Fachbereich Medieninformatik

Hochschule Harz

Huffman-Kodierung

Referat

Henner Wöhler

11459

Abgabe: 15.01.2007

Inhaltsverzeichnis

Einleitung

Im vorliegenden Text wird das Thema Huffman-Kodierung behandelt. Es handelt sich dabei um ein häufig eingesetztes Verfahren zur verlustfreien Kompression von Daten.
Um in das Thema einzuführen, ist es nötig zunächst die Entropiekodierung im Allgemeinen anzusprechen. Außerdem wird kurz die Shannon-Fano-Kodierung erklärt, da die Huffman-Kodierung auf deren Prinzip aufbaut. Anschließend wird die Huffman-Kodierung mit ihren unterschiedlichen Anwendungsverfahren erklärt. Schließlich wird noch kurz ein Einblick in den aktuellen Stellenwert der Huffman-Kodierung geboten.
Auf andere Kodierungsverfahren, zum Beispiel LZW[1] und arithmetische Kodierung, wird nicht eingegangen. Die Literatur in den Quellenangaben liefert zu diesen Themen genügend Informationen.

Da die verschiedenen Prinzipien bildlich am besten verständlich werden, wurden einige Grafiken angefertigt, die die Beispiele verdeutlichen sollen.

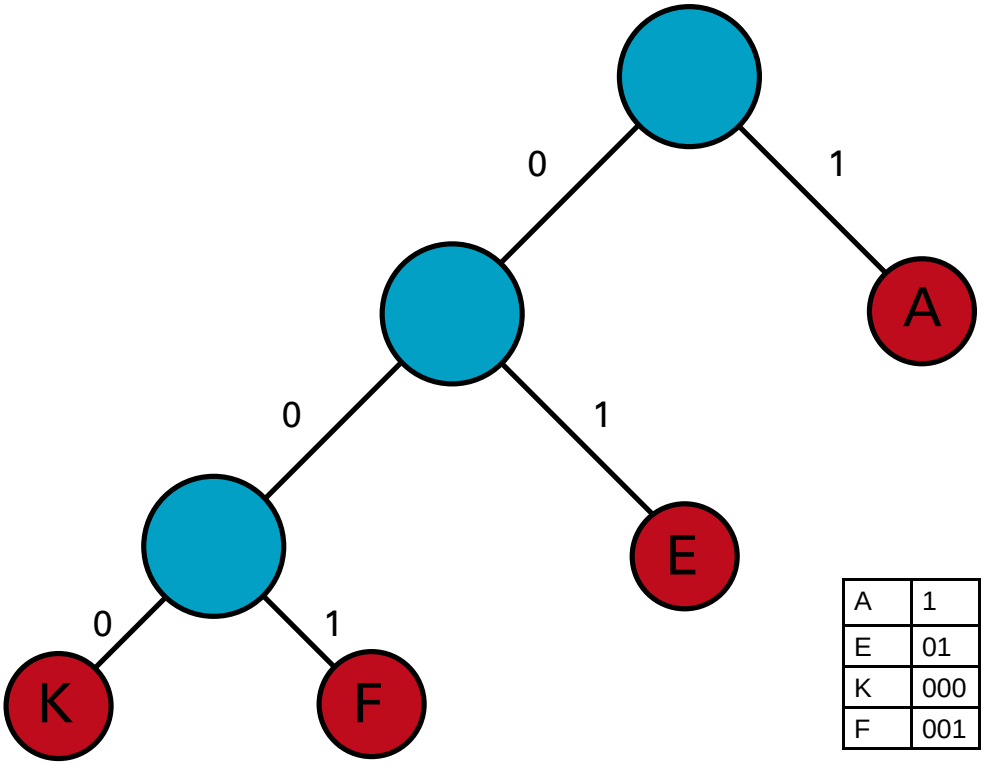
1 Lempel-Ziv-Welch-Kodierung:
Kodierungsverfahren, bei dem ein Wörterbuch mit den häufigsten Zeichenketten erstellt wird, so dass diese nur noch über die Indizes angesprochen werden brauchen.

1. Entropiekodierung

Unter Entropiekodierung versteht man die verlustfreie Kompression von Daten durch Ausnutzen der Auftretenswahrscheinlichkeit verschiedener Symbole.

Die maximale Kompressionsrate entspricht dem mittleren Informationsgehalt des zu kodierenden Signals: der sogenannten Entropie. Ziel einer jeden Kodierung ist es, die mittlere Bitrate an die Entropie anzunähern. Dies ist mit einem Code variabler Länge (variable length code, VLC) möglich. Durch die variable Länge kann man den häufiger auftretenden Symbolen kürzere Codewörter und den seltener auftretenden Symbolen längere Codewörter zuordnen.

Mit Hilfe sogenannter Code-Bäume lassen sich die Kodierungen besonders einfach verdeutlichen. Bei der binären Kodierung ist dies ein binärer Baum, also ein Baum, bei dem nur zweifache Verzweigungen, für Null und Eins, vorkommen (siehe Abb. 1).



A	1
E	01
K	000
F	001

Abb. 1: Beispiel für einen binären Codebaum.

1.1 Morse Code

Einer der ältesten Codes, der mit variabler Länge arbeitet, ist der Morse Code (Samuel Morse, 1791-1872). Er besteht aus kurzen und langen Signalzeichen, die als Punkte und Striche notiert werden. Häufig auftretende Buchstaben, wie zum Beispiel das E, sind dabei kurz gehalten (.), während seltenere, wie das Q, länger sind (- - . -). Problematisch ist, dass mehrere Zeichen hintereinander nicht immer eindeutig zugewiesen werden können. So lässt sich die Zeile „. . _ . . .“ beispielsweise als „fee“, „eeb“ oder „eede“ interpretieren (Beispiel nach [OHM95], Abb. 2). Beim Morse-Code werden deshalb Pausen zwischen den Zeichen gesetzt. Dadurch ist der Code allerdings nicht mehr binär. Diese Problematik lässt sich umgehen, indem man die Codewörter so wählt, dass kein Codewort die Anfangszeichen eines anderen Codewortes enthält. Codes dieser Art werden als präfixfreie Codes bezeichnet.

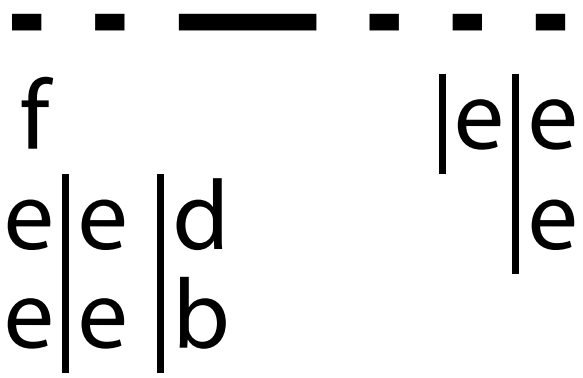


Abb. 2: Reiht man Morsezeichen ohne Pause aneinander lassen sie sich nicht eindeutig entschlüsseln.

1.2 Shannon-Fano-Kodierung

Die präfixfreie Kodierung nach Claude Shannon und Robert Fano (1948) gilt als Vorläufer der Huffman-Kodierung. Sie läuft nach dem, in der Informatik häufig verwendeten, Prinzip „divide et impera“ (teile und herrsche) ab.

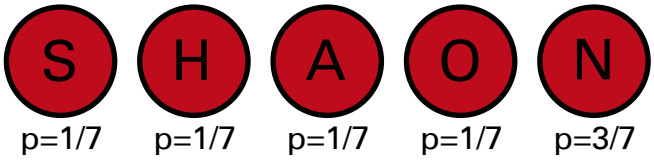
1. Alle zu kodierenden Zeichen werden nach ihrer Wahrscheinlichkeit geordnet.

2. Die Zeichen werden in zwei Gruppen mit möglichst gleichen Wahrscheinlichkeitswerten geteilt.

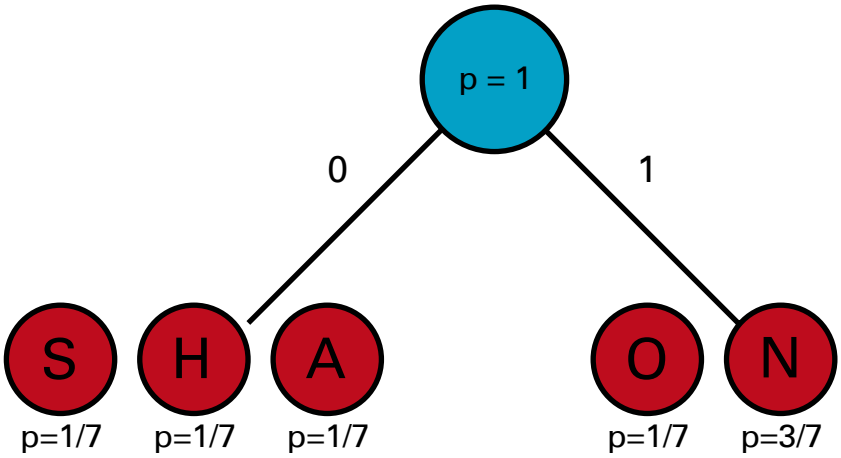
3. Die linke Gruppe bekommt eine Eins, die rechte eine Null zugeordnet.

4. Wenn eine Gruppe noch mehr als ein Zeichen enthält wird bei 2. fortgesetzt. Ansonsten werden die Einsen und Nullen der jeweiligen Zeichen zu Codewörtern aneinandergereiht.

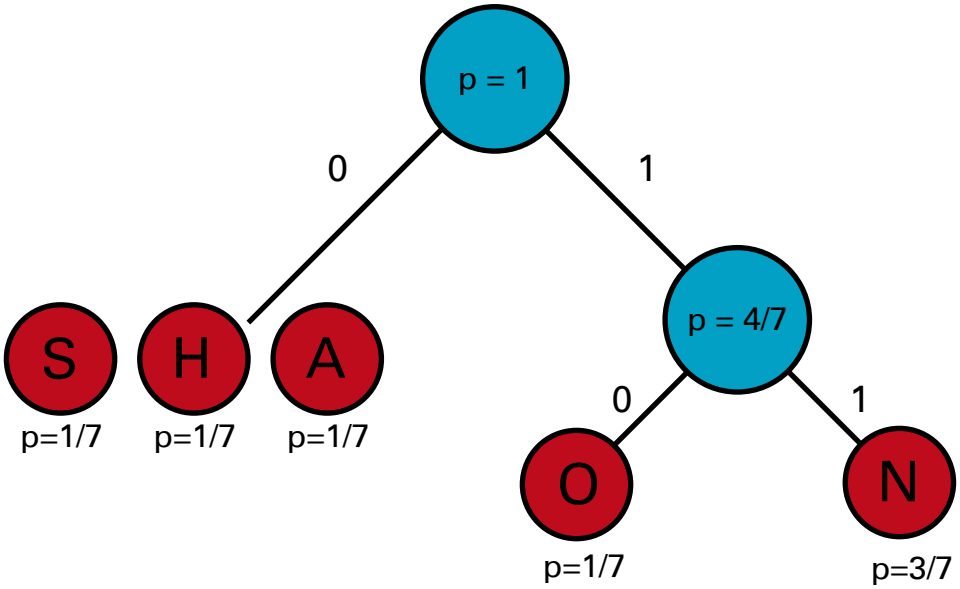
In dem Beispiel aus den Abbildungen wird das Wort „SHANNON“ kodiert.



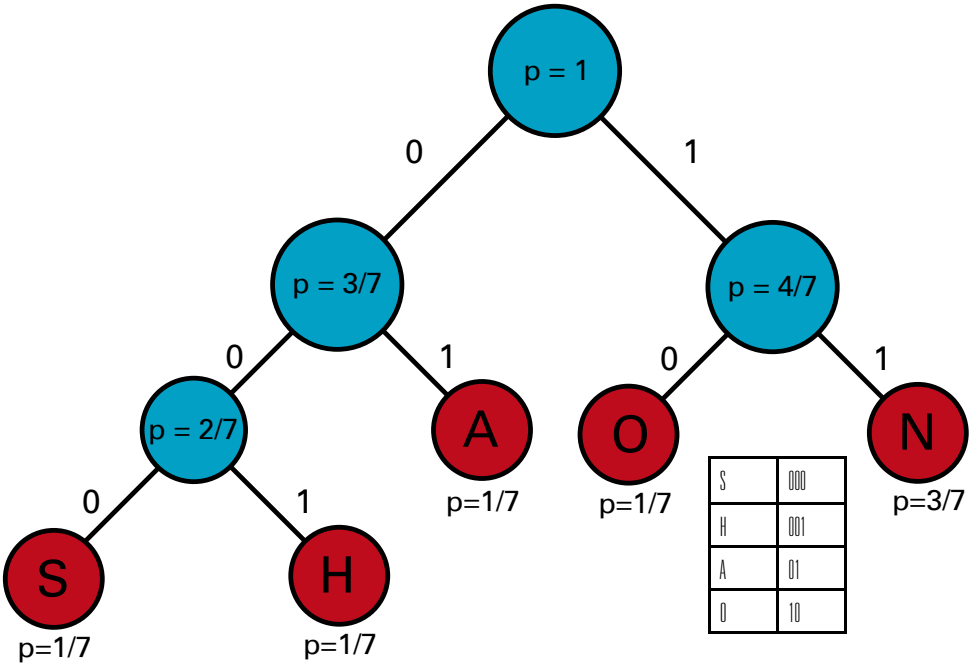
Die Zeichen werden ihren Wahrscheinlichkeitswerten p nach geordnet.



Die Symbole werden in zwei circa gleichgroße Gruppen (p=3/7 und p=4/7) geteilt.



Die Untergruppen werden nach demselben Verfahren geteilt...



...bis alle Symbole aufgelöst sind.

Auch die Shannon-Fano-Kodierung ist noch nicht optimal. So ist es nicht immer eindeutig, welche Symbole welchen Gruppen zugeordnet werden müssen. Dadurch können Ergebnisse entstehen, die nicht optimal sind. Im Beispiel sieht man, dass es trotz gleicher Wahrscheinlichkeitswerte zu unterschiedlich langen Codewörtern kommt.

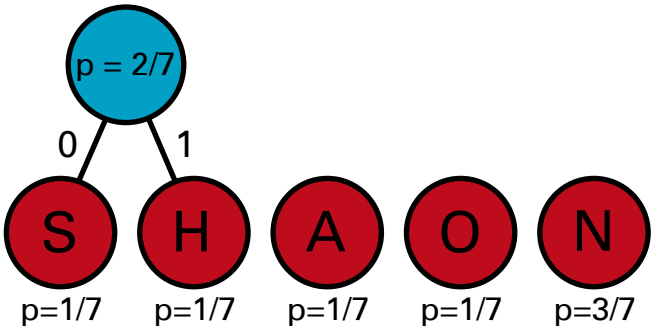
2. Huffman-Kodierung

Die Nachteile der Shannon-Fano-Kodierung wurden 1952 von David A. Huffman behoben. Mit seiner Herangehensweise wird in jedem Fall eine optimale Codewortzuweisung gefunden. Anders als der Shannon-Fano Code, der von der Baumwurzel ausgeht, wird der Huffman Code von den Blättern aus aufgebaut und endet bei der Wurzel. Im Beispiel sieht man, dass die Zeichen mit gleichen Wahrscheinlichkeiten auch gleichlange Codewörter haben. Das N hat die größte Auftretenswahrscheinlichkeit und ist nun nur noch ein statt zwei Bit groß.
Die Kodierung lässt sich wie folgt erklären:

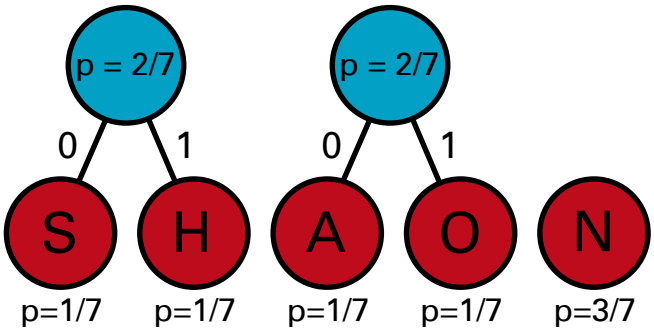
1. Alle Symbole sind Blätter eines Codebaumes.

2. Die beiden Blätter mit den geringsten Wahrscheinlichkeitswerten werden zu einem Blatt zusammengefasst, das ihre Summe zugewiesen bekommt.

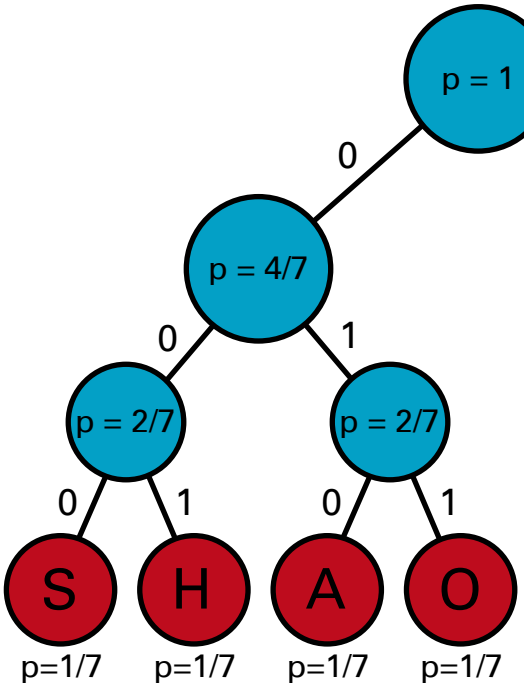
3. Die beiden entstandenen Zweige bekommen eine 1 bzw. 0 zugewiesen.



Die beiden geringsten Wahrscheinlichkeitswerte werden zusammengefasst.



Dies wird fortgeführt...



S	000
H	001
A	010
O	011
N	1

...bis die Wahrscheinlichkeit p = 1 erreicht ist.

2.1 Anwendung der Huffman-Kodierung

Die Huffman-Kodierung wird in vielen Bereichen der Datenkompression angewendet. Meist geschieht dies im Zuge weiterer Kompressionsverfahren. Dateitypen, bei denen die Huffman-Kodierung Verwendung findet sind zum Beispiel: MPEG-1 MPEG-2, JPEG, ZIP und MP3 [BIN06].

Dabei unterscheidet man zwischen drei verschiedenen Verfahren der Kodierung:
Bei dem statischen Verfahren wird dem En- und Dekodierer jeweils ein vorgefertigter Huffman-Baum mitgeliefert. Dieser Baum basiert auf einer allgemeinen Annahme der Verteilung der Zeichen. Dies kann zum Beispiel die Annahme der Buchstabenverteilung von englischsprachigen Texten sein. Dies wird beim Komprimieren solcher Texte auch zu einem akzeptablen Ergebnis führen. Problematisch wird dieses Verfahren, wenn keine nützliche Annahme über die Zeichenverteilung gemacht werden kann. Dies ist zum Beispiel häufig bei Videokompression der Fall.
Ein weiteres Verfahren ist daher die dynamische Zeichenverteilung. Hier wird nicht von einer Standardverteilung ausgegangen, sondern der Baum basiert auf einer Häufigkeitsanalyse, die vor der Enkodierung generiert wird. Dadurch wird eine wesentlich bessere Übereinstimmung mit den realen Daten erreicht. Der Nachteil beim dynamischen Verfahren ist, dass die Daten zur Erstellung des Huffman-Baumes in der Datei gespeichert werden müssen, damit sie wieder dekodiert werden kann. Außerdem wird der am Anfang erstellte Baum für den gesamten zu dekodierenden Datenbestand angewandt. Ändert sich nun die Häufigkeitsverteilung in den Daten, wird dies nicht berücksichtigt. Durch Segmentierung lässt sich diesem Problem aber entgegenwirken.
Bei der adaptiven Huffman-Kodierung ändert sich der Baum entsprechend der eingelesenen Daten kontinuierlich. Dadurch passt sich der Baum Verteilungsänderungen in den Daten immer wieder an. Beim Einlesen eines Zeichens wird zunächst geprüft, ob es bereits im Baum enthalten ist. Ist dies nicht der Fall, wird es mit der geringsten Wertigkeit angehängt. Ist es vorhanden, so wird die Wertigkeit um Eins erhöht. Am Anfang der Kodierung ist die Kompressionsleistung schlecht, da die Zeichen zuerst einmal in den Baum eingefügt werden müssen. Daher ist dieses Verfahren für kleine Dateien ungeeignet. Die Leistung verbessert sich aber, je weiter die Kodierung fortschreitet. Die adaptive Kodierung bietet zwar die beste Kompression, ist aber aufgrund des Verwaltungsaufwands auch die Langsamste.

Obwohl mittlerweile viele neue Kodierungsverfahren entwickelt wurden, spielt die Huffman-Kodierung in der Datenkompression noch immer eine Rolle. Vor allem weil sie so einfach zu implementieren ist und sich die Optimierung des Codes mathematisch nachweisen lässt. Arithmetische Kodierung wird in der Literatur häufig als die bessere Alternative dargestellt. Es hat sich aber gezeigt, dass die Kompressionsleistung nicht immer besser ist und der hohe Implementationsaufwand oft nicht gerechtfertigt ist [BOK93].

3. Quellen

3.1 Literaturverzeichnis

[OHM95]: Jens Rainer Ohm - Digitale Bildcodierung

[STR02]: Tilo Strutz - Bilddatenkompression

[BOK93]: A. Bookstein and S.T. Klein - Is Huffman Coding Dead?

3.2 Internetquellen

[BIN06] http://www.binaryessence.de/ Stand: 27.11.06

Alle Grafiken wurden vom Autor angefertigt.