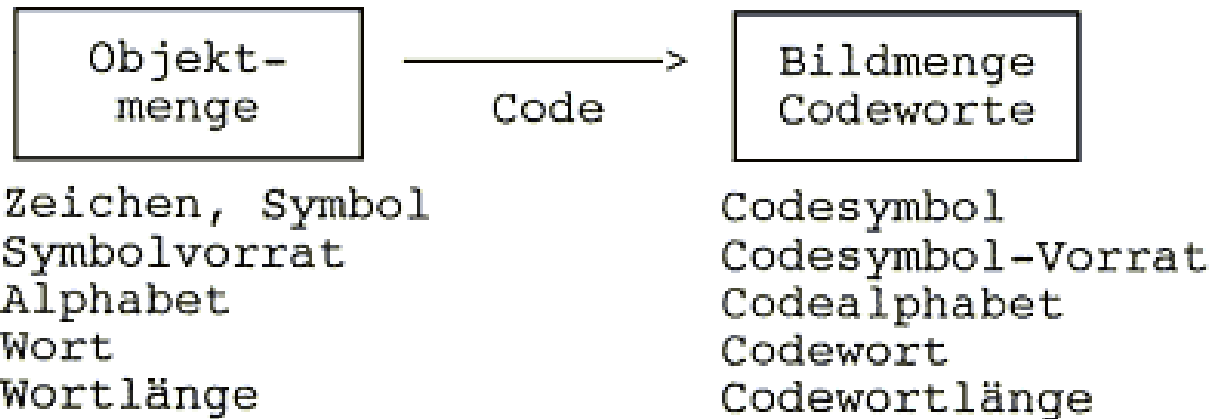
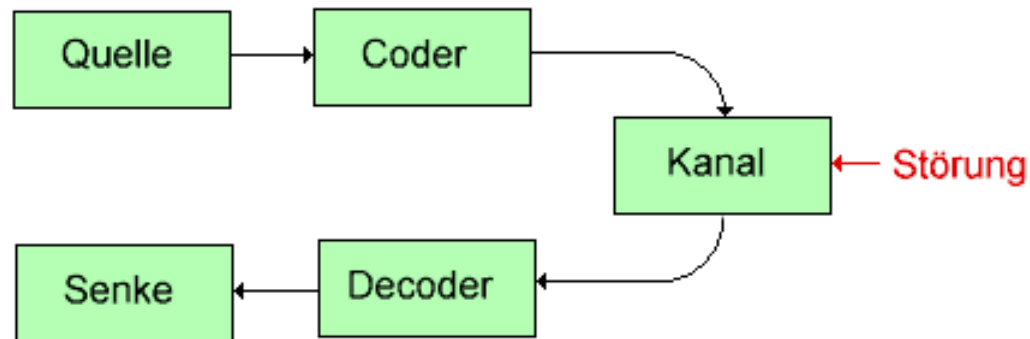


Codes

→ Vorschrift für die eindeutige Zuordnung (= Codierung) der Zeichen einer Zeichenvorrats (Objektmenge) zu den Zeichen eines anderen Zeichenvorrats (Bildmenge).



Zweck der Codierung: Anpassung der Nachricht an technische Systeme (z. B. Morsecode). Die Codierung ändert nur die Darstellungsform einer Nachricht, nicht ihre Bedeutung.



BinärCode

Codealphabet: 0 ; 1

Codeworte : Dualzahlen

Hat ein Codewort n Zeichen, dann können $M=2^n$ (verschieden, gleichlange) Codewörter gebildet werden.

D.h. (umgekehrt): Wenn M Worte codiert werden müssen, dann muss ein Codewort die Länge $n \geq \text{Id}(M)$ ($n \in \mathbb{N}$) haben. ($\text{Id } x = \log_2 x$)

Beispiel: Es sind die Ziffern des dekadischen Codes im BinärCode darzustellen:

- *Es sind 10 Worte zu codieren $\rightarrow M=10$*
- *$\text{Id}(M)=\text{Id}(10)=3,32\dots \rightarrow n=4$*
- *Ein Codewort muss also die Länge $n=4$ haben*

Ist M keine Zweierpotenz, können mehr Codeworte gebildet werden als gebraucht werden.

Die nicht verwendeten Codeworte heißen Pseudowörter \rightarrow Redundanz.

Im Beispiel gibt es 6 Pseudowörter $\rightarrow 2^4=16 \rightarrow 16-10=6$



"Vom schweigsamen König, der gern Schweinebraten aß"

(Frei nach: Walter R. Fuchs: Knauer's Buch der Denkmaschinen, 1968)

In einem fernen Land lebte vor langer, langer Zeit ein kleiner König, der war dick faul und unzufrieden und wollte den lieben langen Tag immer nur essen. Wir wollen hier nur seine Nahrungsgewohnheiten betrachten - insbesondere, da seine Ernährung recht einseitig war. Den lieben langen Tag aß er nur:

(1) Schweinebraten, (2) Schokoladenpudding, (3) Essiggurken, (4) Erdbeertorte.

Zudem war der König sehr maulfaul und mit der Zeit wurde es ihm sogar zu anstrengend, seine Bestellungen aufzugeben (die er sowieso im Telegrammstil kundtat: "Braten, Torte, Gurken"). Eines Tages beschloss er, eine Codierung zu entwickeln, mit der er seine Befehle auch loswurde, ohne den Mund aufzutun. Durch Zufall wurde es sogar eine [Binärcodierung](#):

- Die rechte Hand ein wenig heben heiße: Schweinebraten
- Die linke Hand etwas heben heiße: Schokopudding
- Erst die rechte und dann die linke Hand heben: Gurken
- Zweimal nacheinander die rechte Hand heben: Erdbeertorte

Warum

Der Übersichtlichkeit halber wollen wir die Codierung etwas abkürzen:

- "R" steht für "rechte Hand heben" und
- "L" für "linke Hand heben".

Dann ergibt sich folgende Codezuordnung:

R→Braten L→Pudding RL→Gurken RR→Torte

Was will der König, wenn er dreimal nacheinander die rechte Hand hebt ?



Angenommen der König hob dreimal die rechte Hand (--> RRR). Dann konnte dies bedeuten:

Braten, Braten, Braten oder Braten, Torte oder Torte, Braten

Nun musste der König einmal wirklich viel reden. Er rief seinen Hofmathematiker zu sich und erklärte den Sachverhalt. "Hmmm" überlegte dieser: "Majestät haben Höchstdero Speisewünsche binär codiert. Vorzüglich, Vorzüglich."

"Aber es klappt nicht", raunzte der König, "jedes Mal, wenn ich Schweinebraten und Pudding will, bringen diese Hornochsen mir Gurken!" und er sank erschöpft auf seinen Thron zurück. "Mit Verlaub, die Codierung ist nicht eindeutig, Majestät", wagte der Mathematiker einzuwerfen. "Ich weiß! Lass Dir gefälligst was einfallen!" grunzte der König. Und vor lauter Angst, wieder etwas Falsches zu bekommen, brüllte er: "Braten und Torte, aber fix!".

Der Mathematiker brütete inzwischen über eine eindeutige binäre Codierung nach und gelangte zu folgenden Überlegungen: ?

Es sind vier Worte binär zu codieren, also brauche ich eine Codewortlänge von mindestens 2 Binärzeichen.

Die Codierung der vier Speisewünsche sah folgendermaßen aus:

- Braten → RR
- Pudding → RL
- Gurken → LR
- Torte → LL

Nun war die Codierung unverwechselbar. Die Zeichenfolge "LRLLLRRRLL" konnte nur noch bedeuten: ?

Gurken, Pudding, Torte, 2 x Braten, Torte.



Gründe für die Codierung

- Geeignete Nachrichtendarstellung für die technische Verarbeitung:
 - Erfassung und Aufbereitung
 - Übertragung
 - Sortierung
 - Verknüpfung und Bearbeitung
 - Ausgabe
- Ökonomische Darstellung von Informationen:
 - Übertragungsgeschwindigkeit
 - Energieersparnis
 - Aufwand für Verknüpfung und Verarbeitung
 - Speicherplatzbedarf
- Sicherung der Information gegen Verfälschung:
 - Übertragungsfehler
 - Verarbeitungsfehler
- Schutz der Informationen vor unbefugtem Zugriff (Verschlüsselung, Kryptologie)



BCD-Codes (binary coded decimal)

- Jeder Ziffer einer Dezimalzahl wird ein binäres Codewort zugeordnet.
- Ein Codewort muss die Länge 4 haben (vgl. [Bsp. vorher](#)).
 - Tetradischer Code → ein Codewort heißt auch Tetrade
- Es gibt 16 Tetraden von denen aber nur 10 benötigt werden → 6 Pseudotetraden

8-4-2-1-Code (oft als **der** "BCD-Code" bezeichnet)

- direkte Darstellung der Ziffern 0-9 im Dualcode
- gewichteter Code

Jede Stelle der Tetrade besitzt ein bestimmtes Gewicht (8-4-2-1). Die Summe aus allen Dualziffern, jeweils multipliziert mit ihrem Gewicht, ergibt den Wert der codierten Dezimalzahl, z. B.: 0101 → $0 \cdot 8 + 1 \cdot 4 + 0 \cdot 2 + 1 \cdot 1 = 5$

- monoton wachsend
- einfache Zähl- und Addierschaltungen (Korrektur beim Übertrag notwendig!)
- Unterscheidung gerade/ungerade

Andere (BCD) Codes

- 2-4-2-1-Code (Aiken-Code)
- 5-4-2-1-Code (Biquinär-Code)
- Exzeß-3-Code (Stibitz-Code)
- Mehrstellige BCD-Codes ($n > 4$, nicht tetradisches)
- Einschrittige BCD-Codes
- Mehrschrittige Anordnungs-codes



BCD-Code (8-4-2-1)

| Dez | Tetrade |
|-----|---------|
| 0 | 0000 |
| 1 | 0001 |
| 2 | 0010 |
| 3 | 0011 |
| 4 | 0100 |
| 5 | 0101 |
| 6 | 0110 |
| 7 | 0111 |
| 8 | 1000 |
| 9 | 1001 |

Nicht benötigte Tetraden:

1010
1011
1100
1101
1110
1111



6 Pseudotetraden

Rechnen mit BCD-codierten Zahlen

Dazu zunächst: Rechnen mit Dualzahlen

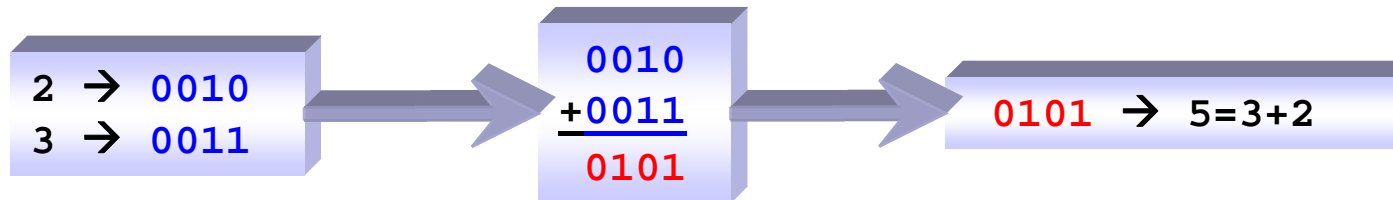
$$0 + 0 = 0$$

$$1 + 0 = 1$$

$$0 + 1 = 1$$

$$1 + 1 = 0 \text{ Übertrag } 1$$

Es sind die Dezimalzahlen 2 und 3 in ihrer BCD-Codierung zu addieren.



Führe diese Überlegungen durch für 3+5 ; 3+9; 6+8 ; 9+7 ; 7+8 und 9+8.
Versuche Rechenregeln für das Rechnen im BCD-Code abzuleiten.

