

# Datenkodierung

*Jeschko Fabian*

## Inhaltsverzeichnis

1	Einführung . . . . .	1
2	Binärcode . . . . .	1
2.1	Binary Coded Decimal . . . . .	2
2.2	Gray Code . . . . .	2
2.3	Huffman Code . . . . .	3
3	Zusammenfassung . . . . .	4

## 1 Einführung

Datenkodierung ist der Prozess, in dem man Informationen jeglicher Art in einen bestimmten Code umwandelt. Dieser Code kann die Daten entweder kompakter, besser erkenntlich, verarbeitbar oder auf viele andere Arten darstellen, je nachdem wie es gerade gewünscht ist. Kodierung findet man überall im Alltag, nachdem die menschliche Sprache, DNA, Computerprogramme alles eine kodierte Ansammlung von Informationen ist.

Üblicherweise meint man jedoch wenn man von Kodierungen spricht die, die im Informatikbereich verwendet werden. Die wichtigsten Kodierungen sind dabei wohl die Binärkodierung, welche von jedem Computer verwendet wird, die Leitungskodierungen, welche für sämtlichen Datenverkehr in der Informatik zuständig sind, Quellcode mit dem Programme, Datenstrukturen und vieles mehr aufgebaut ist. Zudem gibt es noch etliche andere Kodierungen die sehr wichtig sind, wie in etwa der Bytecode, Strichcodes und die Kanalkodierung. Dabei werde ich hier speziell genauer auf den Binärcode eingehen und diverse Abwandlungen davon genauer erläutern.

## 2 Binärcode

Der Binärcode stellt Daten als Sequenzen von zwei Zeichen, meist 0 und 1 bei Computerprogrammen oder High und Low in dem Bereich der Elektronik oder True und False im Bereich der Semantik. Ein Grund warum der

Binärcode so weit verbreitet ist, weil dieser sehr Hardware nahe ist, da man 0 mit keiner Spannung und 1 mit Spannung leicht realisieren kann.<sup>1</sup>

Von dem Binärcode gibt es viele verschiedene Varianten, welche den selben Zeichensatz verwenden, jedoch stabiler sind oder eine bessere Darstellung bieten. Beispiele hierfür sind der BCD-Code, siehe 2.1, welcher für die Darstellung von Dezimalzahlen verantwortlich ist, oder auch der Gray-Code, siehe 2.2, mit welchem die Anfälligkeit auf Fehler deutlich reduziert wird. Außerdem gibt es noch Binärcores mit denen man Fehler erkennen und sogar ausbessern kann.

## 2.1 Binary Coded Decimal

Da Dezimalzahlen und Binärzahlen verschiedene Basen haben und Zehn auch keine Potenz von Zwei ist, geht es schwerer zwischen diesen Systemen umzurechnen.<sup>2</sup> Für diesen Zweck gibt es den BCD-Code, welcher jede Dezimalzahl als vier Bit darstellt. Damit können dann die Zahlen 0 - 9 dargestellt werden als 0000 - 1001 und alles darüber wird als ungültig bezeichnet. Die gültigen Zahlen werden dabei Tetraden genannt, wobei die ungültigen Zahlen Pseudotetraden genannt werden. Somit wird nun jede Ziffer als ein vier Bit Block dargestellt und Zahlen als Folge dieser 4 Bit Blöcke. Da der BCD-Code jede Ziffer einzeln coded, wohingegen das normale Binärsystem die gesamte Zahl coded entstehen nun verschiedene Bitfolgen für die selbe Zahl in den verschiedenen Darstellungen. Die ungültigen Ziffern im BCD-Code lassen deshalb die Bitsequenzen schneller wachsen und es benötigt neue Rechenregeln, damit die Pseudotetraden verhindert werden können. Um negative Zahlen darstellen zu können, wird üblicherweise das erste Bit der vordersten Tetrade 0 für positiv und 1 für negativ gesetzt und man die hinteren 3 Bit nicht mehr verwendet.<sup>3</sup>

## 2.2 Gray Code

Der normale Binärcode kann große Fehler produzieren wenn mehrere Bits gleichzeitig umschalten sollen. Solche Zahlen wären zum Beispiel 1111 und ähnliches das längere 1-Folgen hat. Wenn man nun diese Zahl um 1 erhöht, kann es kurzzeitig dazu kommen, dass die Zahl 1 1111 dargestellt wird, dann 1 0111 und dies bis dann schlussendlich 1 0000 da steht. Besonders wenn

<sup>1</sup> <http://www.itwissen.info/definition/lexikon/Binaercode-BC-binary-code.html>

<sup>2</sup> [http://faculty.kfupm.edu.sa/COE/about/Lesson1\\_7.pdf](http://faculty.kfupm.edu.sa/COE/about/Lesson1_7.pdf)

<sup>3</sup> <http://www.sps-lehrgang.de/bcd-code/>

etwas mit sich ständig wechselndem Wert dargestellt werden sollte in Dezimal, können somit sehr große Abweichungen zustande kommen, obwohl es nur minimale Änderungen sind. Dies kann man jedoch verhindern mit Hilfe des Gray-Codes, durch welchen sich die Binärzahl bei einer inkrementierung immer nur um ein Bit verändert. Die Binärzahlen werden dabei in einem Zykel dargestellt und vom höchstem zum niedrigstem Bit ist es auch nur eine Distanz von 1 Bit. Als Beispiel dient hierfür der Graycode für 3 Bit:

Dezimal	Binär	Gray-Code
0	000	000
1	001	001
2	010	011
3	011	010
4	100	110
5	101	111
6	110	101
7	111	100

Benutzt wird der Graycode oft bei Disks, aufgrund seiner zyklischen Eigenschaft und um die Lage eines sich drehenden Objekts zu erkennen. [DD07]

### 2.3 Huffman Code

Codes sind jedoch nicht auf bessere Darstellung oder weniger Fehleranfälligkeit beschränkt. Der Huffman-Code ist dazu in der Lage Texte zu komprimieren, indem weniger Bits für häufig verwendete Zeichen benutzt werden.

Der Huffman-Code benötigt eine Liste aus Symbolen, die vorkommen zusammen mit ihrer Häufigkeit. Diese Liste wird dann nach der Häufigkeit der Zeichen sortiert bevor man damit weiter arbeiten kann. Wenn dies der Fall ist, kann man die ersten zwei Einträge als Knoten betrachten, die einen gemeinsamen Vaterknoten haben. Diese zwei Knoten werden dann von der Liste entfernt und der Vaterknoten wird in die Liste eingefügt mit der Summe der Häufigkeiten und neu sortiert. Dies wiederholt man dann bis nur noch zwei Knoten vorhanden sind, welche dann wieder einen gemeinsamen Vater haben, welcher dann die Wurzel des daraus entstandenen Baumes ist. Bei diesem Baum werden dann alle Abzweigungen nach links mit 0 nummeriert und alle Abzweigungen nach rechts mit 1. Daraus ergibt sich für jeden Blattknoten ein Code, welcher aus dem Weg zu diesem Knoten besteht. Damit ist eine minimale Länge für die gesamten Daten garantiert. Lesbar ist dieser Code, da keiner der komprimierten Codes Prefix eines anderen ist und somit sofort ausgewertet werden kann. [LJL99]

### 3 Zusammenfassung

Datenkodierung ist ein riesiges Themengebiet, welches unmöglich in einer vier Seiten Arbeit umfangreich erklärt werden kann. Selbst wenn man sich spezialisiert auf die Informatik gibt es dort auch schon verschiedenste Codierungsarten mit unterschiedlichen Zeichensätzen, sowie Hex-Code, Octal-Code, Byte-Code und vieles mehr. Viele dieser Codes können jedoch mit Hilfe von ihrer Grundbasis 2, oder auch Binärcode dargestellt werden, weswegen dieser wohl der wichtigste ist. Diese Codes kann man meines erachtens in die großen Kategorien der Leserlichkeit, Effizienz und Fehlerkontrolle/Fehlerausbesserung/Fehlerermeidung einordnen, zu welchen ich jeweils eine Kodierung erklärt habe. Es ist durchaus ein interessantes Thema und besonders ein Umfangreiches, in welchem es ständig neue Entdeckungen und Kodierungen gibt, welche Vorteilhafter gegenüber anderen sind.

#### Literatur

[DD07] R. W. Doran and R W Doran. Abstract the gray code, 2007.

[LJL99] Glen G. Langdon, Jr., and Glen G. Langdon. Huffman codes, 1999.