# 2. Digitale Codierung und Übertragung

2.1 Informationstheoretische Grundlagen

2.2 Verlustfreie universelle Kompression

2.3 Digitalisierung, Digitale Medien

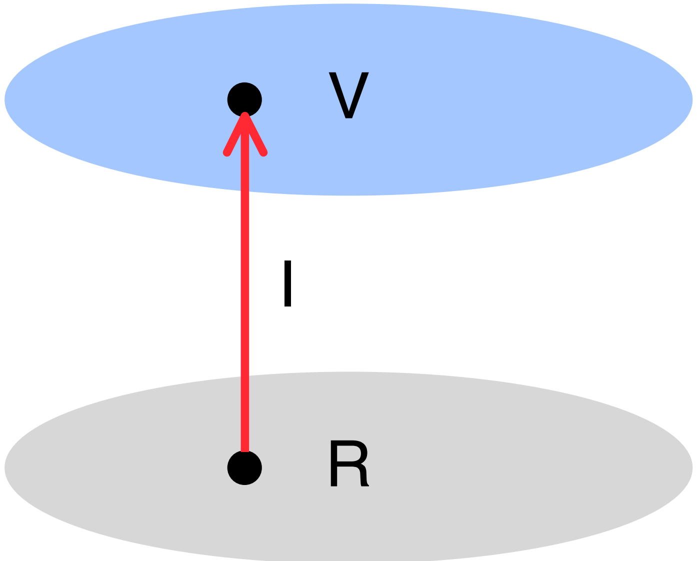
Weiterführende Literatur zum Thema Informationstheorie:

Taschenbuch Medieninformatik Kapitel 2

Herbert Klimant, Rudi Piotraschke, Dagmar Schönfeld:
Informations- und Kodierungstheorie, Teubner 2003

# Information und Repräsentation

- V = Menge von *Werten* (Interpretationen, Bedeutungen)
- R = Menge von *Repräsentationen* (Darstellungswerten)
- Abbildung
  $I : R \rightarrow V$ *Interpretation*
- Umkehrung zur Interpretation: *Repräsentationsbeziehung* $I^{-1} : V \rightarrow R$



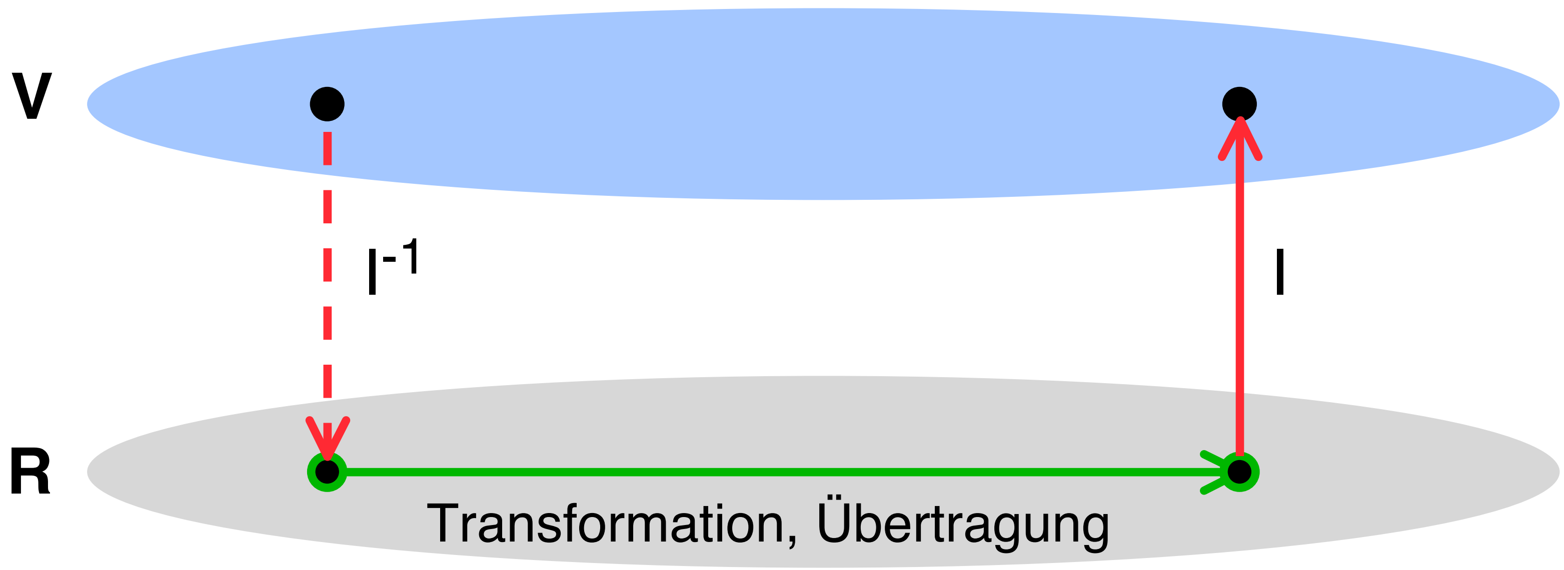

Klassische Beispiele:
V = Zahlwerte, R = Binärzahlen
V = Abbildungen, R = Programme

Hier betrachtete Beispiele:
V = textuelle Aussagen (z.B.)
R = Bilder, R = Klänge, ...

(nach Broy: Informatik Teil I)

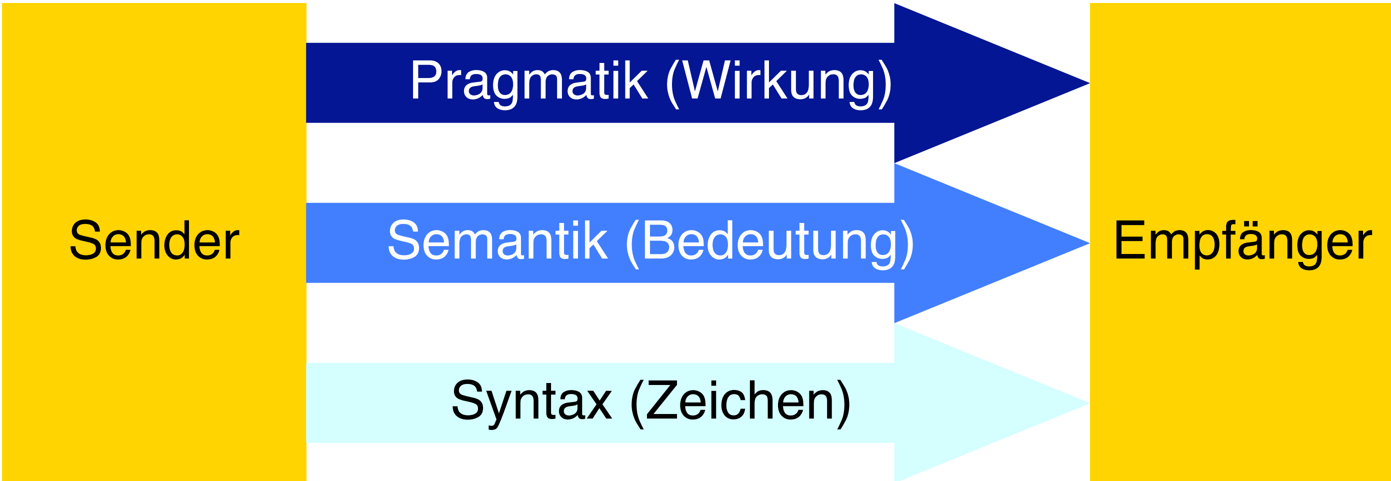
# Informationsverarbeitung

- Information ist ein abstrakter Begriff.
- Computer verarbeiten immer Repräsentationen.
- Informationsverarbeitung ist Repräsentationsverarbeitung.
- Medien sind spezielle Repräsentationen von Information.



# Semiotische Ebenen

- Semiotik = Theorie der Zeichen und Symbole
- Klassische Terminologie der Semiotik: Syntax, Semantik, Pragmatik

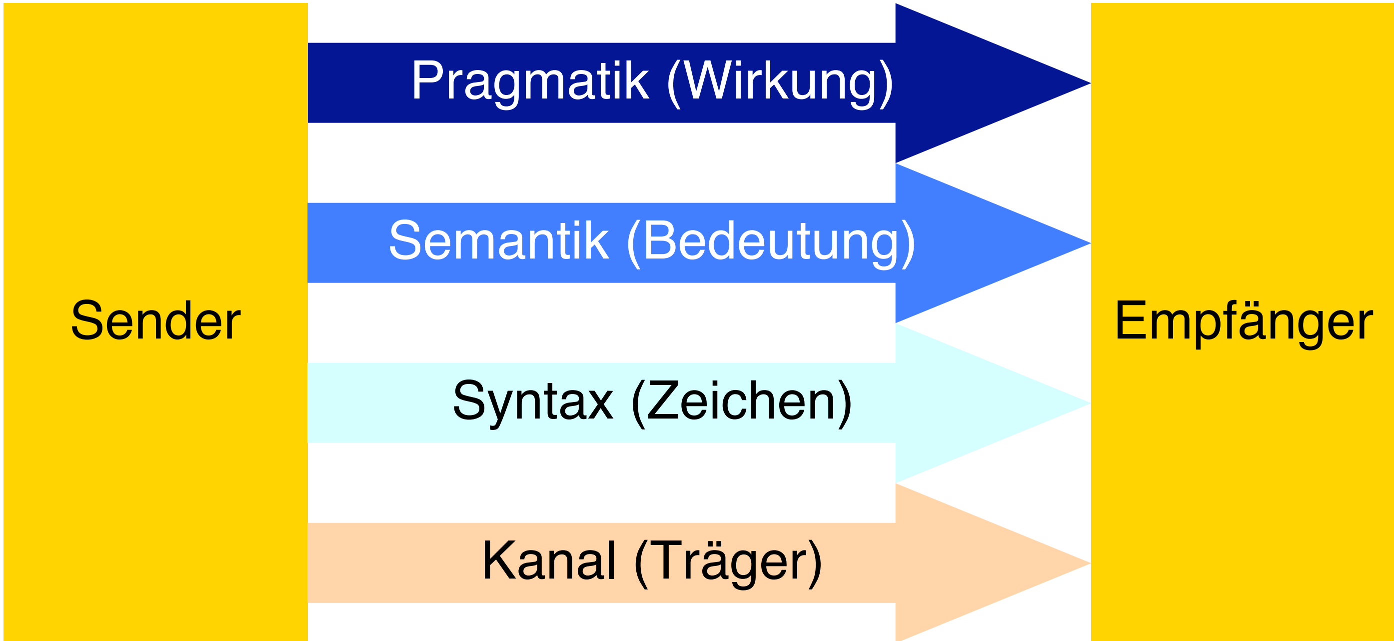



Bezug zur traditionellen Informatik:

Syntax = Repräsentationen (Menge R)
Semantik = Informationsgehalt (Menge V)
Pragmatik wird als irrelevant angesehen

# Semiotische Ebenen in der Medieninformatik

- Für Medien müssen alle semiotischen Ebenen betrachtet werden.
  - Z.B. Wirkung eines Textes abhängig von der grafischen Darstellungsform (Farbe, Größe, Platzierung)

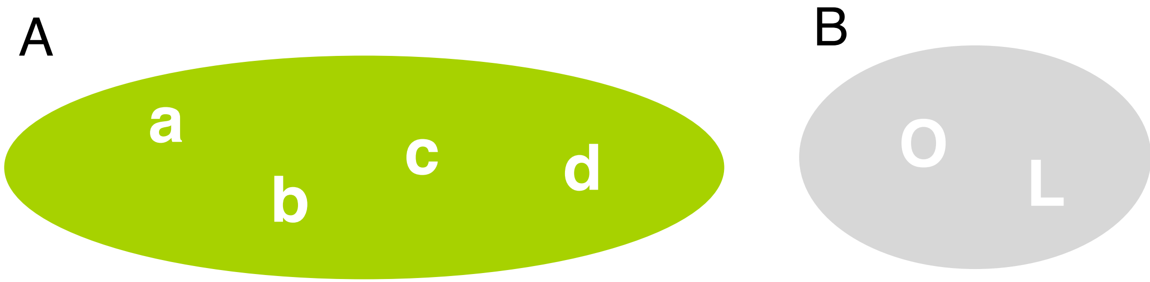



- Für die technische Realisierung sind Eigenschaften des physikalischen Trägers der Repräsentation ebenfalls wesentlich.
  - Z.B. Speicherbedarf, Frequenzspektrum

# Stochastische Informationstheorie: Zeichenvorräte und Codierung

- Ein *Zeichenvorrat* ist eine endliche Menge von *Zeichen*.
- Eine Nachricht (im Zeichenvorrat A) ist eine Sequenz von Zeichen aus A
- Seien A und B Zeichenvorräte.
  Eine *Codierung c* ist eine Abbildung von Nachrichten in A auf Nachrichten in B.

  $c: A \rightarrow B^*$ ($B^*$ : Zeichenreihen über $B$)
- Wir beschränken uns meist auf *binäre* Codierungen, d.h. B = { 0, 1 }
- Die *Informationstheorie* (nach *Shannon*) befasst sich mit Nachrichtenquellen auf der Ebene der Syntax aus *stochastischer* Sicht
  – Zeichen und zugehörige Codierung haben immer identische Interpretation



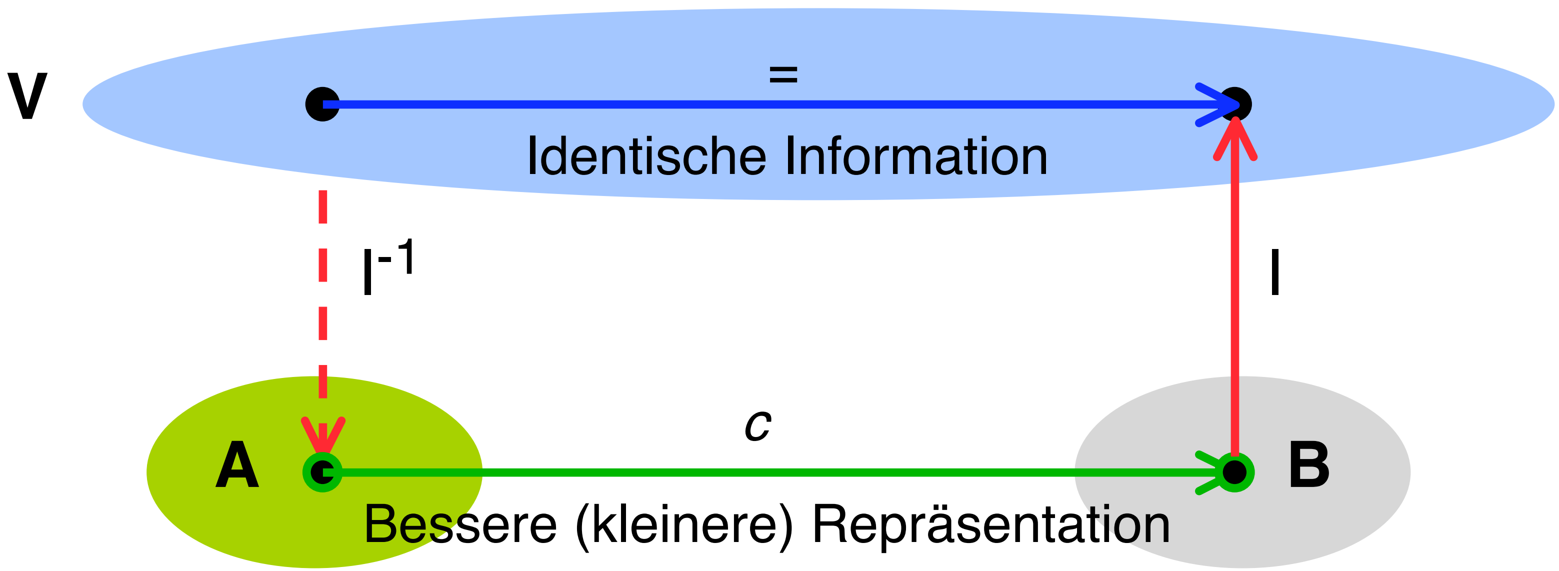

Beispiel:

abca → 00011000

ddc → 111110

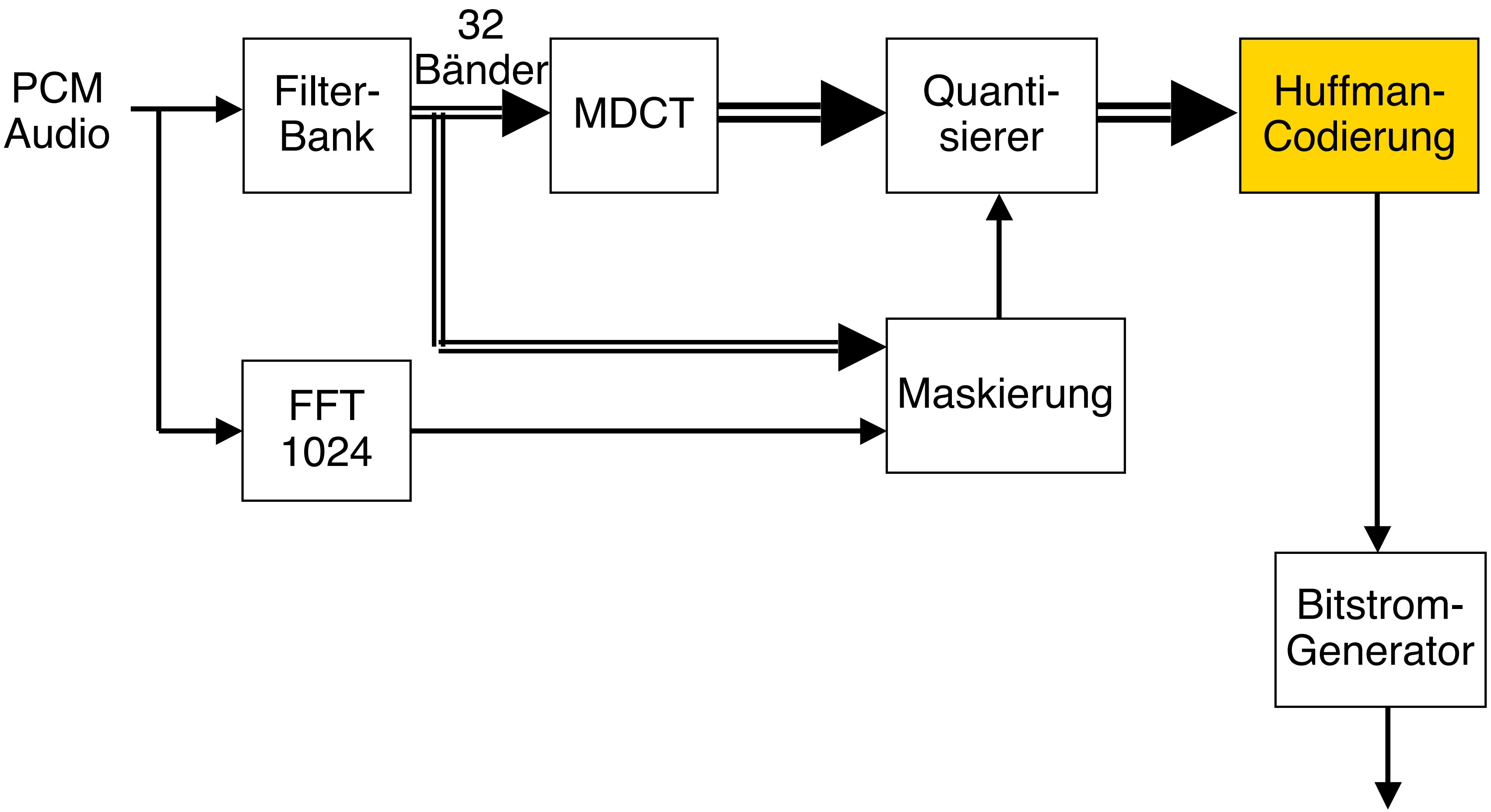
# Interpretation und Codierung

- Es gibt Codierungen verschiedener Effizienz für die gleiche Information.
- Die Informationstheorie betrachtet eine Informationsquelle nach Eigenschaften, die eine bessere (kürzere) Codierung erlauben.
- Informationsquelle wird durch einen Basiszeichenvorrat mit zusätzlichen Informationen (z.B. Häufigkeitsverteilung) erfasst.



# Einschub: Motivation für Informationstheorie

- Aufbau eines MPEG-Layer III (MP3) Encoders
  - Details siehe später!



# Entropie (1)

- Annahme *Stochastische Nachrichtenquelle:* Wir kennen die Häufigkeitsverteilung der Zeichen in den Nachrichten.
- *Entscheidungsgehalt* (*Entropie*) der Nachrichtenquelle:
  - Wie viele Ja/Nein-Entscheidungen entsprechen dem Auftreten eines Einzelzeichens?
  - Eine Ja/Nein-Entscheidung = 1 „bit“
- Beispiele:

Quelle 1

| Zeichen $a$ | A | B | C | D |
|---|---|---|---|---|
| Häufigk. $p_a$ | 1 | 0 | 0 | 0 |
| $x_a$ | 0 | - | - | - |

Quelle 2

| Zeichen $a$ | A | B | C | D |
|---|---|---|---|---|
| Häufigk. $p_a$ | 0.25 | 0.25 | 0.25 | 0.25 |
| $x_a$ | 2 | 2 | 2 | 2 |

$p_a$ = Häufigkeit

$x_a$ = Zahl der Entscheidungen

$$2^{x_a} = 1/p_a$$

$$x_a = \mathrm{ld}\,(1/p_a)$$

(Logarithmus zur Basis 2)

# Entropie (2)

- *Durchschnittlicher* Entscheidungsgehalt je Zeichen: *Entropie H*

$$\boxed{H = \sum_{a \in A} p_a \, ld\left(\frac{1}{p_a}\right)}$$

mit $x_a = \text{ld}\,(1/p_a)$: $H = \sum_{a \in A} p_a x_a$

Quelle 1

| Zeichen $a$ | A | B | C | D |
|---|---|---|---|---|
| Häufigk. $p_a$ | 1 | 0 | 0 | 0 |
| $x_a$ | 0 | - | - | - |

$H = 0$

Quelle 2

| Zeichen $a$ | A | B | C | D |
|---|---|---|---|---|
| Häufigk. $p_a$ | 0.25 | 0.25 | 0.25 | 0.25 |
| $x_a$ | 2 | 2 | 2 | 2 |

$H = 2$

Quelle 3

| Zeichen $a$ | A | B | C | D |
|---|---|---|---|---|
| Häufigk. $p_a$ | 0.5 | 0.25 | 0.125 | 0.125 |
| $x_a$ | 1 | 2 | 3 | 3 |

$H = 1.75$

Entropie ist Maß für „Unordnung“, „Zufälligkeit“

# Wortlängen und Redundanz

- Eine (Binär-)Codierung der Nachrichten einer stochastischen Nachrichtenquelle ergibt eine *durchschnittliche Wortlänge L*.

$$\boxed{L = \sum_{a \in A} p_a \left| c(a) \right|}$$

Quelle 2

| Zeichen *a* | A | B | C | D |
|---|---|---|---|---|
| Häufigk. $p_a$ | 0.25 | 0.25 | 0.25 | 0.25 |
| Code *c(a)* | 00 | 01 | 10 | 11 |

$H =$ 2
$L =$ 2

Quelle 3

| Zeichen *a* | A | B | C | D |
|---|---|---|---|---|
| Häufigk. $p_a$ | 0.5 | 0.25 | 0.125 | 0.125 |
| Code *c(a)* | 00 | 01 | 10 | 11 |

$H =$ 1.75
$L =$ 2

- $\boxed{\textit{Redundanz} = L - H}$
- Redundanz ist ein Maß für die Güte der Codierung: möglichst klein!

# Optimale Codierung

- Eine Codierung ist *optimal*, wenn die Redundanz 0 ist.
- Durch geeignete Codierung (z.B. Wortcodierung statt Einzelzeichencodierung) kann man die Redundanz beliebig niedrig wählen.
- Redundanz ermöglicht andererseits die Rekonstruktion fehlender Nachrichtenteile!
  - B ispi l: Natürlich  Sprach
  - Beispiel: Fehlererkennende und -korrigierende Codes (z.B. Paritätsbits)

Quelle 3

| Zeichen $a$ | A | B | C | D |
|---|---|---|---|---|
| Häufigk. $p_a$ | 0.5 | 0.25 | 0.125 | 0.125 |
| Code $c(a)$ | 00 | 01 | 10 | 11 |

$H =$ 1.75
$L =$ 2

Quelle 3

| Zeichen $a$ | A | B | C | D |
|---|---|---|---|---|
| Häufigk. $p_a$ | 0.5 | 0.25 | 0.125 | 0.125 |
| Code $c'(a)$ | 0 | 10 | 110 | 111 |

$H =$ 1.75
$L =$ 1.75

# 2. Digitale Codierung und Übertragung

2.1 Informationstheoretische Grundlagen

2.2 Verlustfreie universelle Kompression

2.3 Digitalisierung

Weiterführende Literatur zum Thema Kompression:

Taschenbuch Medieninformatik Kapitel 2

Herbert Klimant, Rudi Piotraschke, Dagmar Schönfeld:
Informations- und Kodierungstheorie, Teubner 2003

Khalid Sayood: Introduction to Data Compression, 2nd. ed.,
Morgan Kaufmann 2000

# Darstellungsräume, Darstellungswerte

- Jedes (Einzel-)Medium definiert einen Darstellungsraum (= Menge der möglichen Repräsentationen R).
- Eine konkrete Repräsentation einer Information ist ein Darstellungswert innerhalb des Darstellungsraums.
- Für Perzeptionsmedien:
  - Ein Darstellungsraum richtet sich an einen bestimmten Sinn des Menschen.
- Beispiele:
  - Text: Darstellungsraum = Menge aller möglichen Zeichenfolgen
  - Bild: Darstellungsraum = Menge aller möglichen Belegungen der Wiedergabefläche mit Farbinformationen
  - Sprache: Darstellungsraum = (sehr spezifische und komplexe) Teilmenge der möglichen Verteilungen von Luftdruck über die Zeitachse

# Speicherbedarf multimedialer Information

- Bsp. Schrift
  - Laufschrift (8 bit/Zeichen, 40 Zeichen/s): 320 bit/s
- Bsp. Audio-Signale
  - Sprachsignal niedriger Qualität (Mono, 8 bit, 11 kHz): 88 kbit/s
  - CD-Qualität (Stereo, 16 bit, 44,1 kHz): 1,4 Mbit/s
- Bsp. Bilder (9x13cm = 1062x1536 Pixel)
  - Schwarz/weiß (1 bit Farbtiefe): 200 kByte
  - TrueColor (24 bit Farbtiefe): 4,9 MByte
- Bsp. Video (ohne Ton)
  - 720 x 525 Pixel, 25 Bilder/s, 16 bit Farbtiefe: 151,2 Mbit/s
  - 1280 x 720 Pixel, 60 Bilder/s, 24 bit Farbtiefe: 1,32 Gbit/s
- ***Kompression*** der Information ist extrem wichtig!

Pixel= Bildpunkt

# Kompressionsverfahren: Übersicht

- Klassifikationen:
  - Universell vs. speziell (für bestimmte Informationstypen)
  - Verlustfrei vs. verlustbehaftet
  - In diesem Kapitel: nur universelle & verlustfreie Verfahren

- Im folgenden vorgestellte Verfahren:
  - Statistische Verfahren:
    - » Huffman-Codierung
    - » Arithmetische Codierung
  - Zeichenorientierte Verfahren:
    - » Lauflängencodierung (RLE Run Length Encoding)
    - » LZW-Codierung

# Grundidee zur Huffman-Codierung

- Zeichen größerer Häufigkeit werden durch kürzere Codes repräsentiert
  - vgl. Morse-Code
- Das führt zu einem *Code variabler Wortlänge*:
  - Kein Codewort darf Anfang eines anderen sein *(Fano-Bedingung)*
- In optimalem Code müssen die beiden Symbole der niedrigsten Häufigkeit mit gleicher Länge codiert sein.

  "Beweis"-Skizze:
  - Wären die Längen verschieden, könnte man das längere Wort bei der Länge des kürzeren abschneiden
    - » Dann sind die beiden Codes verschieden (sonst wäre Fano-Bedingung vorher verletzt gewesen)
    - » Kein anderes Codewort kann länger sein (da Zeichen niedrigster Wahrscheinlichkeit), also kann die Kürzung nicht die Fano-Bedingung verletzen
  - Dann hätten wir einen neuen Code mit kleinerer durchschnittlicher Wortlänge!

# Huffman-Codierung (1)

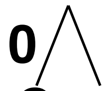
- Gegeben: Zeichenvorrat und Häufigkeitsverteilung
- Ergebnis: Codierung
  (optimal, wenn alle Häufigkeiten Kehrwerte von Zweierpotenzen sind)
- Wiederholte Anwendung dieses Schritts auf die Häufigkeitstabelle:
  – Ersetze die beiden Einträge niedrigster Häufigkeit durch einen Codebaum mit zwei Ästen „0“ und „L“ und trage die Summe der Häufigkeiten als Häufigkeit dafür ein.

| Zeichen | A | B | C | D |
|---|---|---|---|---|
| Häufigkeit | 0.5 | 0.25 | 0.125 | 0.125 |

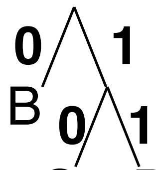
| Zeichen | A | B | 0 1<br>C D |
|---|---|---|---|
| Häufigkeit | 0.5 | 0.25 | 0.25 |

David Huffman 1951

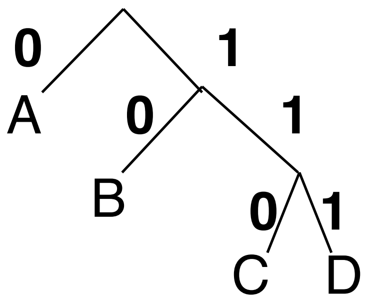
# Huffman-Codierung (2)




| Zeichen | A | B | C D |
|---|---|---|---|
| Häufigkeit | 0.5 | 0.25 | 0.25 |




| Zeichen | A | C D |
|---|---|---|
| Häufigkeit | 0.5 | 0.5 |




Resultierender Codebaum

# Huffman-Codierung (3)

- Eine Nachricht, die sich an die gegebene Häufigkeitsverteilung hält:

  ababacadaabacdba (Länge = 16 Zeichen)

- Codierung mit festen Wortlängen
  (z.B. a = 00, b = 01, c = 10, d = 11)

  Länge 32 bit

- Huffman-Codierung
  ( a = 0, b = 10, c = 110, d = 111)

  0100100110011100100110111100

  Länge 28 bit (d.h. ca. 12.5% Reduktion)

# Experiment: Huffman-Kompression von Bildern

- Grautonbild, 256 x 256 Pixel, 8 bit (d.h. 256 Graustufen)

- Unkomprimiert: 65.536 Bytes
- Mit Huffman kodiert: 40.543 Bytes ca. 38% Reduktion
- Einfacher "Zusatztrick":
  - *Differenz* zwischen benachbarten Pixeln speichern und Huffman dann anwenden

    33.880 Bytes ca. 51% Reduktion
  - Keine universelle Kompression mehr, sondern speziell für Pixelbilder
  - Solche "semantischen Kodierungen" siehe später!