folgenden Satzes verbessert.

2.2.1 Satz. *Zu jeder Hashklasse $\mathcal{H} : U \to R$ gibt es zwei verschiedene Schlüssel $x_1, x_2 \in U$ mit*

$$\textbf{Prob}\left(h(x_1) = h(x_2)\right) \;\geq\; \frac{u-r}{r(u-1)}.$$

Bevor wir den Satz beweisen, wollen wir die *δ-Notation* definieren, die bei der Analyse von Hashklassen oft hilfreich ist. Für $x_1, x_2 \in U$ und $h \in \mathcal{H}$ sei

$$\delta_h(x_1, x_2) \;=\; \begin{cases} 1 & \text{falls } x_1 \neq x_2 \text{ und } h(x_1) = h(x_2) \text{ und} \\ 0 & \text{sonst.} \end{cases}$$

Wenn wir statt h, x_1 oder x_2 Mengen schreiben, so ist die Summe über die Elemente der Mengen gemeint. Also bedeutet z.B. $\delta_{\mathcal{H}}(x_1, M)$ das Gleiche wie

$$\sum_{h \in \mathcal{H}} \sum_{x_2 \in M} \delta_h(x_1, x_2).$$

Beweis zu Satz 2.2.1: Sei $N = \left|\mathcal{H}\right|$ und $\varepsilon = 1/r \cdot (u-r)/(u-1)$. Wir berechnen zunächst eine untere Schranke für die Gesamtzahl von Kollisionen, die in einer Hashklasse zwischen allen Schlüsselpaaren auftreten. Es folgt dann aus dem Schubfachprinzip, daß es zwei Schlüssel geben muß, die unter mindestens εN Funktionen kollidieren.

Die Gesamtzahl von Kollisionen, die unter einer Funktion $h \in \mathcal{H}$ stattfinden, ist gegeben durch

$$\begin{aligned}\delta_h(U,U) &= \sum_{x_1,x_2 \in U} \delta_h(x_1,x_2) = \sum_{y \in R} \left|\{x_1,x_2 \in h^{-1}(y) \mid x_1 \neq x_2\}\right| \\ &= \sum_{y \in R} \left|h^{-1}(y)\right|\left(\left|h^{-1}(y)\right|-1\right) = \sum_{y \in R} \left|h^{-1}(y)\right|^2 - \sum_{y \in R} \left|h^{-1}(y)\right|.\end{aligned}$$

Da $\sum_{y \in R} \left|h^{-1}(y)\right| = u$ ist, ist der letzte Term in der obigen Gleichung offensichtlich dann minimal, wenn alle $h^{-1}(y)$ gleich groß sind, also u/r Elemente besitzen. Es gilt also $\delta_h(U,U) \geq \sum_{y \in R} u/r(u/r-1)$, und somit

$$\delta_{\mathcal{H}}(U,U) \;\geq\; Nu(u/r-1). \tag{2.1}$$

Wir nehmen nun im Widerspruch zur Aussage an, daß für jedes Paar verschiedener Schlüssel $x_1, x_2 \in U$ die Kollisionswahrscheinlichkeit kleiner als ε ist, also $\delta_{\mathcal{H}}(x_1,x_2) < \varepsilon N$. Dann ist offensichtlich $\delta_{\mathcal{H}}(U,U)$ kleiner als

$$\sum_{x_1 \neq x_2} \varepsilon N \;=\; u(u-1)\varepsilon N,$$

was sich zu

$$\delta_{\mathcal{H}}(U,U) \;< Nu(u/r-1)$$

vereinfacht und im Widerspruch zu Ungleichung (2.1) steht. ∎

Wie wir später zeigen werden, gibt es Hashklassen, für die diese untere Schranke auch eine obere ist, wenn r ein Teiler von u ist. Daher hat man universellen Hashklassen, die den optimalen Universalitätsparameter erreichen, eine eigene Bezeichnung gegeben (vgl. Sarwate, 1980).

2.2.2 Definition. Eine $(u-r)/(u-1)$-universelle Hashklasse $U \to R$ heißt *optimal universell*.

Um überhaupt die Existenz von optimal universellen Hashklassen zu zeigen, aber auch für spätere Beweise, ist eine Charakterisierung ihrer kombinatorischen Eigenschaften hilfreich. Obwohl sich die folgende, äußerst nützliche Aussage fast direkt aus dem Beweis zu Satz 2.2.1 ableiten läßt, findet sie sich bisher in der Literatur nicht wieder. Es sei aber hier schon auf die Äquivalenz von optimal universellen Hashklassen und *auflösbaren balanciert unvollständigen Block-Designs* (kurz: auflösbare BIBDs oder RBIBDs, s. Definition 3.3.10) verwiesen, die von Stinson (1994a) gezeigt wurde, und auf die wir in Kapitel 3, § 3 näher eingehen werden.

Wir sagen, eine Hashfunktion $h : U \to R$ hat eine *konstante Korbgröße*, wenn für alle $y \in R$ gilt $\left|h^{-1}(y)\right| = u/r$. Eine Hashklasse $\mathcal{H}$ hat konstante Korbgröße, wenn alle Funktionen aus $\mathcal{H}$ konstante Korbgröße haben.

2.2.3 Lemma. *Eine Hashklasse $U \to R$ ist genau dann optimal universell, wenn sie eine konstante Korbgröße hat, und die Kollisionswahrscheinlichkeit aller Paare verschiedener Schlüssel gleich ist.*

Beweis: Sei wieder $N = \left|\mathcal{H}\right|$ und $\varepsilon = 1/r \cdot (u-r)/(u-1)$. Aus dem Beweis zu Satz 2.2.1 ist ersichtlich, daß als notwendige Bedingung für die optimale Universalität in Ungleichung (2.1) Gleichheit vorliegen muß. Dies ist aber mit den Ausführungen in besagtem Beweis genau dann erfüllt, wenn $\left|h^{-1}(y)\right| = u/r$ für alle $h \in \mathcal{H}$, $y \in R$ gilt. Also ist die konstante Korbgröße eine notwendige Bedingung für die optimale Universalität von $\mathcal{H}$.

$\mathcal{H}$ besitze nun eine konstante Korbgröße, d.h. es ist

$$\delta_{\mathcal{H}}(U,U) = Nu(u/r-1). \tag{2.2}$$

Es genügt zu zeigen, daß $\mathcal{H}$ genau dann optimal universell ist, wenn alle Schlüsselpaare mit gleicher Wahrscheinlichkeit kollidieren. Angenommen $\mathcal{H}$ ist optimal universell und es gibt ein Paar (x_1,x_2) mit einer Kollisionswahrscheinlichkeit kleiner als ε. Dann folgt aber

$$\delta_{\mathcal{H}}(U,U) < u(u-1)N\varepsilon,$$

was im Widerspruch zu (2.2) steht. Sei nun die Bedingung erfüllt, daß alle Schlüsselpaare mit gleicher Wahrscheinlichkeit ε' kollidieren. Dann ist $\delta_{\mathcal{H}}(U,U) = u(u-1)N\varepsilon'$, woraus mit Gleichung (2.2) $\varepsilon' = \varepsilon$ folgt. ∎

Wir können nun leicht die Frage nach der Existenz von optimal universellen Hashklassen beantworten. Dazu sei r ein Teiler von u. Wir betrachten die Menge $\mathcal{H}_{all}$ aller Funktionen $h : \{1,\ldots,u\} \to \{1,\ldots,r\}$ für die gilt $\left|h^{-1}(i)\right| = u/r$ für alle $1 \le i \le r$. Die Voraussetzungen von Lemma 2.2.3 sind trivialerweise erfüllt, und somit ist $\mathcal{H}_{all}$ optimal universell.

Es ist auch offensichtlich, daß optimal universelle Hashklassen nicht existieren, wenn r kein Teiler von u ist.

§ 3. Strenge Universalität

Häufig ist nicht nur eine niedrige Kollisionswahrscheinlichkeit von Schlüsseln interessant. Zusätzlich kann es notwendig sein, daß Schlüssel paarweise unabhängig voneinander über den Wertebereich verteilt werden. Dies kann durch streng universelle Hashklassen erreicht werden, die von Wegman und Carter (1979) definiert wurden.

2.3.1 Definition. Eine Hashklasse $\mathcal{H} : U \to R$ heißt *streng c-universell*, wenn für alle verschiedenen $x_1, x_2 \in U$ und alle $y_1, y_2 \in R$ gilt:

$$\textbf{Prob}\left(h(x_1) = y_1 \wedge h(x_2) = y_2\right) \;\le\; \frac{c}{r^2}.$$

Dabei ist c wieder der *Universalitätsparameter*. Eine streng 1-universelle Hashklasse heißt auch *streng universell*.

Streng universelle Hashklasse bilden also jedes beliebige Paar von verschiedenen Schlüsseln mit gleicher Wahrscheinlichkeit auf jedes Paar von Hashwerten ab. Man erhält so aus einer Menge von Schlüsseln paarweise unabhängige Zufallsvariablen über R. Es ist offensichtlich, daß der Universalitätsparameter einer streng c-universellen Hashklasse nicht kleiner als 1 sein kann.

2.3.2 Bemerkung. Ist eine Hashklasse streng universell, so wird offensichtlich auch jeder einzelne Schlüssel auf jeden Funktionswert mit gleicher Wahrscheinlichkeit $1/r$ abgebildet. Hashklassen mit dieser Eigenschaft nennen wir *gleichverteilend*. Streng c-universelle Hashklassen mit $c > 1$ sind natürlich nicht notwendigerweise gleichverteilend.

2.3.3 Beispiel. Wir betrachten die *lineare Körperklasse*, die aus den linearen Funktionen über einem endlichen Körper konstruiert ist. Sie wird in zahlreichen Arbeiten verwendet, als Beispiel sei hier nur auf Wegman und Carter (1979) verwiesen. Sei $\mathbb{K}$ ein endlicher Körper der Ordnung n, das Universum U eine Teilmenge von $\mathbb{K}$ und R ein beliebiger Wertebereich mit $r \leq n$. Für eine Abbildung $f : \mathbb{K} \to R$ besteht die lineare Körperklasse $\mathcal{H}^{lin}_{\mathbb{K},f}$ aus den Funktionen $x \mapsto f(ax+b)$ mit $a,b \in \mathbb{K}$.Wir zeigen, daß $\mathcal{H}^{lin}_{\mathbb{K},f}$ streng universell ist, wenn $f^{-1}(y)$ für alle $y \in R$ die gleiche Kardinalität n/r hat.

Seien $y_1, y_2 \in R$ beliebig und x_1, x_2 verschiedene Schlüssel aus U. Für zufällige $a,b \in \mathbb{K}$ sei $z_1 = ax_1 + b$ und $z_2 = ax_2 + b$. Analog zum Beweis von Satz 1.1.3 ist klar, daß das Paar (z_1, z_2) jeden Wert aus $\mathbb{K} \times \mathbb{K}$ mit gleicher Wahrscheinlichkeit $1/n^2$ annimmt. Da es genau $(n/r)^2$ Paare gibt, die in $f^{-1}(y_1) \times f^{-1}(y_2)$ liegen, ist also (z_1, z_2) mit Wahrscheinlichkeit $1/r^2$ eines dieser Paare.

Eine wichtige Anwendung von paarweise unabhängigen Zufallsvariablen ist die *Wahrscheinlichkeitsamplifikation*. Wir betrachten einen beliebigen randomisierten Algorithmus mit einseitigem Fehler. Sei z.B. `PrimTest(`x,z`)` der Algorithmus von Solovay und Strassen (1977), der berechnet, ob die Zahl x eine Primzahl ist. Dabei ist z eine Zufallszahl aus $\{1,\ldots,x-1\}$, von der der Erfolg des Algorithmus abhängt: Ist x eine Primzahl, so ist das Ergebnis des Algorithmus unabhängig vom Wert z immer „ja“; ist hingegen x keine Primzahl, so antwortet der Algorithmus „nein“ mit einer Wahrscheinlichkeit von mindestens $1/2$. Um mit hoher Wahrscheinlichkeit annehmen zu können, daß eine Zahl x wirklich prim ist, kann man diesen Algorithmus mehrfach mit unabhängigen Zufallszahlen z widerholen. Man reduziert so durch n-faches Ausführen des Algorithmus die Fehlerwahrscheinlichkeit von $1/2$ auf $1/2^n$. Nachteilig dabei ist, daß man dazu auch die n-fache Menge an Zufallsbits benötigt.

Um mit weniger Zufall auszukommen, kann man statt unabhängiger Eingaben für z auch zweifach unabhängige benutzen: Sei x eine beliebige Zahl, aber nicht prim, und $\mathcal{H}$ eine streng universelle Hashklasse $\{1,\ldots,n\} \to \{1,\ldots,x-1\}$. Wir erzeugen nun n Zufallszahlen $z_1,\ldots,z_n$ für den Primzahltest, indem wir eine zufällige Hashfunktion h für die n Elemente des Universums auswerten. Also $z_i = h(i)$ für $1 \leq i \leq n$. Es sei $Z_i \in \{0,1\}$ eine Zufallsvariable, die genau dann den Wert 1 hat, wenn der Primzahltest `PrimTest(`x,z_i`)` die falsche Antwort „ja“ gibt, und $Z = \sum_{i=1}^{n} Z_i$. Dann ist die Fehlerwahrscheinlichkeit ε (die Wahrscheinlichkeit, daß der Primzahltest auf allen Zufallseingaben z_i eine falsche Antwort gibt), bestimmt durch

$$\varepsilon \;=\; \mathbf{Prob}\,(Z \geq n).$$

Es sind alle Z_i Bernoulli-verteilt mit einer Erfolgswahrscheinlichkeit von höchstens $1/2$. Somit ist ihr Erwartungswert jeweils durch $1/2$ und ihre Varianz durch $1/4$ beschränkt. Aufgrund der Linearität des Erwartungswertes folgt $\mathbf{E}[Z] \leq n/2$, und weil die Z_i paarweise unabhängig sind, ist $\mathbf{Var}[Z] = \sum_{i=1}^{n} \mathbf{Var}[Z_i] \leq n/4$. Wir erhalten mit der Chebychev-Ungleichung

$$\varepsilon \;\leq\; \mathbf{Prob}\left(\left|Z - \mathbf{E}[Z]\right| \geq n - \mathbf{E}[Z]\right) \;\leq\; \frac{\mathbf{Var}[Z]}{\left(n - \mathbf{E}[Z]\right)^2} \;\leq\; \frac{n/4}{(n-n/2)^2} \;=\; \frac{1}{n}.$$

Mit nur einer zufällig gewählten Hashfunktion kann also die Fehlerwahrscheinlichkeit auf $1/n$ reduziert werden. Benutzen wir als Hashklasse die Körperklasse $\mathcal{H}^{lin}_{\mathbb{K},f}$ für einen Körper der Ordnung $k \geq x$ (es gibt offensichtlich so einen Körper mit $x \leq k \leq 2x$), so genügen zwei Zufallszahlen in der Größenordnung von k, um durch n Anwendungen des Algorithmus eine Fehlerwahrscheinlichkeit von höchstens $1/n$ zu erreichen.

§ 4. Ergänzende Bemerkungen

Die Bezeichnung für universelle Hashklassen ist in der Literatur sehr uneinheitlich. So werden 1-universelle Hashklassen z.B. in der ursprünglichen Definition von Carter und Wegman als „universal$_2$“ bezeichnet, wobei die 2 im Index ausdrücken soll, daß man die Kollisionswahrscheinlichkeit von jeweils zwei Schlüsseln betrachtet. In anderen Arbeiten wird statt c-universell oder streng c-universell der Term „ε-almost universal“ bzw. „ε-almost strongly universal“ verwendet, wobei $\varepsilon = c/r$ die Kollisionswahrscheinlichkeit angibt (vgl. z.B. Stinson, 1994b). Der Nachteil dieser Bezeichnungen ist, daß sie nicht das umgekehrt proportionale Verhältnis von Kollisionswahrscheinlichkeit zur Größe des Wertebereichs widerspiegeln. Sie sind daher vor allem in Arbeiten gebräuchlich, die sich mit kryptographischen Aspekten - insbesondere der Authentifizierung von Nachrichten (s. z.B. Rogaway, 1995) - der Hashklassen beschäftigen, wo die Beschränkung der Kollisionswahrscheinlichkeit durch eine Konstante ausreicht.

Die Idee, mit Hilfe zweifach unabhängiger Zufallsvariablen die Erfolgswahrscheinlichkeit von Algorithmen zu erhöhen, ist unter dem Begriff „two-point sampling“ bekannt (vgl. Motwani und Raghavan, 1995), und wurde zuerst von Chor und Goldreich (1989) beschrieben. Sie findet in zahlreichen Algorithmen Anwendung (s. z.B. Alon, Babai und Itai, 1986; Luby, 1986), und wurde zu einer Technik weiterentwickelt, mit der randomisierte Algorithmen derandomisiert werden können (s. auch Nisan, 1992; Alon, Goldreich, Håstad und Peralta, 1992). Zweifach unabhängige Zufallsvariablen spielen in zahlreichen weiteren Gebieten der theoretischen Informatik eine zentrale Rolle. Eine Übersicht über die wichtigsten Arbeiten findet man bei Wigderson (1994).

Die Definition strenger c-Universalität wird häufig auch zur (c,k)-Universalität verallgemeinert (s. z.B. Wegman und Carter, 1979; Alon, Goldreich, Håstad und Peralta, 1992; Dietzfelbinger, 1996). Dabei wird dann die Verteilung von k statt zwei Schlüsseln betrachtet. Für $c = 1$ erhält man so k-fach unabhängige Zufallsvariablen, mit denen sich beispielsweise eine stärkere Wahrscheinlichkeitsamplifikation erreichen läßt. Van Trung (1993) betrachtet auch die Kollisionswahrscheinlichkeiten von mehr als nur zwei Schlüsseln, was aber eher von kombinatorischem als von praktischem Interesse ist.

KAPITEL 3

Kombinatorik

Um effiziente, universelle Hashklassen konstruieren und bewerten zu können, ist es zunächst notwendig, sich allgemein mit ihren kombinatorischen Eigenschaften auseinanderzusetzen. Eine wichtige kombinatorische Eigenschaft, nämlich die untere Schranke für den Universalitätsparameter aus Satz 2.2.1, haben wir bereits kennengelernt. Sie erlaubt uns beispielsweise, neu konstruierte Hashklassen in ihrer Güte bezüglich der maximalen Kollisionswahrscheinlichkeit zu beurteilen. Die Charakterisierung optimal universeller Hashklassen aus Lemma 2.2.3 zeigt, wie diese aufgebaut sein müssen, und führt zu einem besseren Verständnis „guter“ Hashklassen.

In den folgenden Abschnitten werden wir uns zunächst mit unteren Schranken für die Kardinalität von universellen Hashklassen beschäftigen, um danach kombinatorische Methoden zu erarbeiten, die die Konstruktion effizienter Hashklassen erlauben. Dabei werden unter anderem bekannte Beziehungen zwischen universellem Hashing und kombinatorischen Designs aufgezeigt, sowie neue hergestellt.

§ 1. Die Kardinalität von Hashklassen

Beim Entwurf von randomisierten Algorithmen spielt die „Menge“ des verwendeten Zufalls eine wichtige Rolle. So stehen echte Zufallsbits normalerweise nur in begrenztem Umfang zur Verfügung, weil ihre Erzeugung zeitaufwendig ist, und unter Umständen spezieller Hardware bedarf. Für die zufällige Wahl einer Hashfunktion aus einer Hashklasse der Kardinalität n werden mindestens $\lceil \log n \rceil$ Zufallsbits benötigt. Um mit möglichst wenig Zufall auszukommen, ist man daher bestrebt, kleine Hashklassen zu konstruieren.

Dies ist aber auch noch aus einem anderen Grund sinnvoll: Die Anzahl der Bits, die zur Identifikation einer Hashfunktion notwendig sind, bestimmt auch den Speicherbedarf für das Abspeichern der Funktion. Dies ist dann wichtig, wenn ein Algorithmus viele Hashfunktionen gleichzeitig benötigt. Bei den auf Seite 6 erwähnten Wörterbüchern mit konstanter Suchzeit müssen immer mindestens so viele Hashfunktionen gespeichert werden, wie Schlüssel im Wörterbuch vorhanden sind. Die Kardinalität der Hashklasse hat somit auch entscheidenden Einfluß auf den Speicherplatzbedarf solcher Algorithmen.

Mehlhorn (1982) hebt die Bedeutung von kleinen Hashklassen in der Nachrichtenauthentifizierung nach einem von Wegman und Carter (1979) vorgeschlagenen Schema hervor. Bei diesem wählt eine der kommunizierenden Parteien eine Hashfunktion h zufällig aus einer Hashklasse $\mathcal{H} : U \to R$ aus, um sie dann *geheim* der anderen Partei zu übermitteln. Anschließend kann eine beliebige Nachricht $x \in U$ mit einem Authentifizierungstag $h(x)$ versehen und beides öffentlich versendet werden. Dieses Schema ergibt natürlich nur dann Sinn, wenn die Hashfunktionen mit erheblich weniger Bits beschrieben werden können als die Nachrichten, also wenn $\log \left|\mathcal{H}\right|$ wesentlich kleiner als $\log |U|$ ist.

Eine untere Schranke für c-universelle Hashklassen

Wir werden zunächst eine untere Schranke für die Kardinalität von c-universellen Hashklassen bestimmen. In der hier angegebenen allgemeinen Form wurde die Schranke zuerst von Stinson (1994b) gezeigt; die Schranken für 1-universelle und optimal universelle Hashklassen finden sich bereits bei Stinson (1994a) wieder.

Der hier vorgestellte Beweis liefert jedoch zusätzlich eine neue Aussage: Es werden diejenigen Hashklassen kombinatorisch charakterisiert, deren Kardinalität die untere Schranke erreicht. Diese Charakterisierung erlaubt es uns später, die Existenz bzw. Nichtexistenz solcher Hashklassen für bestimmte Universalitätsparameter zu zeigen, sowie die Verbindung zu anderen kombinatorischen Objekten herzustellen.

Es ist hilfreich, Hashklassen als Matrizen zu betrachten. Dazu ordnet man jedem Schlüssel eine Zeile, und jeder Hashfunktion eine Spalte zu. Die Einträge der Matrix entsprechen dann den Funktionswerten der Schlüssel:

3.1.1 Definition. Sei $\mathcal{H} = \{h_1, \ldots, h_n\}$ eine Hashklasse mit Universum $U = \{x_1, \ldots, x_u\}$. Die *Abbildungsmatrix* $(\mathcal{H})$ von $\mathcal{H}$ ist die $u \times n$-Matrix mit

$$(\mathcal{H})_{i,j} = h_j(x_i) \quad \text{für } 1 \leq i \leq u \text{ und } 1 \leq j \leq n.$$

Die *transponierte Hashklasse* $\mathcal{H}^T$ ist diejenige Hashklasse, deren Abbildungsmatrix durch die Transponierte von $(\mathcal{H})$ bestimmt ist.

3.1.2 Satz. *Sei $\mathcal{H}$ eine c-universelle Hashklasse $U \to R$ mit $c = 1 + \delta$ und*

$$N_{univ}(u, r, \delta) := \frac{u(r-1)}{r(r-1) + \delta(u-r)}.$$

(a) Es gilt $|\mathcal{H}| \geq N_{univ}(u, r, \delta)$.

(b) Es gilt $|\mathcal{H}| = N_{univ}(u, r, \delta)$ genau dann, wenn

(b1) $\mathcal{H}^T$ streng universell ist und

(b2) alle Paare verschiedener Schlüssel $x_1, x_2 \in U$ entweder unter keiner oder unter genau $|\mathcal{H}|c/r$ Funktionen aus $\mathcal{H}$ kollidieren.

Wir nennen eine $(1+\delta)$-universelle Hashklasse $U \to R$ *Stinson-minimal*, wenn ihre Kardinalität $N_{univ}(u, r, \delta)$ beträgt. Wir werden später zeigen, daß Stinson-minimale Hashklassen nicht für alle Universalitätsparameter existieren.

Der Satz zeigt beispielsweise, daß 1-universelle Hashklassen ungeeignet für die Nachrichtenauthentifizierung nach dem von Wegman und Carter vorgeschlagenen Schema sind. Denn bei solchen Hashklassen benötigt man zur Beschreibung einer Hashfunktion mindestens $\log u - \log r$ Bits. Das bedeutet, daß entweder die Länge des Authentifizierungstags oder die der Hashfunktion in der Größenordnung der zu authentifizierenden Nachricht ist. Das führt zu einem zusätzlichen Kommunikationsaufwand, der das Verfahren unpraktikabel macht.

Aus kombinatorischer Sicht ist die neue Aussage (b1) interessant. Danach erhält man, wenn man bei einer Stinson-minimalen Hashklasse die Rolle der Schlüssel und der Hashfunktionen vertauscht, eine streng universelle Hashklasse $\mathcal{H} \to R$ der Kardinalität u.

Bevor wir den Satz beweisen, wollen wir kurz die Idee skizzieren: Wir werden eine Menge B von Schlüsseln betrachten, die alle unter einer bestimmten Funktion h_0 aus $\mathcal{H}$ kollidieren. Dann bestimmen wir eine untere und eine obere Schranke für $\delta_{\mathcal{H}\setminus\{h_0\}}(B,B)$. Die untere Schranke ergibt sich leicht aus der maximalen Kollisionswahrscheinlichkeit zweier Schlüssel. Für die obere Schranke hingegen benutzt man die Tatsache, daß sich die $|B|$ Schlüssel in jeder Funktion auf r Funktionswerte verteilen müssen. Dies führt zu einer Mindestzahl von Kollisionen pro Funktion. Also wächst die obere Schranke für $\delta_{\mathcal{H}\setminus\{h_0\}}(B,B)$ mit der Anzahl der Funktionen, und wir erhalten schließlich eine Ungleichung, aus der die Schranke für $\left|\mathcal{H}\right|$ hervorgeht.

Beweis zu Satz 3.1.2: *Zu (a):* Sei N die Kardinalität von $\mathcal{H}$ und $h_0 \in \mathcal{H}$ beliebig. Gemäß dem Schubfachprinzip gibt es ein $y_0 \in R$ mit $\left|{h_0}^{-1}(y_0)\right| \geq u/r$. Sei $B := {h_0}^{-1}(y_0)$, $b := |B|$ und $H := \mathcal{H}\setminus\{h_0\}$. Aufgrund der Voraussetzung kollidieren zwei beliebige Schlüssel $x_1 \neq x_2$ aus B unter den Funktionen aus H höchstens $N \cdot c/r - 1$ mal. Es folgt also

$$\delta_H(B,B) \;=\; \sum_{x_1,x_2 \in B} \delta_H(x_1,x_2) \;\leq\; b(b-1)(N \cdot c/r - 1). \tag{3.1}$$

Wir bemerken, daß Gleichheit in (3.1) genau dann gilt, wenn alle Schlüsselpaare aus B eine Kollisionswahrscheinlichkeit von genau c/r haben.

Sei nun $h \in H$ beliebig. Dann gilt offensichtlich

$$\delta_h(B,B) \;=\; \sum_{y \in R} \left|h^{-1}(y) \cap B\right| \Big(\left|h^{-1}(y) \cap B\right| - 1 \Big).$$

Weil $\sum_{y \in R} \left|h^{-1}(y) \cap B\right| = b$ gilt, ist dies sicherlich minimal, wenn alle $h^{-1}(y) \cap B$ gleiche Kardinalität b/r haben. Somit ist

$$\delta_H(B,B) \;\geq\; \sum_{h \in H} \sum_{y \in R} \frac{b}{r}\left(\frac{b}{r} - 1\right) \;\geq\; (N-1)b\left(\frac{b}{r} - 1\right). \tag{3.2}$$

Hier gilt Gleichheit genau dann, wenn $\left|h^{-1}(y) \cap B\right| = b/r$ für alle $y \in R$ und alle $h \in H$ ist.

Aus den Ungleichungen (3.1) und (3.2) folgt nun

$$b(b-1)(N \cdot c/r - 1) \;\geq\; (N-1)b\left(\frac{b}{r} - 1\right),$$

und wir erhalten folgende untere Schranke für N:

$$N \;\geq\; \frac{b(b-1) - b(b/r-1)}{c/r \cdot b(b-1) - b(b/r-1)} \;=\; \frac{b - b/r}{c/r \cdot b - c/r - b/r + 1} \;=\; \frac{br - b}{bc - b + r - c}.$$

Dieser Term ist - da c auf keinen Fall größer als r ist - monoton steigend in b. Weil B so gewählt wurde, daß $b \geq u/r$ gilt, folgt

$$N \;\geq\; \frac{u - u/r}{u/r \cdot c - u/r + r - c} \;=\; \frac{u(r-1)}{u\delta + u - u + r^2 - \delta r - r} \;=\; \frac{u(r-1)}{r(r-1) + \delta(u-r)}.$$

Damit ist die untere Schranke für $\left|\mathcal{H}\right|$ gezeigt.

Zu (b): Wie aus den obigen Ausführungen ersichtlich ist, gilt $N = N_{univ}(u,r,\delta)$ genau dann, wenn die folgenden drei Bedingungen erfüllt sind:

(i) Alle Schlüsselpaare aus B haben eine Kollisionswahrscheinlichkeit von genau c/r.

(ii) $\left|h^{-1}(y) \cap B\right| = b/r$ für alle $h \neq h_0$ und alle $y \in R$.

(iii) $b = u/r$.

Dabei ist $B = {h_0}^{-1}(y_0)$ für eine beliebige Funktion h_0 und ein y_0 so, daß B mindestens u/r Schlüssel enthält.

Sei zunächst $\mathcal{H}$ eine $(1+\delta)$-universelle Hashklasse mit $\left|\mathcal{H}\right| = N_{univ}(u,r,\delta)$. Wir können h_0 beliebig wählen, und für jedes y_0, das zu einem $b \geq u/r$ führt, gilt nach (iii) sogar $b = u/r$. Also hat $\mathcal{H}$ konstante Korbgröße. Somit ist es egal, welches h_0 und welches y_0 wir wählen, um ein B zu bestimmen. Wir betrachten nun zwei beliebige Schlüssel $x_1 \neq x_2$ aus U. Entweder sie kollidieren gar nicht, oder es gibt ein h_0 und ein y_0 mit $h_0(x_1) = h_0(x_2) = y_0$. Für $B = {h_0}^{-1}(y_0)$ folgt dann aus (i), daß die Kollisionswahrscheinlichkeit dieser Schlüssel genau c/r beträgt, und somit ist (b2) gezeigt. Für (b1) betrachten wir zwei beliebige Hashfunktionen $h_1 \neq h_2$ und zwei $y_1, y_2 \in R$. Wir müssen zeigen, daß $\mathcal{H}^T$ streng universell ist, also daß bei zufälliger Wahl eines Schlüssels $x \in U$

$$\textbf{Prob}\left(h_1(x) = y_1 \wedge h_2(x) = y_2\right) \;=\; 1/r^2$$

gilt. Dazu setzen wir $B = {h_1}^{-1}(y_1)$, woraus mit (ii) und (iii) folgt, daß es genau u/r^2 Schlüssel in ${h_1}^{-1}(y_1) \cap {h_2}^{-1}(y_2)$ gibt. Die Wahrscheinlichkeit, daß sich ein zufällig gewählter Schlüssel in diesem Schnitt befindet, beträgt also wie gefordert $1/r^2$.

Seien nun (b1) und (b2) erfüllt. Wir müssen noch zeigen, daß dann (i)-(iii) für beliebige $B = {h_0}^{-1}(y_0)$ mit $b \geq u/r$ gelten. Aus (b2) folgt unmittelbar (i), denn wenn sich zwei verschiedene Schlüssel in B befinden, so kollidieren sie unter h_0 miteinander, und haben dann nach Voraussetzung (b2) eine Kollisionswahrscheinlichkeit von genau c/r. Aufgrund von Voraussetzung (b1) ist $\mathcal{H}^T$ gleichverteilend (s. Bemerkung 2.3.2), was nichts anderes bedeutet, als daß es genau u/r Schlüssel gibt, die von h_0 auf y_0 abgebildet werden. Somit gilt (iii). Aus der strengen Universalität von $\mathcal{H}^T$ folgt wiederum, daß für alle Funktionen $h \neq h_0$ und beliebige $y \in R$ sowie ein zufällig aus U gewähltes x gilt:

$$\textbf{Prob}\left(h(x) = y \wedge h_0(x) = y_0\right) \;=\; 1/r^2.$$

Also gibt es genau u/r^2 Schlüssel, die sich in $h^{-1}(y) \cap {h_0}^{-1}(y_0)$ befinden. Mit $b = u/r$ folgt somit (ii). ∎

Von besonderer Bedeutung sind Stinson-minimale optimal universelle und 1-universelle Hashklassen. Wir erhalten aus obigem Satz

$$N_{univ}(u,r,\delta) = \quad \frac{u}{r} \qquad \text{für } \delta = 0, \text{ und} \tag{3.3}$$

$$N_{univ}(u,r,\delta) = \quad \frac{u-r}{r-1} \qquad \text{für } \delta = \frac{u-r}{u-1} - 1. \tag{3.4}$$

Bei optimal universellen Hashklassen ist Bedingung (b2) ohnehin immer erfüllt (s. Lemma 2.2.3). Also ist eine optimal universelle Hashklasse $\mathcal{H}$ genau dann Stinson-minimal,

wenn $\mathcal{H}^T$ streng universell ist. Interessanterweise sind diese transponierten Hashklassen genau diejenigen, die die im nächsten Abschnitt gezeigte untere Schranke für die Kardinalität streng universeller Hashklassen erreichen.

Es gibt Stinson-minimale 1-universelle und optimal universelle Hashklassen, wenn u und r Potenzen der gleichen Primzahl sind. Wir werden in § 3 Methoden vorstellen, mit denen man solche Hashklassen konstruieren kann. Hier soll jedoch zunächst gezeigt werden, für welche Universalitätsparameter Stinson-minimale Hashklassen nicht existieren. Dazu benötigen wir folgendes Lemma, was sich auch später noch als nützlich erweisen wird.

3.1.3 Lemma. *Sei $\mathcal{H}$ eine Stinson-minimale c-universelle Hashklasse $U \to R$. Weiterhin seien $x_0 \in U$, $h_0 \in \mathcal{H}$ und $y_0 \in R$ beliebig. Gilt $h_0(x_0) \neq y_0$, so ist die Anzahl der Schlüssel aus ${h_0}^{-1}(y_0)$, die mit x_0 unter irgendeiner Funktion aus $\mathcal{H}$ kollidieren, gleich*

$$\frac{(u-r)(r-c)}{cr(r-1)}.$$

Beweis: Sei $N = \left|\mathcal{H}\right|$. Jeder Schlüssel, der mit x_0 kollidiert, tut dies gemäß Satz 3.1.2 unter genau Nc/r Funktionen aus $\mathcal{H}$. Also ist die Anzahl der Schlüssel aus ${h_0}^{-1}(y_0)$, die mit x_0 überhaupt kollidieren, gegeben durch

$$\frac{\delta_{\mathcal{H}}\left({h_0}^{-1}(y_0), x_0\right)}{Nc/r}. \tag{3.5}$$

Da x_0 nicht in ${h_0}^{-1}(y_0)$ enthalten ist, erhält man

$$\begin{aligned}\delta_{\mathcal{H}}\left({h_0}^{-1}(y_0), x_0\right) &= \sum_{h \in \mathcal{H} \setminus \{h_0\}} \left|\left\{x \in {h_0}^{-1}(y_0) \;\middle|\; h(x) = h(x_0)\right\}\right| \\ &= \sum_{h \in \mathcal{H} \setminus \{h_0\}} \left|{h_0}^{-1}(y_0) \cap h^{-1}\left(h(x_0)\right)\right|.\end{aligned}$$

Wählen wir $y = h(x_0)$ für ein h in dieser Summe, so können wir den dazugehörigen Summanden als $\left|{h_0}^{-1}(y_0) \cap h^{-1}(y)\right|$ schreiben. Dieser Betrag hat den Wert u/r^2. Denn nach Satz 3.1.2 ist $\mathcal{H}^T$ streng universell, was bedeutet, daß genau u/r^2 Schlüssel in einen beliebigen Schnitt ${h_0}^{-1}(y_0) \cap h^{-1}(y)$ fallen, sofern h_0 und h verschieden sind.

Wir erhalten also $\delta_{\mathcal{H}}\left({h_0}^{-1}(y_0), x_0\right) = (N-1)u/r^2$. Mit (3.5) beträgt somit die Anzahl der Schlüssel aus ${h_0}^{-1}(y_0)$, die mit x_0 kollidieren, $u(N-1)/(Ncr)$. Da $\mathcal{H}$ Stinson-minimal ist, und somit $N = N_{univ}(u, r, c-1)$ gilt, folgt die Behauptung aus folgender Umformung:

$$\begin{aligned}\frac{u(N-1)}{Ncr} &= \frac{u}{cr}\left(1 - \frac{1}{N_{univ}(u,r,c-1)}\right) = \frac{u}{cr}\left(1 - \frac{r(r-1)+(c-1)(u-r)}{u(r-1)}\right) \\ &= \frac{u}{cr} \cdot \frac{ur - u - r^2 + r + u - uc - r + cr}{u(r-1)} = \frac{u(r-c)+r(c-r)}{cr(r-1)} \\ &= \frac{(u-r)(r-c)}{cr(r-1)}.\end{aligned}$$

∎

Wir betrachten nun eine Stinson-minimale, c-universelle Hashklasse. Ein beliebiger Korb ${h_0}^{-1}(y_0)$ enthält genau u/r Schlüssel. Also ist die Anzahl seiner Elemente, die mit einem darin nicht enthaltenen Schlüssel x_0 kollidieren, durch den Ausdruck $u/r-k$ für ein $k \in \{0,\ldots,u/r\}$ bestimmt. Wir erhalten

$$\begin{aligned}
& u/r-k \;=\; \frac{(u-r)(r-c)}{cr(r-1)} \\
\iff \quad & (u/r-k)cr(r-1) \;=\; ur-r^2-uc+rc \\
\iff \quad & cr(u-kr-u/r+k) \;=\; c(r-u)+ur-r^2 \\
\iff \quad & c \;=\; \frac{ur-r^2}{ur-kr^2-u+kr-r+u} \;=\; \frac{u-r}{u-kr+k-1} \;=\; \frac{u-r}{u-1-k(r-1)}.
\end{aligned}$$

3.1.4 Korollar. *Sei $\mathcal{H}$ eine Stinson-minimale c-universelle Hashklasse $U \to R$. Dann ist*

$$c \;=\; \frac{u-r}{u-1-k(r-1)}$$

für ein $k \in \{0,\ldots,u/r\}$. ∎

Dieses Korollar läßt sich auch wie folgt interpretieren: Zu jedem Universum U und Wertebereich R gibt es höchstens u/r Universalitätsparameter $c = 1+\delta$, für die es c-universelle Hashklassen der Kardinalität $N_{univ}(u,r,\delta)$ gibt. D.h., daß es für fast alle Universalitätsparameter c eine größere untere Schranke als die aus Satz 3.1.2 (a) geben muß.

Eine untere Schranke für streng c-universelle Hashklassen

Eine untere Schranke für die Kardinalität streng 1-universeller Hashklassen findet sich bei Stinson (1994a). Dort wurde gezeigt, daß genau die Abbildungsmatrizen streng universeller Hashklassen sogenannte *orthogonale Arrays* sind, deren Mindestzahl von Spalten bereits von Plackett und Burman (1945) abgeschätzt wurde. Die Idee des Beweises für orthogonale Arrays führte dann bei Stinson (1994b) zu einer unteren Schranke für gleichverteilende streng c-universelle Hashklassen[1] mit beliebigem Universalitätsparameter c. Van Trung (1994) charakterisiert Hashklassen, die diese Schranke erreichen, mithilfe von kombinatorischen Designs. Wir werden einen auf der gleichen Idee beruhenden Beweis vorstellen, der aber erstmals - ähnlich wie in Satz 3.1.2 - zu einer Charakterisierung durch die transponierte Hashklasse führt.

3.1.5 Satz. *Sei $\mathcal{H}$ eine streng c-universelle und gleichverteilende Hashklasse $U \to R$ mit $c = 1+\delta$, und*

$$N_{strg}(u,r,\delta) \;=\; 1+\frac{u(r-1)^2}{r-1+\delta(u-1)}.$$

(a) Es gilt $\left|\mathcal{H}\right| \geq N_{strg}(u,r,\delta)$.

(b) Es gilt $\left|\mathcal{H}\right| = N_{strg}(u,r,\delta)$ genau dann, wenn

(b1) $\mathcal{H}^T$ optimal universell ist, und

(b2) wenn $\mathcal{H}$ für alle Schlüssel $x_1 \neq x_2$ und beliebige $y_1, y_2 \in R$ entweder keine oder genau $\left|\mathcal{H}\right| c/r^2$ Funktionen enthält, die x_1 auf y_1 und gleichzeitig x_2 auf y_2 abbilden.

[1] Stinson bezeichnet die Hashklassen mit „ε-Almost Strongly Universal“, wobei $\varepsilon = c/r$ ist. Seine Definition beinhaltet jedoch bereits die gleichverteilende Eigenschaft der Hashklasse.

Beweis: *Zu (a):* Sei $N = \left|\mathcal{H}\right|$, h_0 eine beliebige Funktion aus $\mathcal{H}$ und $H = \mathcal{H}\setminus\{h_0\}$. Für $h \in H$ sei $X_h = \left|\{x \in U \mid h(x) = h_0(x)\}\right|$. Es ist also

$$\sum_{h\in H} X_h \;=\; \sum_{x\in U} \left|\{h \in H \mid h(x) = h_0(x)\}\right|.$$

Da $\mathcal{H}$ gleichverteilend ist, enthält $\mathcal{H}$ für jedes x genau N/r Funktionen, die x auf $h_0(x)$ abbilden. Wir erhalten also

$$\sum_{h\in H} X_h \;=\; u(N/r-1).$$

Wir betrachten nun den Term $\sum_{h\in H} X_h(X_h-1)$. Offensichtlich ist dieser minimal, wenn jedes X_h den gleichen Anteil an der Summe hat. Somit folgt

$$\sum_{h\in H} X_h(X_h-1) \;\geq\; u(N/r-1)\left(\frac{u(N/r-1)}{N-1}-1\right). \tag{3.6}$$

Andererseits ist aber $X_h(X_h-1)$ gleich der Anzahl der geordneten Paare von Schlüsseln $x_1 \neq x_2$ aus der Menge $\{x \in U \mid h(x) = h_0(x)\}$. Es gilt demnach

$$\begin{aligned}\sum_{h\in H} X_h(X_h-1) &= \sum_{x_1\neq x_2\in U} \left|\{h \in H \mid h(x_1) = h_0(x_1) \wedge h(x_2) = h_0(x_2)\}\right| \\ &\leq u(u-1)(Nc/r^2-1). \end{aligned}\tag{3.7}$$

Zusammen mit Ungleichung (3.6) folgt also

$$u(u-1)\left(\frac{Nc}{r^2}-1\right) \;\geq\; u(N/r-1)\left(\frac{u(N/r-1)}{N-1}-1\right).$$

Nach Multiplikation beider Seiten mit $(N-1)r^2/u$ folgt hieraus

$$\begin{aligned} 0 &\leq (u-1)(Nc-r^2)(N-1) - r^2(N/r-1)\big(uN/r-u-N+1\big) \\ &= uN^2c - uNc - uNr^2 + ur^2 - N^2c + Nc + Nr^2 - r^2 \\ &\qquad -uN^2 + uNr + N^2r - Nr + uNr - ur^2 - Nr^2 + r^2 \\ &= N(uNc - uc - ur^2 - Nc + c - uN + ur + Nr - r + ur). \end{aligned}$$

Durch weiteres Umformen erhält man

$$\begin{aligned} N &\geq \frac{uc+ur^2-c-2ur+r}{uc-c-u+r} \;=\; 1+\frac{u(r^2-2r+1)}{r-c+u(c-1)} \\ &= 1+\frac{u(r-1)^2}{r-1+\delta(u-1)}. \end{aligned}$$

Somit ist (a) gezeigt.

Zu (b): Aus dem Beweis von (a) ist ersichtlich, daß die untere Schranke für $\left|\mathcal{H}\right|$ genau dann erreicht wird, wenn in den Ungleichungen (3.6) und (3.7) Gleichheit gilt. Für Ungleichung (3.6) bedeutet dies, daß jedes X_h den gleichen Wert hat. Da h_0 beliebig gewählt werden kann, folgt daraus, daß unter $\mathcal{H}^T$ alle Paare verschiedener Schlüssel mit gleicher Wahrscheinlichkeit kollidieren. Außerdem ist $\mathcal{H}$ nach Voraussetzung gleichverteilend, was nichts anderes bedeutet, als daß $\mathcal{H}^T$ konstante Korbgröße hat. Mit Lemma 2.2.3 ist also die Gleichheit in Ungleichung (3.6) äquivalent zur optimalen Universalität von $\mathcal{H}^T$.

Wir müssen abschließend noch die Äquivalenz von (b2) und der Gleichheit in Ungleichung (3.7) zeigen. Seien y_1, y_2 beliebige Werte aus R. Angenommen, es gibt Schlüssel $x_1 \neq x_2$, die mit einer Wahrscheinlichkeit größer als 0 aber kleiner als c/r^2 auf y_1 und y_2 abgebildet werden. Dann wählen wir für h_0 eine der Funktionen mit $h_0(x_1) = y_1$ und $h_0(x_2) = y_2$. Da per Voraussetzung alle anderen Schlüsselpaare höchstens mit einer Wahrscheinlichkeit von c/r^2 auf $h_0(x_1)$ und $h_0(x_2)$ abgebildet werden, folgt mit $H = \mathcal{H} \setminus \{h_0\}$

$$\sum_{x_1 \neq x_2 \in U} \left|\{h \in H \mid h(x_1) = h_0(x_1) \wedge h(x_2) = h_0(x_2)\}\right| < u(u-1)(Nc/r^2 - 1).$$

Also gilt in Ungleichung (3.7) keine Gleichheit. Abschließend nehmen wir an, daß keine Gleichheit in besagter Ungleichung vorliegt. Dann gibt es offensichtlich verschiedene Schlüssel $x_1 \neq x_2$, die auf $y_1 = h_0(x_1)$ und $y_2 = h_0(x_2)$ mit Wahrscheinlichkeit kleiner als c/r^2 abgebildet werden. Da die Schlüssel aber per Definition mindestens einmal (nämlich von h_0) auf y_1 und y_2 abgebildet werden, ist (b2) auch nicht erfüllt. ∎

Wir bezeichnen streng $(1+\delta)$-universelle Hashklassen der Kardinalität $N_{strg}(u, r, \delta)$ als *Stinson-minimal*. Bemerkenswert ist die Ähnlichkeit der Charakterisierung Stinson-minimaler streng c-universeller und Stinson-minimaler c-universeller Hashklassen. Insbesondere erhält man aus Stinson-minimalen streng 1-universellen Hashklassen durch Vertauschen von Schlüsseln und Hashfunktionen Stinson-minimale optimal universelle Hashklassen und umgekehrt.

3.1.6 Korollar. *Eine Stinson-minimale optimal universelle Hashklasse für u Schlüssel und einen Wertebereich R existiert genau dann, wenn es auch eine Stinson-minimale streng universelle Hashklasse für $(u-r)/(r-1)$ Schlüssel und den Wertebereich R gibt.*

Die hier vorgestellten unteren Schranken erlauben es uns, die Qualität von Hashklassen bezüglich ihrer Kardinalität zu beurteilen. Die Charakterisierungen Stinson-minimaler Hashklassen hat zusätzlich gezeigt, daß kleine universelle und kleine streng universelle Hashklassen Objekte mit ähnlicher Struktur zu sein scheinen. Dies wird auch im nächsten Abschnitt verdeutlicht, wo wir zeigen, wie man aus den dort noch zu definierenden abstandsuniversellen Hashklassen sowohl universelle als auch streng universelle konstruieren kann.

§ 2. Konstruktionsmethoden

Arithmetische Operationen wie Multiplikation, Division und Addition werden üblicherweise von der Computerhardware für „kleine“ ganze Zahlen (z.B. 32-bit) gut unterstützt. Deshalb ist beispielsweise die lineare Primzahlklasse für das Hashing mit kleinen Universen (z.B. $u \approx 2^{32}$) recht praktikabel. Allerdings wäre es ineffizient, sie in dieser Form für große Universen zu benutzen. Möchte man beispielsweise Schlüssel, die aus mehreren hundert Bytes

bestehen, auf einen kleinen Wertebereich - z.B. $R = \{0, \ldots, 2^{16} - 1\}$ - abbilden, so wäre zur Auswertung einer Hashfunktion die Multiplikation und Division sehr langer Zahlen erforderlich, obwohl die Hashwerte mit nur 2 Bytes beschrieben werden können. Wir werden uns daher mit Konstruktionsmethoden beschäftigen, die es erlauben, praktikable Hashklassen auch für große Universen (und große Wertebereiche) zu konstruieren. Dazu benötigen wir die Definition eines weiteren Typus von Hashklassen, der die Grundlage für die Konstruktion vieler universeller und streng universeller Hashklassen bildet.

Abstandsuniverselle Hashklassen

3.2.1 Definition. Sei R eine additive abelsche Gruppe und $\mathcal{H}$ eine Hashklasse $U \to R$. $\mathcal{H}$ heißt *c-abstandsuniversell*, wenn für beliebige verschiedene Schlüssel $x_1, x_2 \in U$ und ein beliebiges $d \in R$ gilt

$$\textbf{Prob}\left(h(x_1) - h(x_2) = d\right) \;\leq\; \frac{c}{r}.$$

Wir bezeichnen c als Universalitätsparameter. Eine 1-abstandsuniverselle Hashklasse heißt auch *abstandsuniversell*.

Obwohl abstandsuniverselle Hashklassen erst in den 90er Jahren explizit definiert wurden (eine Definition unter der Bezeichnung „Δ universal“ findet sich z.B. bei Stinson, 1996), haben sie bereits Wegman und Carter (1979) zur Konstruktion von Hashklassen mit großen Universen benutzt. Sie bilden auch heute noch die Grundlage für einige wichtige Konstruktionsmethoden, haben darüber hinaus aber auch Anwendungen in der Authentifizierung von Nachrichten gefunden (s. Krawczyk, 1994, 1995; Rogaway, 1995). Stinson (1996) schreibt ihnen zu, ein „important concept in their own right“ zu sein.

3.2.2 Bemerkung. Folgende Implikationen sind offensichtlich: Jede streng c-universelle Hashklasse ist auch c-abstandsuniversell; jede c-abstandsuniverselle Hashklasse ist auch c-universell.

Als Beispiel betrachten wir die homogenen Funktionen $h_a : \mathbb{K} \to \mathbb{K}$, $x \mapsto ax$ über einem endlichen Körper $\mathbb{K}$. Die Menge $\mathcal{H}^{hom}_{\mathbb{K},id}$, die aus den Funktionen h_a mit $a \in \mathbb{K}$ besteht, ist abstandsuniversell. Dies ergibt sich sofort aus den Körpereigenschaften, nach denen die Gleichung $ax - ax' = d$ für feste $x \neq x'$ und d aus $\mathbb{K}$ genau eine Lösung hat.

Das Universum von $\mathcal{H}^{hom}_{\mathbb{K},id}$ hat in diesem Fall die gleiche Ordnung wie der Wertebereich. Der folgende Satz zeigt, wie man Hashklassen mit kleineren Wertebereichen konstruieren kann. Für eine Funktion $f : X \to Y$ und eine Teilmenge $M \subseteq X$ bezeichne im folgenden $f(M)$ das *Bild* von M unter f, d.h. es sei $f(M) = \{f(y) \mid y \in Y\}$. Weiterhin sei die Operation $\circ$ die Verknüpfung zweier Funktionen $f : X \to Y$ und $g : Y \to Z$, also $g \circ f : X \to Z$ mit $(g \circ f)(x) = g\big(f(x)\big)$.

3.2.3 Satz. *Ist $\mathcal{H}$ eine c-abstandsuniverselle Hashklasse $U \to R$ und $f : R \to R'$ ein surjektiver Gruppenhomomorphismus, so ist die Familie der Funktionen $f \circ h$ mit $h \in \mathcal{H}$ auch c-abstandsuniversell.*

Beweis: Es sei $r = |R|$ und $r' = |R'|$. Wir betrachten zwei verschiedene $x, x' \in U$ und einen beliebigen Abstand $d \in R'$. Es ist allgemein bekannt, daß die Urbilder eines surjektiven Homomorphismus $\varphi : X \to Y$ genau die Nebenklassen des Kerns von φ sind, welcher selbst die Kardinalität $|X|/|\varphi(X)|$ hat (s. z.B. Scheja und Storch, 1994, S. 250). Da alle Nebenklassen die gleiche Kardinalität haben, folgt $|f^{-1}(d)| = r/r'$.

Angenommen, für ein zufällig gewähltes $h \in \mathcal{H}$ gilt nun $(f \circ h)(x) - (f \circ h)(x') = d$. Dann gilt wegen der Homomorphieeigenschaft von f auch $h(x) - h(x') \in f^{-1}(d)$. Da aber $f^{-1}(d)$ genau r/r' Abstände enthält, die von $h(x) - h(x')$ jeweils mit mit einer Wahrscheinlichkeit von höchstens c/r angenommen werden, ist die Wahrscheinlichkeit, daß x und x' unter $f \circ h$ den Abstand d haben, durch $(c/r) \cdot (r/r') = c/r'$ beschränkt. ∎

3.2.4 Beispiel. In Anlehnung an die lineare Körperklasse (s. Beispiel 2.3.3) definieren wir die *homogene Körperklasse* $\mathcal{H}^{hom}_{\mathbb{K},f}$ als die Menge der homogenen Funktionen über einen endlichen Körper $\mathbb{K}$ verknüpft mit einer Funktion f. D.h. $\mathcal{H}^{hom}_{\mathbb{K},f}$ besteht aus den Funktionen $x \mapsto f(ax)$ mit $a \in \mathbb{K}$. Dies ist offensichtlich die Menge der Funktionen aus $\mathcal{H}^{hom}_{\mathbb{K},id}$ verknüpft mit f, und somit gemäß obigem Satz abstandsuniversell, wenn f ein Homomorphismus bezüglich der additiven Gruppe von $\mathbb{K}$ darstellt.

Interessant für f ist z.B. die kanonische Projektion von $\mathbb{K}$ auf einen Unterkörper $\mathbb{K}'$. So lassen sich für beliebige $n \geq m$ abstandsuniverselle Hashklassen bilden, die n-Bit Wörter auf m-Bit Wörter abbilden, indem man die Körper der Ordnung 2^n bzw. 2^m benutzt.

Aus abstandsuniversellen Hashklassen können leicht streng c-universelle konstruiert werden. Dazu betrachten wir eine beliebige c-abstandsuniverselle Hashklasse $\mathcal{H}$ mit Universum U und Wertebereich R. Wir wählen $h \in \mathcal{H}$ und $z \in R$ zufällig, und betrachten die Funktion $f_{h,z} : U \to R$, die x auf $h(x) + z$ abbildet. Offensichtlich nehmen zwei verschiedene $x_1, x_2 \in U$ unter $f_{h,z}$ jeden Abstand $d \in R$ mit einer Wahrscheinlichkeit von höchstens c/r an. Andererseits wird x_1 unabhängig davon durch die Addition mit einem zufälligen z auf jeden Wert $y_1 \in U$ mit gleicher Wahrscheinlichkeit $1/r$ abgebildet. Also ist die Wahrscheinlichkeit, daß x_1 und x_2 auf zwei bestimmte Funktionswerte y_1 und y_2 abgebildet werden, durch c/r^2 beschränkt.

3.2.5 Satz. *Sei $\mathcal{H}$ eine c-abstandsuniverselle Hashklasse $U \to R$. Dann existiert eine gleichverteilende und streng c-universelle Hashklasse $U \to R$ mit $r|\mathcal{H}|$ Funktionen.* ∎

Obwohl viele streng universellen Hashklassen implizit oder explizit auf diesem Prinzip beruhen, wurde dieser Satz erstmals von Stinson (1996) notiert.

3.2.6 Beispiel. Ist $f : \mathbb{K} \to \mathbb{K}'$ ein Körperhomomorphismus zwischen endlichen Körpern, so konstruiert man mit dieser Methode unter Zuhilfenahme der homogenen Körperklasse $\mathcal{H}^{hom}_{\mathbb{K},f}$ (s. Beispiel 3.2.4) eine streng universelle Hashklasse $\mathbb{K} \to \mathbb{K}'$ mit $|\mathbb{K}| \cdot |\mathbb{K}'|$ Funktionen. Die resultierende Hashklasse ist also für $|\mathbb{K}'| < |\mathbb{K}|$ kleiner als die streng universelle Hashklasse $\mathcal{H}^{lin}_{\mathbb{K},f}$ aus Beispiel 2.3.3, die aus $|\mathbb{K}|^2$ Funktionen besteht.

Eine ganz ähnliche Methode wie die aus Satz 3.2.5 erlaubt die Konstruktion universeller Hashklassen aus abstandsuniversellen. Trotz ihrer Einfachheit wurde sie bisher in der Literatur noch nicht berücksichtigt.

3.2.7 Satz. *Sei $\mathcal{H}$ eine c-abstandsuniverselle Hashklasse $U \to R$. Dann existiert eine c-universelle Hashklasse $U \times R \to R$ der Kardinalität $|\mathcal{H}|$.*

Beweis: Für $h \in \mathcal{H}$ sei die Funktion $f_h : U \times R \to R$ gegeben durch $(x_1, x_2) \mapsto h(x_1) + x_2$. Offensichtlich ist die Familie der Funktionen f_h mit $h \in \mathcal{H}$ c-universell. Denn unterscheiden sich zwei verschiedene Schlüssel (x_1, x_2) und (x_1', x_2') in der ersten Komponente, so nimmt

$$f_h(x_1, x_2) - f_h(x_1', x_2') = h(x_1) - h(x_1') + x_2 - x_2'$$

wegen der c-abstandsuniversellen Eigenschaft von $\mathcal{H}$ jeden Wert (also insbesondere auch 0) mit einer Wahrscheinlichkeit von höchstens c/r an. Ist jedoch $x_1 = x_1'$, dann sind x_2 und x_2' verschieden, und mit $h(x_1) = h(x_1')$ folgt $f_h(x_1, x_2) \neq f_h(x_1', x_2')$ für alle $h \in \mathcal{H}$. ∎

Obwohl abstandsuniverselle Hashklassen schon universell sind (s. Bemerkung 3.2.2), hat diese Konstruktion den Vorteil, daß bei der resultierenden Hashklasse das Verhältnis der Universumsgröße zur Kardinalität besser als bei der ursprünglichen ist. Außerdem können mit einer ähnlichen Idee universelle Hashklassen konstruiert werden, deren Funktionen effizienter auswertbar sind, als die der analogen abstandsuniversellen Hashklassen.

Schließlich erhalten wir zusammen mit dem Ergebnis aus § 1 (Satz 3.1.2) sofort eine untere Schranke für die Kardinalität von c-abstandsuniversellen Hashklassen:

3.2.8 Korollar. *Sei $\mathcal{H}$ eine $(1+\delta)$-abstandsuniverselle Hashklasse, dann ist*

$$\left|\mathcal{H}\right| \;\geq\; r \cdot N_{univ}(u, r, \delta) \;=\; \frac{u(r-1)}{r-1+\delta(u/r-1)}.$$

Diese Schranke ist zumindest für $\delta > 0$ besser (und für $\delta = 0$ nicht schlechter) als die bisher größte untere Schranke

$$\left|\mathcal{H}\right| \geq \frac{u(r-1)}{r-1+\delta(u-1)},$$

die Stinson (1996) mithilfe von binären Codes bewiesen hat.

3.2.9 Bemerkung. Diese untere Schranke ist für $\delta = 0$ auch eine obere Schranke, wenn u und r Potenzen der gleichen Primzahl sind, denn dann gibt es Körper $\mathbb{K}$ und $\mathbb{K}'$ der Kardinalitäten u bzw. r, sowie einen Körperhomomorphismus $f : \mathbb{K} \to \mathbb{K}'$. Demnach ist die Hashklasse $\mathcal{H}^{hom}_{\mathbb{K},f}$ (s. Beispiel 3.2.4) abstandsuniversell und hat die kleinstmögliche Kardinalität $|\mathbb{K}| = u$. Gemäß Satz 3.2.7 läßt sich daraus auch eine 1-universelle Hashklasse $\mathbb{K} \times \mathbb{K}' \to \mathbb{K}'$ mit $|\mathbb{K}|$ Funktionen konstruieren; diese ist also Stinson-minimal.

Da jeder endliche Körper aber die Ordnung einer Primpotenz hat, erhalten wir auf diese Weise aus den Körperklassen nur Hashklassen, bei denen die Kardinalitäten von Universum und Wertebereich Potenzen der gleichen Primzahl sind. Tatsächlich sind bis heute keine Stinson-minimalen 1-universellen Hashklassen bekannt, bei denen das nicht der Fall ist.

Die oben erwähnten Konstruktionsmethoden belegen, daß abstandsuniverselle Hashklassen eine wichtige Grundlage für (streng) universelles Hashing sind. Im nächsten Abschnitt zeigen wir, daß sie sich auch zur Erweiterung des Universums und des Wertebereichs eignen.

Hashklassen für große Universen und Wertebereiche

Ausgehend von einer c-abstandsuniversellen Hashklasse $\mathcal{H}$ mit Universum U und Wertebereich R kann man leicht Hashklassen für große Universen konstruieren: Um Schlüssel aus U^n c-abstandsuniversell auf R abzubilden, wählen wir zufällig und unabhängig voneinander n Hashfunktionen $h_1,\ldots,h_n$ aus $\mathcal{H}$. Dann bilden wir $x_1,\ldots,x_n$ auf $h(x_1)+\ldots+h(x_n)$ ab. Es läßt sich leicht zeigen, daß eine Hashklasse, die das verwirklicht, auch c-abstandsuniversell ist (s. Wegman und Carter, 1979); ihre Kardinalität beträgt $\left|\mathcal{H}\right|^n$.

Allerdings läßt sich mit dieser Konstruktion der Wertebereich nicht vergrößern. Natürlich kann man, um einen Wertebereich R^m zu erreichen, wieder m Hashfunktionen $h_1,\ldots,h_m$ zufällig (und unabhängig) wählen, und $\big(h_1(x),\ldots,h_m(x)\big)$ als Hashwert berechnen. Die dazugehörige Hashklasse ist dann offensichtlich c^m-abstandsuniversell, hat aber den Nachteil, daß sie sehr viele Funktionen enthält. Erweitert man mit diesen Methoden das Universum U zu U^n und den Wertebereich R zu R^m (indem man beide Methoden hintereinander ausführt), so erhält man eine Hashklasse der Größe $\left|\mathcal{H}\right|^{nm}$.

Wir zeigen nun, wie man für $m \le n$ mit wesentlich weniger Funktionen auskommt.

3.2.10 Satz (Woelfel, 1999). *Sei $\mathcal{H}$ eine c-abstandsuniverselle Hashklasse $U \to R$. Dann gibt es für beliebige $1 \le m \le n$ eine c^m-abstandsuniverselle Hashklasse $U^n \to R^m$ der Kardinalität $\left|\mathcal{H}\right|^{n+m-1}$.*

Die Idee für die Konstruktion basiert auf der Faltung von Vektoren. Es bezeichne $\mathrm{conv}_{k,l}$ für $0 \le k \le l$ das Ergebnis der Faltung zweier Vektoren $\underline{x}$ und $\underline{y}$ über einem Ring R in den Spalten $k,\ldots,l$. Also sei für $\underline{x}=(x_0,x_1,\ldots)$ und $\underline{y}=(y_0,y_1,\ldots)$

$$\mathrm{conv}_{k,l}(\underline{x},\underline{y}) := (z_k,z_{k+1},\ldots,z_l) \quad \text{mit} \quad z_i = \sum_{j=0}^{i} x_j y_{i-j}.$$

Mansour, Nisan und Tiwari (1993) haben gezeigt, daß die Menge der Funktionen

$$h_{\underline{a}} : (\mathbb{Z}_2)^n \to (\mathbb{Z}_2)^m, \quad \underline{x} \mapsto \mathrm{conv}_{n-1,n+m-2}(\underline{a},\underline{x}) + \underline{b}$$

für $\underline{a} \in (\mathbb{Z}_2)^{n+m-1}$ und $\underline{b} \in (\mathbb{Z}_2)^m$ eine streng universelle Hashklasse ist („$+$“ bezeichne dabei die komponentenweise Addition im Ring $(\mathbb{Z}_2)^m$).

Wir benutzen nun eine Art Faltung von Hashfunktionen mit Schlüsseln; der Beweis zu Satz 3.2.10 ist dann aber völlig anders als der von Mansour et al. Sei $\underline{h}=(h_0,h_1,\ldots)$ ein Vektor von Hashfunktionen aus $\mathcal{H}$ und $\underline{x}=(x_0,x_1,\ldots)$ ein Vektor von Schlüsseln aus U. Wir definieren die Abbildung

$$\mathrm{conv}^*_{k,l}(\underline{h},\underline{x}) := (y_k,y_{k+1},\ldots,y_l) \quad \text{mit} \quad y_i = \sum_{j=0}^{i} h_j(x_{i-j}).$$

Beweis zu Satz 3.2.10: Sei $N=n+m-1$. Für $\underline{h} \in \mathcal{H}^N$ sei $f_{\underline{h}} : U^n \to R^m$ die Abbildung $\underline{x} \mapsto \mathrm{conv}^*_{n-1,N-1}(\underline{h},\underline{x})$. Wir beweisen, daß die Menge aller Funktionen $f_{\underline{h}}$ mit $\underline{h} \in \mathcal{H}^N$ eine c^m-abstandsuniverselle Hashklasse bildet. Dazu betrachten wir zwei beliebige verschiedene Schlüssel $\underline{x}=(x_0,\ldots,x_{n-1})$ und $\underline{x}'=(x'_0,\ldots,x'_{n-1})$. Es sei $\underline{h}=(h_0,\ldots,h_{N-1})$ zufällig aus $\mathcal{H}^N$ gewählt und

$$\underline{d} = (d_{n-1},\ldots,d_{N-1}) = f_{\underline{h}}(x) - f_{\underline{h}}(x').$$

Wir zeigen, daß d_i ($n-1 \leq i < N$) unabhängig von $d_{n-1},\ldots,d_{i-1}$ jeden Wert aus R mit einer Wahrscheinlichkeit von höchstens c/r annimmt. Damit sind dann alle d_i unabhängig voneinander, und d nimmt jeden Wert mit einer Wahrscheinlichkeit von höchstens $(c/r)^m$ an, woraus die Aussage des Satzes folgt.

O.B.d.A. werden $h_0,\ldots,h_{N-1}$ in dieser Reihenfolge zufällig gewählt. Sei t der kleinste Index, in dem sich x_t und x'_t unterscheiden. Dann hängt d_i nur von den Funktionen $h_0,\ldots,h_{i-t}$ ab, da für $j > i-t$ offensichtlich $h_j(x_{i-j}) - h_j(x'_{i-j}) = 0$ ist. Da $h_0,\ldots,h_{i-t-1}$ vor h_{i-t} gewählt wurden, sind bei der zufälligen Wahl von h_{i-t} bereits die Werte $d_{n-1},\ldots,d_{i-1}$ fest. Da aber $h_{i-t}(x_t) - h_{i-t}(x'_t)$ als Summand zum Wert von d_i beiträgt, x_t und x'_t verschieden sind, und h_{i-t} aus einer c-abstandsuniversellen Hashklasse zufällig gewählt wurde, nimmt d_i (unabhängig von $d_{n-1},\ldots,d_{i-1}$) jeden Wert mit einer Wahrscheinlichkeit von höchstens c/r an. ∎

Für $m = 1$ entspricht diese Konstruktion offensichtlich der Methode von Wegman und Carter (1979), die wir oben erwähnt haben.

Es lassen sich nun mithilfe der Faltung von Vektoren über einem endlichen Körper $\mathbb{K}$ universelle, streng universelle und sogar optimal universelle Hashklassen bilden, die wir *Faltungsklassen* nennen.

3.2.11 Satz. *Sei $\mathbb{K}$ ein endlicher Körper.*

(a) Die Klasse der Funktionen $f_{\underline{a}} : \mathbb{K}^n \to \mathbb{K}^m$, $\underline{x} \mapsto \mathrm{conv}_{n-1,n+m-2}(\underline{a},\underline{x})$ mit $\underline{a} \in \mathbb{K}^{n+m-1}$ ist abstandsuniversell.

(b) Sei $f_{\underline{a}} : \mathbb{K}^n \to \mathbb{K}^m$ definiert wie in Teil (a). Die Klasse der Funktionen $f_{\underline{a},\underline{b}} : \mathbb{K}^n \to \mathbb{K}^m$, $\underline{x} \mapsto f_{\underline{a}}(x) + \underline{b}$ mit $\underline{a} \in \mathbb{K}^{n+m-1}$ und $\underline{b} \in \mathbb{K}^m$ ist streng universell.

(c) Die Klasse der Funktionen $h_{\underline{a}} : \mathbb{K}^n \to \mathbb{K}^m$, $\underline{x} \mapsto \mathrm{conv}_{n-m,n-1}(\underline{a},\underline{x})$ mit allen Werten $\underline{a} = (1,a_1,\ldots,a_{n-1}) \in \mathbb{K}^n$ ist universell.

(d) Sei $h_{\underline{a}} : \mathbb{K}^n \to \mathbb{K}^m$ definiert wie in Teil (c). Ist m ein Teiler von n, dann gibt es eine Teilmenge $A \subseteq \mathbb{K}^n$, so daß die Klasse der Funktionen $h_{\underline{a}}$ mit $\underline{a} \in A$ optimal universell ist.

Teil (b) des Satzes (strenge Universalität) ist allgemein bekannt (vgl. Dietzfelbinger, 1996), und entspricht für $\mathbb{K} = \mathbb{Z}_2$ der oben erwähnten Aussage von Mansour et al. Neu ist jedoch die Beweisführung, die darauf basiert, daß zunächst in Teil (a) mit den obigen Konstruktionsmethoden eine abstandsuniverselle Hashklasse konstruiert wird. Bemerkenswert sind auch die neuen Ergebnisse (c) und (d). Gemäß Teil (c) erhält man eine 1-universelle Hashklasse $\mathbb{K}^n \to \mathbb{K}^m$, deren Funktionen von der Faltung zweier Vektoren nur die Spalten $n-m,\ldots,n-1$ berechnen müssen. Teil (d) beschreibt eine optimal universelle Hashklasse, deren Funktionen fast genauso aufgebaut sind wie die der 1-universellen aus Teil (c). Da diese Konstruktion auf Methoden beruht, die wir erst in § 3 entwickeln, erfolgt der Beweis auch erst dort (s. Satz 3.3.17).

Beweis zu Satz 3.2.11 (a)–(c): Teil (a) ist nichts anderes als die Konstruktion aus Satz 3.2.10 mithilfe der homogenen Körperklasse $\mathcal{H}^{hom}_{\mathbb{K},id}$ (s. S. 23). Teil (b) entspricht dann der daraus resultierenden Konstruktion gemäß Satz 3.2.5. Es muß also nur noch (c) gezeigt werden. Dazu betrachten wir analog zum Beweis von Satz 3.2.10 zwei verschiedene Schlüssel $\underline{x},\underline{x}' \in \mathbb{K}^n$, den kleinsten Index t in dem sich x_t und x'_t unterscheiden, sowie

$\underline{d} = (d_{n-m}, \ldots, d_{n-1})$ als Differenz der Funktionswerte einer zufällig gewählten Hashfunktion $h_{\underline{a}}$. Ist $t < n-m$, dann können wir völlig analog zum obigen Beweis fortfahren, und es folgt sogar, daß $\underline{d}$ jeden Abstand mit gleicher Wahrscheinlichkeit annimmt. Sei also $t \geq n-m$. Es folgt $d_t = a_0 x_t - a_0 x'_t = x_t - x'_t \neq 0$. Also kollidieren dann $\underline{x}$ und $\underline{x}'$ unter keiner der Hashfunktionen. ∎

Benutzt man als Körper den Restklassenring $\mathbb{Z}_p$ für eine Primzahl p, so erhält man Hashklassen, deren Funktionen mit ganzzahliger Arithmetik ausgewertet werden können. Abgesehen von der direkten Anwendung haben die Faltungsklassen aber noch eine andere Bedeutung für das Hashing mit ganzzahliger Arithmetik. Betrachtet man die Multiplikation im Ring $\mathbb{Z}_{2^k}$, so läßt sich diese auch als Faltung über $(\mathbb{Z}_2)^k$ auffassen, bei der zusätzlich ein Übertrag zu berücksichtigen ist. Diese Idee bildet in Kapitel 4 die Grundlage für die Konstruktion von universellen Hashklassen, die hauptsächlich aus linearen Funktionen über $\mathbb{Z}_{2^k}$ bestehen. Die Konstruktionsmethode aus Satz 3.2.10 läßt sich auf diese Hashklassen dann besonders effizient anwenden.

§ 3. Konstruktion optimal universeller Hashklassen

Wie bereits erwähnt, ist die Äquivalenz von optimal universellen Hashklassen zu RBIBDs seit der Arbeit von Stinson (1994a) bekannt (s. auch S. 32f.). Obwohl RBIBDs gut erforschte kombinatorische Objekte sind, für die es zahlreiche Existenzaussagen gibt, findet man in der Literatur kaum explizite Konstruktionen von Hashklassen mit optimalem Universalitätsparameter. Wir werden hier eine Technik aufzeigen, mit deren Hilfe man solche Hashklassen aus abstandsuniversellen erhält.

Dazu definieren wir zunächst neue, eingeschränkte c-universelle Hashklassen, die wir $(0|c)$-universell nennen. Dann werden wir deren kombinatorische Eigenschaften untersuchen und dabei feststellen, daß insbesondere $(0|1)$-universelle Hashklassen eine wichtige Bedeutung haben. Diese wird besonders durch die Beziehung zu wichtigen kombinatorischen Objekten, nämlich gruppenteilbaren Designs verdeutlicht. Schließlich entwickeln wir eine allgemeine Methode zur Konstruktion optimal universeller Hashklassen aus abstandsuniversellen.

$(\mathbf{0}|\mathbf{c})$-universelle Hashklassen

Wir erinnern noch mal an Satz 3.2.7. Um aus einer c-abstandsuniversellen Hashklasse $\mathcal{H}$ mit Universum U und Wertebereich R eine c-universelle Hashklasse $U \times R \to R$ zu konstruieren, haben wir die Menge der Funktionen $f_h : U \times R \to R$, $(x,z) \mapsto h(x) + z$ mit $h \in \mathcal{H}$ betrachtet. Der Beweis zu Satz 3.2.7 hat gezeigt, daß zwei verschiedene Schlüssel (x_1, z_1) und (x_2, z_2) höchstens dann kollidieren, wenn sie sich in der x-Komponente unterscheiden. Wir können die so konstruierten Hashklassen genauer mit der folgenden Definition beschreiben.

3.3.1 Definition. Eine Hashklasse $U \to R$ heißt $(0|c)$-*universell*, wenn es eine Äquivalenzrelation $\sim$ gibt, die U in Äquivalenzklassen gleicher Größe unterteilt, so daß für alle verschiedenen Schlüssel $x_1, x_2 \in U$ gilt

$$\mathbf{Prob}\left(h(x_1) = h(x_2)\right) \;\leq\; \begin{cases} 0 & \text{falls } x_1 \sim x_2, \text{ und} \\ c/r & \text{sonst.} \end{cases}$$

Kann man dabei sogar „$\leq$“ durch „$=$“ ersetzten, so heißt die Hashklasse *exakt* $(0|c)$-universell.

Bezeichnen wir das Universum einer $(0|c)$-universellen Hashklasse als $U_1 \times U_2$, so meinen wir damit, daß die dazugehörige Äquivalenzrelation $\sim$ auf $U_1 \times U_2$ wie folgt definiert ist: Für zwei Schlüssel (x_1, x_2) und (x_1', x_2') aus $U_1 \times U_2$ gilt $(x_1, x_2) \sim (x_1', x_2')$ genau dann, wenn $x_1 = x_1'$ ist. Die Anzahl der Äquivalenzklassen ist also durch $\left|U_1\right|$ und ihre Größe durch $\left|U_2\right|$ gegeben. Im folgenden sei immer $u_1 = \left|U_1\right|$ und $u_2 = \left|U_2\right|$.

3.3.2 Bemerkung. Die Konstruktion aus Satz 3.2.7 liefert tatsächlich $(0|c)$-universelle Hashklassen $U \times R \to R$ (dies ist direkt aus dem dazugehörigen Beweis ersichtlich). Aus einer 1-abstandsuniversellen Hashklasse entsteht auf diese Weise sogar eine exakt $(0|1)$-universelle.

Einige $(0|c)$-universelle Hashklassen wurden bereits vorgestellt. So sind optimal universelle Hashklassen gemäß Lemma 2.2.3 genau die exakt $(0|c)$-universellen Hashklassen mit $c = (u-r)/(u-1)$. Die Äquivalenzklassen enthalten dann jeweils ein Element. Auch streng universelle Hashklassen sind exakt $(0|1)$-universell mit einelementigen Äquivalenzklassen.

3.3.3 Bemerkung. Die 1-universelle Faltungsklasse aus Satz 3.2.11 (c) ist tatsächlich sogar $(0|1)$-universell: Im dazugehörigen Beweis wurde gezeigt, daß zwei Vektoren aus $\mathbb{K}^n$ nicht miteinander kollidieren, wenn sie sich in den ersten $n-m$ Spalten nicht unterscheiden. Andernfalls kollidieren sie genau mit einer Wahrscheinlichkeit von $1/|\mathbb{K}^m|$. Es handelt sich also um eine exakt $(0|1)$-universelle Hashklasse $\mathbb{K}^{n-m} \times \mathbb{K}^m \to \mathbb{K}^m$.

Eine weitere interessante Beziehung zwischen 1-universellen und $(0|1)$-universellen Hashklassen liefert der folgende Satz:

3.3.4 Satz. *Jede Stinson-minimale* 1*-universelle Hashklasse ist exakt* $(0|1)$*-universell.*

Ein wesentlicher Teil dieses Satzes ist bereits durch die Charakterisierung Stinson-minimaler Hashklassen (s. Satz 3.1.2) bewiesen: Zwei verschiedene Schlüssel kollidieren entweder gar nicht oder genau mit Wahrscheinlichkeit c/r. Es muß also nur noch die Existenz einer geeigneten Äquivalenzrelation über U gezeigt werden.

Beweis zu Satz 3.3.4: Sei $\mathcal{H}$ die Stinson-minimale Hashklasse mit Universum U und Wertebereich R. Es genügt zu zeigen, daß

$$x_1 \sim x_2 \quad :\Longleftrightarrow \quad \left(x_1 = x_2 \;\vee\; \delta_{\mathcal{H}}(x_1, x_2) = 0\right)$$

eine Äquivalenzrelation auf U definiert, unter der alle Äquivalenzklassen gleiche Größe haben. Symmetrie und Reflexivität der Relation $\sim$ sind trivialerweise erfüllt. Bleibt noch die Transitivität zu zeigen. Angenommen, es gilt $x_1 \sim x_2$ und $x_2 \sim x_3$, aber nicht $x_1 \sim x_3$. Dann

gibt es eine Hashfunktion h_0 und einen Hashwert y_0 mit $h_0(x_1) = h_0(x_3) = y_0$. Mit Lemma 3.1.3 gibt es $u/r - 1$ Schlüssel in ${h_0}^{-1}(y_0)$, die mit x_2 kollidieren. Da aber $\mathcal{H}^T$ streng universell (Satz 3.1.2) und somit gleichverteilend ist (Bemerkung 2.3.2), hat $\mathcal{H}$ konstante Korbgröße. Mit anderen Worten $\left|{h_0}^{-1}(y_0)\right| = u/r$. Also kann es höchstens ein Element in dieser Menge geben, das nicht mit x_2 kollidiert. Dies steht im Widerspruch zur Annahme, nach der sowohl x_1 als auch x_3 nicht mit x_2 kollidieren.

Schließlich zeigt folgende Überlegung, daß alle Äquivalenzklassen die gleiche Größe r haben: Wir betrachten ein Element x_0 einer beliebigen Äquivalenzklasse $[x_0]$, sowie eine beliebige Funktion $h_0 \in \mathcal{H}$. Sei $y_0 = h_0(x_0)$. Nach Lemma 3.1.3 befinden sich in jedem Korb ${h_0}^{-1}(y)$ mit $y \neq y_0$ genau $u/r - 1$ Schlüssel, die mit x_0 kollidieren. Somit gibt es in jedem solchen Korb genau einen, und demnach insgesamt $r - 1$ Schlüssel, die nicht mit x_0 kollidieren. Es folgt, daß die Äquivalenzklasse $[x_0]$ genau r Elemente enthält. ∎

Gruppenteilbare Designs

Im letzten Abschnitt wurden einige Beziehungen zwischen den dort neu definierten $(0|c)$-universellen Hashklassen und den bekannten Hashklassen aus Kapitel 2 aufgezeigt. Die Definition der $(0|c)$-universellen Hashklassen ist insbesondere auch deswegen sinnvoll, weil sie äquivalent zu gut erforschten kombinatorischen Objekten, nämlich auflösbar gruppenteilbaren Designs sind. Die Notation der folgenden Definitionen orientiert sich an dem Handbuch von Colbourn und Dinitz (1996) sowie dem Lehrbuch von Beth, Jungnickel und Lenz (1999).

3.3.5 Definition. Eine *Inzidenzstruktur* ist ein Paar $\mathbf{D} = (V, \mathbf{B})$, wobei V eine Menge ist, deren Elemente als *Punkte* bezeichnet werden. $\mathbf{B}$ ist eine Familie von Teilmengen von V; die Elemente von $\mathbf{B}$ heißen *Blocks*. Stellt eine Teilmenge aus $\mathbf{B}$ eine Partitionierung der Punkte dar, so heißt sie *parallele Klasse*. Eine Inzidenzstruktur heißt *auflösbar*, wenn es eine Partitionierung von $\mathbf{B}$ in parallele Klassen gibt. Sie heißt *affin auflösbar*, wenn zudem alle Paare von Blocks aus verschiedenen parallelen Klassen die gleiche Anzahl von Punkten gemeinsam haben.

Ist ein Punkt $p \in V$ Element eines Blocks $B \in \mathbf{B}$, so sagt man, „der Punkt p liegt auf dem Block B“, oder „p und B sind inzident“. Wir sagen, „zwei Punkte p_1 und p_2 schneiden sich in einem Block B“, wenn sowohl p_1 als auch p_2 auf B liegen.

3.3.6 Definition. Ein *gruppenteilbares Design* (kurz: GDD) mit Parametern (v, k, g, λ) ist eine Inzidenzstruktur $(V, \mathbf{B})$ mit folgenden Eigenschaften:

(a) $|V| = v$.

(b) Alle Blocks aus $\mathbf{B}$ haben die Kardinalität k.

(c) Es gibt eine Partitionierung der Punkte in *Gruppen* der Kardinalität g, so daß sich zwei Punkte aus der gleichen Gruppe in keinem Block, und zwei Punkte verschiedener Gruppen in genau λ Blocks schneiden.

Gruppenteilbare Designs mit Parametern (v, k, g, λ) bezeichnen wir als $GD_\lambda[k, g; v]$. Besonders wichtig sind für uns auflösbare GDDs (kurz: RGDDs). Ein auflösbares $GD_\lambda[k, g; v]$ bezeichnen wir auch als $RGD_\lambda[k, g; v]$.

Wir können nun die Beziehung zu Hashklassen herstellen:

3.3.7 Satz. *Gibt es ein $RGD_\lambda[k,g;v]$ mit N parallelen Klassen, so gibt es auch eine exakt $(0|c)$-universelle Hashklasse $U_1 \times U_2 \to R$ mit konstanter Korbgröße und Kardinalität N, für die gilt:*

$$u_1 = v/g, \quad u_2 = g, \quad r = v/k, \quad c = \frac{v\lambda}{kN}.$$

Gibt es umgekehrt eine exakt $(0|c)$-universelle Hashklasse $U_1 \times U_2 \to R$ mit konstanter Korbgröße und Kardinalität N, dann gibt es auch ein $RGD_\lambda[k,g;v]$ mit N parallelen Klassen, für das gilt:

$$k = \frac{u_1 u_2}{r}, \quad g = u_2, \quad v = u_1 u_2, \quad \lambda = \frac{Nc}{r}.$$

Beweis: Wir konstruieren die Hashklasse aus dem als Inzidenzstruktur $(V,\mathbf{B})$ gegebenen $RGD_\lambda[k,g;v]$, indem wir jeder parallelen Klasse des RGDD eine Hashfunktion zuordnen. Seien also $P_1,\ldots,P_N$ die N parallelen Klassen, und für $1 \le i \le N$ seien $B_{i,1},\ldots,B_{i,v/k}$ die Blocks von P_i (da auf jedem Block k Punkte liegen, besteht eine parallele Klasse aus genau v/k Blocks). Die Hashklasse $\mathcal{H}$ sei dann die Familie der Hashfunktionen $h_1,\ldots,h_N$ mit

$$h_i : V \to \{1,\ldots,v/k\}, \quad x \mapsto j \text{ falls } x \in B_{i,j}.$$

Diese Abbildungen sind eindeutig definiert, weil jeder Punkt aus V nur auf genau einem der Blocks $B_{i,j}$ ($1 \le j \le v/k$) aus P_i liegt. Zudem hat die Hashklasse offensichtlich konstante Korbgröße.

Zur Bestimmung der Mengen U_1 und U_2 nehmen wir nun o.B.d.A. an, daß die Punkte aus V wie folgt bezeichnet sind: Jede der v/g Gruppen wird mit einer eindeutigen Zahl aus $\{1,\ldots,v/g\}$ numeriert, und den Punkten der i-ten Gruppe werden eindeutig die Tupel $(i,1),\ldots,(i,g)$ zugeordnet. Für $U_1 = \{1,\ldots,v/g\}$ und $U_2 = \{1,\ldots,g\}$ ist also $V = U_1 \times U_2$. Da zwei verschiedene Punkte der gleichen Gruppe (also (i,j) und (i,j') mit $j \neq j'$) sich in keinem Block schneiden, kollidieren die dazugehörigen Schlüssel also auch unter keiner Funktion aus $\mathcal{H}$. Andererseits schneiden sich zwei Punkte verschiedener Gruppen in genau λ Blocks, die per Definition natürlich alle aus verschiedenen parallelen Klassen stammen. Somit gibt es genau λ Funktionen in $\mathcal{H}$, in denen die dazugehörigen Schlüssel kollidieren. $\mathcal{H}$ ist also eine exakt $(0|c)$-universelle Hashklasse $U_1 \times U_2 \to R$, mit $u_1 = v/g$, $u_2 = g$ und $r = v/k$.

Es muß abschließend nur noch c bestimmt werden. Da sich zwei Punkte verschiedener Gruppen in genau λ Blocks schneiden, ist die Kollisionswahrscheinlichkeit der dazugehörigen Schlüssel durch λ/N gegeben. Der Universalitätsparameter berechnet sich also zu $c = (\lambda/N)\cdot r$, was mit $r = v/k$ die Behauptung ergibt.

Es ist leicht nachzuvollziehen, daß sich diese Konstruktion auch umkehren läßt. Man erhält dann aus einer Hashklasse ein RGDD mit den in der Behauptung angegebenen Parametern. ∎

Gruppenteilbare Designs wurden zuerst von Bose und Connor (1952) untersucht. Inzwischen sind zahlreiche notwendige und hinreichende Bedingungen für die Existenz von GDDs und RGDDs bekannt (für einen umfassenden Überblick siehe Colbourn und Dinitz, 1996). Folgende, allgemein bekannte Aussage (s. z.B. Beth, Jungnickel und Lenz, 1999, S. 37) läßt uns auf den Universalitätsparameter einer exakt $(0|c)$-universellen Hashklasse mit konstanter Korbgröße folgern.

3.3.8 Aussage. *Jedes $GD_\lambda[k,g;v]$ besteht aus genau $b = \lambda v(v-g)/\big(k(k-1)\big)$ Blocks.*

Da sich in jeder parallelen Klasse eines $RGD_\lambda[k,g;v]$ genau v/k Blocks befinden, enthält das RGDD also genau $b/(v/k) = \lambda(v-g)/(k-1)$ parallele Klassen. Die Kardinalität der diesem RGDD nach Satz 3.3.7 entsprechenden Hashklasse beträgt also $N = \lambda(v-g)/(k-1)$. Mit $\lambda = Nc/r$, folgt

$$N = \frac{Nc(v-g)}{r(k-1)}, \quad \text{bzw.} \quad c = \frac{r(k-1)}{v-g} = \frac{r(u_1u_2/r-1)}{u_1u_2-u_2} = \frac{u_1-r/u_2}{u_1-1}.$$

3.3.9 Korollar. *Für jede exakt $(0|c)$-universelle Hashklasse $U_1 \times U_2 \to R$ mit konstanter Korbgröße gilt $c = (u_1 - r/u_2)/(u_1-1)$.*

Ein direkter Beweis dieses Korollars ist natürlich auch möglich, und zwar auf eine analoge Weise wie beim Beweis zu Satz 2.2.1. Ähnlich wie in Lemma 2.2.3 kann man so auch die Umkehrung des Korollars erhalten, d.h. jede $(0|c)$-universelle Hashklasse $U_1 \times U_2 \to R$ mit $c = (u_1 - r/u_2)/(u_1-1)$ ist sogar exakt $(0|c)$-universell und hat konstante Korbgröße. Da diese Aussage jedoch im Folgenden nicht weiter benötigt wird, soll der Beweis (der aufgrund der angesprochenen Analogie zu früheren Beweisen reine Routine ist), hier nicht geführt werden.

Optimal und exakt $(0|1)$-universelle Hashklassen mit konstanter Korbgröße sind besonders interessante kombinatorische Objekte. Dies wird auch daran deutlich, daß die entsprechenden gruppenteilbaren Designs spezielle Bezeichnungen und Eigenschaften haben.

3.3.10 Definition.

(a) Ein *transverselles Design* (kurz: TD) mit Parametern (v,k,g,λ) ist ein $GD_\lambda[k,g;v]$, bei dem sich jeder Block und jede Gruppe in genau einem Punkt schneiden.

(b) Ein *balanciert unvollständiges Blockdesign* (kurz: BIBD) mit Parametern (v,k,λ) ist ein $GD_\lambda[k,1;v]$.

Jeder Punkt eines TD mit Parametern (v,k,g,λ) ist offensichtlich eindeutig durch den Schnitt einer Gruppe mit einem Block bestimmt. Also hat das TD genau kg Punkte. Wir bezeichnen daher ein TD mit diesen Parametern als $TD_\lambda[k,g]$, und ein auflösbares $TD_\lambda[k,g]$ als $RTD_\lambda[k,g]$. Analog wird ein (auflösbares) BIBD mit Parametern (v,k,λ) als $BIBD_\lambda[k;v]$ bzw. $RBIBD_\lambda[k;v]$ bezeichnet.

Sowohl BIBDs als auch TDs sind seit langem bekannt und gut erforscht, so daß ein Zusammenhang zu universellen Hashklassen neue Erkentnisse über deren kombinatorische Eigenschaften erlaubt.

3.3.11 Korollar. *Eine exakt $(0|1)$-universelle Hashklasse $U \to R$ der Kardinalität N und mit konstanter Korbgröße ist äquivalent zu einem $RTD_\lambda[k,g]$ mit $k = u/r$, $g = r$ und $\lambda = N/r$. Insbesondere ist eine Stinson-minimale 1-universelle Hashklasse äquivalent zu einem $RTD_\lambda[k,g]$ bei dem zusätzlich $k = \lambda g$ gilt.*

Beweis: Der erste Teil ergibt sich unmittelbar aus Satz 3.3.7, wenn man berücksichtigt, daß in dem RTD $v = kg$, also in der Hashklasse $u_1 = u/r$ und $u_2 = r$ gilt. Eine Stinson-minimale 1-universelle Hashklasse $\mathcal{H}$ hat nach Satz 3.1.2 (2a) konstante Korbgröße (dies impliziert die Gleichverteiltheit von $(\mathcal{H})^T$). Aus Satz 3.3.4 folgt dann sofort mit dem bereits bewiesenen ersten Teil des Satzes, daß eine Stinson-minimale 1-universelle Hashklasse ein RTD ist. Da die Hashklasse per Definition aus u/r Funktionen besteht, folgt $k = \lambda g$ durch Einsetzen der

Parameter. Andererseits ist nach dem ersten Teil des Satzes ein $RTD_\lambda[k,g]$ äquivalent zu einer $(0|1)$-universellen Hashklasse, die für $k = \lambda g$ aus $u_1 = u/r$ Funktionen besteht. Diese ist wegen $N_{univ}(u,r,0) = u/r$ also Stinson-minimal. ∎

Interessanterweise haben also exakt $(0|1)$-universelle Hashklassen mit konstanter Korbgröße eine Äquivalenzklassengröße von genau r.

Nach einer wichtigen Ungleichung von Bose (1942) gilt für jedes $RBIBD_\lambda[k;v]$ mit b Blocks

$$b \;\geq\; (v-1)\left(1+\frac{\lambda}{k-1}\right), \tag{3.8}$$

und Gleichheit genau dann, wenn das BIBD affin auflösbar ist.

Durch eine einfache Anwendung der Sätze 3.3.7 und 3.1.2 erhält man als Folgerung die Äquivalenz von (affin) auflösbaren BIBDs und (Stinson-minimalen) optimal universellen Hashklassen. Diese Aussagen wurden bereits von Stinson (1994a) durch eine direkte Beweisführung gezeigt.

3.3.12 Korollar. *Eine optimal universelle Hashklasse $U \to R$ mit N Hashfunktionen ist äquivalent zu einem $RBIBD_\lambda[k;v]$ mit*

$$v = u, \quad k = u/r \quad \textit{und} \quad \lambda = \frac{N(u-r)}{r(u-1)}.$$

Die Hashklasse ist genau dann Stinson-minimal, wenn das äquivalente RBIBD affin auflösbar ist.

Beweis: Wie bereits auf Seite 29 erwähnt sind optimal universelle Hashklassen tatsächlich exakt $(0|c)$-universell mit $c = (u-r)/(u-1)$ und einelementigen Äquivalenzklassen. Da sie gemäß Lemma 2.2.3 auch eine konstante Korbgröße haben, folgt aus Satz 3.3.7 die Äquivalenz zu einem $RBIBD_\lambda[k;v]$ mit den angegebenen Parametern.

Weiterhin erhält man aus Aussage 3.3.8 (mit $g = 1$) für die Anzahl b der Blocks des RBIBD

$$b \;=\; \lambda\frac{v(v-1)}{k(k-1)} \;=\; (v-1)\frac{\lambda v/k}{k-1}.$$

Also ist das BIBD genau dann affin auflösbar, d.h. es gilt in Boses Ungleichung (3.8) Gleichheit, wenn $\lambda v/k = k-1+\lambda$ bzw. $\lambda = (k-1)/(v/k-1)$ ist. Die äquivalente Hashklasse hingegen ist gemäß Satz 3.1.2 genau dann Stinson-minimal, wenn $N = (u-1)/(r-1)$ gilt, was nach Einsetzen der Parameter des BIBDs

$$\lambda \;=\; \frac{u-r}{r(r-1)} \;=\; \frac{v-v/k}{v/k(v/k-1)} \;=\; \frac{k-1}{v/k-1}$$

ergibt. ∎

Verbesserung von Universalitätsparametern

Da gruppenteilbare Designs bereits intensiv erforscht wurden, sind zahlreiche Konstruktionsmethoden bekannt. Wir können nun versuchen, die Ideen dieser Methoden auf Hashklassen zu übertragen. Eine äußerst wichtige Methode von Wilson (1972) benutzt etwas allgemeiner definierte GDDs, um daraus neue GDDs und insbesondere BIBDs zu konstruieren. Die Beschreibung der Konstruktionsmethode würde den Rahmen dieser Diplomarbeit sprengen; die im folgenden beschriebene Technik zur Verbesserung von Universalitätsparametern beruht aber auf einer ähnlichen Idee.

Wir betrachten eine $(0|c_1)$-universelle Hashklasse $\mathcal{H}_1$ mit Universum $U_1 \times U_2$ und Wertebereich R, wobei $u_2 \geq r$ ist. Alle Paare verschiedener Schlüssel (x_1,x_2) und (x_1',x_2') aus $U_1 \times U_2$, die sich in der ersten Komponente nicht unterscheiden, d.h. für die $x_1 = x_1'$ gilt, haben eine Kollisionswahrscheinlichkeit von 0. Die Idee der folgenden Konstruktion ist es, den Universalitätsparameter dadurch zu verbessern, daß man Funktionen hinzufügt, unter der diese Schlüsselpaare öfters als alle anderen kollidieren.

Angenommen, es gibt eine c_2-universelle Hashklasse $\mathcal{F}$ mit Universum U_1 und Wertebereich R, wobei $c_2 < c_1$ ist (da das Universum von $\mathcal{F}$ kleiner als das von $\mathcal{H}_1$ ist, können wir auch auf einen kleineren Universalitätsparameter hoffen - schließlich ist ja beispielsweise der optimale Universalitätsparameter streng monoton steigend in u). Aus dieser können wir leicht eine Hashklasse $\mathcal{H}_2$ mit Universum $U_1 \times U_2$ konstruieren, indem wir für jedes $f \in \mathcal{F}$ eine Abbildung $h_f : U_1 \times U_2, (x_1,x_2) \mapsto f(x_1)$ bilden. Die Kollisionswahrscheinlichkeit zweier verschiedener Schlüssel (x_1,x_2) und (x_1',x_2') ist dann bei der Hashklasse $\mathcal{H}_2$ offensichtlich durch c_2/r beschränkt, wenn $x_1 \neq x_1'$ ist. Andernfalls ist die Kollisionswahrscheinlichkeit 1, d.h. die Schlüssel kollidieren in jedem Fall. Durch Vereinigung von Vielfachen der Hashklassen $\mathcal{H}_1$ und $\mathcal{H}_2$ können wir dann insgesamt eine Hashklasse mit Universum $U_1 \times U_2$ erzeugen, deren Universalitätsparameter kleiner als c_1 ist.

Dazu betrachten wir folgendes Zufallsexperiment: Zunächst entscheiden wir uns mit Wahrscheinlichkeit ε ($0 \leq \varepsilon \leq 1$) für die Familie $\mathcal{H}_2$ und mit Wahrscheinlichkeit $1-\varepsilon$ für die Familie $\mathcal{H}_1$. Anschließend ziehen wir zufällig (gemäß der Gleichverteilung) eine Hashfunktion h aus der gewählten Hashklasse. Die Wahrscheinlichkeit, daß zwei beliebige verschiedene Schlüssel (x_1,x_2) und (x_1',x_2') unter h kollidieren, läßt sich leicht mit folgender Fallunterscheidung bestimmen:

1. Fall: $x_1 = x_1'$. Dann unterscheiden sich die Schlüssel in der zweiten Komponente, d.h. es gilt $x_2 \neq x_2'$. Wurde h aus $\mathcal{H}_1$ gewählt, so befinden sich die Schlüssel in der gleichen Äquivalenzklasse und kollidieren nicht. Wurde h hingegen aus $\mathcal{H}_2$ gewählt, kollidieren die Schlüssel auf jeden Fall unter h. Also beträgt die Wahrscheinlichkeit, daß die Schlüssel unter h kollidieren, genau ε.

2. Fall: $x_1 \neq x_1'$. Die Wahrscheinlichkeit, daß die Schlüssel unter einer zufälligen Funktion aus $\mathcal{H}_1$ bzw. aus $\mathcal{H}_2$ kollidieren, ist durch c_1/r bzw. c_2/r beschränkt. Wir erhalten also

$$\textbf{Prob}\left(h(x_1,x_2) = h(x_1',x_2')\right) \;\leq\; (1-\varepsilon)\cdot\frac{c_1}{r} + \varepsilon\cdot\frac{c_2}{r} \;=\; \frac{c_1}{r} - \varepsilon\cdot\frac{c_1-c_2}{r}.$$

Berücksichtigt man beide Fälle, so ist die Kollisionswahrscheinlichkeit zweier beliebiger Schlüssel durch das Maximum von ε und $c_1/r - \varepsilon(c_1-c_2)/r$ beschränkt. Da für $c_2 < c_1$ der eine Term monoton steigend und der andere monoton fallend ist, nimmt das Maximum den minimalen Wert an, wenn $\varepsilon = c_1/r - \varepsilon(c_1-c_2)/r$ gilt. Dies ist äquivalent zu

$$\varepsilon\left(1+\frac{c_1-c_2}{r}\right) \;=\; \frac{c_1}{r},$$

und ergibt nach ε aufgelöst

$$\varepsilon \;=\; \frac{c_1/r}{1+(c_1-c_2)/r} \;=\; \frac{c_1}{r+c_1-c_2}.$$

Hat ε diesen Wert, so kollidieren also zwei Schlüssel unter der zufällig gewählten Funktion h höchstens mit einer Wahrscheinlichkeit von ε. Dies ist insbesondere für den Fall $c_1 = 1$, $c_2 = (u_1 - r)/(u_1 - 1)$ und $u_2 = r$ interessant (d.h. c_2 ist der optimale Universalitätsparameter einer Hashklasse $U_1 \to R$). Dann ist nämlich

$$\varepsilon \;=\; \frac{1}{r+1-(u_1-r)/(u_1-1)} \;=\; \frac{u_1-1}{u_1 r - r + u_1 - 1 - u_1 + r} \;=\; \frac{u_1-1}{u_1 r - 1}$$

Also hat die so konstruierte Hashklasse $U_1 \times R \to R$ einen Universalitätsparameter von

$$\varepsilon r = \frac{(u_1 r - r)}{(u_1 r - 1)},$$

und ist demnach optimal universell.

Wir haben damit folgendes Lemma bewiesen:

3.3.13 Lemma. *Sei $\mathcal{H}_1 : U_1 \times U_2 \to R$ eine $(0|c_1)$-universelle Hashklasse der Kardinalität N_1 und $\mathcal{H}_2 : U_1 \times U_2 \to R$ eine Klasse von N_2 Funktionen mit folgenden Eigenschaften:*

1. *Zwei Schlüssel (x_1, x_2) und (x_1', x_2') aus $U_1 \times U_2$ mit $x_1 \neq x_1'$ kollidieren unter einer zufällig aus $\mathcal{H}_2$ gewählten Funktion höchstens mit einer Wahrscheinlichkeit von c_2/r.*
2. *Das Verhältnis von $N_2/(N_1+N_2)$ beträgt $\varepsilon := c_1/(r+c_1-c_2)$.*

Dann ist $\mathcal{H} = \mathcal{H}_1 \cup \mathcal{H}_2$ εr-universell. Insbesondere ist $\mathcal{H}$ optimal universell, wenn $c_1 = 1$, $c_2 = (u_1 - r)/(u_1 - 1)$ und $u_2 = r$ gilt. ∎

Wir gehen nun davon aus, daß wir bereits eine optimal universelle Hashklasse $\mathcal{F}$ mit Universum U und Wertebereich R kennen. Für $U = R$ bildet ja z.B. die Identität eine optimal universelle Hashklasse. Außerdem sei $\mathcal{H}_1 : U \times R \times R$ eine $(0|1)$-universelle Hashklasse. Indem wir $\mathcal{H}_2 : U \times R$ aus $\mathcal{F}$ wie oben beschrieben bilden (aus den Funktionen $h_f(x_1, x_2) := f(x_1)$ für $f \in \mathcal{F}$), und dann für entsprechend gewählte Vielfache von $\mathcal{H}_1$ bzw. $\mathcal{H}_2$ obiges Lemma anwenden, erhalten wir also eine optimal universelle Hashklasse $U \times R \to R$. Das bedeutet, daß wir das Universum „vergrößern“ können, ohne die Eigenschaft der optimalen Universalität zu verlieren.

Die Kardinalität der so konstruierten Hashklasse beschreibt folgendes Lemma. Dabei bezeichnen $\mathrm{kgV}(m,n)$ und $\mathrm{ggT}(m,n)$ das kleinste gemeinsame Vielfache bzw. den größten gemeinsamen Teiler zweier natürlicher Zahlen m und n. Die Formel

$$m \cdot n \;=\; \mathrm{kgV}(m,n) \cdot \mathrm{ggT}(m,n) \tag{3.9}$$

ist allgemein bekannt.

3.3.14 Lemma. *Ist $\mathcal{G}$ eine $(0|1)$-universelle Hashklasse $U \times R \to R$ und $\mathcal{F}$ eine optimal universelle Hashklasse $U \to R$, dann gibt es eine optimal universelle Hashklasse $\mathcal{H} : U \times R \to R$ der Kardinalität*

$$N \;=\; \frac{|\mathcal{G}||\mathcal{F}|(ur-1)/(u-1)}{\mathrm{ggT}\Big(|\mathcal{G}|, |\mathcal{F}|(ur-u)/(u-1)\Big)}.$$

Insbesondere ist $\mathcal{H}$ Stinson-minimal, wenn auch $\mathcal{G}$ und $\mathcal{F}$ Stinson-minimal sind.

Beweis: Seien $c_1 = 1$ und $c_2 = (u-r)/(u-1)$ die Universalitätsparameter von $\mathcal{G}$ bzw. $\mathcal{F}$, sowie $N_1 = |\mathcal{G}|$ und $N_2 = |\mathcal{F}|$. Offensichtlich ist $N_2 c_2$ eine natürliche Zahl, da zwei verschiedene Schlüssel aus U unter genau $N_2 c_2 / r$ Funktionen aus $\mathcal{F}$ kollidieren. Also ist $k := \mathrm{kgV}\big(N_1 c_1, N_2(r-c_2)\big)$ wohldefiniert, und es sind auch

$$k_1 := \frac{k}{N_1 c_1} \quad \text{und} \quad k_2 := \frac{k}{N_2(r-c_2)}$$

natürliche Zahlen.

Die Hashklasse $\mathcal{H}_1$ enthalte nun genau die Funktionen aus $\mathcal{G}$, und zwar jeweils k_1-mal. $\mathcal{H}_2$ enthalte genau die Funktionen $h_f : U \times R \to R$, $(x_1, x_2) \mapsto f(x_1)$ mit $f \in \mathcal{F}$ jeweils k_2-mal. Die folgende Rechnung zeigt, daß das Verhältnis der Kardinalitäten von $\mathcal{H}_1$ und $\mathcal{H}_2$ den Voraussetzungen aus Lemma 3.3.13 entspricht:

$$\frac{\big|\mathcal{H}_2\big|}{\big|\mathcal{H}_1\big| + \big|\mathcal{H}_2\big|} = \frac{k_2 N_2}{k_1 N_1 + k_2 N_2} = \frac{k/(r-c_2)}{k/c_1 + k/(r-c_2)} = \frac{c_1}{r - c_2 + c_1}.$$

Da die Hashklasse $\mathcal{H}_2$ auch die erste Voraussetzung aus Lemma 3.3.13 bezüglich der Kollisionswahrscheinlichkeit ihrer Schlüssel erfüllt, können wir das Lemma anwenden, und es folgt die optimale Universalität von $\mathcal{H} = \mathcal{H}_1 \cup \mathcal{H}_2$.

Die Kardinalität von $\mathcal{H}$ berechnet sich unter Zuhilfenahme von Gleichung (3.9) wie folgt:

$$\begin{aligned} \big|\mathcal{H}\big| &= k_1 N_1 + k_2 N_2 = \mathrm{kgV}\big(N_1 c_1, N_2(r-c_2)\big)\left(\frac{1}{c_1} + \frac{1}{r-c_2}\right) \\ &= \frac{N_1 c_1 \cdot N_2(r-c_2)}{\mathrm{ggT}\big(N_1 c_1, N_2(r-c_2)\big)} \cdot \frac{r - c_2 + c_1}{c_1(r-c_2)} = \frac{N_1 N_2 (r - c_2 + c_1)}{\mathrm{ggT}\big(N_1, N_2(r-c_2)\big)}. \end{aligned}$$

Es ist $r - c_2 = r - (u-r)/(u-1) = (ur-u)/(u-1)$, und mit $c_1 = 1$ erhält man $r - c_2 + c_1 = (ur-1)/(u-1)$. Somit ist wie behauptet $\big|\mathcal{H}\big| = N$.

Sind die Hashklassen $\mathcal{G}$ und $\mathcal{F}$ Stinson-minimal, d.h. es gilt $N_1 = u$ und $N_2 = (u-1)/(r-1)$, so ist $N_2(ur-u)/(u-1) = u$. Also ist auch u der größte gemeinsame Teiler von N_1 und $N_2(ur-u)/(u-1)$. Demnach erhält man

$$N = \frac{N_1 N_2 (ur-1)/(u-1)}{u} = \frac{ur-1}{r-1}.$$

Dies ist die Kardinalität einer Stinson-minimalen optimal universellen Hashklasse mit Universum $U \times R$ und Wertebereich R. ∎

Für sich alleine genommen ist diese Methode noch zu schwach, um allgemein effiziente optimal universelle Hashklassen zu konstruieren. Schließlich werden eine $(0|1)$-universelle Hashklasse und eine optimal universelle Hashklasse mit ganz speziellen Universumsgrößen benötigt, um daraus eine neue optimal universelle zu erhalten. Für praktische Anwendungen universeller Hashklassen sieht die Problemstellung aber häufig wie folgt aus: Gegeben ist die *Wortbreite w* eines Rechners, d.h. alle Daten sind aus Wörtern zusammengesetzt, die aus $W = \{0, \ldots, 2^w - 1\}$ stammen. Es sollen nun Daten verarbeitet werden, die aus bis zu n Wörtern bestehen, indem sie mit einer Hashfunktion auf Werte abgebildet werden, die aus $m \leq n$ Wörtern zusammengesetzt sind.

Wünschenswert wäre also z.B. für eine Menge W eine universelle Hashklasse $W^n \to W^m$ mit möglichst kleinem Universalitätsparameter. Ein wichtiger Spezialfall dabei ist $m = 1$, bei dem die Hashwerte durch ein Wort dargestellt und so effizient weiterverarbeitet werden können. Mit den in § 2 vorgestellten Techniken ist es nun möglich, aus einer einzigen abstandsuniversellen Hashklasse $W \to W$ optimal universelle Hashklassen mit beliebigen Universen W^{km} und Wertebereichen W^m zu konstruieren. Man erhält so sogar aus abstandsuniversellen Hashklassen, die minimal bezüglich Korollar 3.2.8 sind, Stinson-minimale optimal universelle Hashklassen.

3.3.15 Satz. *Sei $\mathcal{H}$ eine abstandsuniverselle Hashklasse $W \to W$ der Kardinalität N, und für beliebige $k, m \in \mathbb{N}$ sei $R = W^m$ und $U = W^{km} = R^k$. Dann gibt es eine optimal universelle Hashklasse $U \to R$ der Kardinalität*

$$\frac{N^{km-1}}{r^{k-1}} \cdot \frac{u-1}{r-1}.$$

Beweis: Wir konstruieren zunächst aus $\mathcal{H}$ mit Satz 3.2.10 für alle $i \in \mathbb{N}$ abstandsuniverselle Hashklassen $W^{im} \to W^m$. Diese haben Kardinalität $N^{(i+1)m-1}$. Gemäß Satz 3.2.7 gibt es dann für jedes i eine universelle Hashklasse $\mathcal{G}_i$ mit Universum $W^{(i+1)m}$, gleichem Wertebereich und gleicher Kardinalität. Jedes $\mathcal{G}_i$ ist nach Bemerkung 3.3.2 sogar eine exakt $(0|1)$-universelle Hashklasse $R^i \times R \to R$.

Wir bezeichnen die zu konstruierende optimal universelle Hashklasse $R^k \to R$ mit $\mathcal{F}_k$ und zeigen ihre Existenz durch vollständiger Induktion über k. Für $k = 1$ bestehe $\mathcal{F}_k$ nur aus N^{m-1} Identitäten $id : R \to R$. Da die Identität offensichtlich eine optimal universelle Hashklasse bildet, ist auch $\mathcal{F}_k$ optimal universell. Außerdem hat $\mathcal{F}_k$ wegen $u = r$ die behauptete Kardinalität.

Es erfüllen nun $\mathcal{G}_k$ und $\mathcal{F}_k$ die Voraussetzungen von Lemma 3.3.14, und wir können aus diesen eine optimal universelle Hashklasse $\mathcal{F}_{k+1}$ mit Universum $U' = U \times R$ konstruieren. Es genügt zu zeigen, daß $\mathcal{F}_{k+1}$ die gewünschte Kardinalität besitzt. Offensichtlich ist $U' = R^k \times R$, und wir setzen $u' = ur = r^{k+1}$. Aufgrund der Induktionsvoraussetzung gilt

$$\left|\mathcal{F}_k\right| \cdot \frac{ur-u}{u-1} \quad = \quad \frac{N^{km-1}}{r^{k-1}} \cdot \frac{u-1}{r-1} \cdot \frac{ur-u}{u-1} \quad = \quad \frac{N^{km-1}}{r^{k-1}} \cdot \frac{r^{k+1}-r^k}{r-1} \quad = \quad N^{km-1}r\,.$$

Es folgt somit

$$\begin{aligned} \mathrm{ggT}\left(\left|\mathcal{G}_k\right|, \left|\mathcal{F}_k\right| \cdot \frac{ur-u}{u-1}\right) &= \mathrm{ggT}\left(N^{(k+1)m-1}, N^{km-1}r\right) \\ &= \mathrm{ggT}\left(N^{km-1}N^m, N^{km-1}r\right) \;=\; N^{km-1}r. \end{aligned}$$

Die letzte Gleichung gilt, weil N^m ein Vielfaches von r ist. Der Grund dafür ist, daß wegen der Abstandsuniversalität von $\mathcal{H}$ alle Schlüssel aus W jeden beliebigen Abstand d unter genau N/w Funktionen aus $\mathcal{H}$ annehmen. Somit ist w ein Teiler von N und $r = w^m$ auch ein Teiler von N^m.

Abschließend müssen wir noch das Ergebnis der letzten Gleichung in die Formel aus Lemma 3.3.14 einsetzen, und wir erhalten

$$\begin{aligned} \left|\mathcal{F}_{k+1}\right| &= \frac{\left|\mathcal{G}_k\right|\left|\mathcal{F}_k\right|(ur-1)/(u-1)}{N^{km-1}r} \;=\; \frac{N^{(k+1)m-1}}{N^{km-1}r} \cdot \frac{N^{km-1}(u-1)}{r^{k-1}(r-1)} \cdot \frac{ur-1}{u-1} \\ &= \frac{N^{(k+1)m-1}}{r^k-1}\,\frac{u'-1}{r-1} \end{aligned}$$

Da dies die Induktionsbehauptung erfüllt, ist der Beweis vollständig. ∎

Wir betrachten nun die Hashklasse $\mathcal{H}^{hom}_{\mathbb{K},id}$ für einen endlichen Körper $\mathbb{K}$ der Ordnung r. Diese ist abstandsuniversell und hat die Kardinalität r (s. S. 23). Mit der Konstruktion aus Satz 3.3.15 erhalten wir für jedes $n \in \mathbb{N}$ eine optimal universelle Hashklasse $\mathbb{K}^n \to \mathbb{K}$ der Kardinalität $(r^n-1)/(r-1)$. Diese ist Stinson-minimal, und gemäß Korollar 3.1.6 ist ihre transponierte Hashklasse Stinson-minimal streng universell.

3.3.16 Korollar.

(a) Gibt es eine abstandsuniverselle Hashklasse $R \to R$ der Kardinalität r, dann gibt es für alle $n \in \mathbb{N}$ eine Stinson-minimale optimal universelle Hashklasse $R^n \to R$.

(b) Ist r Potenz einer Primzahl, dann gibt es eine Stinson-minimale optimal universelle Hashklasse $R^n \to R$.

(c) Ist r Potenz einer Primzahl und $u = (r^n-1)/(r-1)$, dann gibt es eine Stinson-minimale streng universelle Hashklasse $U \to R$.

Die Existenzaussage aus Teil (b) wurde bereits von Sarwate (1980) durch die direkte Angabe solcher Hashklassen, sowie von Stinson (1994a) mithilfe der Äquivalenz zu affin auflösbaren BIBDs gezeigt. Interessanterweise sind für den Fall, daß $r < u$ keine Primpotenz ist, keine Stinson-minimalen optimal universellen Hashklassen bekannt und folglich auch keine abstandsuniversellen Hashklassen $R \to R$ der Kardinalität r. In der Theorie der kombinatorischen Designs wird deshalb allgemein vermutet, daß es keine affin auflösbaren BIBDs mit Parametern (v,k,λ) gibt, wenn v und k nicht Potenzen der gleichen Primzahl sind (s. z.B. Shrikhande, 1976).

Genau die Idee der hier beschriebenen Konstruktionsmethode liegt auch der optimal universellen Faltungsklasse aus Satz 3.2.11 zugrunde. Mit den gerade erarbeiteten Methoden können wir den noch offen gebliebenen Beweis jetzt führen. Dazu formulieren wir die Aussage in folgendem Satz etwas präziser.

3.3.17 Satz. *Sei $U_n = \mathbb{K}^n$ und $R = \mathbb{K}^m$, wobei $\mathbb{K}$ ein endlicher Körper und m ein Teiler von $n \in \mathbb{N}$ ist. Für $A \subseteq \mathbb{K}^n$ bestehe $\mathcal{H}^n_A$ aus den Funktionen $h^n_{\underline{a}} : \mathbb{K}^n \to \mathbb{K}^m$, $\underline{x} \mapsto \mathrm{conv}_{n-m,n-1}(\underline{a},\underline{x})$ mit $\underline{a} \in A$. Weiterhin sei A_k für $0 \le k < n$ die Menge der Vektoren*

$$(\underbrace{0,\ldots,0}_{k\text{-mal}},1,a_{k+1},\ldots,a_{n-1}) \in \mathbb{K}^n,$$

mit $a_{k+1},\ldots,a_{n-1} \in \mathbb{K}$. Dann ist $\mathcal{H}^n_A$ für $A = A_0 \cup A_m \cup \ldots \cup A_{n-m}$ optimal universell.

Offensichtlich folgt hieraus sofort Satz 3.2.11 (d).

Die Idee für den Beweis beruht darauf, daß einerseits die Familie $\mathcal{H}^n_{A_0}$ eine $(0|1)$-universelle Hashklasse $U_{n-m} \times R \to R$ bildet und andererseits die Familie $\mathcal{H}^n_{A_m \cup \ldots \cup A_{n-m}}$ die erste Komponente von $U_{n-m} \times R$ optimal universell auf R abbildet. Da auch das Verhältnis der Kardinalitäten beider Familien schon richtig ist, erfüllen diese die Voraussetzungen von Lemma 3.3.13, und ihre Vereinigung ist wieder optimal universell.

Beweis: Es sei zunächst an das Ergebnis aus Satz 3.2.11 (c) und Bemerkung 3.3.3 erinnert. Demnach ist die Hashklasse $\mathcal{H}^n_{A_0}$ exakt $(0|1)$-universell mit Universum $\mathbb{K}^{n-m} \times \mathbb{K}^m$.

Sei i so, daß $n = im$, also $U_n = \mathbb{K}^n = R^i$. Wir zeigen die Behauptung mit vollständiger Induktion über i. Für $i = 1$ (d.h. $n = m$) unterscheiden sich zwei Schlüssel aus $\mathbb{K}^n$ offensichlich nie in den ersten $n-m$ Spalten, und kollidieren aufgrund der $(0|1)$-Universalität von $\mathcal{H}^n_{A_0}$ unter keiner ihrer Funktionen. Per Definition ist aber $A = A_0$, und so ist die Kollisionswahrscheinlichkeit zweier Schlüssel 0. Demnach ist $\mathcal{H}^n_A$ für $n = m$ optimal universell.

Sei nun die Behauptung für das Universum $\mathbb{K}^{n-m}$ gezeigt. Wir teilen die Hashklasse $\mathcal{H}^n_A$ in die Klassen $\mathcal{G}$ und $\mathcal{F}$ auf, wobei $\mathcal{G}$ die Funktionen h^n_a mit $\underline{a} \in A_0$ und $\mathcal{F}$ die mit $\underline{a} \in A_m \cup \ldots \cup A_{n-m}$ beinhaltet. Es genügt zu zeigen, daß beide Hashklassen die Voraussetzungen von Lemma 3.3.13 erfüllen. Für $\mathcal{G}$ ist dies mit der $(0|1)$-Universalität schon gezeigt.

Wir untersuchen nun $\mathcal{F}$ etwas näher. Bei jedem Vektor $\underline{a} \in A_m \cup \ldots \cup A_{n-m}$ haben die ersten m Spalten einen Wert von 0. Um zu zeigen, wie sich diese führenden Nullen auf die Funktion $h^n_{\underline{a}}$ auswirken betrachten wir die Faltung $*$ von Vektoren über $\mathbb{K}$, d.h.

$$(x_0, x_1, \ldots) * (y_0, y_1, \ldots) \;=\; (z_0, z_1, \ldots) \quad \text{mit} \quad z_i = \sum_{j=0}^{i} x_j y_{i-j}.$$

Aus dieser Definition folgt unmittelbar, daß

$$(\underbrace{0, \ldots, 0}_{m\text{-mal}}, x_m, x_{m+1} \ldots) * (y_0, y_1, \ldots) \;=\; (\underbrace{0, \ldots, 0}_{m\text{-mal}}, z_0, z_1, \ldots)$$

gilt, wobei $(z_0, \ldots)$ daß Ergebnis der Faltung von $(x_m, \ldots)$ mit $(y_0, \ldots)$ ist. Die Funktion $\mathrm{conv}_{n-m,n-1}$ berechnet nur die Einträge der Spalten mit Index $n-m, \ldots, n-1$. Wir erhalten also folgende Formel für $\underline{a} = (0, \ldots, 0, a_m, \ldots, a_{n-1})$ und $\underline{x} = (x_0, \ldots, x_{n-1})$:

$$\mathrm{conv}_{n-m,n-1}(\underline{a}, \underline{x}) \;=\; \mathrm{conv}_{n-2m,n-m-1}\big((a_m, \ldots, a_{n-1}), (x_0, \ldots, x_{n-m-1})\big).$$

Dies bedeutet, daß für jedes $(a_0, \ldots, a_{n-1}) \in A_m \cup \ldots \cup A_{n-1}$ gilt:

$$h^n_{(a_0,\ldots,a_{n-1})}(x_0, \ldots, x_{n-1}) \;=\; h^{n-m}_{(a_m,\ldots,a_{n-1})}(x_0, \ldots, x_{n-m-1}).$$

D.h. die Funktionen aus $\mathcal{F}$ „sehen“ nur die ersten $n-m$ Spalten der Schlüssel, und bilden diese mit den Funktionen aus $\mathcal{H}^{n-m}_{A'}$ ab, wobei $A' = A_0 \cup \ldots \cup A_{n-2m}$ ist. Da diese Hashklasse $\mathcal{H}^{n-m}_{A'}$ per Induktionsvoraussetzung optimal universell ist, folgt, daß die ersten $n-m$ Spalten der Schlüssel aus $\mathbb{K}^n$ unter den Funktionen aus $\mathcal{F}$ optimal universell auf $R = \mathbb{K}^m$ abgebildet werden. Also beträgt die Kollisionswahrscheinlichkeit zweier Schlüssel aus $\mathbb{K}^n$, die sich in den ersten $n-m$ Spalten unterscheiden,

$$\frac{1}{r} \cdot \frac{|\mathbb{K}^{n-m}| - r}{|\mathbb{K}^{n-m}| - 1},$$

wenn die Funktion aus $\mathcal{F}$ zufällig gewählt wird. Damit entspricht die Kollisionswahrscheinlichkeit den Voraussetzungen von Lemma 3.3.13, und wir müssen nur noch zeigen, daß das Verhältnis von $|\mathcal{F}|$ zu $|\mathcal{G}| + |\mathcal{F}|$ gleich ε (mit ε wie in Lemma 3.3.13) ist.

Dazu sei $K = |\mathbb{K}|$. Es wurde i anfangs so gewählt, daß $n = im$ gilt. Also folgt

$$\begin{aligned}\frac{|\mathcal{F}|}{|\mathcal{G}|+|\mathcal{F}|} &= \frac{|A_m \cup \ldots \cup A_{n-m}|}{|A_0|+|A_m \cup \ldots \cup A_{n-m}|} = \frac{K^{n-m-1}+K^{n-2m-1}+\ldots+K^{m-1}}{K^{n-1}+K^{n-m-1}+\ldots+K^{m-1}} \\ &= \frac{(K^m)^1+(K^m)^2+\ldots+(K^m)^{i-1}}{(K^m)^1+(K^m)^2+\ldots+(K^m)^i} = \frac{r^0+\ldots+r^{i-2}}{r^0+\ldots+r^{i-1}} = \frac{r^{i-1}-1}{r^i-1}.\end{aligned}$$

Die letzte Gleichheit folgt mit der geometrischen Reihe $x^0+\ldots+x^n = (x^{n+1}-1)/(x-1)$. Das Verhältnis entspricht also mit $u_1 = r^{i-1}$ genau dem Wert von ε. ∎

§ 4. Ergänzende Bemerkungen

Die erste Arbeit, die sich mit optimal universellem Hashing beschäftigt, stammt von Sarwate (1980). In dieser wird u.a. eine untere Schranke für die Kardinalität optimal universeller Hashklassen vorgestellt, die in manchen Fällen sogar besser als die aus Satz 3.1.2 ist. Mehlhorn (1982) gibt die ersten unteren Schranken für c-universelle Hashklassen an; diese sind aber für $c \leq 1$ erheblich schlechter als die hier vorgestellten. Van Trung (1994) hat Stinson-minimale streng c-universelle Hashklassen mithilfe sogenannter quasi-symmetrischer Designs charakterisiert. Anhand dieser Charakterisierung zeigt sich, daß solche Hashklassen nur für ganz bestimmte Parameter c, u und r existieren können.

Neben den hier vorgestellten Äquivalenzen von kombinatorischen Objekten und universellen Hashklassen sind inzwischen zahlreiche weitere Beziehungen bekannt. So sind die Abbildungsmatrizen streng universeller Hashklassen orthogonale Arrays (Stinson, 1994a) und die abstandsuniverseller Hashklassen sind Differenzmatrizen (Stinson, 1996). Stinson (1994b) bzw. Atici und Stinson (1996) zeigen auch Beziehungen zwischen streng c-universellen Hashklassen und Authentifizierungs-Codes auf, und bei Bierbrauer, Johansson, Katatianskii und Smeets (1994), Stinson (1996) sowie Bierbrauer (1997) werden universelle Hashklassen mit Mitteln der algebraischen Kodierungstheorie konstruiert.

Fast alle bekannten Konstruktionen von RBIBDs oder optimal universellen Hashklassen sind algebraischer Natur, und es gibt in der Literatur kaum Hinweise auf ihre Praktikabilität. Unsere Konstruktionsmethode aus § 3 ist ein erster Schritt, effizientes optimal universelles Hashing zu erreichen, und hat bereits bei den Faltungsklassen eine sinnvolle Anwendung gefunden. Es hat sich weiterhin gezeigt, daß abstandsuniverselle Hashklassen eine wichtige Grundlage für viele Konstruktionsmethoden bilden. Dies wird auch im nächsten Kapitel bestätigt, wo die Anwendung der hier vorgestellten Techniken zu effizienten Hashklassen mit ganzzahliger Arithmetik führt.

KAPITEL 4

Ganzzahlige Arithmetik

Im vorigen Kapitel wurde bei den dort vorgestellten Konstruktionsmethoden insbesondere Wert auf die kleine Kardinalität von Hashklassen gelegt. Für viele praktische Anwendungen ist aber die effiziente Auswertung der Hashfunktionen noch wichtiger. So wären z.B. Wörterbuchimplementationen nicht praktikabel, wenn deren Hashfunktionen auf eine komplizierte Arithmetik angewiesen sind, die nicht direkt von der Rechnerhardware unterstützt wird (wie z.B. Körperarithmetik). Die Auswertung der Funktionen wäre dann so langsam, daß andere Wörterbuchtechniken, die nicht auf universellem Hashing (sondern z.B. auf fest gewählten Hashfunktionen) beruhen, auch für „schlechte“ Eingaben zu effizienteren Ergebnissen führen würden.

Viele der bekannten und häufig verwendeten Hashklassen haben jedoch noch andere Nachteile. So kommen zwar die Primzahlklassen (s. Satz 1.1.3 und Beispiel 2.1.2) mit ganzzahliger Arithmetik aus, es ist aber die Kenntnis einer Primzahl in der Größenordnung des Universums notwendig. Ist die Universumsgröße vor der Ausführung des Algorithmus nicht bekannt, muß eine - unter Umständen sehr große - Primzahl „online“ gefunden werden. Dies ist jedoch nicht unproblematisch, da keine effizienten deterministischen Algorithmen zum Auffinden von großen Primzahlen bekannt sind. Zur Wahrscheinlichkeitsamplifikation für einen Primzahltest, wie in Kapitel 2, § 3 (S. 13) beschrieben, sind solche Hashklassen dann natürlich nicht geeignet. Ähnlich verhält es sich mit den Körperklassen, bei denen ein irreduzibles Polynom in der Größenordnung des Universums zur Auswertung der Hashfunktionen benötigt wird (zu der Problematik von Primzahl- und Körperklassen siehe auch Alon, Goldreich, Håstad und Peralta 1992, 1993 sowie Matias und Vishkin 1991).

Die Faltungsklassen hingegen haben zwar interessante kombinatorische Eigenschaften (s. Satz 3.2.11), und können für geeignete Körper (z.B. $\mathbb{K} = \mathbb{Z}_2$) ohne Kenntnis großer Primzahlen oder irreduzibler Polynome ausgewertet werden, andererseits ist aber die Faltung zweier Vektoren für die üblichen Prozessoren erheblich aufwendiger als z.B. die Multiplikation zweier ganzer Zahlen, die direkt unterstützt wird. Ähnlich verhält es sich mit anderen bekannten Hashklassen, die ohne Primzahlen oder irreduzible Polynome auskommen. So können beispielsweise die von Krawczyk (1994, 1995) vorgeschlagenen, auf Toeplitz-Matrizen beruhenden Hashklassen effizient mithilfe spezieller Hardware (Schieberegister) implementiert werden, sind aber wegen der benötigten Matrizenmultiplikation für normale Prozessoren nicht praktikabel.

Speziell für die Nachrichtenauthentifizierung wurden auch Hashklassen mit sehr effizient auswertbaren Funktionen entwickelt. Diese Lösungen führen jedoch alle zu einem sehr großen Universalitätsparameter, der die Hashklassen für andere Anwendungen wie Wörterbuchimplementationen ungeeignet macht. So ist das von Rogaway (1995) vorgeschlagene

„Bucket Hashing“ (s. auch Johansson, 1997) zwar sehr effizient im Sinne der Auswertungszeit der Hashfunktionen, erreicht aber nur einen Universalitätsparameter in der Größenordnung von $\Omega(r/\log r)$.

Das Ziel der folgenden Abschnitte ist daher, Hashklassen vorzustellen und neu zu entwickeln, die mit ganzzahliger Arithmetik ohne Primzahlen auskommen, effizient auswertbar sind und einen möglichst kleinen Universalitätsparameter besitzen. Aus den in Kapitel 3, § 1 genannten Gründen muß dabei auch darauf geachtet werden, daß die Kardinalität der Hashklassen möglichst klein ist.

§ 1. Die Ganzzahlklassen

Wenn nicht ausdrücklich anders erwähnt, bestehen in diesem Kapitel das Universum U und der Wertebereich R immer aus den Zahlen $0,\ldots,u-1$ bzw. $0,\ldots,r-1$ für feste $u \geq r \geq 2$. Die ersten Hashklassen, die kleine Universalitätsparameter mit ganzzahliger Arithmetik ohne Primzahlen erreichen, wurden von Dietzfelbinger (1996) und Dietzfelbinger, Hagerup, Katajainen und Penttonen (1997) vorgestellt. In der Arbeit von Dietzfelbinger et al. wurden für $u = 2^n$ und $r = 2^m$ Funktionen $f_a : U \to R$ untersucht, die durch die Abbildungsvorschrift $x \mapsto \big((ax) \bmod 2^n\big) \operatorname{div} 2^{n-m}$ definiert sind. Es konnte gezeigt werden, daß die sogenannte „multiplikative“ Hashklasse, die aus den Funktionen f_a mit $a \in \{1,3,\ldots,u-1\}$ besteht, 2-universell ist. Die „lineare“ Hashklasse aus der Arbeit von 1996 besteht für ein $k \in \mathbb{N}$ aus den Funktionen $g_{a,b} : U \to R, x \mapsto \big((ax+b) \bmod kr\big) \operatorname{div} k$ mit $a,b \in \{0,\ldots,kr-1\}$. Dietzfelbinger hat gezeigt, daß diese Hashklasse für $k \geq u-1$ im allgemeinen streng $5/4$-universell und, wenn u und r Potenzen der gleichen Primzahl sind, sogar streng universell ist.

Besonders effizient lassen sich die Funktionen der multiplikativen Hashklasse implementieren. Denn bei dieser können die Modulooperation und die Division durch ein bitweises „Und“ mit 2^{n-1} bzw. eine Rechtsverschiebung um $n-m$ Bits ersetzt werden (s. Bemerkung 1.1.4). Somit können die Funktionen f_a äußerst schnell mit einer Multiplikation und zwei bitweisen Operationen ausgewertet werden. Ähnliches gilt natürlich für die lineare Hashklasse, wenn k und r Zweierpotenzen sind. Es ist zu erwarten, daß die Funktionen dieser Hashklassen dann sogar effizienter auswertbar sind als die der Primzahlklassen, welche ja neben der Multiplikation zumindest eine Operation modulo einer Primzahl, also eine Division benötigen.

Insbesondere die multiplikative Hashklasse hat bereits verschiedene Anwendungen gefunden (s. Dietzfelbinger, Hagerup, Katajainen und Penttonen, 1997; Andersson, Hagerup, Nilsson und Raman, 1995), und experimentelle Studien zeigen ihre Praktikabilität für dynamische Wörterbuchimplementationen (Dietzfelbinger und Hühne, 1996). Der Universalitätsparameter von 2 ist hingegen noch nicht ganz zufriedenstellend, und es stellt sich die Frage, ob eine Verbesserung des Universalitätsparameters ohne signifikante Einbußen bei der Effizienz möglich sind.

In den folgenden Abschnitten werden wir eine Vielzahl von Hashklassen entwerfen und analysieren, die alle folgendermaßen parameterisiert beschrieben werden können: Es seien u und r fest. Für $k,a,b \in \mathbb{N}$ seien die Hashfunktionen $h^k_{a,b} : U \to R$ gegeben durch die Abbildung

$$x \mapsto \big((ax+b) \bmod (kr)\big) \operatorname{div} k.$$

Der Übersichtlichkeit wegen schreiben wir $h_{a,b}$ statt $h^k_{a,b}$, wenn klar ist, um welchen Parameter k es sich handelt. Für zwei Familien $A,B \subseteq \{0,\ldots,kr-1\}$ bestehe die Hashklasse $\mathcal{H}^k_{A,B}$

aus den Funktionen $h^k_{a,b}$ mit $a \in A$ und $b \in B$. Wir nennen die Hashklassen $\mathcal{H}^k_{A,B}$ mit $B = \{0\}$ die *homogenen Ganzzahlklassen* und mit $B \supsetneq \{0\}$ die *linearen Ganzzahlklassen*.

Offensichtlich entspricht die oben erwähnte, von Dietzfelbinger untersuchte lineare Hashklasse der Familie $\mathcal{H}^k_{U,U}$ mit $k \geq u-1$, und die multiplikative Hashklasse der Familie $\mathcal{H}^{u/r}_{U,\{0\}}$ (wobei u und r Zweierpotenzen sind).

Auf den ersten Blick würde man im Falle von 2er-Potenzen u und r wahrscheinlich die lineare Hashklasse $\mathcal{H}^u_{U,U}$ der multiplikativen $\mathcal{H}^{u/r}_{U,\{0\}}$ vorziehen, da sie ja sogar 1-universell ist und nur eine zusätzliche Addition benötigt. Sieht man aber von der Tatsache ab, daß die Kardinalität der linearen Hashklasse größer als die der multiplikativen ist, so gibt es noch einen anderen Grund, der für die Verwendung der letzteren spricht. So ist unter Umständen die Auswertung ihrer Funktionen wesentlich effizienter, wie das folgende praxisrelevante Beispiel belegt: Es sollen 64-Bit Schlüssel auf einen Wertebereich mit 32-Bit abgebildet werden, d.h. $u = 2^{64}$ und $r = 2^{32}$. Der Prozessor unterstützt eine 64-Bit Multiplikation, die, wie allgemein üblich, aber nur die untere Worthälfte (also die 64 weniger signifikanten Bits) des Produkts zweier 64-Bit-Zahlen berechnet. Während die Multiplikation der Funktionen aus $\mathcal{H}^{u/r}_{U,\{0\}}$ direkt vom Prozessor ausgeführt werden kann, da sie nur eine Wortbreite von 64 Bit benötigt, ist bei der linearen Hashklasse $\mathcal{H}^k_{U,U}$ wegen der Multiplikation über $\log(kr) = 96$ Bits zusätzlicher Aufwand notwendig. Es ist zu erwarten, daß dadurch die Auswertungszeit für eine Funktion der linearen Hashklasse um ein Vielfaches höher als für eine der multiplikativen Hashklasse ist.

Der Parameter k der Hashklasse $\mathcal{H}^k_{A,B}$ kann somit entscheidenden Einfluß auf die Effizienz haben und muß in den folgenden Konstruktionen berücksichtigt werden. Ist $B = \{0\}$, so können die Funktionen zudem ohne Addition ausgewertet werden, was aber bei gängigen Prozessoren wegen der Dominanz der Rechenzeit für die Multiplikation weniger ins Gewicht fallen dürfte. Die Kardinalität der Hashklasse ist offensichtlich durch $|A| \cdot |B|$ bestimmt und die Anzahl der notwendigen Zufallsbits für die Auswahl einer zufälligen Hashfunktion durch $\lceil \log(|A| \cdot |B|) \rceil$. Somit ist es das Ziel dieses Kapitels, Hashklassen $\mathcal{H}^k_{A,B}$ mit möglichst kleinen Werten k und $|A| \cdot |B|$ sowie guten kombinatorischen Eigenschaften (d.h. niedrigen Universalitätsparametern) zu finden. Dabei spielt der Fall, in dem k und r Zweierpotenzen sind, eine besonders wichtige Rolle, da dann die Hashfunktionen ohne Division ausgewertet werden können. Außerdem entsprechen Universums- und Wertebereichsgrößen in der Ordnung von Zweierpotenzen den natürlichen Anforderungen von Anwendungen, die normalerweise Daten mit solchen Wortbreiten verarbeiten.

Um dieses Ziel zu erreichen, werden die Hashklassen für verschiedenste Werte von A, B und k analysiert. Fast alle Beweise beruhen auf neuen, d.h. in den o.g. Arbeiten nicht verwendeten Methoden und werden später mit den im vorigen Kapitel erarbeiteten Techniken kombiniert. So können wir die bekannten Ergebnisse verbessern und erhalten neue Hashklassen mit interessanten Eigenschaften, wie z.B. eine 1-universelle und eine optimal universelle Ganzzahlklasse, deren Funktionen ähnlich effizient ausgewertet werden können, wie die der multiplikativen Hashklasse. Alle in diesem Kapitel vorgestellten Ergebnisse sind - sofern nicht ausdrücklich auf andere Arbeiten hingewiesen - erstmals bei (Woelfel, 1999) veröffentlicht.

§ 2. Abstandsuniverselle und streng universelle Ganzzahlklassen

In Kapitel Kapitel 3, § 2 wurde deutlich, daß abstandsuniverselle Hashklassen die Grundlage für die Konstruktion universeller, streng universeller und sogar optimal universeller Hashklassen bilden können. Wir werden uns daher zunächst mit den abstandsuniversellen Eigenschaften der Ganzzahlklassen beschäftigen.

Die Multiplikation

Zur Analyse ist es hilfreich, wenn man die Funktionen $h_{a,b}$ als die Hintereinanderausführung einer Multiplikation über $\mathbb{Z}_{kr}$ und einer Addition von b verbunden mit der Division betrachtet. Da $\mathbb{Z}_{kr}$ ein Ring ist, gestattet eine solche Betrachtungsweise die Anwendung der grundlegenden algebraischen Methoden.

Im folgenden bezeichne v immer den Wert $kr \geq u$ und V den Restklassenring $\mathbb{Z}_v$. Insbesondere werden die Schlüssel aus U als Elemente des Rings V angesehen. Wenn nicht ausdrücklich anders darauf hingewiesen wird, sind alle Additionen und Multiplikationen mit Elementen aus V Operationen in diesem Ring (d.h. also modulo v). Benutzen wir Operationen oder Relationen, die nur über $\mathbb{Z}$ oder $\mathbb{N}$, aber nicht über V definiert sind (z.B. die Relation „$<$“) im Zusammenhang mit Elementen aus V, so sind diese auf die entsprechenden Repräsentanten aus $\{0,\ldots,v-1\}$ von V anzuwenden. Sind beispielsweise $x,y \in V$, so ist mit $x > y$ die Relation $x \bmod v > y \bmod v$ oder mit $\mathrm{ggT}(x,y)$ der größte gemeinsame Teiler von $x \bmod v$ und $y \bmod v$ gemeint (sofern dieser definiert ist). Schließlich betrachten wir R immer als abelsche Gruppe $\mathbb{Z}_r$, und bezeichnen die Addition über $\mathbb{Z}_r$ mit $\oplus$.

Für $a,b \in V$ seien die Abbildungen $f_a : V \to V$ und $g_b : V \to R$ definiert als

$$f_a : x \mapsto ax, \qquad \text{und} \qquad g_b : x \mapsto (x+b) \operatorname{div} k.$$

Offensichtlich ist also $h_{a,b} = g_b \circ f_a$.

Wir untersuchen zunächst die Verteilung von $f_a(x_2) - f_a(x_1)$ für verschiedene $x_1, x_2 \in V$ und ein zufällig gewähltes $a \in V$. Das folgende Lemma sagt aus, daß für $x_1 < x_2$ der Abstand $f_a(x_2) - f_a(x_1)$ nur Werte annimmt, die ein Vielfaches des größten gemeinsamen Teilers von $x_2 - x_1$ und v sind, und daß jeder dieser Werte mit gleicher Wahrscheinlichkeit angenommen wird. Fast die gleiche Aussage hat bereits Dietzfelbinger (1996, Lemma 5a) gezeigt. Für beliebige $x \in V$ bezeichne Vx die Menge $\{ax \mid a \in V\}$.

4.2.1 Lemma. *Seien $d, x_1, x_2 \in V$ mit $\delta = x_2 - x_1 > 0$ und $\gamma = \mathrm{ggT}(\delta, v)$. Dann gilt*

(a) $V\delta = \{i\gamma \mid 0 \leq i < v/\gamma\}$

(b) $\underset{a \in V}{\mathbf{Prob}}\big(f_a(x_2) - f_a(x_1) = d\big) = \underset{a \in V}{\mathbf{Prob}}(a\delta = d) = \begin{cases} \gamma/v & \text{falls } d \in V\delta \\ 0 & \text{sonst.} \end{cases}$

Beweis: *Zu (a):* Bekanntlich ist $V\delta$ ein Hauptideal des Rings V und bildet somit ein V-Modul (s. z.B. Scheja und Storch, 1994, S. 164ff.). Sind also $p,q \in V\delta$ und $z \in V$, dann ist auch $pz + q \in V\delta$. Wir berechnen nun gemäß dem euklidischen Algorithmus den größten gemeinsamen Teiler von v und δ mit der rekursiven Formel $\mathrm{ggT}(x,x) = x$ und $\mathrm{ggT}(x,y) = \mathrm{ggT}(x-y,y)$ für $x > y$. Da per Definition v und δ Elemente des V-Moduls $V\delta$ sind, sind dies auch alle bei der Berechnung des größten gemeinsamen Teilers erzielten Zwischenergebnisse und insbesondere auch $\mathrm{ggT}(\delta, v) = \gamma$. Folglich ist auch jedes $z\gamma$ mit $z \in V$ ein Element aus $V\delta$, und es gilt $V\gamma \subseteq V\delta$. Andererseits ist natürlich δ - und somit auch jedes Vielfache von δ - ein Vielfaches von γ. Also folgt $V\delta \subseteq V\gamma$ und somit zusammen $V\delta = V\gamma$. Da γ ein Teiler von v ist, ist $V\gamma$ genau die behauptete Menge $\{i\gamma \mid 0 \leq i < v/\gamma\}$.

Zu (b): Offensichtlich ist f_a ein Gruppenhomomorphismus über V (bezüglich der Addition), d.h. es gilt

$$f_a(x_2) - f_a(x_1) = f_a(x_2 - x_1) = a\delta.$$

Wir müssen also die Anzahl derjenigen $a \in V$ bestimmen, für die $a\delta = d$ gilt. Ist $d \notin V\delta$, so gibt es trivialerweise kein solches a. Sei also $d \in V\delta$. Wir betrachten die Abbildung $\varphi_\delta : V \to V\delta$, $x \mapsto x\delta$. Dann ist die Menge der $a \in V$, für die $a\delta = d$ gilt, gleich ${\varphi_\delta}^{-1}(d)$. Da φ_δ ein Gruppenhomomorphismus und nach Teil (a) surjektiv ist, folgt daß $\varphi_\delta^{-1}(d)$ genau die Kardinalität des Kerns von φ_δ hat (s. dazu auch den Beweis von Satz 3.2.3). Diese beträgt bekanntlich $|V|/|V\delta| = v/(v/\gamma) = \gamma$. Also gilt $a\delta = d$ für genau γ der v möglichen a aus V. ∎

Homogene Funktionen

Wir betrachten nun die Hashklasse $\mathcal{H}^k_{V,\{0\}}$. Es stellt sich heraus, daß diese c-abstandsuniversell für $k \geq u-1$ ist, mit einem durch 3 beschränkten Universalitätsparameter c. Sind $k \geq u/p$ und r Potenzen der gleichen Primzahl p (z.B. $p = 2$), dann ist die Hashklasse sogar 2-abstandsuniversell.

Es sei an dieser Stelle auf die Ähnlichkeit zwischen den Ganzzahlklassen und den Faltungsklassen hingewiesen: Betrachten wir den Fall $u = 2^n$, $r = 2^m$ und $k = 2^{n-1}$. Die Funktionen $h_{a,0}$ bestehen dann aus einer Multiplikation mit einem zufälligen Faktor $a \in \{0,\ldots,2^{n+m-1}-1\}$, aus deren Ergebnis dann die Bits mit den Indizes $n-1,\ldots,n+m-2$ „ausgeschnitten“ werden und den Hashwert bilden (das am wenigsten signifikante Bit habe den Index 0). Ersetzt man also die Multiplikation durch die Faltung der Bitstrings, so erhält man genau die abstandsuniverselle Faltungsklasse aus Satz 3.2.11 (a) für den Körper $\mathbb{K} = \mathbb{Z}_2$.

Der wesentliche Unterschied zwischen beiden Hashklassen ist, daß bei der Faltungsklasse das „Ausschneiden“ des Hashwerts nach der Faltung einen Gruppenhomomorphismus der additivien Gruppe $\mathbb{K}^{n+m-1}$ in die additive Gruppe $\mathbb{K}^m$ bildet, während dies für die Division bei den Ganzzahlklassen nicht der Fall ist. Dies führt dazu, daß ein Universalitätsparameter von 1 bei den Ganzzahlklassen nicht garantiert werden kann.

Um den Universalitätsparameter der Ganzzahlklassen im allgemeinen Fall genau zu beschreiben, benötigen wir die Definition von $\Gamma(u,r,k)$. Dieser Wert ist das Maximum aller derjenigen größten gemeinsamen Teiler von $x \in U$ und v, die keine Teiler von k sind. Formal definiert sei

$$\Gamma(u,r,k) \;:=\; \max\{0,\ \gamma = \mathrm{ggT}(x,kr) \mid 0 \leq x < u \ \wedge\ \gamma \nmid k\}.$$

Auch wenn dieser Term zunächst etwas kompliziert aussieht, sind seine wesentlichen Eigenschaften leicht beschrieben: Offensichtlich ist $\Gamma(u,r,k)$ durch $u-1$ beschränkt, so daß für $k \geq u-1$ einerseits $0 \leq \Gamma(u,r,k)/k \leq 1$ gilt, und andererseits $\Gamma(u,r,k)/k$ in k gegen 0 konvergiert. Ist zudem kr Potenz einer Primzahl p, und gilt $k \geq u/p$, so ist offensichtlich jeder Teiler von kr, der kleiner als u ist, auch eine Potenz von p, und somit ein Teiler von k. Also ist in diesem Fall jedes $\gamma = \mathrm{ggT}(x,kr)$ mit $0 \leq x < u$ ein Teiler von k, und es folgt $\Gamma(u,r,k) = 0$.

4.2.2 Satz.

(a) Sei $k \geq u-1$. Dann ist $\mathcal{H}^k_{V,\{0\}}$ c-abstandsuniversell, wobei $c = 2 + \Gamma(u,r,k)/k$ gilt. Der Universalitätsparameter ist also durch 3 beschränkt und konvergiert in k gegen 2.

(b) Ist $v = kr$ Potenz einer Primzahl p und $k \geq u/p$, dann ist $\mathcal{H}^k_{V,\{0\}}$ 2-abstandsuniversell.

Beweis: Seien $d \in R$ und $x_1, x_2 \in U$ mit $\delta = x_2 - x_1 > 0$ und $\gamma = \mathrm{ggT}(\delta, v)$. Wir betrachten zunächst die Division durch k der Funktionen $h_{a,0}$, die durch g_0 berechnet wird. Ist z eine positive ganze Zahl, dann folgt aus $z \operatorname{div} k = i$, daß $ki \le z < k(i+1)$ ist. Sind also z_1, z_2 Elemente von V mit $g_0(z_2) - g_0(z_1) = d$, dann gibt es $y_1, y_2 \in R$ mit

$$y_2 - y_1 \;=\; d, \quad z_1 \in \{ky_1, \ldots, k(y_1+1)-1\} \quad \text{und} \quad z_2 \in \{ky_2, \ldots, k(y_2+1)-1\}.$$

Die letzte Anforderung läßt sich wegen $y_2 = y_1 + d$ auch als

$$z_2 \in \{ky_1 + kd, \ldots, k(y_1+1) + kd - 1\}$$

schreiben, und es folgt

$$z_2 - z_1 \in \{k(d-1)+1, \ldots, k(d+1)-1\}.$$

Die Wahrscheinlichkeit, mit der $f_a(x_2) - f_a(x_1)$ einen Wert in dieser Menge annimmt, ist also eine obere Schranke für die Wahrscheinlichkeit, daß $h_{a,0}(x_1)$ und $h_{a,0}(x_2)$ einen Abstand von d haben. Nach Lemma 4.2.1 (b) genügt es also, die Wahrscheinlichkeit nach oben zu beschränken, mit der $f_a(\delta)$ einen Wert zwischen $k(d-1)+1$ und $k(d+1)-1$ annimmt. Jeder solche Wert wird mit einer Wahrscheinlichkeit von γ/v angenommen, wenn er ein Vielfaches von γ ist, und sonst gar nicht. Da es höchstens $\lceil 2k/\gamma \rceil$ Vielfache von γ in $\{k(d-1)+1, \ldots, k(d+1)-1\}$ gibt, erhalten wir

$$\mathbf{Prob}\left(h_{a,0}(x_2) - h_{a,0}(x_1) = d\right) \;\le\; \left\lceil \frac{2k}{\gamma} \right\rceil \cdot \frac{\gamma}{v}.$$

Man beachte, daß aufgrund der Voraussetzungen von (a) und (b) γ in jedem Fall höchstens so groß wie k ist. Ist nun γ ein Teiler von k, so ist die rechte Seite der Ungleichung gleich $2/r$. Ist γ hingegen kein Teiler von k, dann ist der Term nicht größer als $(2k+\gamma)/v = (2+\gamma/k)/r$. Beide Terme sind also durch $\big(2+\Gamma(u,r,k)/k\big)/r$ beschränkt, was bedeutet, daß die Hashklasse $2+\Gamma(u,r,k)/k$-universell ist. Somit ist (a) gezeigt, und die Behauptung (b) folgt mit $\Gamma(u,r,k) = 0$ für Primpotenzen kr. ∎

Mit der Methode von Satz 3.2.5 können wir nun leicht aus diesen c-abstandsuniversellen Hashklassen streng c-universelle konstruieren, indem wir zu den Hashwerten einen zufälligen Wert $i \in R$ addieren. Da

$$h_{a,0}(x) \oplus i \;=\; \big((ax) \bmod v\big) \operatorname{div} k \oplus i \;=\; \big((ax+ki) \bmod v\big) \operatorname{div} k \;=\; h_{a,ki}(x)$$

gilt, können wir die so resultierenden Hashklassen auch mit unserem Schema der Ganzzahlklassen beschreiben:

4.2.3 Korollar. *Sei* $B = \{ik \mid 0 \le i < r\}$.

(a) Ist $k \ge u-1$, *dann ist* $\mathcal{H}^k_{V,B}$ *streng* c*-universell mit* $c = 2+\Gamma(u,r,k)$.

(b) Ist $v = kr$ *Potenz einer Primzahl* p *und* $k \ge u/p$, *dann ist* $\mathcal{H}^k_{V,B}$ *streng 2-universell.*

Inhomogene Funktionen

Im letzten Abschnitt wurden abstandsuniverselle Hashklassen entwickelt, die ohne Addition auskamen. Natürlich ist der Zeitbedarf für eine Addition im Vergleich zu einer Multiplikation auf der gängigen Hardware eher unbedeutend, so daß man diesen in vielen Fällen in Kauf nehmen kann, wenn man dadurch einen besseren Universalitätsparameter erreicht.

Warum eine zusätzliche Addition zu einer Verbesserung des Universalitätsparameters führen kann, soll am Beispiel der Faltungsklassen erläutert werden. Deren Abstandsuniversalität (Satz 3.2.11 (a)) läßt sich nämlich auch wie folgt erklären: Einerseits sind die Funktionen dieser Hashklasse - wie man leicht nachweisen kann - gleichverteilend für alle Schlüssel bis auf das 0-Element. Andererseits sind sie aber auch Homomorphismen. Das bedeutet, daß jeder Schlüssel $d \neq 0$ unter einer zufälligen Funktion jeden Funktionswert mit gleicher Wahrscheinlichkeit annimmt, und wegen der Homomorphieeigenschaft somit jedes Schlüsselpaar mit Abstand d auch jeden Abstand mit gleicher Wahrscheinlichkeit annimmt (dieser Zusammenhang zwischen Homomorphismen und gleichverteilenden Hashklassen findet sich auch bei anderen Konstruktionen, wie z.B. den Körperklassen wieder; auf ihn wurde explizit von Krawczyk (1994) hingewiesen).

Auch die 2-abstandsuniverselle Ganzzahlklasse (für eine Primpotenz kr) des letzten Satzes ist - wie man leicht aus Lemma 4.2.1 ableiten kann - für alle Schlüssel aus $U\backslash\{0\}$ gleichverteilend. Ihre Funktionen sind aber - im Unterschied zu denen der Faltungsklasse - keine Homomorphismen. Der Grund ist bei der Division, also bei den Funktionen g_0, zu suchen, da es Werte $x_1, x_2 \in M$ gibt mit $g_0(x_2) - g_0(x_1) \neq g_0(x_2 - x_1)$ (solche Werte sind z.B. 3 und 4 für $k = 4$, da $\lfloor 4/4 \rfloor - \lfloor 3/4 \rfloor = 1$, aber $\lfloor (4-3)/4 \rfloor = 0$ ist). Genauer gesagt ist $g_0(x_2) - g_0(x_1)$ entweder gleich $g_0(x_2 - x_1)$ oder $g_0(x_2 - x_1) + 1$. Indem wir nun als Abbildung $V \to R$ eine Funktionen g_b mit zufälligem b wählen, verteilen sich die Werte $g_b(x_2) - g_b(x_1)$ mit einer „kontrollierten“ Wahrscheinlichkeit auf die zwei möglichen Ergebnisse, wie das folgende Lemma verdeutlicht. Der technische Beweis des Lemmas folgt zusammen mit den restlichen Beweisen diese Abschnitts weiter unten.

4.2.4 Lemma. *Seien $x_1, x_2 \in V$ mit $\delta = x_2 - x_1$ und $\varepsilon = (\delta \bmod k)/k$. Dann gilt für ein beliebiges $d \in R$ und zufälliges $b \in B = \{0, \ldots, k-1\}$*

$$\textbf{Prob}\left(g_b(x_2) - g_b(x_1) = d\right) \;=\; \begin{cases} 1-\varepsilon & \text{falls } d = \delta \operatorname{div} k \\ \varepsilon & \text{falls } d = \delta \operatorname{div} k + 1 \\ 0 & \text{sonst.} \end{cases}$$

Diese Wahrscheinlichkeitsverteilung genügt, um die gleichen Ergebnisse wie ein Homomorphismums zu erzielen. Somit erhalten wir, indem wir die Funktionen $h_{a,b}$ mit zufälligem b benutzen, den bestmöglichen Universalitätsparameter von 1 für den Fall, daß kr Potenz einer Primzahl ist. Ist dies nicht der Fall, so ist die Hashklasse nicht ganz gleichverteilend, und der Universalitätsparameter ist etwas größer. Wir können ihn dann mit einem etwas technischen Term beschreiben. Sei für $\gamma \leq 0$

$$\vartheta(k,\gamma) \;=\; \begin{cases} \dfrac{1}{4\lfloor k/\gamma \rfloor \left(\lfloor k/\gamma \rfloor + 1\right)} & \text{falls } \gamma > 0 \\ 0 & \text{falls } \gamma = 0. \end{cases}$$

Der Universalitätsparameter der Hashklassen des nächsten Satzes ist durch $1 + \vartheta\left(k, \Gamma(u,r,k)\right)$ beschränkt, so daß wir den Wert von $\vartheta := \vartheta\left(k, \Gamma(u,r,k)\right)$ untersuchen sollten. Ist kr Potenz

der Primzahl p und $k \geq u/p$, so ist wegen $\Gamma(u,r,k) = 0$ auch $\vartheta = 0$. Im allgemeinen Fall ist $\Gamma(u,r,k)$ durch $u-1$ beschränkt. Also erhalten wir für $k \geq u-1$ sofort eine obere Schranke von $1/8$ für ϑ und zudem $\vartheta = O(1/k^2)$. Somit konvergiert ϑ in k relativ schnell gegen 0.

4.2.5 Satz. *Sei $B = \{0,\ldots,k-1\}$.*

(a) Ist $k \geq u-1$, dann ist $\mathcal{H}^k_{V,B}$ c-abstandsuniversell mit einem durch $9/8$ beschränkten Universalitätsparameter $c = 1+\vartheta\big(k,\Gamma(u,r,k)\big) = 1+O(1/k^2)$.

(b) Ist v Potenz einer Primzahl p und $k \geq u/p$, dann ist $\mathcal{H}^k_{V,B}$ 1-abstandsuniversell.

Der Beweis folgt im nächsten Abschnitt.

Mit der Technik aus Satz 3.2.5 erhalten wir aus dieser Konstruktion wieder streng c-universelle Hashklassen, wobei c wie im obigen Satz ist. Dabei handelt es sich um die von Dietzfelbinger (1996) untersuchte „lineare“ Hashklasse (s. S. 42).

4.2.6 Korollar.

(a) Ist $k \geq u-1$, dann ist $\mathcal{H}^k_{V,V}$ streng c-universell mit $c = 1+\vartheta\big(k,\Gamma(u,r,k)\big) \leq 9/8$.

(b) Ist v Potenz einer Primzahl p und $k \geq u/p$, dann ist $\mathcal{H}^k_{V,V}$ streng c-universell.

Dietzfelbinger hat für den allgemeinen Fall (kr keine Primpotenz) eine obere Schranke von $5/4$ für den Universalitätsparameter bewiesen. Genauer besagt seine Schranke, daß die lineare Hashklasse einen Universalitätsparameter von höchstens

$$1+\left(\frac{\max\big\{\mathrm{ggT}\,(x,kr)\,\big|\,1\leq x\leq u-1\big\}}{2k}\right)^2$$

besitzt, was z.B. für $k = u-1$ einen Wert von genau $5/4$ ergibt. Unsere neuen Methoden liefern also für den allgemeinen Fall eine leichte Verbesserung. Für den Spezialfall, bei dem kr eine Primpotenz ist, hat Dietzfelbinger das gleiche Ergebnis erzielt (wenn auch mit einer anderen Beweismethode). Später verbessern wir solche Hashklassen jedoch hinsichtlich ihrer Kardinalität.

Analyse der Division nach einer zufälligen Addition

Wir werden uns in diesem Abschnitt mit dem Beweis zu Satz 4.2.5 beschäftigen. Bevor wir Lemma 4.2.4 beweisen, führen wir einen Term ein, der beschreibt, für welche x_1, x_2 die Ganzzahldivision „abstandserhaltend“ ist, d.h. unter welchen Umständen $(x_2-x_1)\operatorname{div}\zeta = x_2\operatorname{div}\zeta - x_1\operatorname{div}\zeta$ für einen ganzzahligen Divisor ζ gilt.

Für natürliche Zahlen x_1, x_2 und ζ sei

$$T_\zeta(x_1,x_2) \;:=\; x_2\operatorname{div}\zeta - x_1\operatorname{div}\zeta - (x_2-x_1)\operatorname{div}\zeta.$$

4.2.7 Lemma. *Es gilt*

$$T_\zeta(x_1,x_2) \;=\; \begin{cases} 0 & \text{falls } x_1 \bmod \zeta \leq x_2 \bmod \zeta \text{ ist, und} \\ 1 & \text{sonst.} \end{cases}$$

Insbesondere ist $T_\zeta(x_1,x_2) = 0$, wenn ζ ein Teiler von $\big|x_2-x_1\big|$ ist.

Beweis: Für $i = 1,2$ sei $\mu_i = x_i \bmod \zeta$ und $\lambda_i = x_i \operatorname{div} \zeta$. Dann ist $x_i = \lambda_i \zeta + \mu_i$, und es folgt

$$T_\zeta(x_1,x_2) \;=\; \lambda_2 - \lambda_1 - (\lambda_2\zeta + \mu_2 - \lambda_1\zeta - \mu_1) \operatorname{div} \zeta \;=\; -(\mu_2 - \mu_1) \operatorname{div} \zeta.$$

Für $\mu_1 \le \mu_2$ ist dies offensichtlich 0 und ansonsten 1. Ist ζ ein Teiler von $|x_2 - x_1|$, dann folgt leicht, daß $x_1 \bmod \zeta = x_2 \bmod \zeta$ ist, und somit ist mit dem bisher Bewiesenen wie behauptet $T_\zeta(x_1,x_2) = 0$. ∎

Der Term $T_\zeta(x_1,x_2)$ spielt bei folgendem, wie auch bei späteren Beweisen eine wichtige Rolle.

Beweis zu Lemma 4.2.4: Sei $\delta^* = \delta \bmod k$. Es gilt offensichtlich

$$g_b(x_2) - g_b(x_1) \;=\; (x_2 + b) \operatorname{div} k - (x_1 + b) \operatorname{div} k \;=\; \delta \operatorname{div} k + T_k(x_1 + b, x_2 + b).$$

Da $T_k(x_1 + b, x_2 + b)$ nach Lemma 4.2.7 aus $\{0,1\}$ ist, genügt es zu zeigen, daß für genau δ^* der k möglichen b-Werte $T_k(x_1 + b, x_2 + b) = 1$ gilt. Gemäß Lemma 4.2.7 ist dies äquivalent dazu, daß $(x_1 + b) \bmod k$ für genau δ^* der b-Werte größer als $(x_2 + b) \bmod k$ ist. Offensichtlich nimmt $(x_1 + b) \bmod k$ für alle $b \in B$ jeden Wert aus $\{0,\ldots,k-1\}$ genau einmal an, und wir können die $b \in B$ mit $b_0,\ldots,b_{k-1}$ so bezeichnen, daß $(x_1 + b_i) \bmod k = i$ für alle $0 \le i < k$ gilt. Somit hat $(x_2 + b_i) \bmod k$ dann einen Wert von $(i + \delta) \bmod k = (i + \delta^*) \bmod k$. Da $i > (i + \delta^*) \bmod k$ genau dann gilt, wenn $i + \delta^* \ge k$ ist, folgt

$$T_k(x_1 + b_i, x_2 + b_i) \;=\; 1 \quad\Longleftrightarrow\quad i + \delta^* \;\ge\; k.$$

Dies ist aber genau für $i \in \{k - \delta^*,\ldots,k-1\}$, also für δ^* der $b_i \in B$ der Fall. ∎

Wir benutzen das eben bewiesene Lemma im Beweis zum nächsten, das im Wesentlichen die Verteilung von $g_b(x_2) - g_b(x_1)$ untersucht, wenn x_2 und x_1 Funktionswerte der Abbildung f_a für zufälliges a sind. Die Aussage ist etwas allgemeiner gehalten als für den Beweis zu Satz 4.2.5 notwendig, da sie in dieser Form später noch einmal benötigt wird.

4.2.8 Lemma. *Seien für $1 \le i \le n$ die Paare (x_i, x_i') aus V^2 so, daß alle zugehörigen $\delta_i := x_i' - x_i$ paarweise verschieden sind, und eine Menge $\{j\gamma + C \mid 0 \le j < v/\gamma\}$ für ein festes $C \in V$ und ein $1 \le \gamma \le k$ bilden. Dann gilt für beliebiges $d \in R$ und zufällig gewähltes $(i,b) \in \{1,\ldots,n\} \times \{0,\ldots,k-1\}$*

$$\mathbf{Prob}\left(g_b(x_i') - g_b(x_i) = d\right) \;\le\; \begin{cases} 1/r & \text{falls } \gamma \text{ ein Teiler von } k \text{ ist} \\ \big(1 + \vartheta(k,\gamma)\big)/r & \text{sonst.} \end{cases}$$

Beweis: Sei $\varepsilon_i = (\delta_i \bmod k)/k$ für $1 \le i \le n$. Wir betrachten zunächst die Verteilung von $g_b(x_i') - g_b(x_i)$ für ein festes i und zufälliges b. Es hat $\delta_i \operatorname{div} k$ den Wert $d-1$, falls $\delta_i \in \{k(d-1),\ldots,kd-1\}$ und den Wert d, falls $\delta_i \in \{kd,\ldots,k(d+1)-1\}$. Mit Lemma 4.2.4 beträgt die Wahrscheinlichkeit, daß der Abstand $g_b(x_i') - g_b(x_i)$ den Wert d annimmt, in diesen Fällen ε_i bzw. $1 - \varepsilon_i$.

Seien nun o.B.d.A. die δ_i so mit den Indizes $1,\ldots,n$ versehen, daß $\delta_1 < \ldots < \delta_t$ genau die Elemente aus $\{k(d-1),\ldots,kd-1\}$ und $\delta_{t+1} < \ldots < \delta_{t+t'}$ genau die Elemente aus $\{kd,\ldots,k(d+1)-1\}$ sind. Wir erhalten so für jetzt zufälliges i und zufälliges b

$$p := \mathbf{Prob}\left(g_b(x_i') - g_b(x_i) = d\right) \;=\; \frac{1}{n}\left(\sum_{i=1}^{t} \varepsilon_i + \sum_{i=t+1}^{t+t'} (1 - \varepsilon_i)\right). \tag{4.1}$$

Da es in einer Menge von k aufeinanderfolgenden Zahlen entweder $\lfloor k/\gamma \rfloor$ oder $\lceil k/\gamma \rceil$ Zahlen der Form $j\gamma + C$ für festes C gibt, sind $t, t' \in \{\lfloor k/\gamma \rfloor, \lceil k/\gamma \rceil\}$. Wir unterscheiden also die drei Fälle $t = t'$, $t = t' - 1 = \lfloor k/\gamma \rfloor$ und $t = t' + 1 = \lceil k/\gamma \rceil$.

1. Fall: $t = t'$. Dieser Fall tritt offenbar zumindest dann ein, wenn γ ein Teiler von k ist. Um die Summe aus (4.1) abzuschätzen, summieren wir die Paare ε_i und $1-\varepsilon_{i+t}$ für $1 \le i \le t$. Da für diese i

$$\delta_i = (d-1)k + \delta_i \bmod k \qquad \text{und} \qquad \delta_{i+t} = dk + \delta_{i+t} \bmod k$$

gilt, folgt

$$\delta_i \bmod k - \delta_{i+t} \bmod k \;=\; \delta_i - \delta_{i+t} + k \;=\; -\gamma t + k.$$

Somit ist

$$\varepsilon_i + 1 - \varepsilon_{i+t} \;=\; 1 + \frac{\delta_i \bmod k - \delta_{i+t} \bmod k}{k} \;=\; 2 - \frac{\gamma t}{k}. \tag{4.2}$$

Eingesetzt in (4.1) erhalten wir mit $n = v/\gamma = kr/\gamma$

$$p \;=\; \frac{1}{n} \sum_{i=1}^{t} \Big(2 - \frac{\gamma t}{k}\Big) \;=\; \frac{2t}{n} - \frac{\gamma t^2}{kn} \;=\; \frac{2t}{r} \cdot \frac{\gamma}{k} - \frac{t^2}{r} \cdot \Big(\frac{\gamma}{k}\Big)^2. \tag{4.3}$$

Wir maximieren p in Abhängigkeit von γ/k, indem wir die reelle Funktion $f(x) = \lambda x - \tau x^2$ betrachten. Deren Differenzialquotient berechnet sich zu $\lambda - 2\tau x$ und hat eine Nullstelle für $x = \lambda/(2\tau)$. Da die Funktion f offensichtlich konkav ist, hat sie an der Stelle $x = \lambda/(2\tau)$ ein globales Maximum, und wir erhalten $f(x) \le f\big(\lambda/(2\tau)\big) = \lambda^2/(4\tau)$. Mit $\lambda = 2t/r$ und $\tau = t^2/r$ folgt somit aus (4.3)

$$p \;\le\; \frac{(2t/r)^2}{4t^2/r} \;=\; \frac{1}{r}.$$

Da $t = t'$ schon dann gilt, wenn γ ein Teiler von k ist, ist somit bereits der erste Teil der Behauptung (γ teilt k) gezeigt.

2. Fall: $t = t' - 1 = \lfloor k/\gamma \rfloor$. Wir summieren diesmal über die Paare ε_i und $1 - \varepsilon_{i+t+1}$ für $i = 1, \ldots, t$. Schließlich muß zur Bestimmung von p noch der Summand $1 - \varepsilon_{t+t'}$ addiert werden. D.h. wir formen (4.1) um zu

$$p \;=\; \frac{1}{n} \cdot \left[\left(\sum_{i=1}^{t} \varepsilon_i + 1 - \varepsilon_{i+t+1} \right) + 1 - \varepsilon_{t+t'} \right].$$

Analog zu (4.2) ist $\varepsilon_i + 1 - \varepsilon_{i+t+1}$ gleich $2 - \gamma(t+1)/k$, und wir erhalten, indem wir $1 - \varepsilon_{t+t'}$ nach oben durch 1 abschätzen,

$$p \;\le\; \frac{2t}{n} - \frac{\gamma t(t+1)}{kn} + \frac{1}{n} \;=\; \frac{2t+1}{r} \cdot \frac{\gamma}{k} - \frac{t(t+1)}{r} \cdot \Big(\frac{\gamma}{k}\Big)^2.$$

Für $t = 0$ ist p dann offensichtlich durch $\gamma/(rk)$ beschränkt, was aufgrund der Voraussetzung $\gamma \le k$ nicht größer als $1/r$ ist. Wir können also $t > 0$ annehmen. Durch Einsetzen der entsprechenden Parameter in die Funktion $f(x) = \lambda x - \tau x^2$ erhalten wir analog zum ersten Fall jetzt eine obere Schranke von

$$p \;\le\; \frac{(2t+1)^2/r^2}{4t(t+1)/r} \;=\; \frac{1}{r} \cdot \frac{4t^2+4t+1}{4t^2+4t} \;=\; \frac{1}{r} \cdot \left(1 + \frac{1}{4t(t+1)}\right).$$

Mit $t = \lfloor k/\gamma \rfloor$ ist $4t(t+1) = \vartheta(k,\gamma)$, und es folgt die Behauptung.

3. Fall: $t = t' + 1 = \lceil k/\gamma \rceil$. Wir summieren jetzt über die Paare ε_i und $1 - \varepsilon_{i+t}$, aber diesmal für $i = 1, \ldots, t'$, so daß der Summand ε_t zusätzlich addiert werden muß. Wir erhalten analog zu den vorigen Fällen mit $\varepsilon_i + 1 - \varepsilon_{i+t} = 2 - \gamma t/k$

$$p \;\leq\; \frac{t'}{n}\Big(2 - \frac{\gamma t}{k}\Big) + \frac{\varepsilon_t}{n}.$$

Indem wir ε_t nach oben durch 1 abschätzen, folgt wegen $t = t' + 1$

$$p \;\leq\; \frac{2t'}{n} - \frac{\gamma t'(t'+1)}{kn} + \frac{1}{n}.$$

Dies ist wegen $t' = \lfloor k/\gamma \rfloor$ die gleiche obere Schranke wie im vorigen Fall, und die Behauptung folgt analog zu diesem. ∎

Aufgrund dieser Vorarbeit folgt Satz 4.2.5 fast unmittelbar:

Beweis zu Satz 4.2.5: Seien x_1, x_2 Schlüssel aus U mit $x_1 < x_2$, sowie $d \in R$. Sei weiterhin $\delta_a = f_a(x_2) - f_a(x_1)$ für $a \in V$. Gemäß Lemma 4.2.1 sind die δ_a gleichverteilt über $V\delta = \{j\gamma \mid 0 \leq j < v/\gamma\}$, wobei $\gamma = \mathrm{ggT}\big(x_2 - x_1, v\big)$ ist. Aufgrund der Voraussetzungen gilt dann $\gamma \leq k$ (für $k \geq u - 1$ ist dies offensichtlich; für den Fall, daß v Potenz der Primzahl p und $k \geq u/p$ ist, ist dann γ auch eine Potenz von p und somit höchstens so groß wie k). Für das Zufallsexperiment mit Ergebnis δ_a bei einer zufälligen Wahl von $a \in V$ macht es keinen Unterschied, ob die δ_a die Menge $V\delta$ bilden und dabei paarweise verschieden oder gleichverteilt über diese Menge sind. Aus diesem Grund können wir Lemma 4.2.8 auf die Paare $\big(f_a(x_1), f_a(x_2)\big)$ mit $a \in V$ anwenden, und erhalten

$$\mathbf{Prob}\,\big(h_{a,b}(x_2) - h_{a,b}(x_1) = d\big) \;\leq\; \begin{cases} 1/r & \text{falls } \gamma \text{ ein Teiler von } k \text{ ist} \\ \big(1 + \vartheta(k,\gamma)\big)/r & \text{sonst.} \end{cases}$$

Ist kr eine Primpotenz und $k \geq u/p$, dann ist γ offensichtlich immer ein Teiler von k, und die Hashklasse somit 1-abstandsuniversell. Andernfalls ist mit $k \geq u - 1$ der Wert γ durch $\Gamma(u, r, k)$ beschränkt, und es folgt die Behauptung. ∎

Verbesserung der Kardinalität für Potenzen von Primzahlen

Wie bereits erwähnt, ist der Fall, in dem $v = kr$ eine Zweierpotenz ist, aus Gründen der Effizienz der Wichtigste. In solchen Fällen läßt sich die Kardinalität der abstandsuniversellen Hashklasse aus Satz 4.2.5 noch verbessern. Dazu seien $v = 2^n$ und $r = 2^m$ (also $k = 2^{n-m}$ und $u \leq k$). Wir zeigen, daß für die Addition eines zufälligen Parameters $b \in \{0, \ldots, k-1\}$ nur die vordere Hälfte der Bits von b zufällig gewählt werden muß. Genauer gesagt, genügt es, b zufällig aus der Menge

$$B(\zeta, k) \;:=\; \{i\zeta \mid 0 \leq i < k/\zeta\}$$

auszuwählen, wobei $\zeta = p^{\lceil (n-m)/2 \rceil}$ ist. Tatsächlich gilt dies nicht nur für Zweierpotenzen kr, sondern sogar für Potenzen beliebiger Primzahlen.

4.2.9 Satz. *Seien $r = p^m$ und $u \leq kp$ mit $k = p^{n-m}$ für eine Primzahl p. Ist $\zeta = p^\tau$ mit $0 \leq \tau \leq \lceil (n-m)/2 \rceil$ und $B = B(\zeta, k)$, so ist die Ganzzahlklasse $\mathcal{H}^k_{V,B}$ abstandsuniversell.*

Bevor wir den Satz beweisen, konstruieren wir aus dieser abstandsuniversellen Hashklasse auf die übliche Weise gemäß Satz 3.2.5 eine streng universelle. Die Menge der Summanden b ermittelt sich zu

$$\{b+jk \mid b \in B(\zeta,k), 0 \le j < k\} \;=\; \{i\zeta \mid 0 \le i < v/\zeta\} \;=\; B(\zeta,v).$$

4.2.10 Korollar. *Seien $r = p^m$ und $u \le kp$ für $k = p^{n-m}$ und eine Primzahl p. Ist $\zeta = p^\tau$ mit $0 \le \tau \le \lceil (n-m)/2 \rceil$ und $B = B(\zeta,v)$, so ist die Ganzzahlklasse $\mathcal{H}^k_{V,B}$ streng universell.*

Der Beweis von Satz 4.2.9 ist größtenteils technischer Natur. Zunächst benötigen wir eine Verallgemeinerung von Lemma 4.2.4, die auf die Auswahl von $b \in B(\zeta,k)$ angepaßt ist. Es sei an die Definition von $T_\zeta(x_1,x_2)$ auf S. 48 erinnert.

4.2.11 Lemma. *Es seien $x_1,x_2 \in V$ mit $\delta = x_2 - x_1$, sowie $\mu_i = x_i \bmod k$ $(i = 1,2)$ und ζ ein Teiler von k. Ist $d \in R$ beliebig und b zufällig aus $B(\zeta,k)$, dann gilt*

$$\mathbf{Prob}\left(g_b(x_2) - g_b(x_1) = d\right) \;=\; \begin{cases} 1-\varepsilon_\zeta(x_1,x_2) & \text{falls } d = \delta \operatorname{div} k, \\ \varepsilon_\zeta(x_1,x_2) & \text{falls } d = \delta \operatorname{div} k + 1 \text{ und} \\ 0 & \text{sonst,} \end{cases}$$

wobei

$$\varepsilon_\zeta(x_1,x_2) \;=\; \frac{\Big((\mu_2-\mu_1) \operatorname{div} \zeta + T_\zeta(\mu_1,\mu_2)\Big) \bmod (k/\zeta)}{k/\zeta}.$$

Beweis: Sei $\Delta_b = g_b(x_2) - g_b(x_1)$. Aus der Definition von $T_k(x_1,x_2)$ folgt unmittelbar:

$$\Delta_b \;=\; \delta \operatorname{div} k + T_k(x_1+b, x_2+b).$$

Nach Lemma 4.2.7 ist $\Delta_b = \delta \operatorname{div} k$ genau dann, wenn $(\mu_1+b) \bmod k \le (\mu_2+b) \bmod k$ ist. Ansonsten ist $\Delta_b = \delta \operatorname{div} k + 1$. Es genügt also zu zeigen, daß die Wahrscheinlichkeit, daß $\Delta_b = \delta \operatorname{div} k$ ist, genau $1-\varepsilon_\zeta(x_1,x_2)$ beträgt.

Wir betrachten zunächst den Fall, daß $\mu_1 \le \mu_2$ ist. Dann ist

$$(\mu_1+b) \bmod k > (\mu_2+b) \bmod k$$

genau für die $b \in \{0,\ldots,k-1\}$, für die $\mu_1 + b < k \le \mu_2 + b$ gilt. Für $b \in B(\zeta,k)$ ist b zudem ein Vielfaches von ζ, und es folgt

$$\Delta_b \;=\; \delta \operatorname{div} k + 1 \quad\Longleftrightarrow\quad \mu_1 + b \;<\; k \;\le\; \mu_2 + b \quad\Longleftrightarrow\quad \mu_1 \operatorname{div} \zeta \;<\; \frac{k}{\zeta} - \frac{b}{\zeta} \;\le\; \mu_2 \operatorname{div} \zeta.$$

Ist hingegen $\mu_1 > \mu_2$, dann gilt $(\mu_1+b) \bmod k \le (\mu_2+b) \bmod k$ genau dann, wenn $\mu_2 + b < k \le \mu_1 + b$ ist. Also gilt in diesem Fall

$$\Delta_b \;=\; \delta \operatorname{div} k \quad\Longleftrightarrow\quad \mu_2 + b \;<\; k \;\le\; \mu_1 + b \quad\Longleftrightarrow\quad \mu_2 \operatorname{div} \zeta \;<\; \frac{k}{\zeta} - \frac{b}{\zeta} \;\le\; \mu_1 \operatorname{div} \zeta.$$

In beiden Fällen beträgt die Anzahl der $b \in B(\zeta,k)$, für die $\Delta_b = \delta \operatorname{div} k$ ist, genau

$$\begin{aligned}(\mu_1 \operatorname{div} \zeta - \mu_2 \operatorname{div} \zeta) \bmod (k/\zeta) \;&=\; k/\zeta - (\mu_2 \operatorname{div} \zeta - \mu_1 \operatorname{div} \zeta) \bmod (k/\zeta) \;= \\ &\qquad k/\zeta - \Big((\mu_2-\mu_1) \operatorname{div} \zeta + T_\zeta(\mu_1,\mu_2)\Big) \bmod (k/\zeta).\end{aligned}$$

Da $B(\zeta,k)$ genau k/ζ Elemente hat, und dieser Term dividiert durch k/ζ genau $1-\varepsilon_\zeta(x_1,x_2)$ ergibt, folgt hieraus die Behauptung. ∎

Wir benötigen noch zusätzlich folgende einfache Rechenregel:

4.2.12 Aussage. *Ist m ein Teiler von $n \in \mathbb{N}$, dann gilt*

$$(x \bmod n) \operatorname{div} m \;=\; (x \operatorname{div} m) \bmod (n/m)$$

Beweis: Sei $i = (x \bmod n) \operatorname{div} m$. Dann ist $x \bmod n \in \{im, \ldots, i(m+1)-1\}$, und somit hat x eine Darstellung als $jn + im + k$ mit $0 \le j$ und $0 \le k < m$. Dann können wir aber $x \operatorname{div} m$ als $jn/m + i + k \operatorname{div} m$ schreiben, und erhalten so $(x \operatorname{div} m) \bmod (n/m) = i$. ∎

Beweis zu Satz 4.2.9: Wie bei den bisherigen Beweisen seien auch hier $d \in R$, $x_1, x_2 \in V$ mit $\delta = x_2 - x_1 > 0$ und $\gamma = \mathrm{ggT}(\delta, v)$. Wir bemerken zunächst, daß γ ein Teiler von k ist. Dies folgt unmittelbar daraus, daß kr eine Potenz der Primzahl p, und γ ein Teiler von kr und zudem kleiner als u, also auch kleiner als kp ist.

Nach Lemma 4.2.1 ist $f_a(x_2) - f_a(x_1)$ gleichverteilt über $V\delta$. Wie bereits mehrfach erwähnt (s. z.B. den Beweis zu Satz 4.2.2), gilt $h_{a,b}(x_2) - h_{a,b}(x_1) = d$ nur dann, wenn $f_a(x_2) - f_a(x_1)$ aus der Menge $\{k(d-1), \ldots, k(d+1)-1\}$ stammt. Wir bezeichnen die Elemente aus $V\delta \cap \{k(d-1), \ldots, kd-1\}$ mit $\Delta_1 < \ldots < \Delta_t$, und die aus $V\delta \cap \{kd, \ldots, k(d+1)-1\}$ mit $\Delta'_1 < \ldots < \Delta'_t$. Da γ ein Teiler von k ist, folgt sofort, daß

$$\Delta_1 = k(d-1), \quad \Delta_2 = k(d-1) + \gamma, \quad \ldots, \quad \Delta_t = kd - \gamma$$

gilt. Analoges gilt auch für die Δ'_i. Somit ist $t = t' = k/\gamma$ und insbesondere $\Delta'_i = \Delta_i + k$.

Es sei nun

$$p_j \;:=\; \mathbf{Prob}\left(h_{a,b}(x_2) - h_{a,b}(x_1) = d \;\middle|\; f_a(x_2) - f_a(x_1) \in \left\{\Delta_j, \Delta'_j\right\}\right).$$

Wir zeigen im Folgenden, daß $p_j = 1/2$ für alle $1 \le j \le t$ ist. Da $f_a(x_2) - f_a(x_1)$ gemäß Lemma 4.2.1 jeden Abstand Δ_j bzw. Δ'_j mit einer Wahrscheinlichkeit von genau γ/v annimmt, folgt dann die Behauptung:

$$\mathbf{Prob}\left(h_{a,b}(x_2) - h_{a,b}(x_1) = d\right) \;=\; \sum_{j=1}^{t} p_j 2\frac{\gamma}{v} \;=\; t\frac{\gamma}{v} \;=\; \frac{1}{r}.$$

Sei also $1 \le j \le t$ beliebig und $s = \delta/\gamma$. Dann ist $\mathrm{ggT}(s, v) = \mathrm{ggT}(\delta/\gamma, v) = 1$. Dies bedeutet aber nach Lemma 4.2.1, daß $Vs = V$ ist, und somit gibt es ein bezüglich der Multiplikation inverses Element s^{-1} in V (d.h. es gilt $s^{-1}s = 1$). Wir definieren nun für jedes $a \in V$ einen „Partner“ $a^* = a + s^{-1}k/\gamma$, und zeigen später folgende Eigenschaften von a und dessen Partner a^*:

(i) Es ist $a^*\delta = a\delta + k$, d.h. es gilt

$$f_a(x_2) - f_a(x_1) \;=\; \Delta_j \quad\Longleftrightarrow\quad f_{a^*}(x_2) - f_{a^*}(x_1) \;=\; \Delta'_j.$$

(ii) Es ist $\varepsilon_\zeta(ax_1, ax_2) = \varepsilon_\zeta(a^*x_1, a^*x_2)$.

Wir betrachten die Teilmengen A_j und A'_j von V, für die $f_a(x_2) - f_a(x_1)$ mit $a \in A_j$ auf Δ_j und mit $a \in A'_j$ auf Δ'_j abgebildet wird. Aussage (i) bedeutet, daß die Elemente aus A'_j genau die Partner der Elemente aus A_j sind. Somit ist gemäß Lemma 4.2.11

$$\begin{aligned} p_j &= \frac{1}{2\left|A_j\right|} \sum_{a \in A_j} \Big(\mathbf{Prob}\left(h_{a,b}(x_2) - h_{a,b}(x_1) = d\right) + \mathbf{Prob}\left(h_{a^*,b}(x_2) - h_{a^*,b}(x_1) = d\right) \Big) \\ &= \frac{1}{2\left|A_j\right|} \sum_{a \in A_j} \left(\varepsilon_\zeta(ax_1, ax_2) + 1 - \varepsilon_\zeta(a^* x_1, a^* x_2)\right). \end{aligned}$$

Mit Aussage (ii) folgt dann sofort, daß $p_j = 1/2$ ist, und somit die Behauptung. Es müssen also nur noch (i) und (ii) gezeigt werden.

Da $a\delta = f_a(x_2) - f_a(x_1)$ ist, folgt (i) sofort aus dieser Gleichung:

$$f_{a^*}(x_2) - f_{a^*}(x_1) = a^*\delta = \left(a + s^{-1}\frac{k}{\gamma}\right)\delta = a\delta + s^{-1}\frac{k}{\gamma}\gamma s = a\delta + k = f_a(x_2) - f_a(x_1) + k.$$

Für (ii) betrachten wir $\mu_i = ax_i \bmod k$ und $\mu_i^* = a^* x_i \bmod k$ $(i = 1,2)$. Da ζ ein Teiler von k ist, folgt mit Aussage 4.2.12

$$\begin{aligned} \left((\mu_2 - \mu_1) \operatorname{div} \zeta\right) \bmod (k/\zeta) &= \left((\mu_2 - \mu_1) \bmod k\right) \operatorname{div} \zeta \\ &= \Big[\left((ax_2) \bmod k - (ax_1) \bmod k\right) \bmod k\Big] \operatorname{div} \zeta \\ &= (\Delta_j \bmod k) \operatorname{div} \zeta \;=\; \left((\Delta_j + k) \bmod k\right) \operatorname{div} \zeta \\ &= (\Delta'_j \bmod k) \operatorname{div} \zeta \\ &\;\;\vdots \quad \textit{(analoge Umformung)} \\ &= \left((\mu_2^* - \mu_1^*) \operatorname{div} \zeta\right) \bmod (k/\zeta). \end{aligned}$$

Angenommen, es gilt auch

$$T_\zeta(\mu_1, \mu_2) \;=\; T_\zeta(\mu_1^*, \mu_2^*), \tag{4.4}$$

dann folgt aus der Definition von ε_ζ sofort die Behauptung (ii).

Wir müssen also nur noch Gleichung (4.4) zeigen. Ist $\zeta \le \gamma$, dann ist ζ ein Teiler von γ, da beides Potenzen der gleichen Primzahl p sind. Da sowohl δ als auch k Vielfache von γ sind, ist dann ζ ein Teiler von beiden und demnach auch ein Teiler von $\left|(ax_2) \bmod k - (ax_1) \bmod k\right|$. Nach Lemma 4.2.7 ist somit $T_\zeta(\mu_1, \mu_2) = 0$, und auf völlig analoge Weise folgt auch $T_\zeta(\mu_1^*, \mu_2^*) = 0$, so daß (4.4) gilt.

Sei nun $\zeta > \gamma$, d.h. γ ist ein echter Teiler von ζ. Da nach Voraussetzung $k = p^{n-m}$ und $\zeta \le p^{\lceil (n-m)/2 \rceil}$ ist, folgt $k/\gamma \ge p^{n-m-\lceil (n-m)/2 \rceil} \ge \zeta$. Somit ist ζ ein Teiler von k/γ. Per Definition ist $a^* x_1 = ax_1 + s^{-1} x_1 (k/\gamma)$, und es folgt

$$(ax_1) \bmod \zeta \;=\; (a^* x_1) \bmod \zeta. \tag{4.5}$$

Weiterhin ist ζ ein Teiler von k und, da mit (i) $a^*\delta = a\delta + k$ ist, gilt

$$(a\delta) \bmod \zeta \;=\; (a^*\delta) \bmod \zeta. \tag{4.6}$$

Nach Lemma 4.2.7 ist nun $T_\zeta(\mu_1,\mu_2)=0$ genau dann, wenn $(ax_2) \bmod \zeta \le (ax_1) \bmod \zeta$, also wenn

$$(ax_1 + a\delta) \bmod \zeta \;\le\; (ax_1) \bmod \zeta.$$

Analog ist $T_\zeta(\mu_1^*,\mu_2^*)=0$ genau dann, wenn

$$(a^*x_1 + a^*\delta) \bmod \zeta \;\le\; (a^*x_1) \bmod \zeta.$$

Nach (4.5) und (4.6) sind aber beide Ungleichungen äquivalent, und somit folgt (4.4). ∎

§ 3. Universelle und optimal universelle Ganzzahlklassen

Bei den Faltungsklassen (s. Satz 3.2.11) haben wir gesehen, daß die universelle Variante sich von der abstandsuniversellen vor allem in der Länge des zufälligen Vektors unterscheidet, über den die Faltung ausgeführt wird. Während bei der abstandsuniversellen Hashklasse die Faltung mit einem Vektor bestehend aus $n+m-1$ Spalten (für $U=\mathbb{K}^n$ und $R=\mathbb{K}^m$) berechnet werden muß, genügen für die universelle Hashklasse bereits n Spalten. Aufgrund der bereits angesprochenen Ähnlichkeit zwischen den Faltungsklassen und den Ganzzahlklassen kann man annehmen, daß es auch universelle Ganzzahlklassen gibt, die mit einer Multiplikation über eine kürzere Wortbreite auskommen. Während die Ganzzahlklassen $\mathcal{H}^k_{A,B}$ mit $k \ge u/p$ (und Primpotenz kr) den abstandsuniversellen Faltungsklassen entsprechen, untersuchen wir jetzt in Analogie zu der universellen Faltungsklasse die Ganzzahlklassen mit $k \ge u/r$.

1- und 2-universelle Ganzzahlklassen

Wir werden hier nur den wichtigsten Fall untersuchen, bei dem kr Potenz einer Primzahl ist. Dementsprechend sei von nun an $r=p^m$ und $v=p^n$ für eine Primzahl p und zwei natürliche Zahlen $m<n$. Bei der 1-universellen Faltungsklasse war der multiplikative Faktor ein Vektor, dessen erste Spalte den Eintrag 1 hatte. Dies entspricht bei der Ganzzahlklasse einem Parameter a, dessen letzte Ziffer in einer Zahlendarstellung zur Basis p den Wert 1 hat. D.h., wir können a als $ip+1$ für $0 \le i < p^{n-1}$ darstellen. Im Folgenden bezeichnen wir also die Menge $\{ip+1 \mid 0 \le i < p^{n-1}\}$ mit V^*.

Ist $p=2$, dann handelt es sich bei V^* offensichtlich um die Menge der ungeraden Zahlen aus V. Die wesentlichen Eigenschaften der Verteilung zweier Schlüssel unter der Abbildung f_a mit $a \in V^*$ zeigt folgendes Lemma. Es handelt sich dabei um eine Variante von Lemma 4.2.1.

4.3.1 Lemma. *Sei $d \in V$ und $x_1,x_2 \in V$ mit $\delta = x_2 - x_1 > 0$ und $\gamma = \mathrm{ggT}(\delta,v)$. Weiterhin sei $z = (\delta/\gamma) \bmod p$. Es gilt*

(a) $V^*\delta \;=\; \{\gamma(ip+z) \mid 0 \le i < v/(\gamma p)\}$.

(b) $\underset{a\in V^*}{\mathbf{Prob}}\big(f_a(x_2)-f_a(x_1)=d\big) \;=\; \underset{a\in V^*}{\mathbf{Prob}}\big(a\delta=d\big) \;=\; \begin{cases} \gamma p/v & \text{falls } d \in V^*\delta \\ 0 & \text{sonst.} \end{cases}$

Beweis: Die Verteilung von $a\delta$ mit $a \in V^*$ entspricht per Definition der Verteilung von $(ip+1)\delta$ mit $0 \le i < v/p$ und diese wiederum der Verteilung von $ip\delta+\delta$ mit $i \in V$. Weiterhin gilt $\gamma p = \mathrm{ggT}(p\delta, v)$, und somit folgt gemäß Lemma 4.2.1, daß $a\delta$ genau die Werte aus $\{i\gamma p+\delta \mid i \in V\}$ annimmt und zwar jeweils mit gleicher Wahrscheinlichkeit. Wir können δ als

$$\delta \operatorname{div}(\gamma p)\cdot\gamma p + \delta \bmod(\gamma p)$$

schreiben. Außerdem folgt $\delta \bmod(\gamma p) = \gamma z$ aus Aussage 4.2.12 und der Definition von z, und somit gilt $V^*\delta = \{\gamma(ip+jp+z) \mid i \in V\}$, wobei $j = \delta \operatorname{div}(\gamma p)$ ist. Das beweist Teil (a), und da jeder Wert aus dieser Menge mit gleicher Wahrscheinlichkeit von $a\delta$ angenommen wird, auch Teil (b). ∎

Der Grund für die Wahl der Menge V^* wird aus folgenden Überlegungen deutlich: Wir haben bereits öfters in diesem Kapitel die Eigenschaft benutzt, daß $h_{a,b}(x_2) - h_{a,b}(x_1)$ nur dann einen Wert d haben kann, wenn der Abstand $f_a(x_2) - f_a(x_1)$ in der Menge

$$\{k(d-1)+1,\ldots,k(d+1)-1\}$$

enthalten ist (s. z.B. den Beweis zu Satz 4.2.2). Für den hier zu betrachtenden Fall bedeutet dies, daß zwei Schlüssel x_1 und x_2 unter einer Funktion $h_{a,b}$ höchstens dann kollidieren (also deren Funktionswerte den Abstand 0 haben), wenn $f_a(x_2) - f_a(x_1)$ aus $\{-k+1,\ldots,k-1\}$ sind. Dies können wir aber bereits für bestimmte Schlüsselpaare ausschließen, wenn wir a aus V^* wählen.

Seien z.B. $x_1, x_2 \in U$ mit $\delta = x_2 - x_1 > 0$ und wie üblich $\gamma = \mathrm{ggT}(\delta, v)$ sowie $z = (\delta/\gamma) \bmod p$. Jeder Abstand $f_a(x_2) - f_a(x_1)$ kann gemäß Lemma 4.3.1 für $a \in V^*$ als $\gamma(ip+z)$ für geeignete i geschrieben werden. Es ist aber $ip+z \neq 0$, da $0 < z < p$ gilt (wäre $z = 0$, dann wäre p ein Teiler von δ/γ, und somit γ nicht der $\mathrm{ggT}(\delta, v)$). Nehmen wir nun den Fall an, in dem $\gamma \ge k$, also aufgrund der Primpotenz-Eigenschaften ein Vielfaches von k ist. Dann ist $f_a(x_2) - f_a(x_1)$ für jedes $a \in V^*$ ungleich 0 und zudem ein Vielfaches von γ und somit auch ein Vielfaches von k. Demnach ist $f_a(x_2) - f_a(x_1)$ nicht in der Menge $\{-k+1,\ldots,k-1\}$ enthalten, und x_1 und x_2 kollidieren unter keiner der Funktionen $h_{a,b}$.

Das bedeutet, daß wir bei der Bestimmung des Universalitätsparameters von Hashklassen $\mathcal{H}^k_{V^*,B}$ nur Schlüsselpaare mit $\gamma < k$ berücksichtigen müssen. Wir halten das Ergebnis dieser Überlegung in folgendem Lemma fest.

4.3.2 Lemma. *Seien $x_1, x_2 \in U$ mit $x_2 > x_1$. Ist $\mathrm{ggT}(x_2 - x_1, v) \ge k$, dann kollidieren x_1 und x_2 unter keiner Funktion $h_{a,b}$ mit $a \in V^*$ und $b \in V$.* ∎

Analog zu den c-abstandsuniversellen Ganzzahlklassen aus den Sätzen 4.2.2 (b) ($c = 2$) und 4.2.9 ($c = 1$) erhalten wir nun c-universelle Hashklassen $\mathcal{H}^k_{A,B}$, wenn wir $k = p^{n-m}$ und $A = V^*$ wählen.

4.3.3 Satz. *Sei $r = p^m$, $v = kr = p^n$ und $u \le p^n$ für eine Primzahl p.*

(a) Die Hashklasse $\mathcal{H}^k_{V^*,\{0\}}$ ist 2-universell.

(b) Sei $\zeta = p^\tau$ mit $0 \le \tau \le \lceil (n-m)/2 \rceil$ und $B = B(\zeta, k) = \{i\zeta \mid 0 \le i < k/\zeta\}$. Dann ist die Hashklasse $\mathcal{H}^k_{V^*,B}$ 1-universell.

Bei Teil (a) handelt es sich für $p = 2$ um die „multiplikative“ Hashklasse, für die Dietzfelbinger, Hagerup, Katajainen und Penttonen (1997) bereits die 2-Universalität gezeigt haben. Neu hingegen ist die Verallgemeinerung für beliebige Primpotenzen, sowie die inhomogene Konstruktion aus Teil (b). Es handelt sich bei dieser um die erste 1-universelle Hashklasse beruhend auf ganzzahliger Arithmetik, die mit einer Multiplikation über die Wortlänge der Schlüssel und (für $p = 2$) ohne Division auskommt. Die Beweise sind nach der bisher geleisteten Vorarbeit fast völlig analog zu denen aus § 2.

Beweis zu Satz 4.3.3: Seien x_1, x_2, δ und γ wie üblich definiert. Gemäß Lemma 4.3.2 müsssen wir nur den Fall $\gamma < k$ (also γp ist ein Teiler von k) betrachten.

Zu (a): Der Beweis ist analog zu dem von Satz 4.2.2: Wir zählen die Anzahl der $a \in V^*$ mit $f_a(\delta) \in \{-k+1,\ldots,k-1\}$. Nach Lemma 4.3.1 liegen höchstens $2k/(\gamma p)$ Werte aus V^* in dieser Menge, und jeder solche Wert wird mit einer Wahrscheinlichkeit von genau $\gamma p/v$ angenommen. Da γp ein Teiler von k ist, folgt

$$\textbf{Prob}\left(h_{a,0}(x_1) = h_{a,0}(x_2)\right) \;\leq\; \frac{\gamma p}{v} \cdot \frac{2k}{\gamma p} \;=\; \frac{2k}{v} \;=\; \frac{2}{r}.$$

Zu (b): Ersetzt man im Beweis zu Satz 4.2.9 V durch V^*, so kann dieser wörtlich für den hier gegebenen Fall übernommen werden, wenn γ ein Teiler von k ist. Ein einziges Detail muß geklärt werden, nämlich daß für jedes Element $a \in V^*$ auch dessen Partner a^* in V^* enthalten ist. Der Partner a^* von a war definiert als $a + s^{-1}k/\gamma$, wobei s^{-1} das inverse Element von $s = \delta/\gamma$ ist. Ist a ein Element von V^*, dann hat es eine Darstellung als $ip+1$ für ein $0 \leq i < v/p$. Somit ist $a^* = ip + 1 + s^{-1}k/\gamma$, und hat - da k ein Vielfaches von γp ist - eine Darstellung als $jp+1$ ($0 \leq j < v/p$). Somit ist auch a^* in V^* enthalten. ∎

Optimal universelle Ganzzahlklassen

Die Grundlage für die Konstruktion optimal universeller Ganzzahlklassen nach den Methoden aus Kapitel 3 liefert Lemma 4.3.2. Demnach kollidieren zwei Schlüssel $x_1 < x_2$ unter keiner der Funktionen $h^k_{a,b}$, wenn $\gamma = \text{ggT}(x_2 - x_1, 2^n)$ nicht kleiner als k ist. Wir erhalten so auf natürliche Weise eine Äquivalenzrelation $\sim$, für die verschiedene Schlüssel einer Äquivalenzklasse nicht miteinander kollidieren. Dazu sei $x_1 \sim x_2$ genau dann, wenn $x_1 = x_2$ ist, oder wenn $\text{ggT}\left(\left|x_2 - x_1\right|, v\right) \geq k$ gilt.

Um zu zeigen, daß $\sim$ tatsächlich eine Äquivalenzrelation ist, muß - da Symmetrie und Reflexivität schon per Definition gegeben sind - nur die Transitivität gezeigt werden. Angenommen, es gilt sowohl $\gamma = \text{ggT}\left(\left|x_2 - x_1\right|, v\right) \geq k$ als auch $\gamma' = \text{ggT}\left(\left|x_3 - x_1\right|, v\right) \geq k$. Da dann sowohl γ und γ' als auch k Potenzen der Primzahl p sind, ist k ein Teiler von γ und von γ'. Somit teilt k auch $\left|x_3 - x_1\right| + \left|x_2 - x_1\right|$ und auch $\left|\left|x_3 - x_1\right| - \left|x_2 - x_1\right|\right|$. Einer dieser beiden Werte ist aber $\left|x_3 - x_2\right|$, was somit ein Vielfaches von k ist. Demnach ist auch $\text{ggT}\left(\left|x_3 - x_2\right|, v\right) \geq k$, und damit die Relation $\sim$ transitiv.

4.3.4 Lemma. *Sei $v = u$. Dann sind die Hashklassen aus Satz 4.3.3 sind $(0|2)$- bzw. $(0|1)$-universell, wobei $\sim$ die dazugehörige Äquivalenzrelation bildet und die Kardinalität der Äquivalenzklassen r beträgt.*

Beweis: Es genügt zu zeigen, daß die Äquivalenzklassen die Kardinalität r haben. Sei $x \in U$ beliebig. Jedes $y \in V$ mit $x \sim y$ können wir als $x + ik$ schreiben, da k ein Teiler des $\mathrm{ggT}(|x-y|, 2^n)$ ist. Andererseits ist $|x-y|$ ein Vielfaches von k für jedes $y = x + ik \neq x$. Also ist die Äquivalenzklasse von x die Menge $\{x + ik \mid i \in V\}$, welche die Kardinalität $v/k = r$ hat. ∎

Wir können also hoffen, daß wir mithilfe der Methoden aus Kapitel 3, § 3 optimal universelle Hashklassen erhalten. Auch hier ist wieder eine Ähnlichkeit zu den Faltungsklassen gegeben. So haben wir zur Konstruktion der optimal universellen Faltungsklasse die Tatsache ausgenutzt, daß die 1-universelle Faltungsklasse tatsächlich $(0|1)$-universell ist (s. Bemerkung 3.3.3). Zwei Schlüssel $(x_0, \ldots, x_{n-1})$ und $(x'_0, \ldots, x'_{n-1})$ liegen bei dieser in der gleichen Äquivalenzklasse, wenn $(x_0, \ldots, x_{n-m-1})$ und $(x'_0, \ldots, x'_{n-m-1})$ gleich sind.

Bei den Ganzzahlklassen verhält es sich fast genauso, denn die Äquivalenzrelation $\sim$ läßt sich auch wie folgt charakterisieren: Sind $x = \langle x_{n-1} \ldots x_0 \rangle$ und $x' = \langle x'_{n-1} \ldots x'_0 \rangle$ zwei Schlüssel aus U ($u = p^n$), dargestellt zur Basis p, so gilt $x \sim x'$ genau dann, wenn $\langle x_{n-m-1} \ldots x_0 \rangle$ und $\langle x'_{n-m-1} \ldots x'_0 \rangle$ gleich sind. Insofern ist es nicht überraschend, daß auch folgende Konstruktion der optimal universellen Faltungsklasse aus Satz 3.3.17 sehr ähnelt.

4.3.5 Satz. *Sei $r = p^m$ und $u = v = kr = p^n$, wobei m ein Teiler von n ist, sowie $\zeta = p^\tau$ mit $0 \leq \tau \leq \lceil (n-m)/2 \rceil$. Weiterhin sei $A_i = V^* p^i$ für $0 \leq i < n$. Ist $A = A_0 \cup A_m \cup \ldots \cup A_{n-m}$ und $B = B(\zeta, k) = \{i\zeta \mid 0 \leq i < k/\zeta\}$, dann ist die Ganzzahlklasse $\mathcal{H}^k_{A,B}$ optimal universell.*

Beweis: Sei $n = lm$. Wir zeigen die Behauptung analog zum Beweis von Satz 3.3.17 mit vollständiger Induktion über l unter Anwendung von Lemma 3.3.13 Für $l = 1$ ist $k = 1$ und $A = V^*$. Für beliebige Schlüssel $x_1 < x_2$ ist dann $\mathrm{ggT}(x_2 - x_1, v) \geq 1 = k$, und gemäß Lemma 4.3.2 kollidieren die Schlüssel unter keiner der Hashfunktionen aus $\mathcal{H}^k_{A,B}$. Demnach ist die Hashklasse optimal universell.

Sei nun $l > 1$. Wir teilen die Hashklasse $\mathcal{H}^k_{A,B}$ in die Hashklassen $\mathcal{H}_1 = \mathcal{H}^k_{A_0,B}$ und $\mathcal{H}_2 = \mathcal{H}^k_{A_m \cup \ldots \cup A_{n-m},B}$ auf. Da $A_0 = V^*$ ist, ist $\mathcal{H}_1$ nach Lemma 4.3.4 $(0|1)$-universell mit Äquivalenzklassengröße r. Es genügt also zu zeigen, daß $\mathcal{H}_2$ die in Lemma 3.3.13 beschriebenen Eigenschaften hat. Dabei ist $u_1 = v/r$ und $u_2 = r$.

Die erste Eigenschaft besagt, daß zwei Schlüssel verschiedener Äquivalenzklassen höchstens mit einer Wahrscheinlichkeit von $(v/r - r)/(v - r)$ kollidieren. Dies ist die Kollisionswahrscheinlichkeit einer optimal universellen Hashklasse mit Universumsgröße v/r und Wertebereich r.

Seien $x_1 < x_2$ Elemente verschiedener Äquivalenzklassen, also mit $\mathrm{ggT}(x_2 - x_1, v) < k$. Jedes $a \in A_m \cup \ldots \cup A_{n-m}$ ist ein Vielfaches von $p^m = r$, so daß wir mit Aussage 4.2.12 für $a' = a/r$ und $b' = b \operatorname{div} r$ schreiben können:

$$\begin{aligned} h^k_{a,b}(x) &= \big((ax+b) \bmod v\big) \operatorname{div} k = \Big(\big((ax+b) \operatorname{div} r\big) \bmod (v/r)\Big) \operatorname{div} (k/r) \\ &= \Big(\big((ax/r) + b \operatorname{div} r\big) \bmod (v/r)\Big) \operatorname{div} (k/r) = h^{k/r}_{a',b'}(x) \\ &= h^{k/r}_{a',b'}\big(x \bmod (v/r)\big). \end{aligned}$$

Da der größte gemeinsame Teiler von $x_2 - x_1$ und v kleiner als k ist, folgt, daß $x_1 \bmod (v/r)$ und $x_2 \bmod (v/r)$ verschiedene Elemente aus $V' = \{0, \ldots, v/r - 1\}$ sind (denn sonst wäre $x_2 - x_1$ ein Vielfaches von $v/r = k$). Somit ist die Kollisionswahrscheinlichkeit von x_1

und x_2 unter der Hashklasse $\mathcal{H}_2$ nicht größer als die maximale Kollisionswahrscheinlichkeit zweier Elemente aus V' unter der Hashklasse $\mathcal{H}^{k/r}_{A',B'} : V' \to R$, wobei A' die Familie $\{a/r \mid a \in A_m \cup \ldots \cup A_{n-m}\}$ und B' die Familie $\{b \operatorname{div} r \mid b \in B\}$ ist. Wir können A' schreiben als $A'_0 \cup \ldots \cup A'_{n-2m}$ mit

$$\begin{aligned} A'_i &= \{a/r \mid a \in A_{i+m}\} = \{a/r \mid a \in V^* p^{i+m}\} \\ &= \left\{ \frac{p^{i+m}(jp+1)}{r} \;\middle|\; 0 \le j < \frac{v}{p^{i+m+1}} \right\} = \left\{ p^i(jp+1) \;\middle|\; 0 \le j < \frac{v/r}{p^{i+1}} \right\} \\ &= (V')^* p^i. \end{aligned}$$

Die Familie B' besteht aus den Zahlen $(i\zeta) \operatorname{div} r$ für $0 \le i < k/\zeta$. Ist $\zeta > r$, so gilt demnach

$$B' = \{\zeta/r, 2(\zeta/r), \ldots, (k/\zeta - 1)(\zeta/r)\} = B(\zeta/r, k/r)$$

Ist hingegen $\zeta \le r$, so nimmt ein zufällig aus B' gewähltes b jeden Wert $0 \le i < k/r$ mit gleicher Wahrscheinlichkeit an. In jedem Fall ändert sich also an der Wahrscheinlichkeitsverteilung für ein zufällig gewähltes b nichts, wenn wir B' durch $B(\zeta', k/r)$ ersetzen, wobei $\zeta' = \max\{1, \zeta/r\}$ ist. Dann gilt $\zeta' = p^{\tau'}$, wobei $\tau' = \max\{0, \tau - m\}$. Wegen $\tau - m \le \lceil (n-m)/2 - m \rceil \le \lceil (n-2m)/2 \rceil$ folgt $0 \le \tau' \le \lceil (n-2m)/2 \rceil$.

Wir fassen zusammen: Die Kollisionswahrscheinlichkeit der Schlüssel x_1 und x_2 unter der Hashklasse $\mathcal{H}_2$ ist nicht größer als die zweier verschiedener Schlüssel unter der Ganzzahlklasse $\mathcal{H}^{k/r}_{A',B'} : V' \to R$ mit $V' = \{0, \ldots, v/r - 1\}$, wobei $A' = A'_0 \cup \ldots \cup A_{n-2m}$ und $B' = B(\zeta', k/r)$ sind mit $\zeta' = p^{\tau'}$ und $0 \le \tau' \le \lceil (n-2m)/2 \rceil$. Per Induktionsvoraussetzungen ist diese Hashklasse $\mathcal{H}^{k/r}_{A',B'}$ optimal universell, und x_1 und x_2 kollidieren daher höchstens mit einer Wahrscheinlichkeit von $(v/r - r)/(v - r)$. Somit erfüllt $\mathcal{H}_2$ die erste Eigenschaft von Lemma 3.3.13.

Wir müssen also nur noch die zweite Eigenschaft zeigen, d.h., daß das Verhältnis von $|\mathcal{H}_2|$ zu $|\mathcal{H}_1| + |\mathcal{H}_2|$ gleich $\varepsilon = (u_1 - 1)/(u_1 r - 1)$ ist (mit $u_1 = v/r$). Es enthält A_i genau die Elemente $\{(jp+1)p^i \mid 0 \le j < v/(p^{i+1})\}$. Also ist $|A_i| = v/(p^{i+1}) = p^{n-i-1}$, und es folgt

$$\begin{aligned} |A_m| + |A_{2m}| + \ldots + |A_{n-m}| &= p^{n-m-1} + p^{n-2m-1} + \ldots + p^{m-1} \\ &= p^{m-1} \cdot \left(r^0 + r^1 + \ldots + r^{l-2} \right) \\ &= \frac{r}{p} \cdot \frac{r^{l-1} - 1}{r - 1} = \frac{v - r}{pr - p}. \end{aligned}$$

Wir erhalten dann wie folgt das gewünschte Ergebnis:

$$\begin{aligned} \frac{|\mathcal{H}_2|}{|\mathcal{H}_1| + |\mathcal{H}_2|} &= \frac{|A_m| + \ldots + |A_{n-m}|}{|A_0| + |A_m| + \ldots + |A_{n-m}|} = \frac{(v-r)/(pr-p)}{v/p + (v-r)/(pr-p)} \\ &= \frac{v - r}{vr - v + v - r} = \frac{v/r - 1}{v - 1} = \varepsilon. \end{aligned}$$

■

Die optimal universelle Ganzzahlklasse benutzt für die Menge der multiplikativen Faktoren eine Teilmenge von $V = U$. Ihre Funktionen können also genauso effizient ausgewertet werden wie die der 1-universellen Hashklasse. Bislang sind keine anderen optimal universellen Hashklassen bekannt, deren Funktionen mit reiner Ganzzahlarithmetik und einer einzigen Multiplikation ausgewertet werden können.

§ 4. Hashing langer Schlüssel

Die Hashklassen des letzten Paragraphen sind besonders gut geeignet, wenn die Multiplikation direkt vom Prozessor unterstützt wird. Dies ist aber nur dann der Fall, wenn der Ring V in einem Computerwort dargestellt werden kann, d.h. daß die von der Multiplikation unterstützte Bitlänge $\log v$ nicht überschreitet. Üblicherweise ist damit die Bitlänge der Schlüssel auf kleine Werte wie z.B. 64 Bit beschränkt. Sollen längere Schlüssel abgebildet werden, so müssen spezielle Verfahren zur Multiplikation langer Zahlen implementiert werden. Dies erhöht den Programmieraufwand, und verschlechtert die Effizienz der Hashklasse.

Mit Satz 3.2.10 kann die Multiplikation langer Zahlen vermieden werden, indem man aus der abstandsuniversellen Ganzzahlklasse $U \to R$ eine abstandsuniverselle Hashklasse $U^n \to R^m$ ($m \le n$) konstruiert. Wir könnten nun die in Kapitel 3 vorgestellten Methoden direkt anwenden, um streng universelle, 1-universelle und sogar optimal universelle Hashklassen zu erhalten, die hauptsächlich auf ganzzahliger Arithmetik beruhen. Wir werden hier jedoch die Konstruktionsmethode noch etwas genauer untersuchen und so auf die Ganzzahlklassen „zuschneiden“, daß das Ergebnis besonders effizient ist.

Angenommen, man wendet die Technik aus Satz 3.2.10 direkt auf die 1-abstandsuniverselle Ganzzahlklasse $U \to R$ aus Satz 4.2.5 an, um das Universum zu U^n und den Wertebereich zu R^m zu erweitern. Um eine Spalte eines Hashwerts $(y_0,\dots,y_{m-1})$ aus R^m zu berechnen, müssen dann n Hashfunktionen $h^k_{a,b}$ der abstandsuniversellen Ganzzahlklasse ausgewertet und addiert werden. Dies beinhaltet n Multiplikationen und Additionen über dem Ring V, n Divisionen (bzw. bitweise Rechtsverschiebungen), sowie $n-1$ Additionen über R. Wir zeigen nun, daß man mit n Multiplikationen und n Additionen über dem Ring V, sowie einer Division (Rechtverschiebung) auskommt. Außerdem sind die im folgenden präsentierten Hashklassen wesentlich kleiner als die, die aus einer direkten Konstruktion gemäß Satz 3.2.10 resultieren würden.

Es sei wie üblich $U = \{0,\dots,u-1\}$, $R = \{0,\dots,r-1\}$ und V der Ring $\mathbb{Z}_v$ mit $v = kr$ für ein noch zu bestimmendes k. Wir betrachten nun das Universum U^n und den Wertebereich R^m für $m \le n$ (wobei R^m die abelsche Gruppe mit der komponentenweisen Addition modulo r ist). Um den Hashwert eines Schlüssels $\underline{x} = (x_0,\dots,x_{n-1})$ zu berechnen, benutzen wir, ähnlich wie bei den Faltungsklassen, die Konvolution über dem Ring V. Nach dieser wird für jede Spalte eine zufällige Addition und eine Division mit k ausgeführt. Wir definieren dazu zunächst für $\underline{b} = (b_0,\dots,b_{m-1}) \in V^m$ die Funktion

$$\mathbf{g}_{\underline{b}} : V^m \to R^m, \quad (x_0,\dots,x_m) \mapsto (y_0,\dots,y_m) \quad \text{mit} \quad y_i = g_{b_i}(x_i),$$

wobei $g_b(x)$ wie bisher üblich als $(x+b) \operatorname{div} k$ definiert ist.

Die im folgenden Satz betrachteten *langen Ganzzahlklassen* bestehen aus der Faltung von Elementen aus U^n mit zufälligen Vektoren, verknüpft mit der Funktion $\mathbf{g}_{\underline{b}}$. Dabei sei $\mathrm{conv}_{k,l}$ über dem Ring V definiert wie auf Seite 26.

4.4.1 Satz. *Sei $c = 1 + \vartheta\big(k, \Gamma(u,r,k)\big)$, sowie entweder $k \ge u-1$ oder $v = kr$ Potenz der Primzahl p und $k \ge u/p$ (es sei daran erinnert, daß dann c durch $9/8$ beschränkt ist, gegen 1 konvergiert und für den Fall, daß $v = kr$ Potenz der Primzahl p ist, sogar $c = 1$ gilt).*

(a) Die Klasse der Funktionen $f_{\underline{a},\underline{b}} : U^n \to R^m$, $\underline{x} \mapsto \mathbf{g}_{\underline{b}}\big(\mathrm{conv}_{n-1,n+m-2}(\underline{a},\underline{x})\big)$ mit $\underline{a} \in V^{n+m-1}$ und $\underline{b} \in \{0,\dots,k-1\}^m$ ist c^m-abstandsuniversell.

(b) Sei $f_{\underline{a},\underline{b}}$ Definiert wie in (a). Die Klasse der Funktionen $f_{\underline{a},\underline{b}}$ mit $\underline{a} \in V^{n+m-1}$ und $\underline{b} \in V^m$ ist streng c^m-universell.

(c) Die Klasse der Funktionen $h_{\underline{a},\underline{b}} : U^n \to R^m, \underline{x} \mapsto \mathbf{g}_{\underline{b}}\big(\mathrm{conv}_{n-m,n-1}(\underline{a},\underline{x})\big)$ mit allen Werten $\underline{a} = (k, a_1, \ldots, a_{n-1}) \in V^n$ und $\underline{b} \in \{0,\ldots,k-1\}^m$ ist c^m-universell.

(d) Sei $h_{\underline{a},\underline{b}}$ Definiert wie in (c). Ist $v = kr$ Potenz der Primzahl p und n ein Vielfaches von m, dann gibt es eine Teilmenge $\mathcal{A} \subseteq V^n$, so daß die Klasse der Funktionen $h_{\underline{a},\underline{b}}$ mit $\underline{a} \in \mathcal{A}$ und $\underline{b} \in \{0,\ldots,k-1\}^m$ optimal universell ist.

Beweis: *Zu (a).* Wir verbinden die Idee von Satz 3.2.10 mit den Methoden aus § 2. Seien $\underline{x} = (x_0,\ldots,x_{n-1})$ und $\underline{x}' = (x'_0,\ldots,x'_{n-1})$ verschiedene Schlüssel aus U^n sowie $\underline{d} = (d_0,\ldots,d_{m-1})$ der Abstand $f_{\underline{a},\underline{b}}(\underline{x}) - f_{\underline{a},\underline{b}}(\underline{x}')$.

O.B.d.A. seien $a_0,\ldots,a_{n+m-2}$ in dieser Reihenfolge zufällig gewählt. Wir untersuchen zunächst das Ergebnis der Konvolution und definieren daher

$$(y_{n-1},\ldots,y_{n+m-2}) \;=\; \mathrm{conv}_{n-1,n+m-2}(\underline{a},\underline{x})$$

und analog

$$(y'_{n-1},\ldots,y'_{n+m-2}) = \mathrm{conv}_{n-1,n+m-2}(\underline{a},\underline{x}').$$

Wir betrachten $\Delta_i = y'_i - y_i$ für $n-1 \le i \le n+m-2$. Sei $n-1 \le i < n+m-1$ beliebig und t der kleinste Index, in dem sich x_t und x'_t unterscheiden (o.B.d.A. $x_t < x'_t$). Da $a_0,\ldots,a_{i-t-1}$ vor a_{i-t} gewählt wurden, sind bei der zufälligen Wahl von a_{i-t} bereits $\Delta_{n-1},\ldots,\Delta_{i-1}$ fest. Wir erhalten, da für $j > i-t$ die Werte x_{i-j} und x'_{i-j} gleich sind,

$$\Delta_i \;=\; \sum_{j=0}^{i} a_j(x_{i-j} - x'_{i-j}) \;=\; \sum_{j=0}^{i-t} a_j(x_{i-j} - x'_{i-j}) \;=\; C + a_{i-t}(x'_t - x_t)$$

für ein von $a_0,\ldots,a_{i-t-1}$ abhängendes aber bei der Wahl von a_{i-t} festes C. Mit Lemma 4.2.1 ist dann Δ_i gleichverteilt über $\{j\gamma + C \mid 0 \le j < v/\gamma\}$, wobei $\gamma = \mathrm{ggT}(x'_t - x_t, v)$ ist. Damit erfüllt die Wahrscheinlichkeitsverteilung des Paars (y_i, y'_i) die Voraussetzungen von Lemma 4.2.8, und mit $l = i-n+1$ und $d_l = g_{b_l}(y_i) - g_{b_l}(y'_i)$ folgt, daß d_l jeden Wert aus R mit einer Wahrscheinlichkeit von höchstens c/r annimmt. Da die Verteilung von d_l nur von der Verteilung von Δ_i und der von b_l abhängt (also auch unabhängig von C ist), ist d_l unabhängig von $d_0,\ldots,d_{l-1}$. Somit sind alle d_l unabhängig voneinander, und der Abstand $(d_0,\ldots,d_{m-1})$ wird mit einer Wahrscheinlichkeit von höchstens $(c/r)^m$ angenommen.

Zu (b). Dies ist die übliche Konstruktion einer streng c^m-universellen Hashklasse aus der c^m-abstandsuniversellen von Teil (a) gemäß Satz 3.2.5.

Zu (c). Analog zum Beweis von (a) betrachten wir zwei Schlüssel $\underline{x}$ und $\underline{x}'$ sowie den kleinsten Index t, in dem sich x_t und x'_t unterscheiden. Ist $t < n-m$, dann kann die Behauptung analog zum Beweis von (a) geführt werden, und die Hashwerte nehmen sogar jeden Abstand mit einer Wahrscheinlichkeit von höchstens $(c/r)^m$ an. Sei also $t \ge n-m$. Es folgt für y_i und y'_i wie oben:

$$y_t - y'_t \;=\; \sum_{j=0}^{t} a_j(x_{t-j} - x'_{t-j}) \;=\; a_0(x_t - x'_t) \;=\; k(x_t - x'_t).$$

Also ist $g_b(y_t) - g_b(y'_t) = x_t - x'_t \neq 0$ für alle b, und die Schlüssel $\underline{x}$ und $\underline{x}'$ kollidieren nicht.

Zu (d). Der Beweis von (c) hat eigentlich gezeigt, daß die Klasse der Funktionen $h_{\underline{a},\underline{b}}$ $(0|c^m)$-universell ist. Insbesondere ist sie also für eine Primpotenz kr $(0|1)$-universell. Die Behauptung folgt somit völlig analog zum Beweis von Satz 3.3.17, indem wir

$$\mathcal{A} = \mathcal{A}_0 \cup \mathcal{A}_m \cup \ldots \cup \mathcal{A}_{n-m}$$

definieren, wobei $\mathcal{A}_i$ die Menge der Vektoren

$$(\underbrace{0,\ldots,0}_{i\text{-mal}},k,a_{i+1},\ldots,a_{n-1}) \in V^n$$

mit $a_{i+1},\ldots,a_{n-1} \in V$ ist. ∎

Diese langen Ganzzahlklassen sind ganz ähnlich aus der inhomogenen abstandsuniversellen Ganzzahlklasse konstruiert, wie die Faltungsklassen aus der abstandsuniversellen Körperklasse. Tatsächlich sind sie für $u = r = p$ (p prim) und $k = 1$ genau die Faltungsklassen für den Körper $\mathbb{K} = \mathbb{Z}_p$.

Das Ergebnis aus Teil (b) wurde für $m = 1$ bereits von Dietzfelbinger (1996) ohne Beweis notiert (mit einer oberen Schranke von $5/4$, wenn kr keine Primpotenz ist). Ähnliche Konstruktionen für Werte $m > 1$ sind vorher allerdings noch nicht untersucht worden.

Die Konvolutionen $\mathrm{conv}_{n-1,n+m-2}(\underline{a},\underline{x})$ können offensichtlich mit $n \cdot m$ Multiplikationen und $(n-1)m$ Additionen über dem Ring V berechnet werden. Für die Konvolution $\mathrm{conv}_{n-m,n-1}$ genügen sogar

$$(n-m+1)+(n-m+2)+\ldots+(n-m+m) \;=\; m(n-m)+\frac{m(m+1)}{2} \;=\; m\left(n-\frac{m-1}{2}\right)$$

Multiplikationen.

In der Praxis ist dabei für Zweierpotenzen kr nur eine oder sogar keine Modulooperation notwendig. Wir betrachten beispielsweise einen Prozessor mit einer Wortbreite von 64 Bit und eine Anwendung, bei der aus n Halbworten bestehende Schlüssel gegeben sind, und auf ein Halbwort abgebildet werden sollen (bei den meisten Prozessoren lassen sich Halbwörter direkt adressieren). Um eine Hashfunktion $h_{\underline{a},\underline{b}}$ einer langen Ganzzahlklasse auszuwerten, multipliziert man nacheinander jedes der n Halbwörter des Schlüssels mit dem entsprechenden Wort des Vektors $\underline{a}$, summiert diese auf und addiert anschließend das Ergebnis zu dem entsprechenden Halbwort aus $\underline{b}$. Dabei erhält man automatisch Ergebnisse modulo 2^{64}, und es genügt, das signifikante Halbwort des Ergebnisses als Hashwert zu nehmen. Die Zeit für die Auswertung einer solchen Hashfunktion entspricht also der von n 64-Bit Multiplikationen und Additionen.

KAPITEL 5

Schluß

Im Rahmen dieser Diplomarbeit wurden neue Ergebnisse über universelles Hashing vorgestellt. Diese werden abschließend zusammengefaßt und hinsichtlich ihrer praktischen Bedeutung bewertet.

Kombinatorik und Konstruktionsmethoden

In Kapitel 3 wurden zunächst bekannte untere Schranken für die Kardinalität (streng) c-universeller Hashklassen hergeleitet (Satz 3.1.2 und Satz 3.1.5). Dabei konnten zum ersten Mal Hashklassen, die diese Schranken erreichen, mithilfe streng universeller bzw. optimal universeller Hashklassen charakterisiert werden. Anschließend wurden einige wichtige Konstruktionsmethoden vorgestellt, so u.a. eine einfache Methode, mit der man c-universelle Hashklassen aus c-abstandsuniversellen erhält (Satz 3.2.7). Trotz ihrer Einfachheit konnte damit die bisher beste untere Schranke für c-abstandsuniverselle Hashklassen verbessert werden (Korollar 3.2.8).

Eine wichtige Bedeutung messen wir der neuen Konstruktionsmethode aus Satz 3.2.10 zu. Diese erlaubt es, kleine Hashklassen nicht nur für große Universen, sondern auch für große Wertebereiche zu entwerfen. Ihre Güte bezüglich der Kardinalität der daraus resultierenden Hashklassen läßt sich wie folgt beurteilen. Angenommen, man geht von einer 1-abstandsuniversellen Hashklasse $\mathcal{H} : W \to W$ mit $N = |W|$ Funktionen aus (also einer minimalen abstandsuniversellen Hashklasse), dann kann man daraus gemäß besagtem Satz eine abstandsuniverselle Hashklasse $\mathcal{H}' : W^n \to W^m$ mit N^{n+m-1} Funktionen konstruieren ($n \geq m$). Jede andere Konstruktionsmethode muß aber aufgrund der unteren Schranke für abstandsuniverselle Hashklassen zumindest zu einer Kardinalität von N^n führen. Das bedeutet, daß selbst für große Wertebereiche, also z.B. $m = n$, die Hashfunktionen aus $\mathcal{H}'$ mit höchstens doppelt so vielen Bits beschrieben werden können, wie eine Hashklasse, deren Kardinalität der unteren Schranke entspricht.

Weiterhin haben wir eine Technik entworfen, mit deren Hilfe man optimal universelle Hashklassen konstruieren kann (Satz 3.3.15). Dabei handelt es sich um die erste allgemeine Methode, die auf Hashing und nicht auf kombinatorischen Designs basiert (es gibt zahlreiche algebraische Methoden zur Konstruktion von RBIBDs, die aber keinerlei Aufschluß über die Praktikablität der resultierenden Hashklassen geben). Dazu war es notwendig, die sogenannten $(0|c)$-universellen Hashklassen neu zu definieren, die man auf natürlichem Wege aus c-abstandsuniversellen Hashklassen erhält (Bemerkung 3.3.2). Sie sind neben ihrer Bedeutung für Konstruktionsmethoden auch als kombinatorische Objekte interessant. So konnten wir eine Äquivalenz zwischen exakt $(0|c)$-universellen Hashklassen mit konstanter Korbgröße und RGDDs aufzeigen (Satz 3.3.7) und begründen, daß dies eigentlich eine Verallgemeinerung der bekannten Äquivalenz von optimal universellen Hashklassen und RBIBDs

darstellt (Korollar 3.3.12). Schließlich konnte gezeigt werden, daß jede Stinson-minimale 1-universelle Hashklasse tatsächlich exakt $(0|1)$-universell ist (Satz 3.3.4), also einem RTD entspricht (Korollar 3.3.11).

Zusammen mit den schon bekannten Zusammenhängen zwischen kombinatorischen Designs und universellem Hashing belegen diese neuen Ergebnisse, daß kleine universelle Hashklassen eine feste Struktur aufweisen. Sie erlauben uns auf die gut entwickelten Methoden der Mathematik zurückzugreifen, um Hashklassen zu analysieren und zu konstruieren.

Die in Kapitel 3 vorgestellten Techniken haben bereits dort eine wichtige Anwendung gefunden. So haben wir neue abstandsuniverselle, universelle und optimal universelle Hashklassen vorgestellt, die auf der Faltung von Vektoren beruhen (Satz 3.2.11). Auch die schon vorher bekannte strenge Universalität einer der Faltungsklassen geht aus unserer Konstruktionsmethode hervor.

Die Faltung über Bit-Vektoren läßt sich effizient auf Hardwarebasis (z.B. mithilfe einfacher Schieberegister) realisieren. Weiterhin gibt es Schaltkreise der Tiefe $O(\log n)$ und der Größe $O(n \log^2 n \log\log n)$, die die Faltung zweier n-Bit Vektoren berechnen (vgl. Wegener, 1996, S. 110). Damit können für $u = 2^n$ die Funktionen der Faltungsklassen in Zeit $O(n \log^2 n \log\log n)$ ausgewertet werden. Da dieses asymptotische Ergebnis aber unpraktikabel hohe Konstanten aufweist, und die Faltung von Prozessoren nicht direkt unterstützt wird, sind solche Funktionen für eine Software-Implementierung im allgemeinen nicht geeignet.

Das Entscheidende an den Faltungsklassen ist jedoch, daß sie das Zusammenspiel der einzelnen Methoden verdeutlichen: Man geht von einer abstandsuniversellen Klasse $R \to R$ aus, erweitert Universum und Wertebereich, und konstruiert schließlich 1-universelle oder streng universelle Hashklassen für beliebige Universen R^n und Wertebereiche R^m. Die 1-universellen Hashklassen sind dann sogar $(0|1)$-universell, was mit der trivialen Existenz optimal universeller Hashklassen $R \to R$ die rekursive Konstruktion weiterer optimal universeller Hashklassen $R^n \to R^m$ (für Vielfache n von m) ermöglicht. Während fast alle bisherigen Arbeiten die Universalität oder strenge Universalität von Hashklassen immer direkt bewiesen haben, hat man jetzt ein umfassendes Konstruktionsschema zur Hand.

Bedeutung der Ganzzahlklassen

Auch die „multiplikative“ Hashklasse von Dietzfelbinger et al. (1997) und die „lineare“ Hashklasse von Dietzfelbinger (1996) wurden von den Autoren direkt und unabhängig voneinander analysiert. Wir haben beide in das allgemeine Schema der Ganzzahlklassen $\mathcal{H}^k_{A,B}$ eingebunden, und diese wiederum mit den Methoden aus Kapitel 3 untersucht. Ausgehend von der Erfahrung, daß abstandsuniverselles Hashing die Grundlage für streng universelles, universelles und optimal universelles Hashing ist, haben wir zunächst abstandsuniverselle Ganzzahlklassen konstruiert. Aus diesen sind dann die bereits bekannten, sowie zahlreiche neue Hashklassen hervorgegangen. Eine Übersicht über alle Ergebnisse liefert Tabelle 1. Dabei werden die Fälle berücksichtigt bei denen k minimal für das vorgegebene Universum u und für den vorgegebenen Wertebereich r ist.

Für die bekannten streng universellen Ganzzahlklassen haben sich einige Verbesserungen ergeben. So konnte die obere Schranke des Universalitätsparameters der „linearen Hashklasse“ im allgemeinen Fall von 5/4 auf 9/8 verbessert werden (Korollar 4.2.6). Außerdem konnten wir ihre Kardinalität erheblich reduzieren, wenn k und r Potenzen der gleichen Primzahl sind. So zeigt Korollar 4.2.10, daß man in diesem Fall mit $3/2\log(u/p) + 2\log r$ statt $2\log(u/p) + 2\log r$ Zufallsbits auskommt. Bei unserer Konstruktion wird eine Teilmenge

Universalität	c	u, r [†]	k	A [‡]	B [§]	Kardinalität	Quelle	Erste Referenz[¶]
c-univ.	2	Potenzen von 2	u/r	V^*	$\{0\}$	$u/2$	4.3.3 (a)	DHKP
	2	Potenzen von $p \neq 2$	u/r	V^*	$\{0\}$	u/p	4.3.3 (a)	W
	1	Potenzen von p	u/r	V^*	$\{i\zeta \mid 0 \leq i < k/\zeta\}$	$\left(u\sqrt{u/r}\right)/p$	4.3.3 (b)	W
optimal univ.	$(u-r)/(u-1)$	$r = p^m$, $u = p^{lm}$	u/r	$\subseteq V$	$\{i\zeta \mid 0 \leq i < k/\zeta\}$	$\leq u\sqrt{u/r}$	4.3.5	W
abstandsuniv.	3	beliebig	$u-1$	V	$\{0\}$	$(u-1)r$	4.2.2 (a)	W
	2	Potenzen von p	u/p	V	$\{0\}$	ur/p	4.2.2 (b)	W
	9/8	beliebig	$u-1$	V	$\{0,\ldots,k-1\}$	$(u-1)^2 r$	4.2.5	W
	1	Potenzen von p	u/p	V	$\{i\zeta \mid 0 \leq i < k/\zeta\}$	$(u/p)^{3/2} r$	4.2.9	W
streng c-univ.	3	beliebig	$u-1$	V	$\{ik \mid 0 \leq i < r\}$	$(u-1)r^2$	4.2.3 (a)	W
	2	Potenzen von p	u/p	V	$\{ik \mid 0 \leq i < r\}$	ur^2/p	4.2.3 (b)	W
	5/4 (9/8)	beliebig	$u-1$	V	V	$\big((u-1)r\big)^2$	4.2.6	D (W)
	1	Potenzen von p	u/p	V	V	$(ur/p)^2$	4.2.10	D
	1	Potenzen von p	u/p	V	$\{i\zeta \mid 0 \leq i < v/\zeta\}$	$(u/p)^{3/2} r^2$	4.2.10	W

Tabelle 1: Alle bekannten Ergebnisse für die Ganzzahlklassen $\mathcal{H}^k_{A,B}$.

[†] Es ist p eine Primzahl.
[‡] Es ist $v = kr$, $V = \{0,\ldots,v-1\}$ und $V^* = \{ip+1 \mid 0 \leq i < v/p\}$.
[§] Für $v = p^n$ und $r = p^m$ ist $\zeta = p^\tau$ mit $0 \leq \tau \leq \lceil (n-m)/2 \rceil$.
[¶] D=Dietzfelbinger (1996), DHKP=Dietzfelbinger, Hagerup, Katajainen und Penttonen (1997), W=Woelfel (1999).

der linearen Hashklasse benutzt, so daß sich für die Effizienz der Hashfunktionen kein Unterschied ergibt.

Weiterhin haben wir gezeigt, daß man Zufallsbits einsparen kann, wenn man auf die zufällige Addition ganz verzichtet. So kommt man gemäß Korollar 4.2.3 mit $\log u + 2\log r$ Bits aus, wenn man einen Universalitätsparameter von 3 im allgemeinen Fall und von 2 für Primpotenzen k und r in Kauf nimmt.

Für die Multiplikation bei den streng c-universellen Hashklassen wird eine Wortbreite von mindestens $\log u + \log r$ Bit benötigt. Wenn die Verteilung der Schlüsselpaare nicht interessiert, sondern nur die Kollisionswahrscheinlichkeit, sollte man daher die c-universellen Hashklassen benutzen, die mit einer Wortbreite von $\log u$ Bit auskommen. Die einzige bekannte Hashklasse, die diese Eigenschaft hat und mit ganzzahliger Arithmetik ohne Primzahlen auskommt, war bisher die 2-universelle Ganzzahlklasse von Dietzfelbinger et al. Wir haben die ursprüngliche Aussage, die nur für Zweierpotenzen k und r galt, für beliebige Potenzen von p verallgemeinert (4.3.3 (a)). Diese Verallgemeinerung ist jedoch nur von geringer praktischer Relevanz.

Zu den wichtigsten Ergebnissen dieser Arbeit zählen hingegen die 1-universelle und die optimal universelle Hashklasse (Satz 4.3.3 (b) und Satz 4.3.5). Beide kommen für $u \leq 2^n$ mit einer Multiplikation, einer Addition und zwei bitweisen Operationen über n Bits aus. Es ist keine andere 1-universelle Hashklasse bekannt, deren Funktionen so effizient ausgewertet werden können, wenn die Schlüssel in ein Computerwort passen, d.h. direkt vom Prozessor verarbeitet werden können. Die lineare Primzahlklasse beispielsweise benötigt neben der Multiplikation und der Addition zusätzlich mindestens eine Division (die Operation modulo p), die nicht durch eine bitweise Operation ersetzt werden kann.

Alle Ganzzahlklassen können für $r = 2^m$ und $v = 2^n$ mit einer Multiplikation, einer Addition und zwei bitweisen Operationen ausgewertet werden. Mithilfe der Schönhage-Strassen Multiplikation (Schönhage und Strassen, 1971) können diese Hashklassen somit auf Schaltkreisen der Tiefe $O(\log n)$ und der Größe $O(n \log n \log\log n)$ realisiert werden (vgl. auch Dietzfelbinger, 1996). Bekannte Schaltkreise für Hashklassen, die eine „echte“ Division benötigen (wie z.B. die Primzahlklassen), sind größer oder zumindest tiefer (s. Wegener, 1996).

Die langen Ganzzahlklassen (Satz 4.4.1) sind gut geeignet, lange Schlüssel mit ganzzahliger Arithmetik abzubilden, ohne auf die Multiplikation langer Zahlen angewiesen zu sein. Üblicherweise wird man bei Prozessoren, die Wörter mit w Bits verarbeiten können, Universen W^n und Wertebereiche W^m mit $W = \{0, \ldots, 2^w - 1\}$ benutzen. Für $m = 1$ wurde die streng universelle lange Ganzzahlklasse bereits von Dietzfelbinger (1996) vorgestellt. Die Ergebnisse für $m > 1$ erlauben aber jetzt auch eine effiziente Auswertung der Hashfunktionen, wenn die Bitbreite der Hashwerte länger als w ist. In den meisten Fällen ist m konstant, und die Funktionen können in Zeit $O(n)$ berechnet werden. Ist hingegen m sehr groß (z.B. $m \approx n$), dann ist eine direkte Implementation der Faltung nicht effizient, so daß man möglicherweise doch die „kurzen“ Ganzzahklassen mit einem effizienten Algorithmus zur Multiplikation langer Zahlen benutzen sollte.

Optimal universelles oder 1-universelles Hashing?

Für alle drei Typen von Hashklassen (Faltungsklassen, Ganzzahlklassen und lange Ganzzahlklassen) haben wir sowohl 1-universelle als auch optimal universelle Familien vorgestellt. Gerade die effizienten optimal universellen Hashklassen geben aufgrund ihrer Äquivalenz

zu RBIBDs eine Antwort auf eine wichtige Fragestellung. So gibt es zwar zahlreiche Anwendungen für RBIBDs und andere kombinatorische Designs (vgl. Colbourn und Van Oorschot, 1989), aber kaum Hinweise auf deren praktische Implementierbarkeit. Hofmeister und Lefmann (1996) benutzen beispielsweise RBIBDs für Algorithmen zum Auffinden großer Schnitte in Graphen. Sie bemerken allerdings:

> *Although various algebraic construction methods for BIBDs are known in the literature [...], not a lot of attention has been paid to their exact running times. It might be interesting to investigate this more closely.*

Und später schreiben sie über die aufgrund asymptotischer Existenzbeweise existierenden RBIBDs:

> *The question remains how we can compute such a resolvable BIBD by an efficient algorithm.*

Diese Frage ist nun mit den effizienten optimal universellen Hashklassen, die wir in dieser Arbeit vorgestellt haben, beantwortet.

Es muß aber noch diskutiert werden, ob in Anwendungen, bei denen es hauptsächlich um die Kollisionswahrscheinlichkeit von Schlüsseln geht, die optimal universellen oder die 1-universellen Hashklassen zu bevorzugen sind. Anhand der Ganzzahlklasse aus Satz 4.3.3 wollen wir abschließend auf Vor- und Nachteile beider eingehen. Sei $u = 2^n$ und $r = 2^m$, wobei m ein Teiler von n ist. Die Funktionen beider Klassen lassen sich gleich schnell berechnen, da sie jeweils sowohl eine Multiplikation als auch eine Addition über n Bits benötigen. Auch in der Kardinalität unterscheiden sich die Hashklassen kaum: Die 1-universelle Ganzzahlklasse, wir nennen sie hier $\mathcal{H}_U$, hat eine Kardinalität von 2^{n-1}, und die der optimal universellen ($\mathcal{H}_{OU}$) ist durch 2^n beschränkt. Aber auch der Universalitätsparameter c_{opt} von $\mathcal{H}_{OU}$ ist nicht erheblich kleiner als bei $\mathcal{H}_U$, denn es gilt

$$c_{opt} \;=\; \frac{2^n - 2^m}{2^n - 1} \;=\; 1 - \frac{2^m - 1}{2^n - 1} \;\approx\; 1 - 2^{m-n}.$$

Da üblicherweise m erheblich kleiner als n ist, liegt dieser Wert sehr nahe bei 1.

Ein Punkt spricht aber für die Verwendung von $\mathcal{H}_U$. Denn bei dieser kann a, der zufällige Wert für die Multiplikation, sehr einfach ausgewählt werden. Dazu muß nur eine ungerade Zahl aus $\{0,\ldots,2^n-1\}$ gewählt werden, was einer zufälligen Belegung der vorderen $n-1$ Bits und einer festen Belegung des letzten Bits mit 1 entspricht. Bei $\mathcal{H}_{OU}$ ist dies schwieriger, da die Menge A, aus der a zufällig gewählt werden muß, aus den Zahlen $i2^{jm+1}+1$ mit $0 \le j < n/m$ und $0 \le i < 2^{n-jm}$ besteht. Dies sind alle Zahlen, deren Bits sich als

$$\big\langle a_{n-1}\ldots a_{jm+1}\,1\,\underbrace{0\ldots0}_{jm\text{-mal}}\big\rangle \tag{5.1}$$

darstellen lassen. Ist j gewählt, so lassen sich zwar die vorderen Bits $a_{n-1}\ldots a_{jm+1}$ einfach zufällig bestimmen, aber j selbst darf nicht gleichverteilt aus $\{0,\ldots,n/m-1\}$ gewählt werden. Statt dessen kommt bei gleichverteilter Wahl von $a = i2^{jm+1}+1$ aus A jedes j mit einer Wahrscheinlichkeit von genau

$$\frac{2^{n-jm-1}}{\sum_{k=0}^{n/m-1} 2^{n-km-1}} \;=\; \frac{2^{n-jm}}{\sum_{k=1}^{n/m} 2^{km}} \;=\; \frac{2^{n-jm}}{(2^{n+m}-1)/(2^m-1)-1} \;=\; \frac{2^{n-jm+m}-2^{n-jm}}{2^{n+m}-2^m} \tag{5.2}$$

vor.

Eine einfache Lösung wäre, a solange hintereinander zufällig gleichverteilt aus $V = \{0,\ldots,2^{n-1}-1\}$ zu wählen, bis $a \in A$ ist. Da $|A| > |V|/2$ gilt, benötigt man im Mittel höchstens 2 Versuche, bis man ein passendes a gefunden hat. Andererseits ist dann aber weder die Zeit, bis die Hashfunktion gefunden wird, noch die Anzahl der benötigten Zufallsbits deterministisch bestimmt.

Eine bessere Möglichkeit besteht darin, daß man $a = \langle a_{n-1}\ldots a_0\rangle$ zufällig aus $V\backslash\{0\}$ wählt, und das kleinste j ermittelt, für das a ein Vielfaches von 2^{jm} ist. Anschließend setzt man $a_{jm} = 1$. Offensichtlich hat dann a eine Darstellung wie in (5.1) und ist somit ein Element aus A. Außerdem ist die Anzahl der Elemente aus $V\backslash\{0\}$, die zu einem bestimmten Wert j führen, durch die Anzahl der a bestimmt, bei denen mindestens die letzten jm und höchstens die letzten $(j+1)m-1$ Bits den Wert 0 haben. Also durch alle Zahlen

$$\langle \underbrace{a_{n-1}\ldots a_{(j+1)m}}_{\text{beliebig}} \underbrace{a_{(j+1)m-1}\ldots a_{jm}}_{\neq 0} \underbrace{a_{jm-1}\ldots a_0}_{=0}\rangle.$$

Es gibt in $V\backslash\{0\}$ offensichtlich $2^{n-(j+1)m}(2^m-1)$ solche Werte, und a hat dann eine Darstellung $i2^{jm}+1$ für ein festes j mit einer Wahrscheinlichkeit von genau

$$\frac{2^{n-(j+1)m}(2^m-1)}{2^n-1} = \frac{2^{n-jm+m}-2^{n-jm}}{2^{n+m}-2^m}.$$

Dies entspricht genau dem Wert aus (5.2). Unter der Bedingung, daß j einen festen Wert hat, ist der Wert von $\left\langle a_{n-1}\ldots a_{n-jm}\right\rangle$ offensichtlich gleichverteilt. Wenn man dann a_{n-jm} auf 1 setzt, nimmt a jeden Wert $i2^{jm}+1$ mit gleicher Wahrscheinlichkeit an und ist somit gleichverteilt über A.

Das Verfahren hat aber den Nachteil, daß die Wahl einer Zahl aus $V\backslash\{0\}$ unter Umständen nichtdeterministisch erfolgen muß, wenn die Zufallszahlen aus Zufallsbits zusammengesetzt sind. Die Wahrscheinlichkeit, daß ein zufälliges $a \in \{0,\ldots,2^n-1\}$ den Wert 0 hat, ist aber für große n vernachlässigbar klein. Ein weiterer Nachteil des Verfahrens besteht darin, daß das kleinste j, für das a ein Vielfache von 2^{jm} ist, bestimmt werden muß. Dies erfordert zusätzlichen Rechenaufwand.

Bei Anwendungen, bei denen nur sehr selten eine zufällige Hashfunktion gewählt werden muß (wie z.B. bei einem Wörterbuch mit verketteten Listen), kann der zusätzliche Aufwand für die Wahl einer Hashfunktion aus $\mathcal{H}_{OU}$ vernachlässigt werden. $\mathcal{H}_{OU}$ ist dann aufgrund der niedrigeren Kollisionswahrscheinlichkeit die bessere Wahl. Bei anderen Wörterbüchern, wie z.B. dem statischen von Fredman et al. (1984), müssen ungefähr so viele Hashfunktionen zufällig gewählt werden, wie Schlüssel im Wörterbuch vorhanden sind. Dann ist es möglicherweise besser, die 1-universelle Hashklasse $\mathcal{H}_U$ zu verwenden.

In jedem Fall sind aber die Ganzzahlklassen für viele Anwendungen die praktikabelsten universellen Hashklassen, die wir kennen. Ihre Hashfunktionen lassen sich genauso effizient (oder sogar effizienter) auswerten, wie solche, die beim deterministischen Hashing eingesetzt werden. Damit ist die Verwendung von universellem Hashing in Wörterbuchimplementationen und anderen Algorithmen praktisch und sinnvoll.

Literaturverzeichnis

A. V. Aho, J. E. Hopcroft und J. D. Ullman (1987). *Data Structures and Algorithms*. Addison Wesley, erste Auflage.

N. Alon, L. Babai und A. Itai (1986). A fast and simple randomized parallel algorithm for the maximal independent set problem. *Journal of Algorithms*, Band 7, S. 567–583.

N. Alon, M. Dietzfelbinger, P. B. Miltersen, E. Petrank und G. Tardos (1997). Is linear hashing good? In *Proceedings of the 29th Annual ACM Symposium on Theory of Computing*, S. 465–474.

N. Alon, O. Goldreich, J. Håstad und R. Peralta (1992). Simple constructions of almost *k*-wise independent random variables. *Random Structures and Algorithms*, Band 3, S. 289–304.

N. Alon, O. Goldreich, J. Håstad und R. Peralta (1993). Addendum to 'Simple constructions of almost *k*-wise independent random variables'. *Random Structures and Algorithms*, Band 4, S. 119–120.

A. Andersson, T. Hagerup, S. Nilsson und R. Raman (1995). Sorting in linear time? In *Proceedings of the 25th Annual ACM Symposium on Theory of Computing*, S. 427–436.

M. Atici und D. R. Stinson (1996). Universal hashing and multiple authentication. In *Advances in Cryptology – CRYPTO '96*, S. 16–30.

T. Beth, D. Jungnickel und H. Lenz (1999). *Design Theory*, Band 1. Cambridge University Press, zweite Auflage.

J. Bierbrauer (1997). Universal hashing and geometric codes. *Designs, Codes and Cryptographie*, Band 11, S. 207–221.

J. Bierbrauer, T. Johansson, G. Katatianskii und B. Smeets (1994). On families of hash functions via geometric codes and concatenation. In *Advances in Cryptology – CRYPTO '93*, S. 331–342.

R. C. Bose (1942). A note on the resolvability of balanced incomplete block designs. *Sankhyā*, Band 6, S. 105–110.

R. C. Bose und W. S. Connor (1952). Combinatorial properties of group divisible incomplete block designs *Ann. Math. Statist*, Band 23, S. 367–383.

G. Brassard und S. Kannan (1988). The generation of random permuations on the fly. *Information Processing Letters*, Band 28, S. 207–212.

J. L. Carter und M. N. Wegman (1979). Universal classes of hash functions. *Journal of Computer and System Sciences*, Band 18, S. 143–154.

B. Chor und O. Goldreich (1989). On the power of two-point based sampling. *Journal of Complexity*, Band 5.

C. J. Colbourn und J. H. Dinitz (Hg.) (1996). *The CRC Handbook of Combinatorial Designs*. CRC Press, erste Auflage.

C. J. Colbourn und P. C. Van Oorschot (1989). Applications of combinatorial designs in computer science. *ACM Computing Surveys*, Band 21, S. 223–250.

T. H. Cormen, C. E. Leiserson und R. L. Rivest (1990). *Introduction to Algorithms*. MIT Press, erste Auflage.

M. Dietzfelbinger (1996). Universal hashing and k-wise independent random variables via integer arithmetic without primes. In *Proceedings of the 13th Annual Symposium on Theoretical Aspects of Computer Science*, S. 569–580.

M. Dietzfelbinger, T. Hagerup, J. Katajainen und M. Penttonen (1997). A reliable randomized algorithm for the closest-pair problem. *Journal of Algorithms*, Band 25, S. 19–51.

M. Dietzfelbinger und F. Meyer auf der Heide (1992). Dynamic hashing in real time. In J. Buchmann, H. Ganziger und W. J. Paul (Hg.), *Informatik-Festschrift zum 60. Geburtstag von Günter Hotz*, S. 95–119. Teubner.

M. Dietzfelbinger und M. Hühne (1996). A dictionary implementation based on dynamic perfect hashing. Manuskript.

M. Dietzfelbinger, A. Karlin, K. Mehlhorn, F. Meyer auf der Heide, H. Rohnert und R. E. Tarjan (1994). Dynamic perfect hashing: Upper and lower bounds. *SIAM Journal on Computing*, Band 23, S. 738–761.

O. Forster (1983). *Analysis 1*. Vieweg, vierte Auflage.

M. L. Fredman, J. Komlós und E. Szemerédi (1984). Storing a sparse table with $O(1)$ worst case access time. *Journal of the Association for Computing Machinery*, Band 31, S. 538–544.

J. Gill (1977). Computational complexity of probabilistic turing machines. *SIAM Journal on Computing*, Band 6, S. 675–695.

Goldreich und Wigderson (1997). Tiny families of functions with random properties: A quality-size trade-off for hashing. *Random Structures and Algorithms*, Band 11.

T. Hofmeister und H. Lefmann (1996). A combinatorial design approach to MAXCUT. *Random Structures and Algorithms*, Band 9, S. 163–173.

J. Illingworth und J. Kittler (1988). A survey of the hough transform. *Computer Vision Graphics Imae Processing*, Band 44, S. 87–116.

R. Impagliazzo und D. Zuckerman (1989). How to recycle random bits. In *Proceedings of the 30th Annual IEEE Symposium on Fountations of Computer Science*, S. 248–253.

T. Johansson (1997). Bucket hashing with a small key size. In *Advances in Cryptology – EUROCRYPT '97*, S. 149–162.

D. E. Knuth (1998). *The Art of Computer Programming*, Band 3, Sorting and Searching. Addison-Wesley, zweite Auflage.

H. Krawczyk (1994). LFSR-based hashing and authentication. In *Advances in Cryptology – CRYPTO '94*, S. 129–139.

H. Krawczyk (1995). New hash functions for message authentication. In *Advances in Cryptology – EUROCRYPT '95*, S. 301–310.

Y. Lamdan und H. J. Wolfson (1988). Geometric hashing: A general and efficient model-based recognition scheme. In *Proceedings of the 2nd International Conference on Computer Vision*, S. 238–249.

M. Luby (1986). A simple parallel algorithm for the maximal independent set. *SIAM Journal on Computing*, Band 15, S. 1036–1053.

Y. Mansour, N. Nisan und P. Tiwari (1993). The computational complexity of universal hashing. *Theoretical Computer Science*, Band 107, S. 121–133.

Y. Matias und U. Vishkin (1991). On parallel hashing and integer sorting. *Journal of Algorithms*, Band 12, S. 573–606.

K. Mehlhorn (1982). On the program size of perfect and universal hash functions. In *Proceedings of the 23rd Annual IEEE Symposium on Fountations of Computer Science*, S. 170–175.

K. Mehlhorn (1988). *Datenstrukturen und effiziente Algorithmen*, Band 1, Sortieren und Suchen. Springer, zweite Auflage.

K. Mehlhorn und U. Vishkin (1984). Randomized and deterministic simulations of PRAMs by parallel machines with restricted granularity of parallel memories. *Acta Informatica*, Band 21, S. 339–374.

R. Motwani und P. Raghavan (1995). *Randomized Algorithms*. Cambridge University Press, erste Auflage.

N. Nisan (1992). Pseudorandom generators for space-bounded computation. *Combinatorica*, Band 12, S. 449–461.

R. L. Plackett und J. P. Burman (1945). The design of optimum multi-factorial experiments. *Biometrika*, Band 33, S. 305–325.

M. O. Rabin (1963). Probabilistic automata. *Information and Control*, Band 6, S. 230–245.

P. Rogaway (1995). Bucket hashing and its application to fast message authentication. In *Advances in Cryptology – CRYPTO '95*, S. 29–42.

D. V. Sarwate (1980). A note on universal classes of hash functions. *Information Processing Letters*, Band 10, S. 41–45.

G. Scheja und U. Storch (1994). *Lehrbuch der Algebra: unter Einschluß der linearen Algebra*, Band 1. Teubner, zweite Auflage.

A. Schönhage und V. Strassen (1971). Schnelle Multiplikation großer Zahlen. *Computing*, Band 7, S. 281–292.

V. Shoup (1996). On fast and provably secure message authentication based on universal hashing. In *Advances in Cryptology – CRYPTO '96*, S. 313–328.

S. S. Shrikhande (1976). Affine resolvable balanced incomplete block designs: A survey. *Aequationes Mathematicae*, Band 14, S. 251–269.

A. Siegel (1989). On universal classes of fast high performance hash functions, their time-space tradeoff, and their applications. In *Proceedings of the 30th Annual IEEE Symposium on Fountations of Computer Science*, S. 20–25.

M. Sipser (1983). A complexity theoretic approach to randomness. In *Proceedings of the 15th Annual ACM Symposium on Theory of Computing*, S. 330–335.

R. Solovay und V. Strassen (1977). A fast monte-carlo test for primality. *SIAM Journal on Computing*, Band 6, S. 84–85.

D. R. Stinson (1994a). Combinatorial techniques for universal hashing. *Journal of Computer and System Sciences*, Band 48, S. 337–346.

D. R. Stinson (1994b). Universal hashing and authentication codes. *Designs, Codes and Cryptography*, Band 4, S. 369–380.

D. R. Stinson (1996). On the connections between universal hashing, combinatorial designs and error-correcting codes. *Congressus Numerantium*, Band 114, S. 7–27.

T. van Trung (1993). Universal hashing and unconditional authentication codes. In *IEEE International Symposium on Information Theory*, S. 228.

T. van Trung (1994). A combinatorial characterization of certain universal classes of hash functions. *Journal of Combinatorial Designs*, Band 2, S. 161–166.

I. Wegener (1996). *Effiziente Algorithmen für grundlegende Funktionen*. Teubner, zweite Auflage.

M. N. Wegman und J. L. Carter (1979). New classes and applications of hash functions. In *Proceedings of the 20th Annual IEEE Symposium on Fountations of Computer Science*, S. 175–182.

A. Wigderson (1994). The amazing power of pairwise independence. In *Proceedings of the 26th Annual ACM Symposium on Theory of Computing*, S. 574–583.

R. M. Wilson (1972). An existence theory for pairwise balanced designs. I. Composition theorems and morphisms. *Journal of Combinatorial Theory (A)*, Band 13, S. 220–245.

P. Woelfel (1999). Efficient strongly universal and optimally universal hashing. In *Mathematical Foundations of Computer Science: 24th International Symposium*, S. 262–272.

Stichwortverzeichnis