64-041 Übung Rechnerstrukturen und Betriebssysteme

Aufgabenblatt 7 Ausgabe: 30.11., Abgabe: 07.12. 24:00

Gruppe	
Name(n)	Matrikelnummer(n)

Aufgabe 7.1 (Punkte 5+5)

Entropie und Redundanz: Wir betrachten die BCD-Codierung der 10 Dezimalziffern.

(a) Die Ziffern seien alle gleich wahrscheinlich. Wie groß ist der mittlere Informationsgehalt (die Entropie) H dieser Symbole?

(b) Wie groß ist die Redundanz ($H_0 - H$) des BCD-Codes?

Aufgabe 7.2 (Punkte 5+10+5+15)

Optimale Codierung: In vielen realen Datensätzen und Messdaten treten die Dezimalziffern nicht mit gleicher Wahrscheinlichkeit auf, wie in der vorigen Aufgabe. Häufig folgen die Wahrscheinlichkeiten der Dezimalziffern dem Benford'schen Gesetz (siehe `de.wikipedia.org/wiki/Benfordsches_Gesetz`) mit folgenden Werten:

a_i	1	2	3	4	5	6	7	8	9	0
$p(a_i)$	0,301	0,176	0,125	0,097	0,079	0,067	0,058	0,051	0,046	0,0

(a) Wie groß ist jetzt der mittlere Informationsgehalt (die Entropie) H der Dezimalziffern?

(b) Konstruieren Sie einen Fano-Code zur Quellencodierung.

(c) Berechnen Sie die mittlere Codewortlänge H_0 und die Redundanz ($H_0 - H$) für ihren Fano-Code?

(d) Wie ändert sich die Redundanz, wenn man einen Huffman-Code nutzt?

Aufgabe 7.3 (Punkte 10+10+5)

2 D-Paritätscode: Wir betrachten den in der Vorlesung vorgestellten zweidimensionalen Paritätscode. Jeweils 64 Datenbits werden als Matrix mit 8×8 Zeilen und Spalten notiert und zu jeder Zeile und Spalte wird ein ungerades Paritätsbit hinzugefügt. Ein weiteres Bit ganz unten rechts berechnet sich als Parität der Spalten-/Zeilen-Paritätsbits.

$d_{0,0}$	$d_{0,1}$	$d_{0,2}$	$d_{0,3}$	$d_{0,4}$	$d_{0,5}$	$d_{0,6}$	$d_{0,7}$	$p_{0,8}$
$d_{1,0}$	$d_{1,1}$	$d_{1,2}$	$d_{1,3}$	$d_{1,4}$	$d_{1,5}$	$d_{1,6}$	$d_{1,7}$	$p_{1,8}$
$d_{2,0}$	$d_{2,1}$	$d_{2,2}$	$d_{2,3}$	$d_{2,4}$	$d_{2,5}$	$d_{2,6}$	$d_{2,7}$	$p_{2,8}$
$d_{3,0}$	$d_{3,1}$	$d_{3,2}$	$d_{3,3}$	$d_{3,4}$	$d_{3,5}$	$d_{3,6}$	$d_{3,7}$	$p_{3,8}$
$d_{4,0}$	$d_{4,1}$	$d_{4,2}$	$d_{4,3}$	$d_{4,4}$	$d_{4,5}$	$d_{4,6}$	$d_{4,7}$	$p_{4,8}$
$d_{5,0}$	$d_{5,1}$	$d_{5,2}$	$d_{5,3}$	$d_{5,4}$	$d_{5,5}$	$d_{5,6}$	$d_{5,7}$	$p_{5,8}$
$d_{6,0}$	$d_{6,1}$	$d_{6,2}$	$d_{6,3}$	$d_{6,4}$	$d_{6,5}$	$d_{6,6}$	$d_{6,7}$	$p_{6,8}$
$d_{7,0}$	$d_{7,1}$	$d_{7,2}$	$d_{7,3}$	$d_{7,4}$	$d_{7,5}$	$d_{7,6}$	$d_{7,7}$	$p_{7,8}$
$p_{8,0}$	$p_{8,1}$	$p_{8,2}$	$p_{8,3}$	$p_{8,4}$	$p_{8,5}$	$p_{8,6}$	$p_{8,7}$	$p_{8,8}$

(a) Wie groß ist die Minimaldistanz d dieses Codes? Begründen Sie Ihre Antwort.

(b) Können mit diesem Code Ein-, Zwei- und Dreibitfehler erkannt und korrigiert werden? Geben Sie jeweils an, welche Arten von Fehlern erkannt, bzw. korrigiert werden können.

(c) Angenommen, Sie empfangen folgende Bits. Wurden alle Bits korrekt übertragen? Wenn nein, welches Bit muss korrigiert werden?

1	1	0	0	0	0	1	1	1
0	0	1	1	1	1	0	0	1
1	0	0	1	0	0	0	1	0
0	1	1	1	0	0	0	1	1
0	0	0	1	0	0	1	1	0
0	0	0	1	1	1	1	1	0
1	1	1	0	1	0	0	1	1
1	1	0	0	0	1	0	0	0
1	1	0	0	0	0	0	0	1

Aufgabe 7.4 (Punkte 10+10+10)

Hamming-Code: Entsprechend dem Schema aus der Vorlesung wird ein (7,4)-Hamming-Code gebildet, der Einzelbitfehler korrigieren kann. Die vier Informationsbits (d_i) werden um drei Prüfbits (p_j) ergänzt und bilden ein Codewort C ($c_1 \dots c_7$). Mit Hilfe der x_i lässt sich dann der (eine) Einzelbitfehler lokalisieren.

Codewortstelle	c_1	c_2	c_3	c_4	c_5	c_6	c_7
Bedeutung	p_1	p_2	d_1	p_3	d_2	d_3	d_4
Prüfgruppe 1	*		*		*		*
Prüfgruppe 2		*	*			*	*
Prüfgruppe 3				*	*	*	*

$$p_1 = d_1 \oplus d_2 \oplus d_4 \qquad x_a = c_1 \oplus c_3 \oplus c_5 \oplus c_7$$
$$p_2 = d_1 \oplus d_3 \oplus d_4 \qquad x_b = c_2 \oplus c_3 \oplus c_6 \oplus c_7$$
$$p_3 = d_2 \oplus d_3 \oplus d_4 \qquad x_c = c_4 \oplus c_5 \oplus c_6 \oplus c_7$$

(a) Sie haben das Wort $c = 1100011$ empfangen. Ist das ein gültiges Codewort?
Wenn ja, schreiben Sie alle gültigen Codeworte auf.
Wenn nein, beschreiben Sie wie der Fehler lokalisiert und korrigiert werden kann.

(b) Schreiben Sie die Generatormatrix G und die Prüfmatrix H für einen (15,11)-Hamming-Code auf.

(c) Codieren Sie mit der Matrix G des vorigen Aufgabenteils:

$d = (d_1, d_2 \dots d_{11}) = (1,0,0,0,1,1,0,0,0,1,1)$