



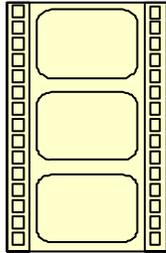
Mathematik für Information und Kommunikation

Am Beispiel des Huffman-
Algorithmus

Thomas Borys und (Christian Urff)

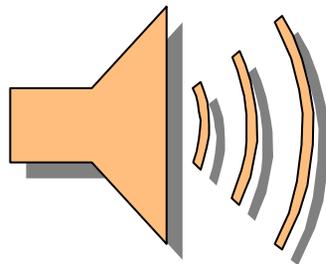
Huffman im Alltag

MPEG



JPEG

Telefax



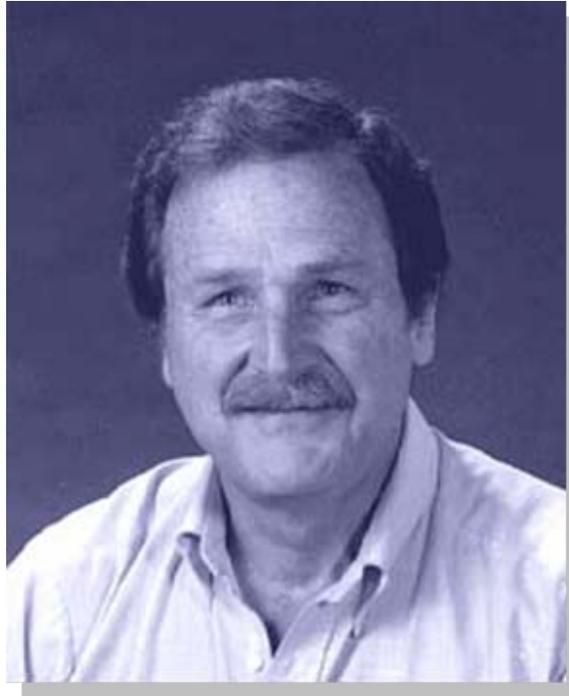
MP3



ZIP

...

David Huffman



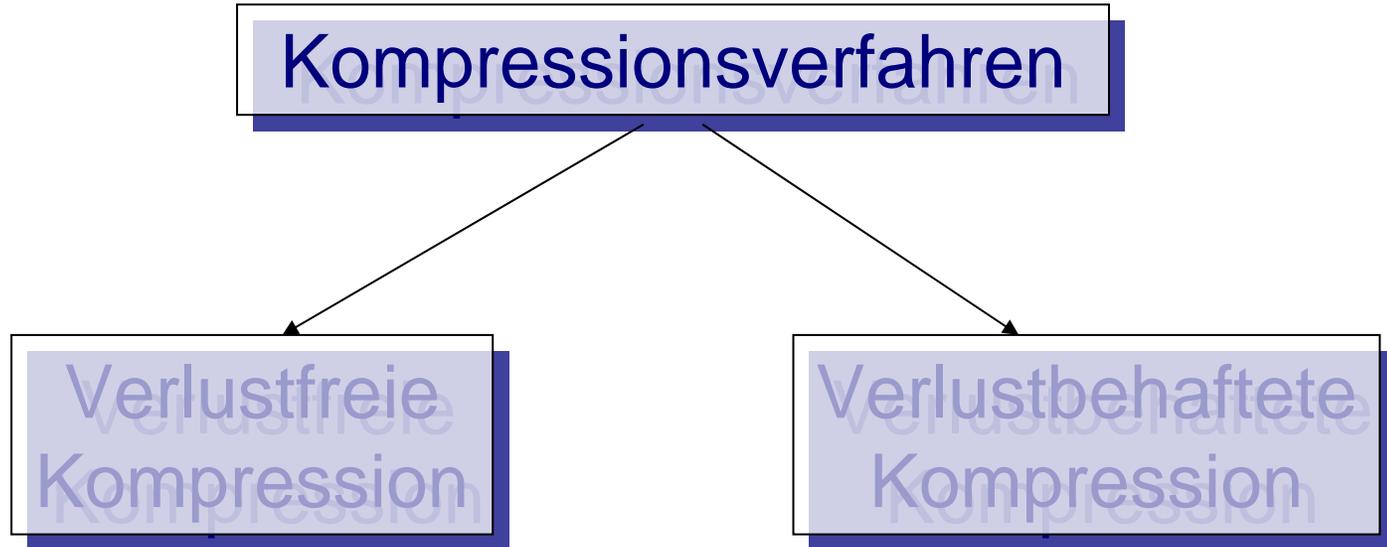
David Huffman
[1925-1999]

www.soe.ucsc.edu/people/faculty/huffman.html

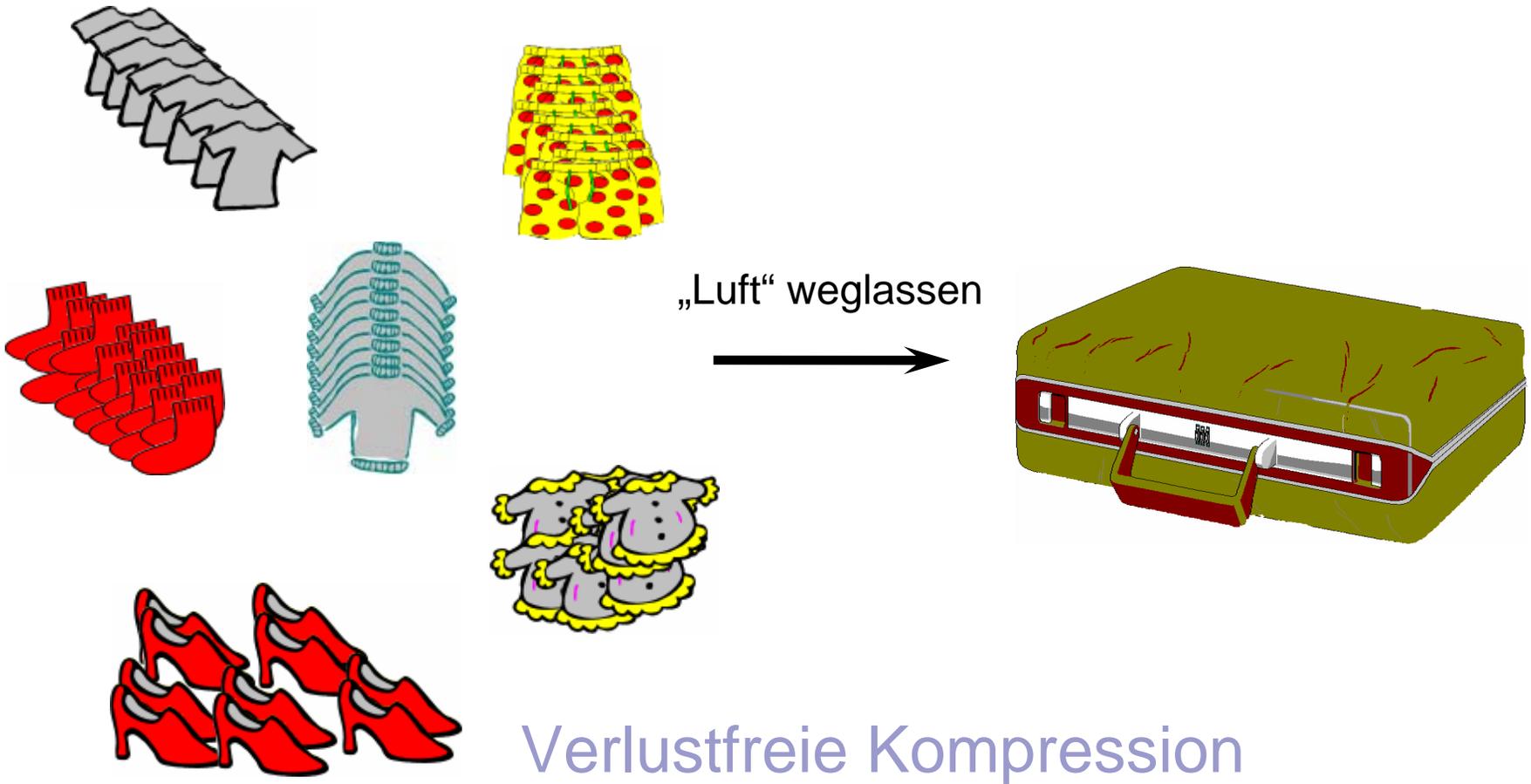
Gliederung

1. Grundidee des Huffman-Algorithmus
2. Der Huffman-Algorithmus exemplarisch an einem Beispiel
3. Eigenschaften der Huffman-Codes
4. Anwendung des Huffman beim Telefax

Grundidee des Huffman-Algorithmus



Grundidee des Huffman-Algorithmus



Grundidee des Huffman-Algorithmus



Unwichtiges weglassen



Zusammenfassung

skda fjsakd lksdaj lskdaf lsdakj ölsadk jflaskf
sadtik lkdjff lksdajf ölsakjf ölsakf lksadjtllksajfdölsakdjfl sdlkfj sdaLökj flsdak
fjsaldök jlsadk jfaölskjdl flsadkjflslökjfl
aölskjfldejlökajdsfleaalktjaejllkasjfoiaesejlkf saeliffjask jflsalifdlksdajf ölskaj
ldksaj ldsakjflsalfllkds lksdjafiejlksdjafiesf lksadjaeslksadjsaejflkdsjfl
lksdjflsaejfltkd isajfelkjldskjfasilff esalkjflisadfelkdjsafi eölakfjesaoi

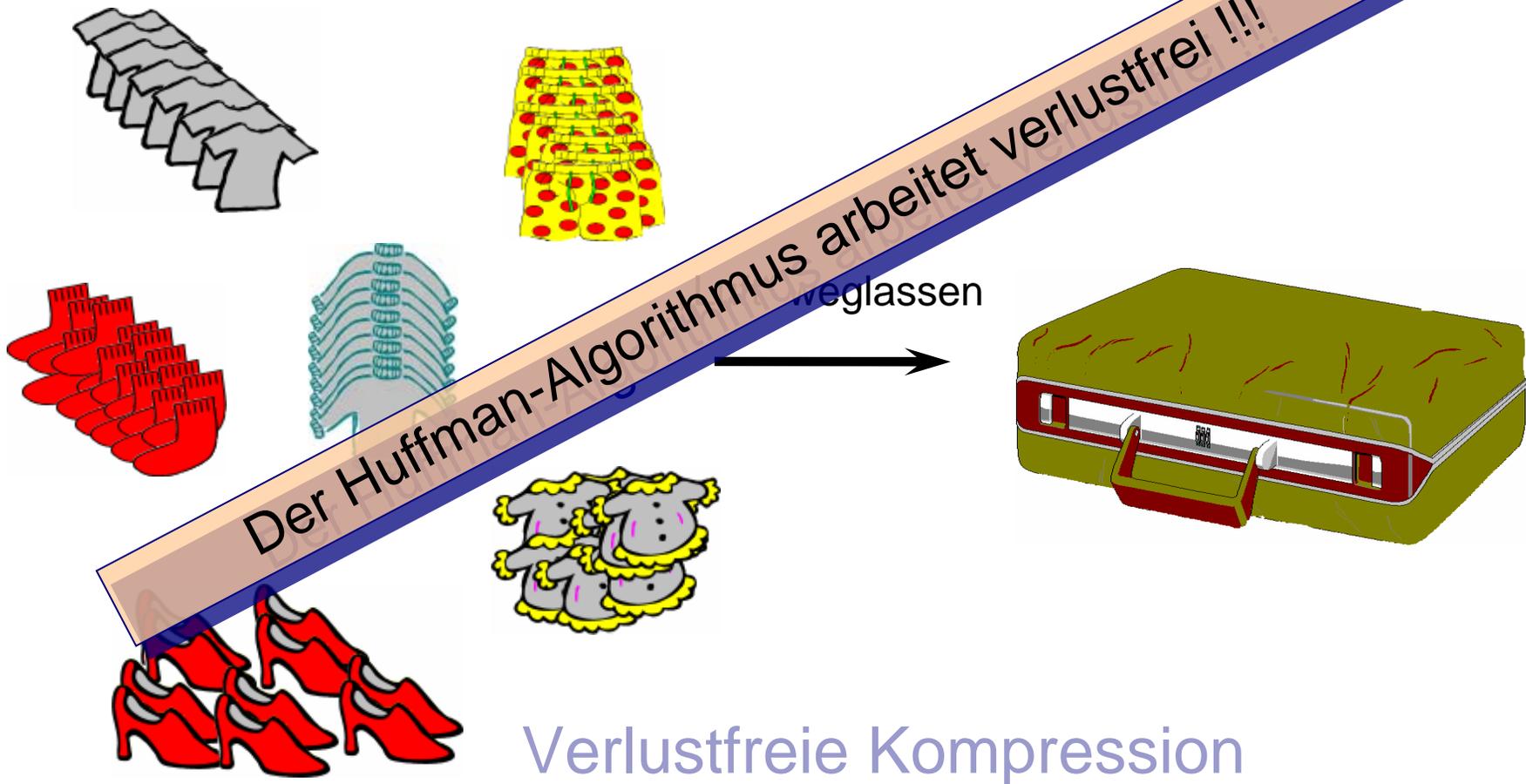
ölsakjflsadjfoiasjelkjesalfjatsalfj sdlakf jldsakjffsaejllksadjfiesä lksdj lidjsfa
lkeaj iaiejlkdf ielkf ösaldkflesajtkd isajfelkjldskjfasilff esalkjflisadfelkdjsafi
eölakfjesaoi ölsakjflsadjfoiasjelkjesalfjatsalfj sdlakf jldsakjffsaejllksadjfiesä

lksdj lidjsfa lkeaj iaiejlkdf ielkf ösaldkflesajtkd isajfelkjldskjfasilff
esalkjflisadfelkdjsafi eölakfjesaoi ölsakjflsadjfoiasjelkjesalfjatsalfj sdlakf
jldsakjffsaejllksadjfiesä lksdj lidjsfa lkeaj iaiejlkdf ielkf ösaldkflesajtkd
isajfelkjldskjfasilff esalkjflisadfelkdjsafi eölakfjesaoi
ölsakjflsadjfoiasjelkjesalfjatsalfj sdlakf jldsakjffsaejllksadjfiesä lksdj lidjsfa
lkeaj iaiejlkdf ielkf ösaldkflesajtkd isajfelkjldskjfasilff esalkjflisadfelkdjsafi
eölakfjesaoi ölsakjflsadjfoiasjelkjesalfjatsalfj sdlakf jldsakjffsaejllksadjfiesä lksdj lidjsfa
lkeaj iaiejlkdf ielkf ösaldkflesajtkd isajfelkjldskjfasilff esalkjflisadfelkdjsafi

jldsakjffsaejllksadjfiesä lksdj lidjsfa lkeaj iaiejlkdf ielkf ösaldkflesajtkd
isajfelkjldskjfasilff esalkjflisadfelkdjsafi eölakfjesaoi
ölsakjflsadjfoiasjelkjesalfjatsalfj sdlakf jldsakjffsaejllksadjfiesä lksdj lidjsfa
lkeaj iaiejlkdf ielkf ösaldkflesajtkd isajfelkjldskjfasilff esalkjflisadfelkdjsafi
eölakfjesaoi ölsakjflsadjfoiasjelkjesalfjatsalfj sdlakf jldsakjffsaejllksadjfiesä
lksdj lidjsfa lkeaj iaiejlkdf ielkf ösaldkfj
esajtkd isajfelkjldskjfasilff esalkjflisadfelkdjsafi eölakfjesaoi
ölsakjflsadjfoiasjelkjesalfjatsalfj sdlakf jldsakjffsaejllksadjfiesä lksdj lidjsfa
lkeaj iaiejlkdf ielkf ösaldkflesajtkd isajfelkjldskjfasilff esalkjflisadfelkdjsafi
eölakfjesaoi ölsakjflsadjfoiasjelkjesalfjatsalfj sdlakf jldsakjffsaejllksadjfiesä
lksdj lidjsfa lkeaj iaiejlkdf ielkf ösaldkfj

Verlustbehaftete Kompression

Grundidee des Huffman-Algorithmus



Grundidee des Huffman-Algorithmus

ABRAKADABRA

ASCII

A=01000001

B=01000010

...

88 Bit



Idee:

häufig vorkommende Zeichen bekommen
einen kürzeren Code, selten vorkommende
Zeichen ein längeres Codewort

z.B. A=0 B=11

Grundidee des Huffman-Algorithmus

Häufig benötigte
Bücher stellt man in
greifbare Nähe
(Augenhöhe)

Selten benötigte
Bücher verstaut man
weiter oben oder
unten



Grundidee des Huffman-Algorithmus

Morse-Code

A	· -	B	- · · ·	C	- · - ·
E	·	F	· · - ·	G	- - ·
I	· ·	J	· - - -	K	- · - -
M	- -	N	- ·	O	- - -
Q	- - - -	R	· - ·	S	· · ·
U	· · -	V	· · · -	W	· - -
Y	- · - -	Z	- · · ·		

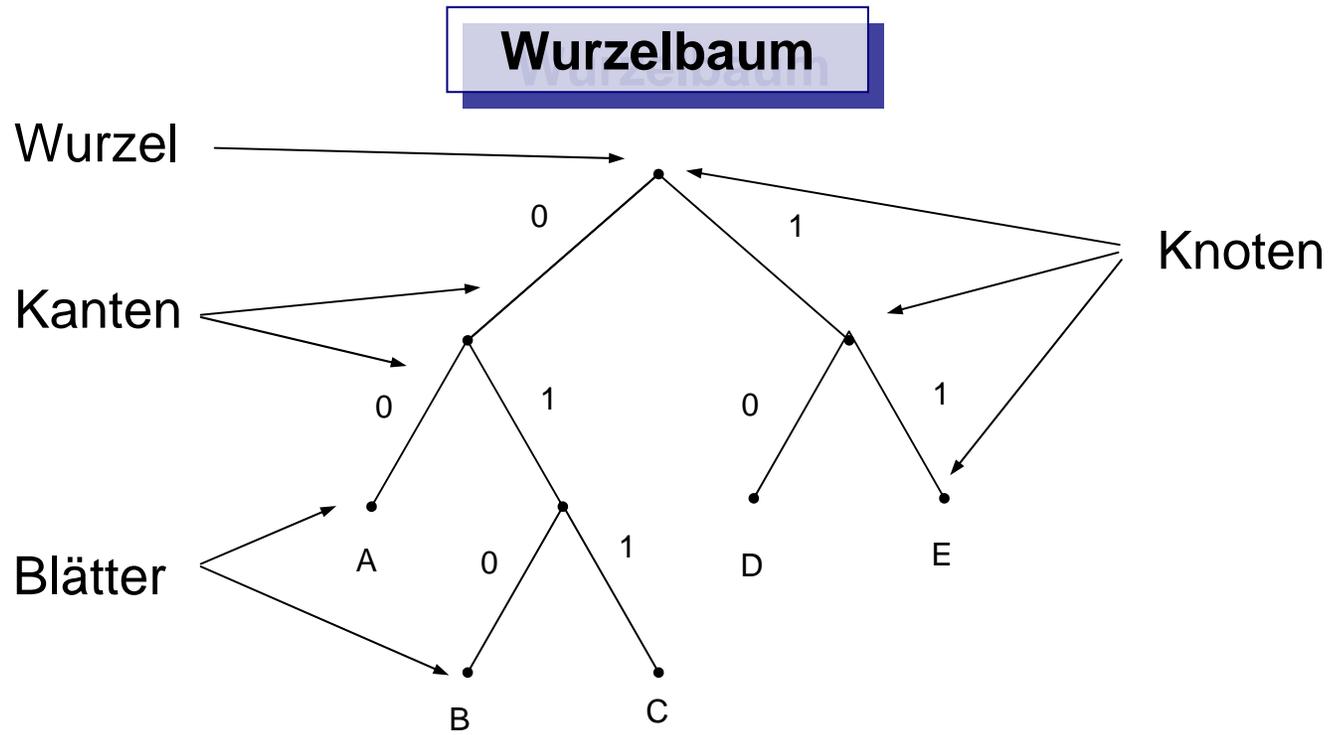


Samuel Morse
[1791-1872]

www.morsehistoricsite.org

Der Huffman-Algorithmus exemplarisch an einem Beispiel

Ziel: Jedem im Text vorkommenden Zeichen wird ein Binärcode zugewiesen!



Der Huffman-Algorithmus exemplarisch an einem Beispiel

Text: ABRAKADABRA

Häufigkeitsanalyse:

Buchstaben	A	B	R	K	D
Häufigkeit	5				

Der Huffman-Algorithmus exemplarisch an einem Beispiel

Text: ABRAKADABRA

Häufigkeitsanalyse:

Buchstaben	A	B	R	K	D
Häufigkeit	5	2			

Der Huffman-Algorithmus exemplarisch an einem Beispiel

Text: ABRAKADABRA

Häufigkeitsanalyse:

Buchstaben	A	B	R	K	D
Häufigkeit	5	2	2		

Der Huffman-Algorithmus exemplarisch an einem Beispiel

Text: ABRAKADABRA

Häufigkeitsanalyse:

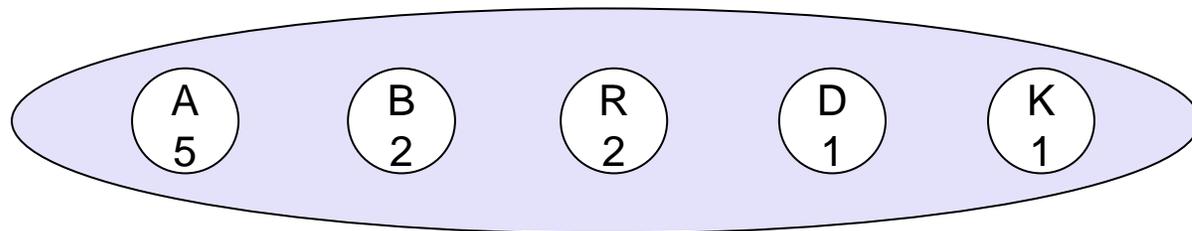
Buchstaben	A	B	R	K	D
Häufigkeit	5	2	2	1	

Der Huffman-Algorithmus exemplarisch an einem Beispiel

Text: ABRAKADABRA

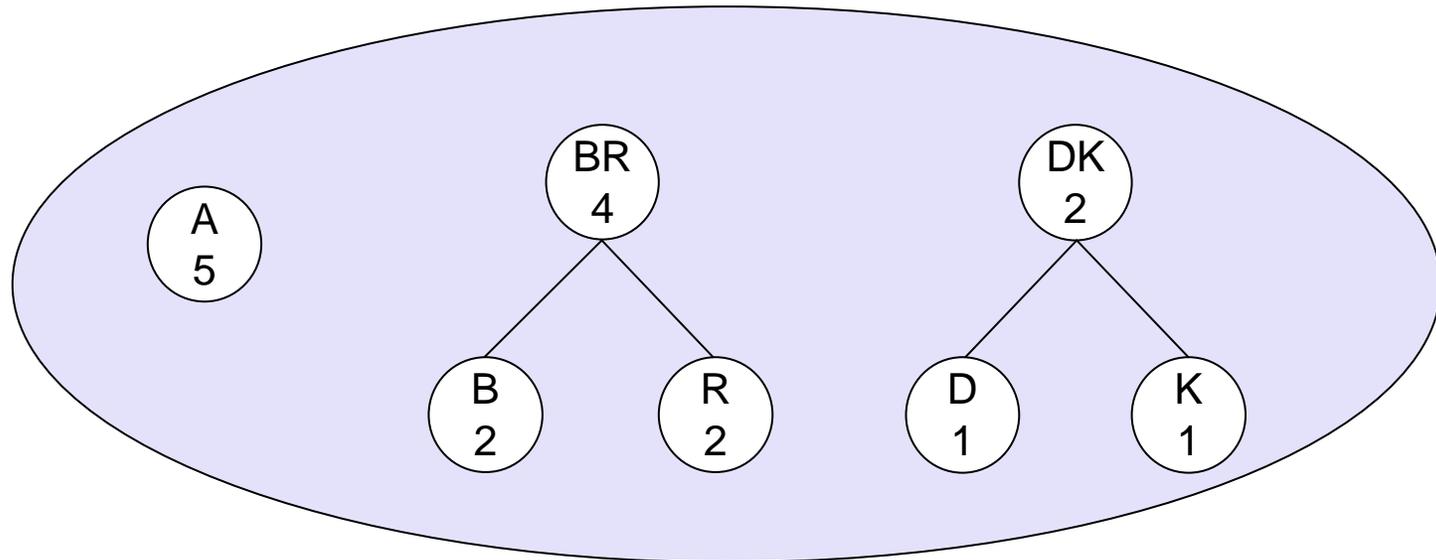
Häufigkeitsanalyse:

Buchstaben	A	B	R	K	D
Häufigkeit	5	2	2	1	1



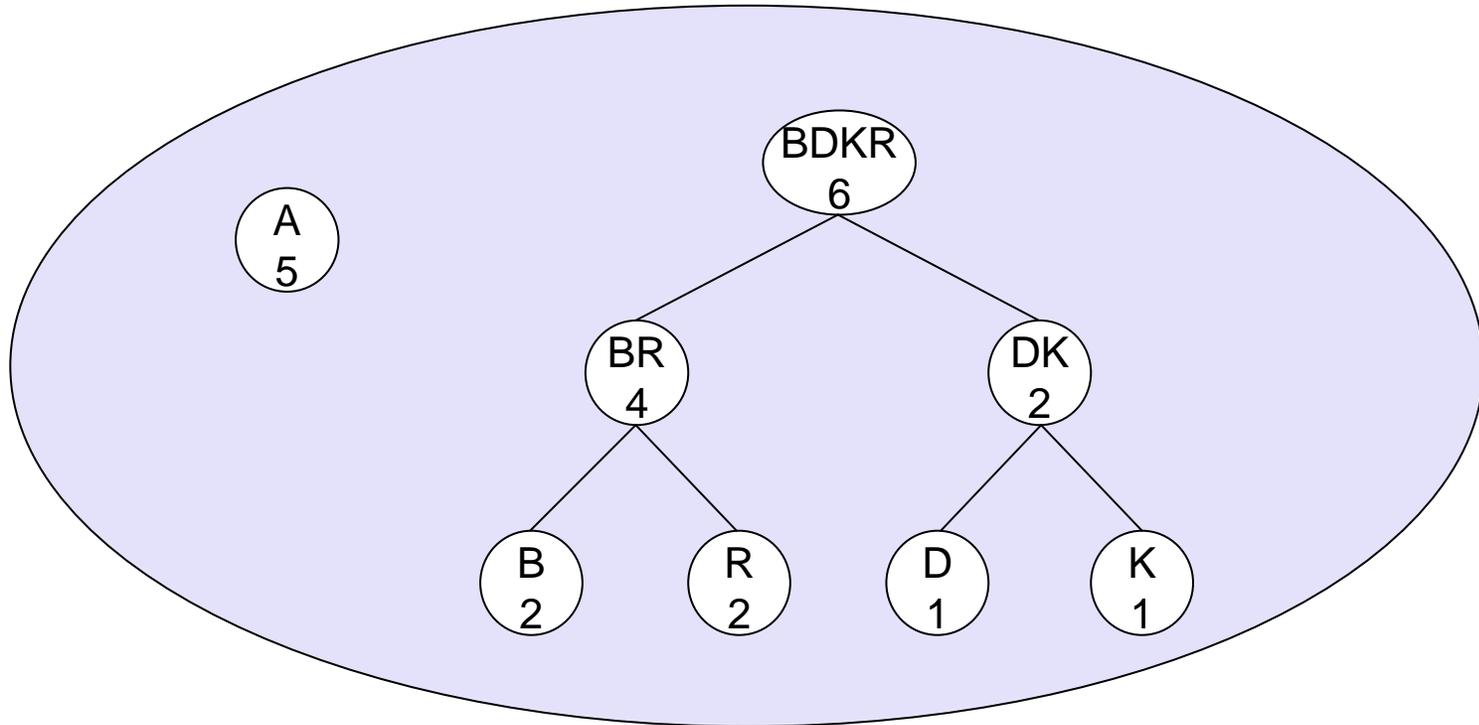
Huffman-Liste 1

Der Huffman-Algorithmus exemplarisch an einem Beispiel



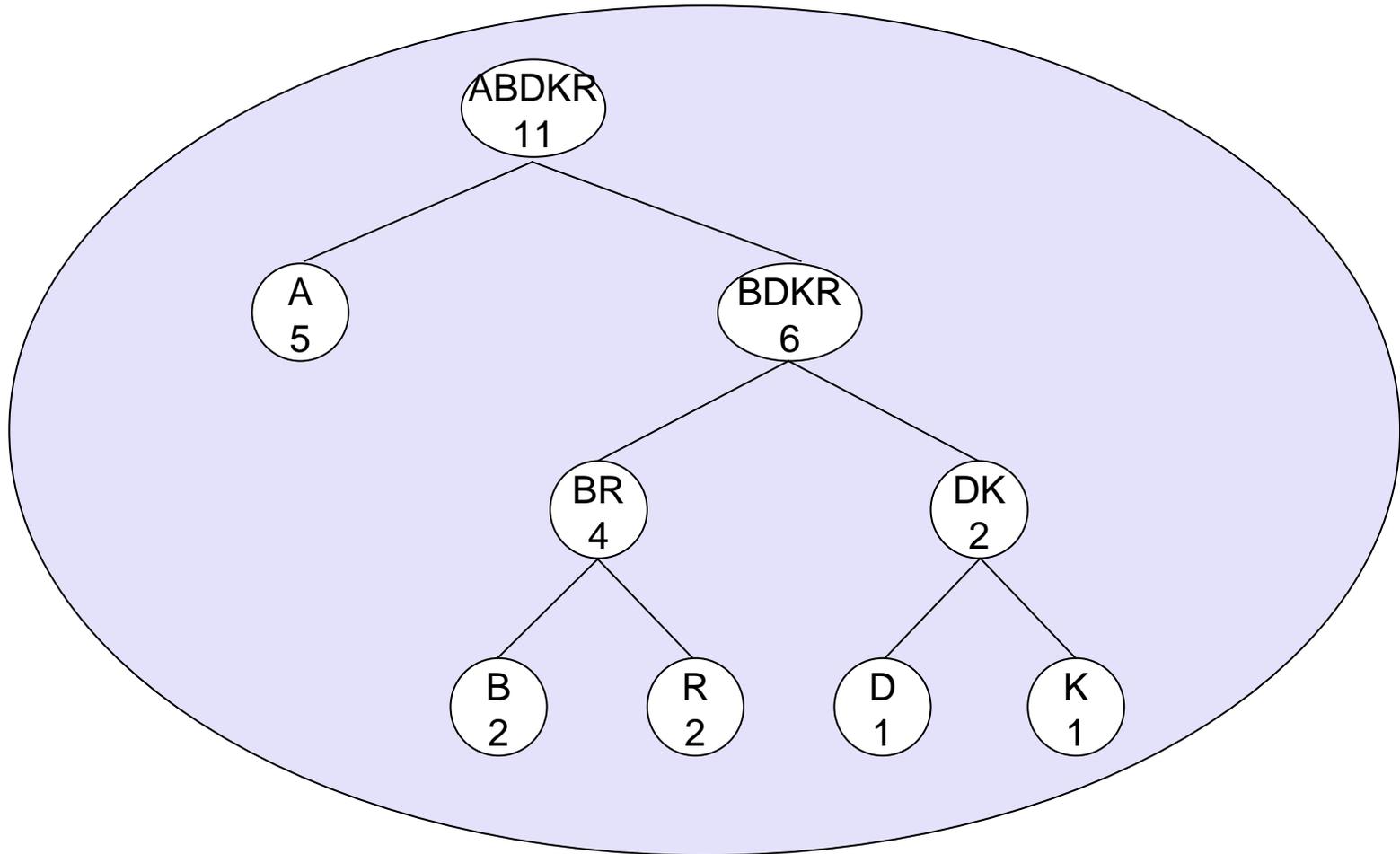
Huffman-Liste 3

Der Huffman-Algorithmus exemplarisch an einem Beispiel



Huffman-Liste 4

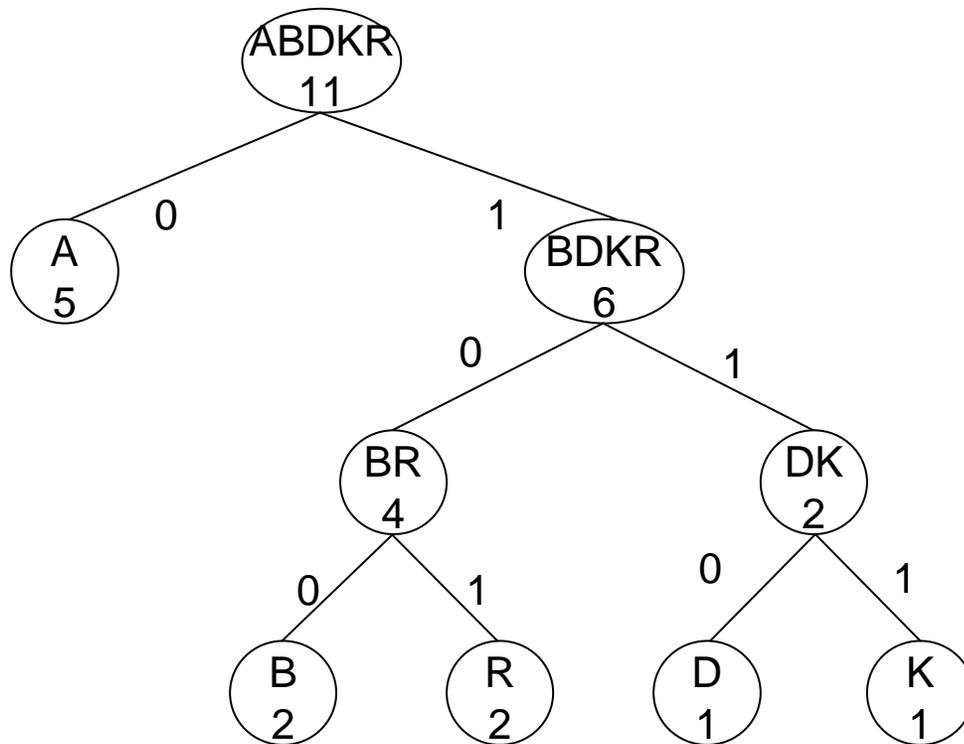
Der Huffman-Algorithmus exemplarisch an einem Beispiel



Huffman-Liste 5

Der Huffman-Algorithmus exemplarisch an einem Beispiel

Codebaum

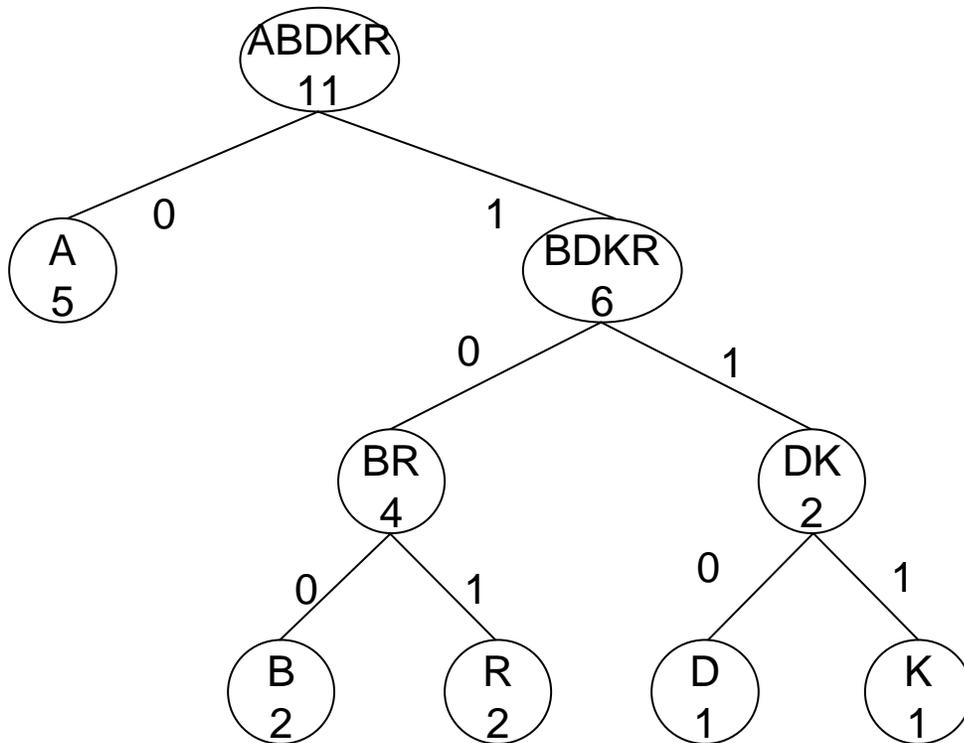


Codetabelle

Buchstaben	Binärkode
A	0

Der Huffman-Algorithmus exemplarisch an einem Beispiel

Codebaum

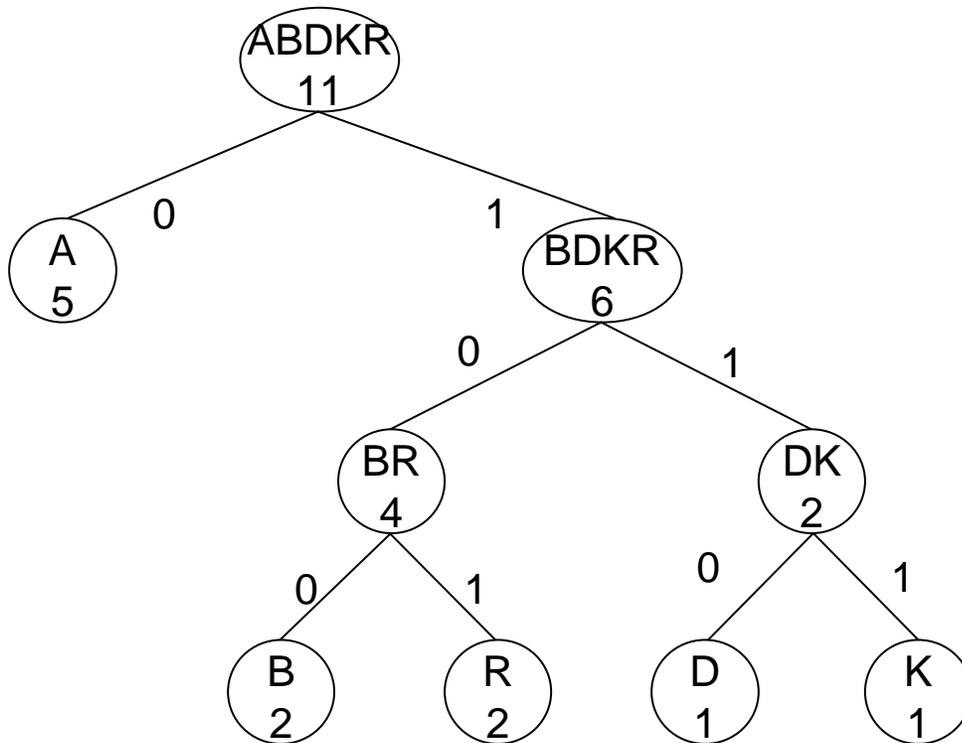


Codetabelle

Buchstaben	Binärkode
A	0
B	100

Der Huffman-Algorithmus exemplarisch an einem Beispiel

Codebaum

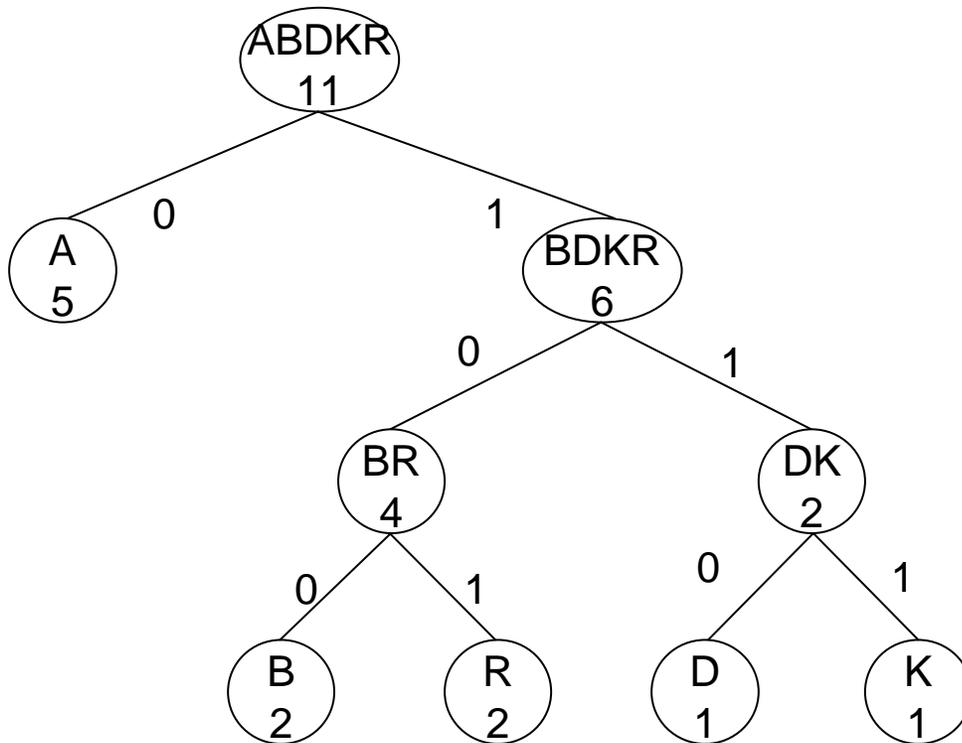


Codetabelle

Buchstaben	Binärkode
A	0
B	100
D	110

Der Huffman-Algorithmus exemplarisch an einem Beispiel

Codebaum

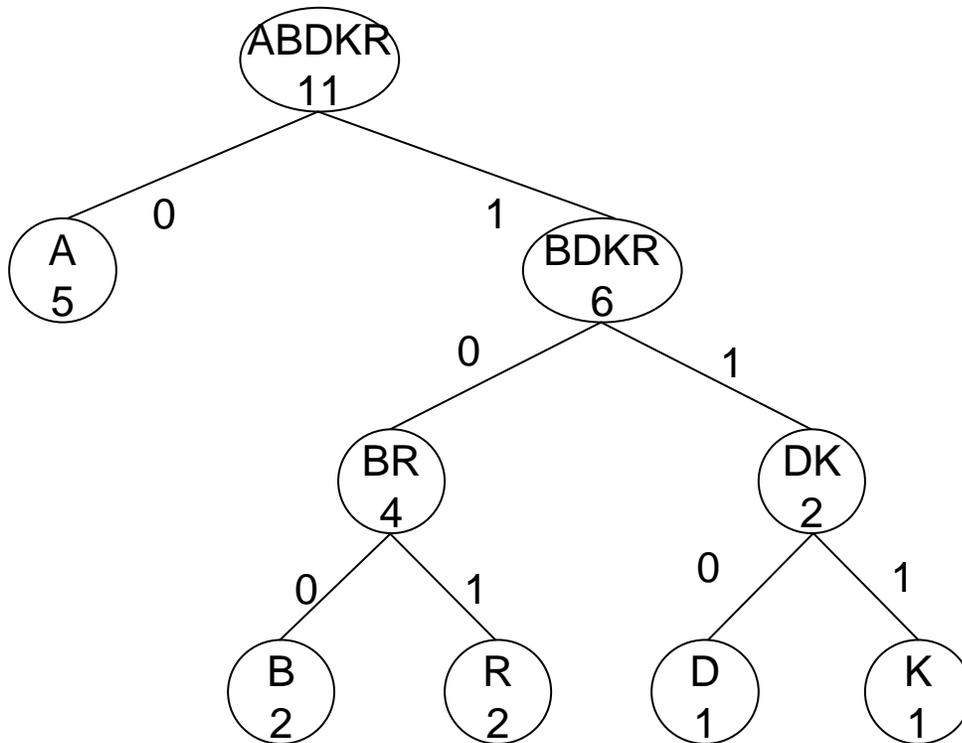


Codetabelle

Buchstaben	Binärkode
A	0
B	100
D	110
K	111
R	101

Der Huffman-Algorithmus exemplarisch an einem Beispiel

Codebaum



Codetabelle

Buchstaben	Binärkode
A	0
B	100
D	110
K	111
R	101

Der Huffman-Algorithmus exemplarisch an einem Beispiel

Codierung des Textes:

A B R A K A D A B R A
└─┘ └─┘ └─┘ └─┘ └─┘ └─┘ └─┘ └─┘ └─┘ └─┘ └─┘
0 100 101 0 111 0 110 0 100 101 0

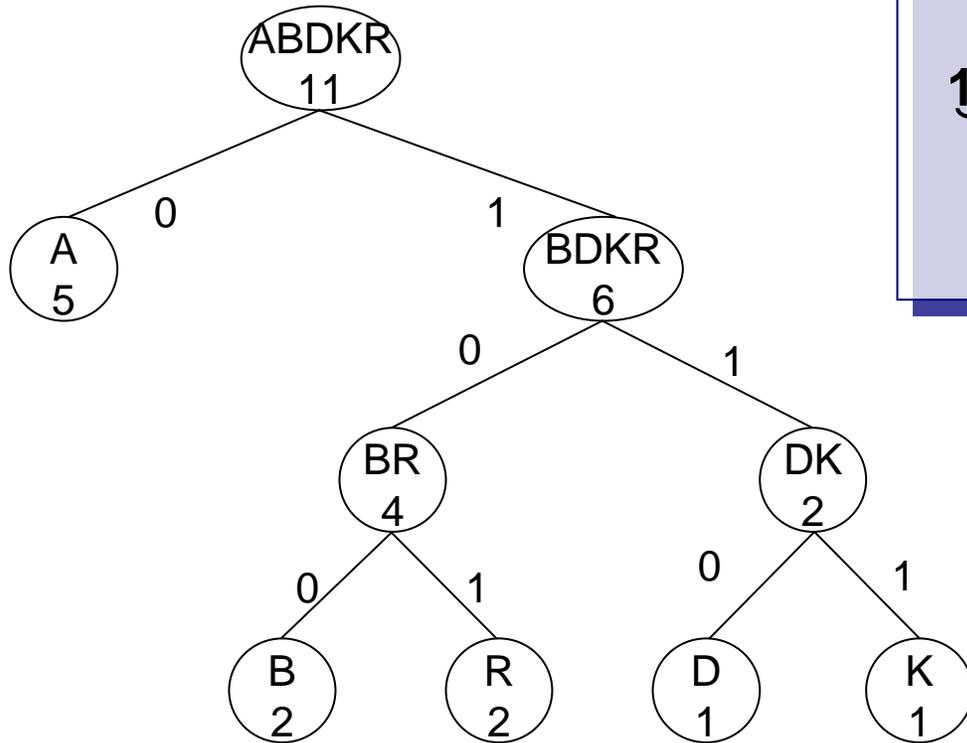
Buchstaben	Binärcode
A	0
B	100
D	110
K	111
R	101

Zusammenfassung des Algorithmus:

Eingabe: Häufigkeitstabelle
Hauptteil: 1. Erstelle die Huffman-Liste.
2. Wiederhole die Zusammenführung der beiden mit der geringsten Häufigkeit beschrifteten Bäume so lange, bis die Huffman-Liste nur noch aus einem Baum, dem Huffman-Baum, besteht.
Ausgabe: Codebaum



Eigenschaften des Huffman-Codes

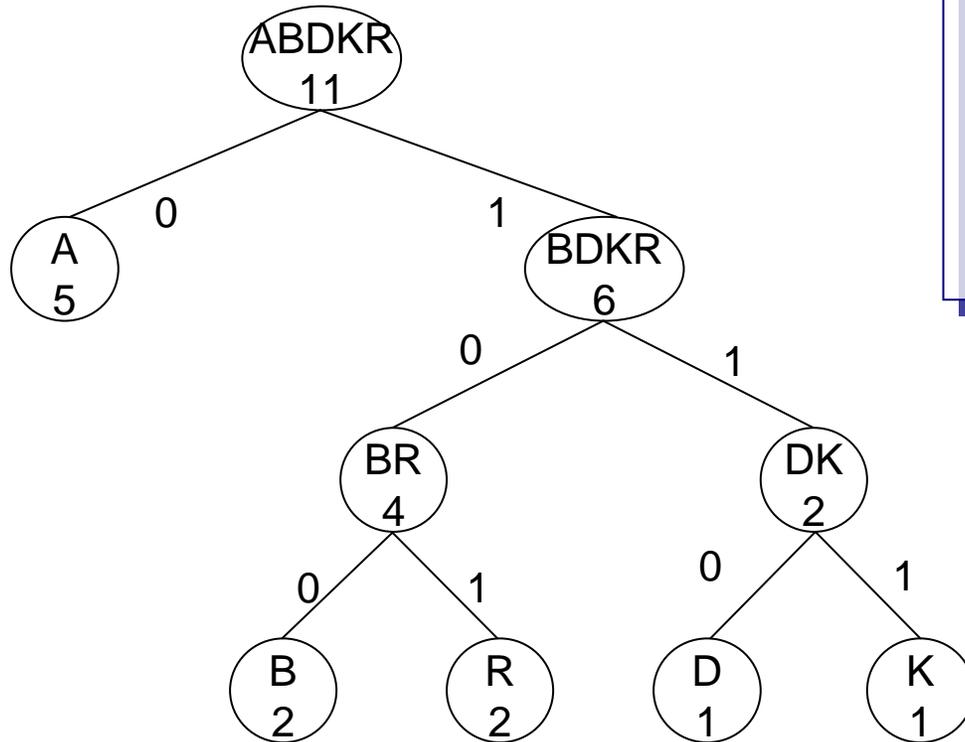


Decodieren wir den Text:

11001110101

D

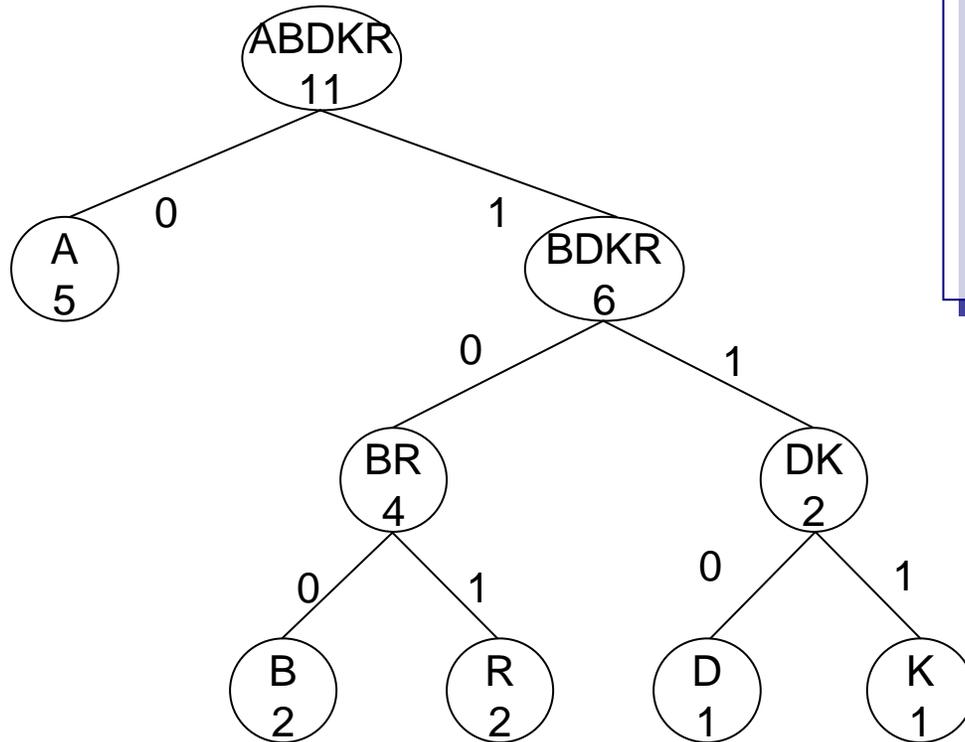
Eigenschaften des Huffman-Codes



Decodieren wir den Text:

$\underbrace{110}_D$ $\underbrace{01110101}_A$

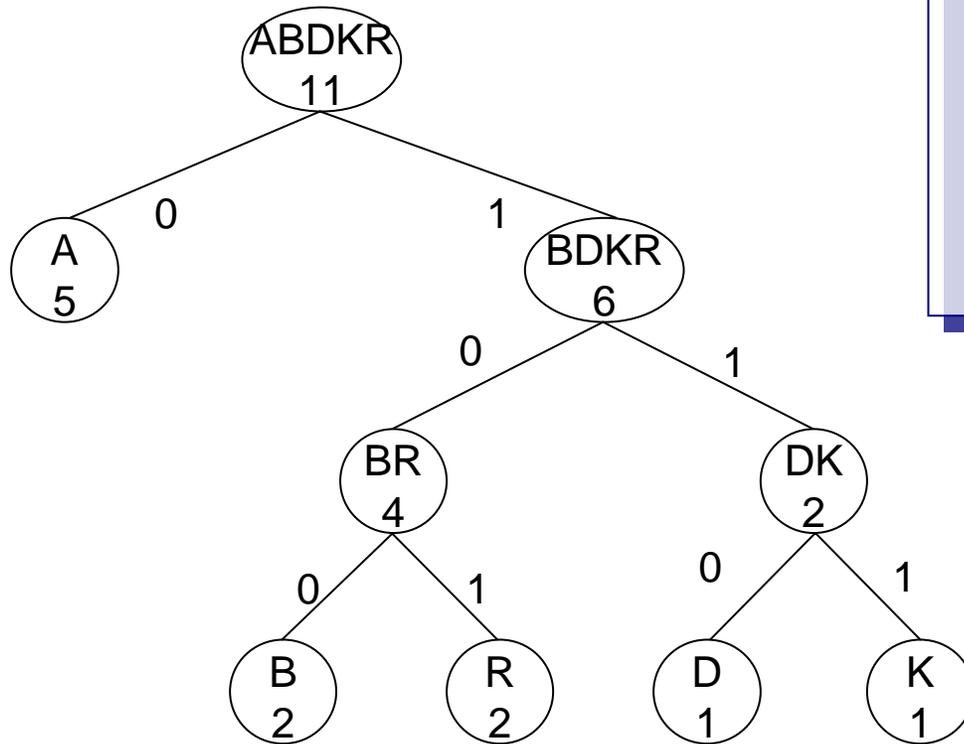
Eigenschaften des Huffman-Codes



Decodieren wir den Text:

110 **0** **1110101**
└─┬─┘ └─┬─┘ └─┬─┬─┬─┬─┬─┬─┘
D A K

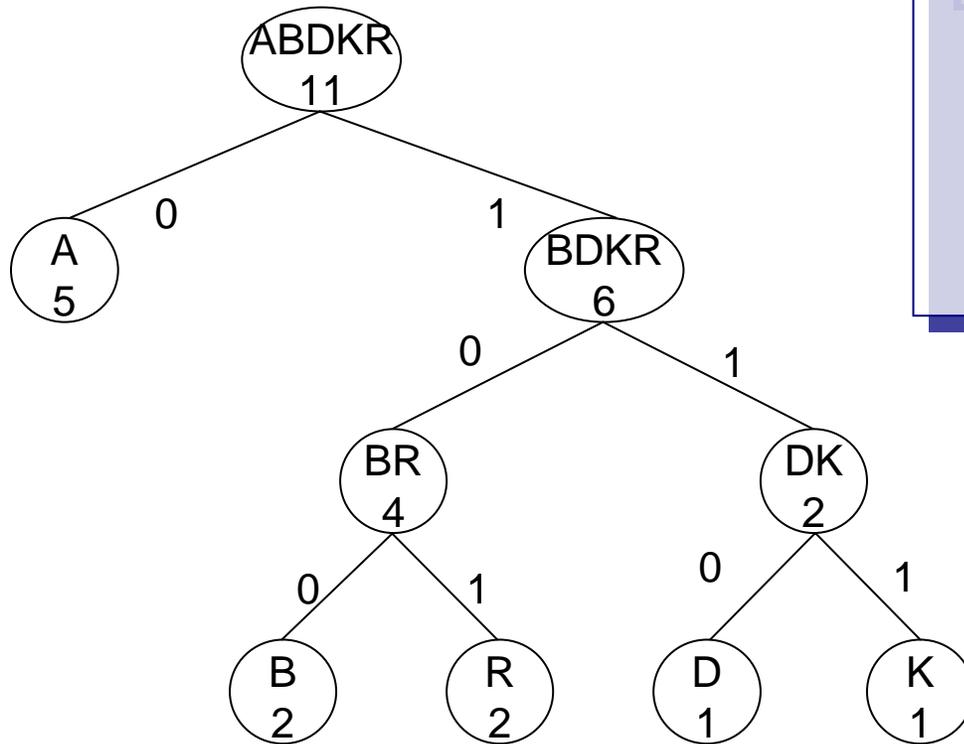
Eigenschaften des Huffman-Codes



Decodieren wir den Text:

$\underbrace{110}_D$ $\underbrace{0}_A$ $\underbrace{111}_K$ $\underbrace{0101}_A$

Eigenschaften des Huffman-Codes



Decodieren wir den Text:

$\underbrace{110}_D$ $\underbrace{0}_A$ $\underbrace{111}_K$ $\underbrace{0}_A$ $\underbrace{101}_R$

Eigenschaften des Huffman-Codes

Der Huffman-Code ist **präfixfrei**.

Vergleich mit dem Telefonsystem:

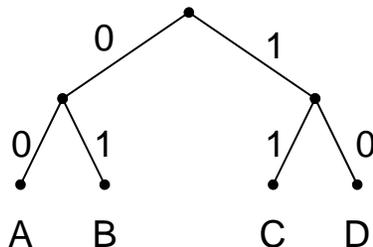
Das Telefonnummernsystem ist auch präfixfrei.

Beispiel:

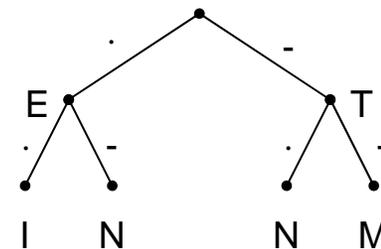
Wählt man 110, „weiß“ das System, dass man fertig mit wählen ist und verbindet einen mit der Polizei. Das liegt daran, dass die Nummer 110 nie Anfangsteil (Präfix) einer anderen Nummer, z.B. gibt es keine Telefonnummer 11011.

Woran erkennt man einen präfixfreien Code?

Ein Codebaum liefert einen präfixfreien Code, wenn die zu codierenden Zeichen nur in den Blättern des Baumes stehen. Beim Morsecode ist dies beispielsweise nicht der Fall, daher muss nach jedem Buchstaben eine kleine Pause mitgeteilt werden.

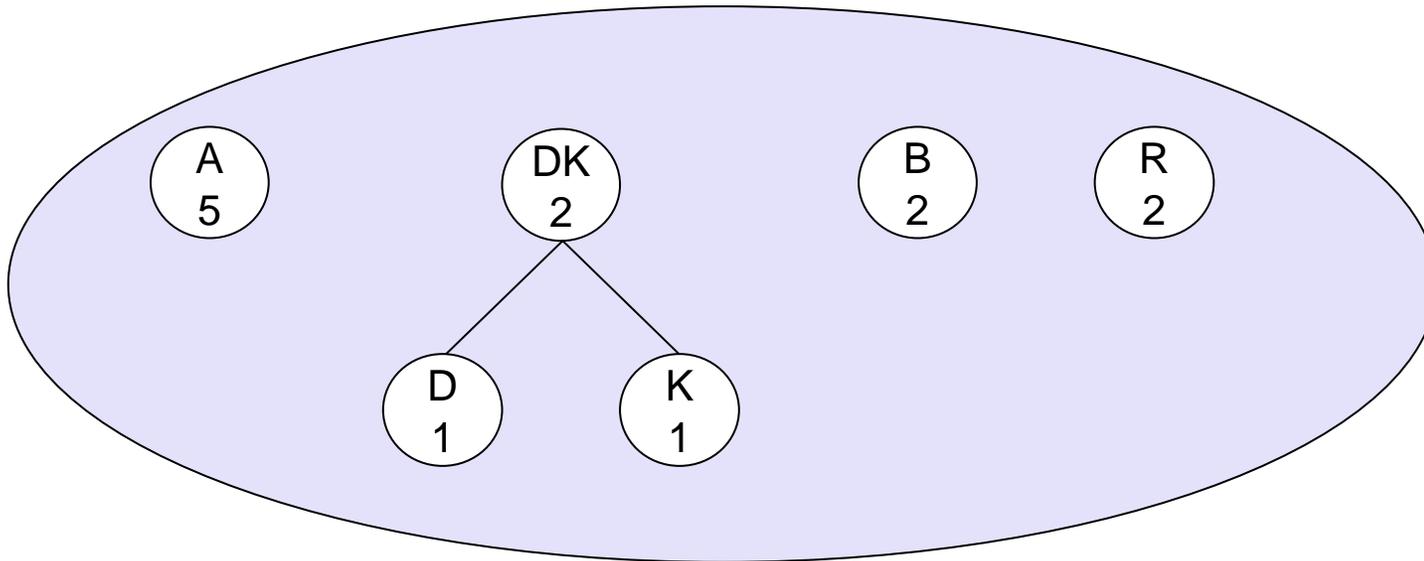


präfixfrei



Morsecode

Eigenschaften Huffman-Codes

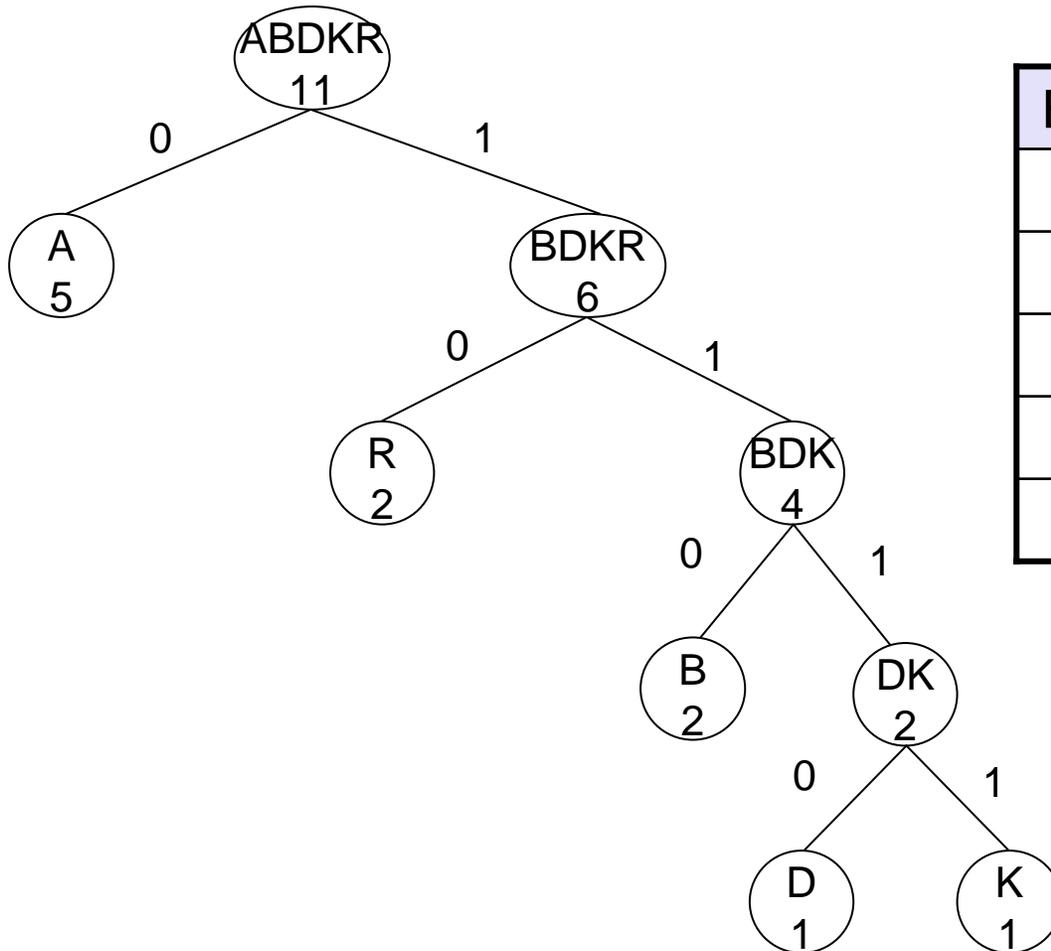


Huffman-Liste 2

In der Huffman-Liste zwei haben wir „B“ und „R“ zu einem Baum zusammengeführt, wir hätten auch „DK“ und „B“ wählen können.

Eigenschaften des Huffman-Codes

Codebaum*



Codetabelle*

Buchstaben	Binärkode
A	0
B	110
D	1110
K	1111
R	10

Eigenschaften des Huffman-Codes

Codierung* des Textes:

A B R A K A D A B R A
0 110 10 0 1111 0 1110 0 110 10 0

Buchstaben	Binärcode
A	0
B	110
D	1110
K	1111
R	10

Mittlere Codewortlänge $23/11 \approx 2,1$.

$$L = \sum_{i=1}^n p_i \cdot l_i$$

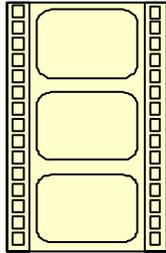
Die Formel liefert den Erwartungswert der Zufallsvariablen „Codewortlänge“.

Der Huffman-Algorithmus minimiert die mittlere Codewortlänge und liefert eine möglichst kurze also eine **optimalen Codierung**.

Die *Huffman-Codewortlänge* ist ein Maß für die Entropie eines Textes.

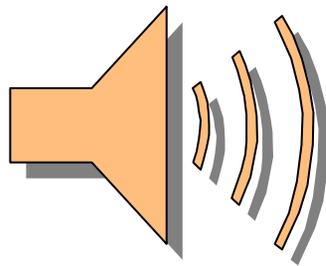
Anwendungsbeispiele

MPEG



JPEG

Telefax



MP3



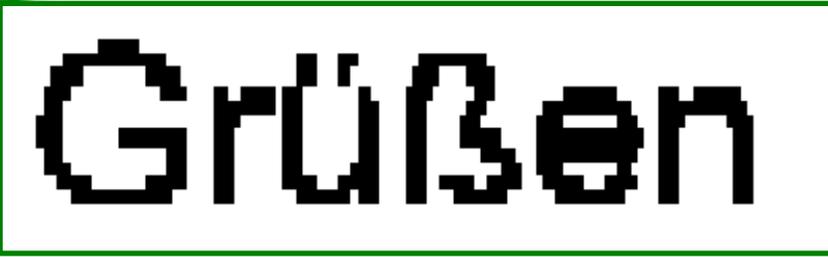
ZIP

...

Telefax-Codierung

- **Speicherplatzbedarf:**
 $1011 * 1728 = 1.747.008$ Bit
 (ca. **1,7 MBit**)
- **Übertragung** würde
 $1747008 \text{ bit} / 2400 \text{ bit/sec} = 727 \text{ sec}$
 bzw. **12 min** dauern.

1011 Zeilen



Grüßen

Christian Urf
 Auf dem Schafwasen 5
 72218 Wildberg
 Tel: 0721-5695063
 E-Mail: christian.urff@web.de

An das
 Prüfungsamt Mathematik und Informatik
 Informatikzentrum
 Universitätsstraße 1
 58084 Hagen

Wildberg, den 09.07.02

Das ist die Betreffzeile

Sehr geehrten Damen und Herren,

hiermit beantrage ich die Anrechnung erbrachter Studienleistungen im Diplomstudiengang Informatik an der Universität Karlsruhe für den Modellstudiengang Bachelor in Informatik an der Fernuniversität Hagen (Studienbeginn: WS 2002/2003).

Beigefügt habe ich Ihnen als Nachweis meiner bisher erbrachten Studienleistungen einen beglaubigten Notenauszug des Prüfungsamtes der Universität Karlsruhe sowie eine Übersicht der in den geprüften Vorlesungen behandelten Themen und meine Vorschläge für eine Anrechnung.

Für Rückfragen stehe ich Ihnen gerne zur Verfügung.

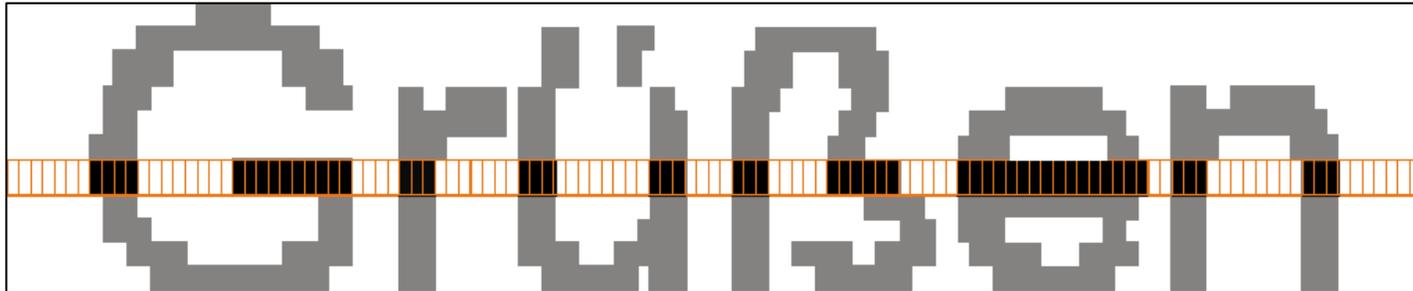
Mit freundlichen Grüßen

Christian Urf

Anlagen:

- Notenauszug des Prüfungsamtes der Universität Karlsruhe
- Übersicht über Themeninhalte der Vorlesungen
- Vorschläge für eine Anrechnung

Lauf­längencodierung

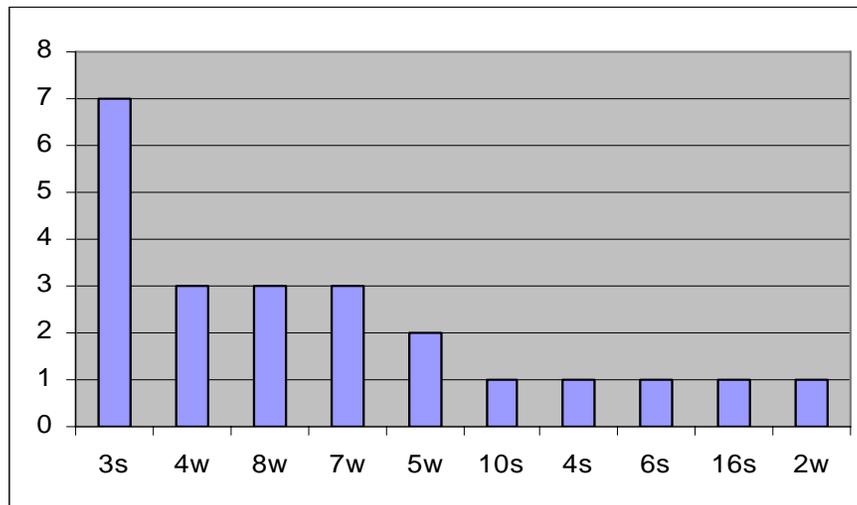


7w, 4s, 8w, 10s, 4w, 3s, 7w, 3s, 8w, 3s, 4w, 3s, 5w, 3s, 5w, 6s, 5w, 16s, 2w, 3s, 8w, 3s, 7w

Lauf­längencodierung (run-length)



Häufigkeitsanalyse:

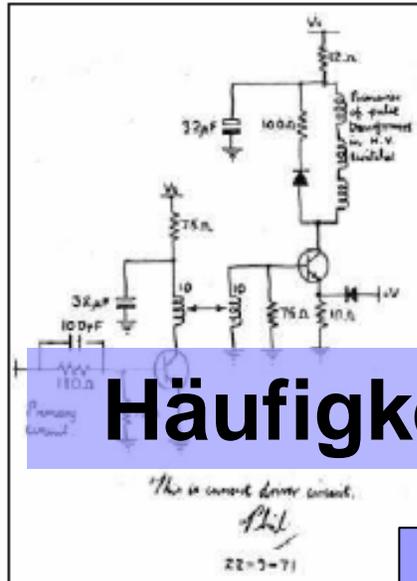


THE SHERRICK COMPANY LIMITED
 100, 102, 104, 106, 108, 110, 112, 114, 116, 118, 120, 122, 124, 126, 128, 130, 132, 134, 136, 138, 140, 142, 144, 146, 148, 150, 152, 154, 156, 158, 160, 162, 164, 166, 168, 170, 172, 174, 176, 178, 180, 182, 184, 186, 188, 190, 192, 194, 196, 198, 200, 202, 204, 206, 208, 210, 212, 214, 216, 218, 220, 222, 224, 226, 228, 230, 232, 234, 236, 238, 240, 242, 244, 246, 248, 250, 252, 254, 256, 258, 260, 262, 264, 266, 268, 270, 272, 274, 276, 278, 280, 282, 284, 286, 288, 290, 292, 294, 296, 298, 300, 302, 304, 306, 308, 310, 312, 314, 316, 318, 320, 322, 324, 326, 328, 330, 332, 334, 336, 338, 340, 342, 344, 346, 348, 350, 352, 354, 356, 358, 360, 362, 364, 366, 368, 370, 372, 374, 376, 378, 380, 382, 384, 386, 388, 390, 392, 394, 396, 398, 400, 402, 404, 406, 408, 410, 412, 414, 416, 418, 420, 422, 424, 426, 428, 430, 432, 434, 436, 438, 440, 442, 444, 446, 448, 450, 452, 454, 456, 458, 460, 462, 464, 466, 468, 470, 472, 474, 476, 478, 480, 482, 484, 486, 488, 490, 492, 494, 496, 498, 500, 502, 504, 506, 508, 510, 512, 514, 516, 518, 520, 522, 524, 526, 528, 530, 532, 534, 536, 538, 540, 542, 544, 546, 548, 550, 552, 554, 556, 558, 560, 562, 564, 566, 568, 570, 572, 574, 576, 578, 580, 582, 584, 586, 588, 590, 592, 594, 596, 598, 600, 602, 604, 606, 608, 610, 612, 614, 616, 618, 620, 622, 624, 626, 628, 630, 632, 634, 636, 638, 640, 642, 644, 646, 648, 650, 652, 654, 656, 658, 660, 662, 664, 666, 668, 670, 672, 674, 676, 678, 680, 682, 684, 686, 688, 690, 692, 694, 696, 698, 700, 702, 704, 706, 708, 710, 712, 714, 716, 718, 720, 722, 724, 726, 728, 730, 732, 734, 736, 738, 740, 742, 744, 746, 748, 750, 752, 754, 756, 758, 760, 762, 764, 766, 768, 770, 772, 774, 776, 778, 780, 782, 784, 786, 788, 790, 792, 794, 796, 798, 800, 802, 804, 806, 808, 810, 812, 814, 816, 818, 820, 822, 824, 826, 828, 830, 832, 834, 836, 838, 840, 842, 844, 846, 848, 850, 852, 854, 856, 858, 860, 862, 864, 866, 868, 870, 872, 874, 876, 878, 880, 882, 884, 886, 888, 890, 892, 894, 896, 898, 900, 902, 904, 906, 908, 910, 912, 914, 916, 918, 920, 922, 924, 926, 928, 930, 932, 934, 936, 938, 940, 942, 944, 946, 948, 950, 952, 954, 956, 958, 960, 962, 964, 966, 968, 970, 972, 974, 976, 978, 980, 982, 984, 986, 988, 990, 992, 994, 996, 998, 1000

THE SHERRICK COMPANY LIMITED
 100, 102, 104, 106, 108, 110, 112, 114, 116, 118, 120, 122, 124, 126, 128, 130, 132, 134, 136, 138, 140, 142, 144, 146, 148, 150, 152, 154, 156, 158, 160, 162, 164, 166, 168, 170, 172, 174, 176, 178, 180, 182, 184, 186, 188, 190, 192, 194, 196, 198, 200, 202, 204, 206, 208, 210, 212, 214, 216, 218, 220, 222, 224, 226, 228, 230, 232, 234, 236, 238, 240, 242, 244, 246, 248, 250, 252, 254, 256, 258, 260, 262, 264, 266, 268, 270, 272, 274, 276, 278, 280, 282, 284, 286, 288, 290, 292, 294, 296, 298, 300, 302, 304, 306, 308, 310, 312, 314, 316, 318, 320, 322, 324, 326, 328, 330, 332, 334, 336, 338, 340, 342, 344, 346, 348, 350, 352, 354, 356, 358, 360, 362, 364, 366, 368, 370, 372, 374, 376, 378, 380, 382, 384, 386, 388, 390, 392, 394, 396, 398, 400, 402, 404, 406, 408, 410, 412, 414, 416, 418, 420, 422, 424, 426, 428, 430, 432, 434, 436, 438, 440, 442, 444, 446, 448, 450, 452, 454, 456, 458, 460, 462, 464, 466, 468, 470, 472, 474, 476, 478, 480, 482, 484, 486, 488, 490, 492, 494, 496, 498, 500, 502, 504, 506, 508, 510, 512, 514, 516, 518, 520, 522, 524, 526, 528, 530, 532, 534, 536, 538, 540, 542, 544, 546, 548, 550, 552, 554, 556, 558, 560, 562, 564, 566, 568, 570, 572, 574, 576, 578, 580, 582, 584, 586, 588, 590, 592, 594, 596, 598, 600, 602, 604, 606, 608, 610, 612, 614, 616, 618, 620, 622, 624, 626, 628, 630, 632, 634, 636, 638, 640, 642, 644, 646, 648, 650, 652, 654, 656, 658, 660, 662, 664, 666, 668, 670, 672, 674, 676, 678, 680, 682, 684, 686, 688, 690, 692, 694, 696, 698, 700, 702, 704, 706, 708, 710, 712, 714, 716, 718, 720, 722, 724, 726, 728, 730, 732, 734, 736, 738, 740, 742, 744, 746, 748, 750, 752, 754, 756, 758, 760, 762, 764, 766, 768, 770, 772, 774, 776, 778, 780, 782, 784, 786, 788, 790, 792, 794, 796, 798, 800, 802, 804, 806, 808, 810, 812, 814, 816, 818, 820, 822, 824, 826, 828, 830, 832, 834, 836, 838, 840, 842, 844, 846, 848, 850, 852, 854, 856, 858, 860, 862, 864, 866, 868, 870, 872, 874, 876, 878, 880, 882, 884, 886, 888, 890, 892, 894, 896, 898, 900, 902, 904, 906, 908, 910, 912, 914, 916, 918, 920, 922, 924, 926, 928, 930, 932, 934, 936, 938, 940, 942, 944, 946, 948, 950, 952, 954, 956, 958, 960, 962, 964, 966, 968, 970, 972, 974, 976, 978, 980, 982, 984, 986, 988, 990, 992, 994, 996, 998, 1000

Phil.

Muster 1



Muster 2

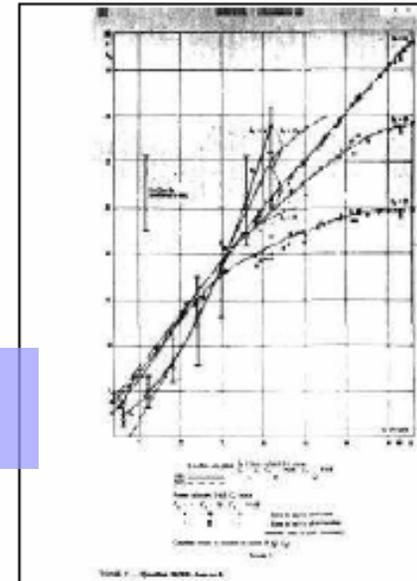
Ohm's Law states that $V = IR$ and that the current through a resistor is directly proportional to the voltage across it. This is shown in the graph below.

Graph 1: $V = IR$

Graph 2: $V = IR$

Graph 3: $V = IR$

Muster 5



Muster 6

NAME	DATE	MARKS	REMARKS
1. 1000	1. 1000	1000	1000
2. 1000	1. 1000	1000	1000
3. 1000	1. 1000	1000	1000
4. 1000	1. 1000	1000	1000
5. 1000	1. 1000	1000	1000
6. 1000	1. 1000	1000	1000
7. 1000	1. 1000	1000	1000
8. 1000	1. 1000	1000	1000
9. 1000	1. 1000	1000	1000
10. 1000	1. 1000	1000	1000

Muster 3

The Huffman algorithm is a lossless data compression algorithm. It is named after David Huffman, who published his work on the subject in 1952. The algorithm is based on the concept of variable-length codes, where more frequent characters are represented by shorter codes and less frequent characters by longer codes. This results in a more compact representation of the data.

The Huffman algorithm is a lossless data compression algorithm. It is named after David Huffman, who published his work on the subject in 1952. The algorithm is based on the concept of variable-length codes, where more frequent characters are represented by shorter codes and less frequent characters by longer codes. This results in a more compact representation of the data.

Muster 4

The Huffman algorithm is a lossless data compression algorithm. It is named after David Huffman, who published his work on the subject in 1952. The algorithm is based on the concept of variable-length codes, where more frequent characters are represented by shorter codes and less frequent characters by longer codes. This results in a more compact representation of the data.

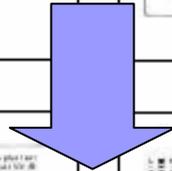
The Huffman algorithm is a lossless data compression algorithm. It is named after David Huffman, who published his work on the subject in 1952. The algorithm is based on the concept of variable-length codes, where more frequent characters are represented by shorter codes and less frequent characters by longer codes. This results in a more compact representation of the data.

Muster 7

WELL, WE ASKED FOR IT!

Muster 8

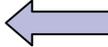
Häufigkeitsanalyse



Huffman-Algorithmus

Telefax-Code:

(Ausschnitt)

Lauflänge	Codes für Schwarz
1s	010
2s	11 
3s	10 
4s	011
5s	0011
6s	0010
7s	00011
8s	000101
9s	000100
10s	0000100
11s	0000101
12s	0000111
13s	00000100
14s	00000111
15s	000011000
16s	0000010111
17s	0000011000
18s	0000001000
19s	00001100111 
20s	00001101000

Effizienz der Kompression

Es lassen sich Kompressionsraten von bis zu **1:50** erreichen.

	Ohne Kompression	Mit Kompression
Datenmenge	1,7 MBit	0,04 MBit
Übertragungsdauer	12 min (720 sec)	15 sec



Mathematik für Information und Kommunikation

Danke für Ihre Aufmerksamkeit.

Am Beispiel des Huffman- Algorithmus

Thomas Borys und (Christian Urff)