64-041 Übung Rechnerstrukturen und Betriebssysteme

Aufgabenblatt 7 Ausgabe: 27.11., Abgabe: 04.12. 24:00

Gruppe	
Name(n)	Matrikelnummer(n)

Aufgabe 7.1 (Punkte 10+10)

Codierung: Die 26 Buchstaben des Alphabets sollen in einem zyklisch-einschrittigen Binärcode „durchgezählt“ werden.

(a) Entwickeln Sie einen Code mit dem rekursiven Verfahren aus der Vorlesung.

(b) Kann man den Code so erweitern, dass auch das Leerzeichen mit codiert wird?
Wenn ja, geben Sie eine gültigen Code an.
Wenn nein, begründen Sie Ihre Antwort.

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

Optimale Codierung: Die folgenden 10 Symbole a_i sind mit ihren Wahrscheinlichkeiten $p(a_i)$ in der Tabelle angegeben:

a_i	a	b	c	d	e	f	g	h	i	j
$p(a_i)$	0,14	0,02	0,08	0,3	0,01	0,05	0,1	0,09	0,06	0,15

(a) Wie groß ist der mittlere Informationsgehalt (die Entropie) H dieser Symbole?

(b) Geben Sie eine Fano-Codierung für diese Symbole an.

(c) Welche mittlere Codewortlänge H_0 ergibt sich?

(d) Wie groß ist die Redundanz ($H_0 - H$) ihres Codes?

Aufgabe 7.3 (Punkte 10+10)

Huffman-Codierung: Das Huffman-Verfahren ist nicht auf Binärsymbole beschränkt. Bei der *Radix-r* Huffman-Codierung wird ein optimaler Code mit Codewörtern variabler Länge auf dem Alphabet der Symbole $\{0, 1, \ldots r-1\}$ aufgebaut. Bei der Konstruktion des Codebaums werden dabei jeweils die r Symbole (oder Knoten) mit der geringsten Symbolhäufigkeit zu einem neuen Knoten zusammengefasst. Nachdem der Baum komplett ist, werden in jeder Stufe die einzelnen Symbole $0, 1, \ldots r-1$ zugeordnet. Die einzige Komplikation besteht darin, dass abhängig von der Anzahl der zu codierenden Symbole eventuell ein Knoten des fertigen Codebaums weniger als r Kinder hat. Damit der erzeugte Code später minimale Länge hat, darf dieser nicht komplett „ausgelastete“ Knoten allerdings nicht an der Wurzel des Baums sitzen (dann würden kurze Codewörter verschwendet), sondern muss ganz zu Anfang der Konstruktion (bei den längsten Codewörtern) eingefügt werden.

(a) Gegeben sei die Basis r und eine Anzahl n von zu codierenden Symbolen a_i mit Wahrscheinlichkeiten $p(a_i)$, mit $i = 0, \ldots n-1$. Überlegen Sie sich, wie viele Symbole m (mit den kleinsten Wahrscheinlichkeiten $p(a_j)$ im allerersten Schritt abhängig von r und n zusammengefasst werden müssen.

(b) Das folgende Beispiel stammt aus dem Originalpaper von Huffman. Wir haben $r = 4$ und die folgenden $n = 8$ Symbole a_i mit Wahrscheinlichkeiten $p(a_i)$:

a_i	a	b	c	d	e	f	g	h
$p(a_i)$	0,22	0,20	0,18	0,15	0,10	0,08	0,05	0,02

Konstruieren Sie den *Radix-4* Huffman-Codebaum für diese Symbole und geben Sie die zugehörige Codierung an.

Aufgabe 7.4 (Punkte 5+10+15)

Hamming-Code: Entsprechend dem Schema aus der Vorlesung wird ein (7,4)-Hamming-Code gebildet, der Einzelbitfehler korrigieren kann. Wie in der Tabelle dargestellt, besitzt er vier Informationsbits (d_i) und drei Prüfbits (p_j). Insgesamt sind $2^4 = 16$ Codewörter möglich, die in der linken Tabelle aufgelistet sind:

	c_1	c_2	c_3	c_4	c_5	c_6	c_7
Nr.	p_1	p_2	d_1	p_3	d_2	d_3	d_4
1			0		0	0	0
2			0		0	0	1
3			0		0	1	0
4			0		0	1	1
5			0		1	0	0
6			0		1	0	1
7			0		1	1	0
8			0		1	1	1
9			1		0	0	0
10			1		0	0	1
11			1		0	1	0
12			1		0	1	1
13			1		1	0	0
14			1		1	0	1
15			1		1	1	0
16			1		1	1	1

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 A	*		*		*		*
Prüfgruppe B		*	*			*	*
Prüfgruppe C				*	*	*	*

Für die Prüfstellen gilt:

$$c_1 = c_3 \oplus c_5 \oplus c_7$$
$$c_2 = c_3 \oplus c_6 \oplus c_7$$
$$c_4 = c_5 \oplus c_6 \oplus c_7$$

Um (einen) Einzelbitfehler zu lokalisieren, bildet man ein Prüfwort (x_a, x_b, x_c), wobei gilt:

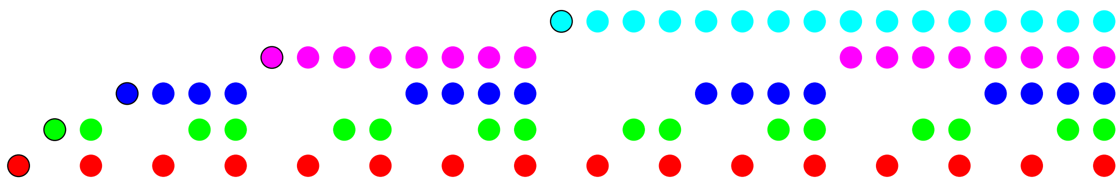
$$x_a = c_1 \oplus c_3 \oplus c_5 \oplus c_7$$
$$x_b = c_2 \oplus c_3 \oplus c_6 \oplus c_7$$
$$x_c = c_4 \oplus c_5 \oplus c_6 \oplus c_7$$

(a) Tragen Sie die Prüfbits in die Tabelle ein.

(b) Zeigen Sie anhand eines Beispiels, wie ein auftretender Einzelbitfehler lokalisiert und damit korrigiert werden kann. Verfälschen Sie dazu die Codewortstelle c_3 des Codewortes Nr. 12 und bilden Sie die Prüfbits.

Zeigen Sie dann, wie man aus dem Prüfwort die fehlerhafte Codewortstelle bestimmt?

(c) Beschreiben Sie, wie man aus einem Schema für Hamming-Codes, wie nachfolgend angegeben, Generatormatrix G und die Prüfmatrix H erstellen kann. Erstellen Sie die Matrizen G und H für den (15,11)-Hamming-Code.



2^0	2^1		2^2				2^3								2^4															
C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}	C_{16}	C_{17}	C_{18}	C_{19}	C_{20}	C_{21}	C_{22}	C_{23}	C_{24}	C_{25}	C_{26}	C_{27}	C_{28}	C_{29}	C_{30}	C_{31}
p_1	p_2	d_1	p_3	d_2	d_3	d_4	p_4	d_5	d_6	d_7	d_8	d_9	d_{10}	d_{11}	p_5	d_{12}	d_{13}	d_{14}	d_{15}	d_{16}	d_{17}	d_{18}	d_{19}	d_{20}	d_{21}	d_{22}	d_{23}	d_{24}	d_{25}	d_{26}