DHBW Lörrach
Duale Hochschule
Baden-Württemberg

Algorithmen und Komplexität
TIF 21 A/B
Dr. Bruno Becker

11. Datenkompression

Datenkompression

- **Redundanz und Informationsgehalt**
- Verlustfreie Datenkompression
- Huffman-Code
- Informationsreduzierende Codes

Störungen bei Datenübertragung

- Ganze Zahlen digital in Binärdarstellung.
- Wird bei Übertragung nur 1 Bit falsch gesetzt, geht die Zahl verloren und der Fehler ist nicht erkennbar

- Beispiel: 49 = 00110001

  00111001 = 57

  10110001 = 177

Störungen bei Datenübertragung (2)

- In anderen Kontexten ist Verlust von Teildaten nicht so kritisch für die Informationserkennung - Bildübertragung

Störungen bei Datenübertragung (3)

- Text aus Lückentext rekonstruieren:

H_ff_ntl_ch w_rd d__ Kl__s_r n_cht z_ schw_r !

Wer _eit_t so _pät _urc_ Nac_t un_ Win_?
Es i_t de_ Vat_r mi_ sei_em K_nd;
Er h_t de_ Kna_en w_hl i_ dem _rm,
Er f_sst _hn s_che_, er h_lt i_n wa_m.

Redundanz und Informationsgehalt

Eine Nachricht enthält **Redundanz,** wenn sie mehr Zeichen enthält, als für die Darstellung der enthaltenen Information nötig wären.

Redundante Zeichen können ohne Informationsverlust weggelassen werden.

Fehlererkennung: Zusätzliche Prüfziffern, die vom Input abhängen

- Beispiel IBAN: DE*25* 8503 0300 4000 438549
  DE*52* 8503 0300 4000 43854*8*

Fehlerkorrigierende Codes:

- Einfaches Beispiel (3,1)-Hamming-Code:
  - Jedes Bit wird als 3-Bitfolge codiert: Zulässig sind nur *000, 111*
  - Bitfolge *001, 010, 100* → *000; 101, 011, 110* → *111*
- Hamming-Code (nicht in dieser Vorlesung)

Datenkompression

- Redundanz und Informationsgehalt
- **Verlustfreie Datenkompression**
- Huffman-Code
- Informationsreduzierende Codes

Verlustfreie Datenkompression

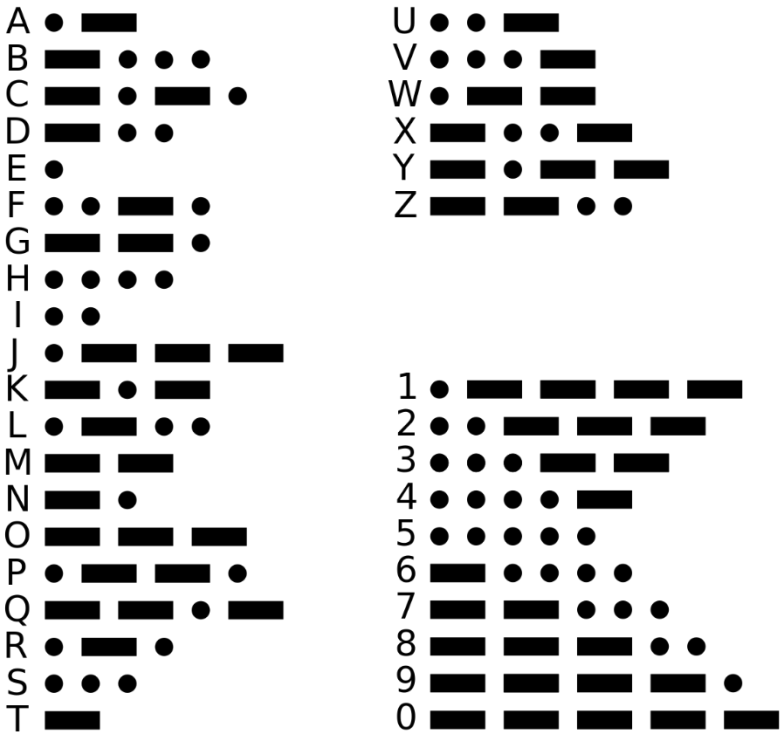
- **Ziel:** Inputdaten **mit möglichst wenig Bits** (*redundanzarm)* binär kodieren
- **Verlustfrei:** Dekodierung stellt Inputdaten exakt wieder her

Motivation: *Häufige Zeichen* werden durch kurze,
seltene Zeichen durch lange Codewörter codiert.

Historisches Vorbild: Morsecode

International Morse Code

1. The length of a dot is one unit.
2. A dash is three units.
3. The space between parts of the same letter is one unit.
4. The space between letters is three units.
5. The space between words is seven units.



Datenkompression

- Redundanz und Informationsgehalt
- Verlustfreie Datenkompression
- **Huffman-Code**
- LZW-Algorithmus
- Informationsreduzierende Codes

Huffman-Code

David A. Huffman (1951): Student, Aufgabe im Rahmen einer Seminararbeit

- **Ziel:** Konstruktion von Codes mit möglichst wenig Redundanz
- **Input:** Menge der Quellsymbole mit Wahrscheinlichkeiten
- **Output:** Ein binärer *Codebaum* zur (De-)Codierung aller Zeichen

Idee:

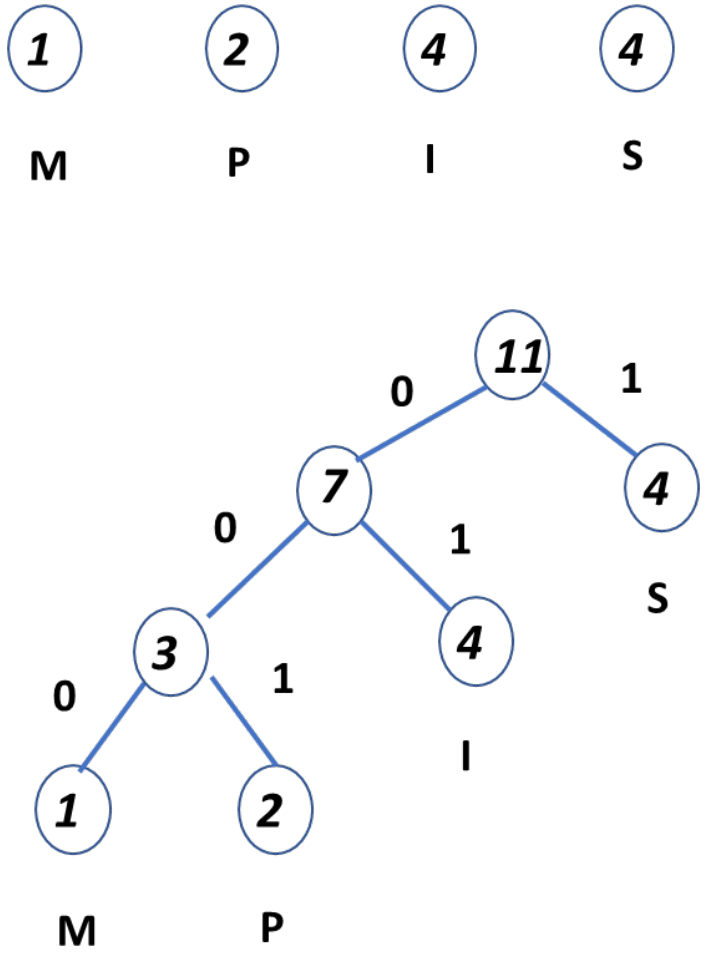
- Baue Baum *bottom-up* auf: Starte mit den Symbolen als Blätter
- Summiere Wahrscheinlichkeiten für Teilbäume
- Vereinige die Teilbäume mit der geringsten Wahrscheinlichkeit

Huffman-Code Beispiel

Text: Mississippi

Zeichen	M	P	I	S
Häufigkeit	1	2	4	4
Code	000	001	01	1

Mississippi →
000 01 1 1 01 1 1 01 001 001 01
00001110111010010010 1
21 Bit statt 88 Bit (ASCII)



Huffman-Code Algorithmus

Input: *X* Quellsymbolalphabet, *p(x) relative Häufigkeit* des Symbols *x,*
$\{0,1\}$ – Codealphabet C -> Binärbaum
(es geht auch allgemeines Codealphabet, dann Baum mit Ordnung $|C|$

Aufbau Baum:

1. Ermittle für jedes Quellsymbol *relative Häufigkeit*
2. Erstelle für jedes Quellsymbol einen einzelnen Knoten und notiere Häufigkeit
3. Wiederhole, bis nur noch 1 Baum vorhanden:
   i. Wähle die beiden Teilbäume mit der geringsten Häufigkeit in der Wurzel, bei mehreren die beiden Teilbäume mit geringster Tiefe
   ii. Fasse die beiden Bäume zu einem Baum zusammen und notiere Häufigkeit an Wurzel

Konstruktion Codebuch

1. Ermittle für jedes Blatt im Baum Codezeichen: Linke Kante 0, Rechte Kante 1
2. Suchpfad von der Wurzel zum Blatt ergibt eindeutiges Codezeichen

Huffman-Code Eigenschaften

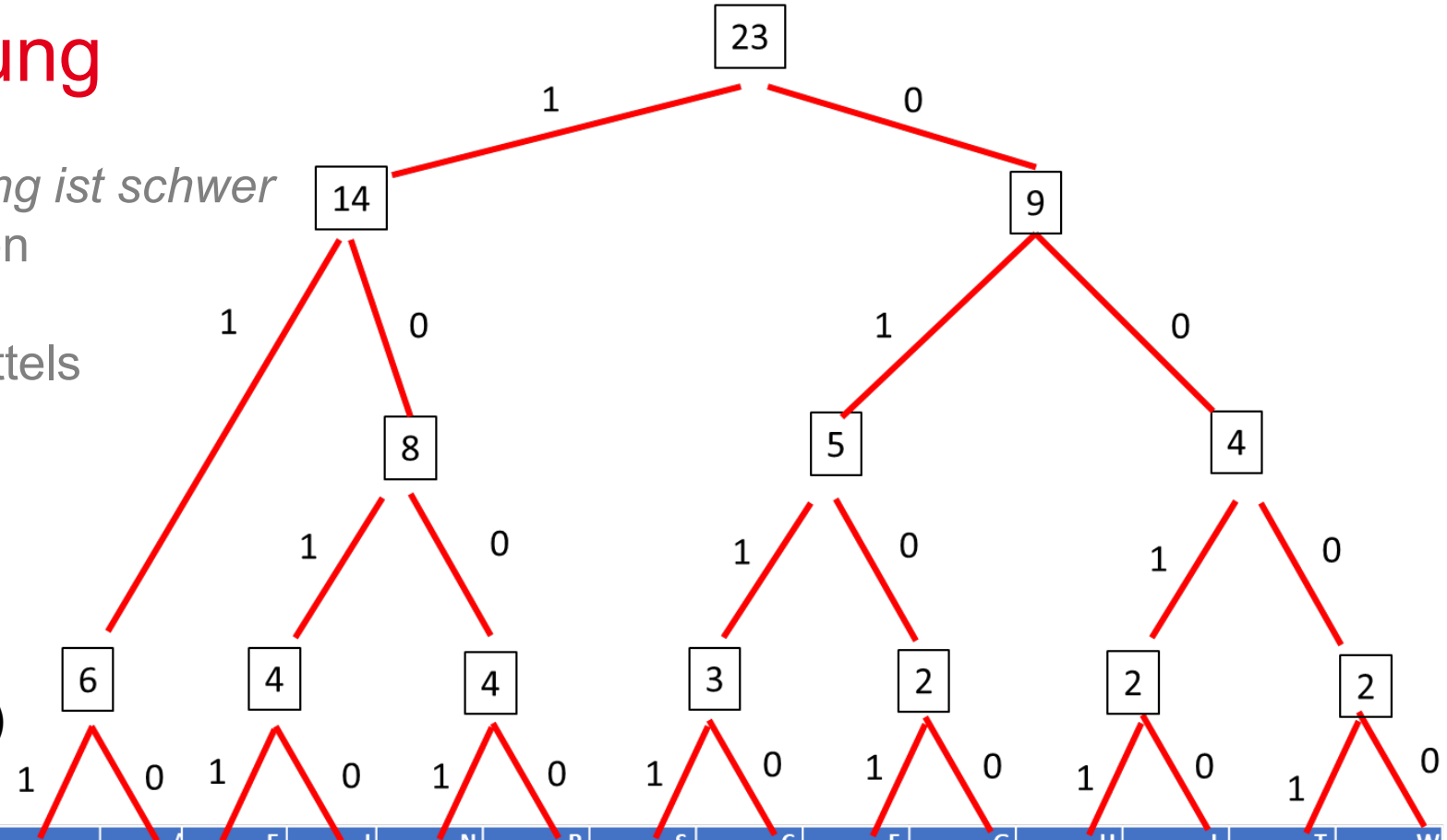
- Wahrscheinlichstes Zeichen erhält kürzesten Code
- D.h. Text entsprechend Häufigkeitsverteilung nicht kürzer codierbar
- Effiziente Codierung und Decodierung mit Codebaum
- Codebaum muss zum Empfänger übertragen werden
- Jeder Code ist ***präfixfrei***, d.h. kein Codewort ist Anfang eines anderen Codewortes (Eigenschaft heißt auch *Fano-Bedingung*)
- **Aufwand:** Welche Datenstruktur?
  → Übungsblatt

  ➔ Gesamt-Aufwand O(N log N)

Huffman-Code Übung

Gegeben sei der Text: *Aller Anfang ist schwer*

- Ermitteln Sie die Häufigkeiten (inkl. Leerzeichen)
- Erstellen Sie einen Code mittels des Huffman-Algorithmus
- Wie lange ist der Code für den Text?



6*3 Zeichen (für A und „ „)
alle anderen Zeichen Länge 4 (17 Zeichen)

➔ Länge 18 + 68 = 86 Zeichen

Zeichen	„ „	A	E	L	N	R	S	C	F	G	H	I	T	W
Häufigk.	3	3	2	2	2	2	2	1	1	1	1	1	1	1
Code	111	110	1011	1010	1001	1000	0111	0110	0101	0100	0011	0010	0001	0000

Huffman-Code Bedeutung und Anwendungen

Man kann zeigen:

- Der Huffman-Code ist das *bestmögliche Verfahren* zur Binärcodierung einzelner, statistisch unabhängiger Zeichen
- Konstruktion *nicht eindeutig,* aber alle Huffman-Codes gleich gut
- Falls alle Wahrscheinlichkeiten Zweierpotenzen sind, ist der Code sogar *redundanzfrei*

Anwendungen

- Ist Bestandteil vieler Verfahren zur Text-, Bild-, und Audiokompression

Weiteres Verfahren zur verlustfreien Datenkompression

LZW-Algorithmus (Lempel, Ziv, Welch 1984)

- Tabellengesteuertes Verfahren zur verlustfreien Datenkompression
- Nur Tabellen-Indizes werden als Binärzahlen fester Länge übertragen
- *Lernender Algorithmus:* Wörterbuch wird adaptiv aufgebaut
- Wird für Bildformate *GIF* und *TIFF* eingesetzt
- Umstrittenes Softwarepatent- seit 2004 ausgelaufen

Datenkompression

- Redundanz und Informationsgehalt
- Verlustfreie Datenkompression
- Huffman-Code
- **Informationsreduzierende Codes**

Informationsreduzierende Codes

Bild-, Video- und Audiodateien enthalten viel Redundanz und sind auch noch bei „gröberer“ Reduktion noch wiedererkennbar

Informationsreduzierende Codes verzichten zugunsten einer besseren Komprimierung auf eine exakte Wiederherstellbarkeit des Originals, akzeptieren also einen (begrenzten) Qualitätsverlust

- Einsatz für Bild-, Video- und Audiodateien
- Kompromiss zwischen Qualität und Komprimierung, ggfs. auch abhängig von Inhalten und Wiedergabegeräten
- Viele Verfahren sind *parametrierbar,* z.B. mit einer gewünschten *Bitrate x kbit/s* oder einer *Qualitätsstufe*

Beispiele Informationsreduzierende Codes

- **MP3** für Audiodaten
  - Entwickelt am Fraunhofer-Institut IIS in Erlangen (ab 1982)
  - Idee: Ausnutzen der Charakteristik des menschlichen Hörvermögens z.B. unpräzise wahrnehmbare Frequenzbereiche
  - Vewendet Huffman-Code als Teilbaustein
- **JPEG** (*Joint Photographic Experts Group*) für Bilddaten
  - Einsatz: Digitalkameras, Internet
  - Idee: Abschneiden hoher Frequenzen im Frequenzspektrum
  - Gut geeignet für Bilder mit kontinuierlichen Übergängen
- **MPEG** (*Moving Pictures Experts Group*) für Videodaten
  - Einsatz: Videokameras, DVD, Internet
  - Idee: Speicherung von Differenzen aufeinanderfolgender Frames
  - Umfangreich parametrierbar

Zusammenfassung

- Redundanz und Informationsgehalt
  - Fehlererkennung (z.B. IBAN) und Fehlerkorrektur
- Verlustfreie Datenkompression
  - Decodierung stellt Information komplett wieder her (z.B. Morse-Alphabet)
- Huffman-Code
  - Bestmögliches Verfahren zur Codierung einzelner, statistisch unabhängiger Zeichen – Minimale Codierungslänge entsprechend den Wahrscheinlichkeiten
- Informationsreduzierende Codes
  - Reduktion von (redundanten) Informationen zur besseren Komprimierung
  - MP3, JPEG, MPEG,…