Übung 13: Quellencodierung

Aufgabe 1: *Huffmann-Algorithmus.*

Betrachten Sie die folgende ternäre, gedächtnislose Quelle mit dem Symbolalphabet *A* = {A,B,C} und den Symbol-Wahrscheinlichkeiten $P_X(A)=2/3$, $P_X(B)=1/6$, $P_X(C)=1/6$.

a) Bestimmen Sie die Information pro Symbol H(X) der Quelle.

Wieviel Information tragen 2 aufeinander folgende Quellensymbole?

b) Entwerfen Sie einen binären Huffman-Code, mit dem Sie 1 Symbol codieren bzw. komprimieren können.

Wie gross ist die Coderate bzw. die mittlere Codewortlänge [bit/Symbol]?

c) Entwerfen Sie einen binären Huffman-Code, mit dem Sie 2 aufeinander folgende Symbole codieren bzw. komprimieren können.

Wie gross ist die Coderate bzw. die mittlere Codewortlänge [bit/Symbol] jetzt?

Aufgabe 2: *BMS.*

Betrachten Sie die folgende binäre, gedächtnisfreie Quelle (BMS).

X[n] = ... 0 0 0 0 0 0 0 1 1 0 0 1 1 0 1 0 0 0 0 0 ...

a) Bestimmen Sie die Information bzw. die Entropie H(X) [bit / Quellensymbol].

Ein Quellencoder fasst jeweils 3 Symbole zusammen und codiert sie mit einem binären, präfixfreien Codewort variabler Länge, siehe Tabelle.

Eingang X[n]	Ausgang Y[n]
0 0 0	0
0 0 1	1 0 0
0 1 0	1 0 1
0 1 1	1 1 1 0 0
1 0 0	1 1 0
1 0 1	1 1 1 0 1
1 1 0	1 1 1 1 0
1 1 1	1 1 1 1 1

b) Ist dieser Quellenencoder verlustlos?

Wenn ja, wie gut ist diese Datenkompression?

Hinweis:
Bestimmen Sie die mittlere Codewortlänge und dann die mittlere Anzahl bit / Symbol.

c) Bestimmen Sie die Wahrscheinlichkeitsverteilung $P_Y(y)$ am Ausgang des Quellen-encoders.

Aufgabe 3: *LZ77.*

Komprimieren Sie mit der LZ77-Methode den folgenden Text:

FISCHERS FRITZ FISCHT FRISCHE FISCHE.

Der Vorschau-Buffer soll 6 Symbole (Bytes) und der Such-Buffer 32 Bytes lang sein.

Hinweis: Bestimmen Sie im Satz oben die Grenzen zwischen Such- und Vorschau-Buffer und markieren Sie sie mit Hoch-Kommas.

Bestimmen Sie zusätzlich die Kompressionsrate R.

Dekodieren Sie zum Schluss die Token-Folge und vergewissern Sie sich, dass der Dekoder sehr einfach zu realisieren ist.

Musterlösung

Aufgabe 1

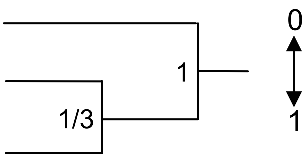
a) $H(X) = 2/3 \cdot \log_2(3/2) + 2 \cdot 1/6 \cdot \log_2(6) = 1.2516$ bit / Symbol

Die Symbole einer DMS sind unabhängig. Deshalb gilt:

$H(X[n-1],X[n]) = H(X[n-1]) + H(X[n]) = 2 \cdot H(X) = 2.5032$ bit / 2 Symbolen

b) Binärer Huffman-Code für 1 Quellensymbol:

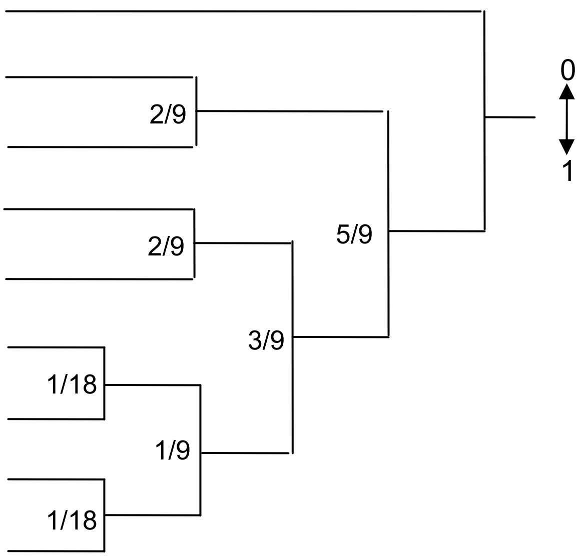
0	A	2/3
10	B	1/6
11	C	1/6



$R = E[L] = 2/3 \cdot 1 + 1/6 \cdot 2 + 1/6 \cdot 2 = 4/3 = 1.33$ bit / Symbol $> H(X) = 1.2516$ bit / Symbol

c) Binärer Huffman-Code für 2 Quellensymbole:

0	AA	4/9
100	AB	1/9
101	AC	1/9
1100	BA	1/9
1101	CA	1/9
11100	BB	1/36
11101	BC	1/36
11110	CB	1/36
11111	CC	1/36



$R = E[L] = 4/9 \cdot 1 + 2 \cdot 1/9 \cdot 3 + 2 \cdot 1/9 \cdot 4 + 4 \cdot 1/36 \cdot 5 = 2.5556$ bit / 2 Symbol
$= 1.2778$ bit / Symbol $> H(X) = 1.2516$ bit / Symbol

Je mehr Symbole gleichzeitig codiert bzw. komprimiert werden, desto besser ist die Kompression, auf Kosten der Code-Komplexität.

Aufgabe 2

a) Um H(X) zu bestimmen, muss man zuerst $P_X(x)$ bestimmen.

Dem Ausgangsmuster kann man entnehmen: $P_X(x) = 0.75$ und $P_X(1) = 0.25$

Eigentlich müsste man noch mehr Ausgangssymbole betrachten.

Aus der Grafik der binären Entropiefunktion h(p) im Skript kann man für p=0.25 ablesen:
=> H(X) = 0.811 bit / Symbol

b) Um die Frage zu beantworten, ob der Quellencode verlustlos ist, müssen wir abklären, wie gross die mittlere Codewortlänge E[W] ist. Ist E[W]/3 ≥ H(X), so ist der Code verlustlos. Andernfalls werden die Quellensymbole zu stark komprimiert, so dass Information verloren geht.

Die mittlere Codewortlänge beträgt, siehe Tabelle unten:

$E[W] = 0.4219 \cdot 1 + 3 \cdot 0.1406 \cdot 3 + 3 \cdot 0.0469 \cdot 5 + 0.0156 \cdot 5 = 2.4688$ bit / 3 Quellensymbole

Eingang X[n]	P(CW gesendet) = P(Eingang)	Länge Codewort [bit]
0 0 0	$0.75^3 = 0.4219$	1
0 0 1	0.1406	3
0 1 0	0.1406	3
0 1 1	0.0469	5
1 0 0	0.1406	3
1 0 1	0.0469	5
1 1 0	0.0469	5
1 1 1	$0.25^3 = 0.0156$	5

=> E[W] / 3 = 0.8229 bit / Symbol > H(X)

Der Quellencode ist verlustlos und nahe am Optimum.

c) Es sollen z.B. 3000 Quellensymbole komprimiert werden.

Dazu sind 1000 Codeworte mit mittlerer Länge E[W]=2.4688 erforderlich

In diesen 1000 Codeworten sind
=> 422*1 Nullen (Eingang 000)
=> 141*2 Nullen (Eingang 001)
=> 141*1 Nullen (Eingang 010)
=> 47*2 Nullen (Eingang 011)
=> 141*1 Nullen (Eingang 100)
=> 47*1 Nullen (Eingang 101)
=> 47*1 Nullen (Eingang 110)

Daraus folgt, dass von 2469 Codebit ca. 422+4*141+4*47 Nullen sind.

Wie erwartet ist $P_Y(0) = 0.4755$ und damit fast 0.5. Der binäre Quellenencoder sollte am Ausgang ja fast keine Redundanz mehr enthalten und damit im Mittel fast gleich viele Nullen wie Einer aufweisen.

Aufgabe 3

F’I’S’C’H’E’R’S ’FR’IT’Z’ FI’SCHT’ FRIS’CHE ’FISCHE.’

<= (0,0,F) (0,0,I) (0,0,S) (0,0,C) (0,0,H) (0,0,E) (0,0,R) (5,1, _) (9,1,R) (10,1,T)
(0,0,Z) (6,2,I) (15,3,T) (13,4,S) (23,3, _) (30,6,.)

Input: 37 Bytes, Output: 16 Tokens mit je 5+3+8 = 2 Bytes => R = 32/37 = 0.865
(kleine Kompression, für bessere Kompression müsste der Input viel länger sein)

Dekoder nach 7 Tokens: FISCHER
Dekoder nach 8 Tokens: FISCHERS_
Dekoder nach 9 Tokens: FISCHERS FR