



Kommunikationstechnik

Bachelor-Studiengang Telematik
Sommersemester 2019
Thema: Kodierung

Quellen- und Leitungskodierung



0

1





- Quellencodierung:
 - Dient der Optimierung des Durchsatzes
 - Dazu gehört auch Kompression
- Kanalcodierung:
 - Dient der Fehlersicherung
 - Paritäts-Bits, Prüfsummen, Redundanz...
- Leitungscodierung
 - Einem oder mehreren Bits wird ein Symbol zugeordnet, das auf der Leitung übertragen wird.

Forderungen an einen Code



- Daten werden kodiert oder müssen kodiert werden, um
 - eine geeignete Darstellung für die technische Verarbeitung zu haben,
 - eine ökonomische Darstellung und Übertragung zu gewährleisten,
 - Informationen vor Verfälschung oder unberechtigtem Zugriff zu schützen (s.a. „Verschlüsselung“).
- Aus diesen Bestrebungen resultieren u.a. folgende Anforderungen an Eigenschaften eines Codes:
 - Umkehrbar eindeutig,
 - geringe Wortlänge,
 - Codierung leicht realisierbar,
 - effizient übertragbar etc.

Kleiner Exkurs: Verschlüsselung



- Ein absolut sicheres Verfahren: Der Schlüssel ist ein Unikat und so lang wie die Nachricht („Onetime-Pad“)
- Ein weniger sicheres Verfahren: Der Schlüssel soll ein Unikat sein und ist wesentlich kürzer als die Nachricht
 - ENIGMA und andere mechanische Verschlüsselungs-Geräte
 - Nutzerfehler erlaubten das „Knacken“
 - Bedeutungslos seit der datentechnischen Verschlüsselung
- Ganz unsichere Verfahren sind schon knackbar, wenn man das Verfahren selbst kennt (z.B. IBM -> HAL)
- Grundregel: Nur die Kenntnis des richtigen Schlüssels, aber nicht allein die des Verfahrens ist ausreichend, um eine verschlüsselte Nachricht zu entschlüsseln (n. Kerckhoff).

Der verfressene König*



- Schweinebraten
- Schokoladenpudding
- Essiggurken
- Erdbeertorte

*Nach Walter R. Fuchs, 1968

Des Königs (Quellen-) Kodierung



Der König beschloss, seine Befehle zu kodieren.

- | | |
|--------------------------------------|--------------------|
| →rechte Hand heben: | Schweinebraten |
| →linke Hand heben: | Schokoladenpudding |
| →erst rechte, dann linke Hand heben: | Essiggurken |
| →zweimal rechte Hand heben: | Erdbeertorte |

Aber es gab Probleme:

Hob der König dreimal die rechte Hand (RRR), bekam er an einem Tag dreimal Schweinebraten, an einem anderen Tag zuerst Erdbeertorte, dann Schweinebraten und an einem dritten erst Schweinebraten, dann Erdbeertorte.

Des Mathematikus Kodierung



- Benötigt wird eine eindeutige, binäre Kodierung.
- Um vier Worte binär zu kodieren, wird eine Codewortlänge von mindestens 2 Zeichen benötigt.
 - Schweinebraten RR
 - Schokoladenpudding RL
 - Essiggurken LR
 - Erdbeertorte LL
- Nun war die Kodierung unverwechselbar.
- Der König bestellte: LRRLLLRRRRLL und erhielt
Gurken, Pudding, Torte, 2 x Braten und Torte

Des Königs Vorlieben



- Die Kodierung war nun eindeutig und der König erhielt immer, was er auch wollte.
- Er wurde immer dicker und dicker und das Heben der Hände strengte ihn an.
- Das warf die Frage auf: Ist die Kodierung optimal oder könnte sie noch kürzer sein?
- Der Mathematiker ließ seinen Assistenten eine Statistik über die königlichen Essenwünsche führen. Ergebnis:
 - Jeden Tag ißt der König durchschnittlich 18 Gerichte.
 - Die Verteilung der Häufigkeiten liegt bei:
9 x Schweinebraten, 6 x Schokopudding, 1 x Essiggurken
und 2 x Erdbeertorte.

Kodierung nach Vorlieben I



- Um die Kodierung optimal zu wählen, beschloss der Mathematiker dem Braten einen möglichst kurzen Code zu geben, während der Code für die Gurken ruhig länger sein konnte.

Merke: Selten auftretende Informationen dürfen ein langes Symbol bekommen, häufig auftretenden Informationen sollten möglichst kurze Symbole haben!

- Der Code soll optimal, eindeutig und zweckmäßig sein.
- Bei der Generierung des Codes muss die **Fano-Bedingung** berücksichtigt werden:
 - Kein Code (kompletter Code!) darf der Beginn eines anderen verwendeten Codes sein.

Kodierung nach Vorlieben II



- Für den Schweinebraten wird festgelegt:
Braten → R
- Damit ist das R verbraucht, und kein Codewort darf mehr mit R beginnen.
- Für den Schokoladenpudding wird festgelegt:
Pudding → LR
- Damit bleibt nur noch ein zweistelliges Wort (LL) übrig, es sind aber noch zwei Worte zu kodieren. Daher werden dreistellige Codes gewählt:
Gurken → LLR
Torte → LLL

Kodierung nach Vorlieben III



- Bei der ersten Kodierung benötigt der König für die Bestellung der 18 Speisen 36 Bit.
- Ausgehend von der vorgegebenen Häufigkeitsverteilung benötigt der König mit dem neuen Code...
 - 9 x Schweinebraten 9 Bit
 - 6 x Schokopudding 12 Bit
 - 1 x Essiggurken 3 Bit
 - 2 x Erdbeertorte 6 Bit
- Es ergeben sich in Summe für diese 18 Speisen 30 Bit.
- Was bestellt der König bei LRRRLLR?

Häufigkeit und Wahrscheinlichkeit I



- Bei der Ermittlung von Häufigkeiten handelt es sich um Erfahrungswerte.
- Führt man ein Ereignis n -mal durch und es führt in m Fällen zu dem zufälligen Ergebnis A , dann liegt die Häufigkeit $h(A) = m/n$ beliebig nahe an der Wahrscheinlichkeit $P(A)$.
- Für unser Beispiel heißt das:
 - A_1 : Der König bestellt Braten
 - A_2 : Der König bestellt Pudding
 - A_3 : Der König bestellt Torte
 - A_4 : Der König bestellt Gurken

Häufigkeit und Wahrscheinlichkeit II



- Nehmen wir an, dass die Werte auf genügend vielen Beobachtungen beruhen, können wir die Wahrscheinlichkeiten durch Häufigkeiten annähern.

$$P(A_1) = 9/18 = 1/2$$

$$P(A_2) = 6/18 = 1/3$$

$$P(A_3) = 2/18 = 1/9$$

$$P(A_4) = 1/18$$

- Die Summe aller Wahrscheinlichkeiten ergibt immer den Wert 1.
- Je kleiner die Wahrscheinlichkeit des Auftretens der Nachricht ist, desto höher ist der *Informationsgehalt*, d.h. die Länge des Codes darf größer sein.

Häufigkeit und Wahrscheinlichkeit III



- Als Formel bedeutet das: Informationsgehalt I der Nachricht A :

$$\rightarrow I(A) = \text{Id} (1/P(A)) \text{ [Bit*]} (= -\text{Id}(P(A)))$$

- Wir wenden nun diese Formel auf die Nachrichten A_1 bis A_4 an:

$$\rightarrow I(A_1) = \text{Id} (1/(1/2)) = \text{Id}(2) = 1,000 \text{ bit}$$

$$\rightarrow I(A_2) = \text{Id} (1/(1/3)) = \text{Id}(3) = 1,585 \text{ bit}$$

($2^{1,585} \approx 3$)

$$\rightarrow I(A_3) = \text{Id} (1/(1/9)) = \text{Id}(9) = 3,170 \text{ bit}$$

$$\rightarrow I(A_4) = \text{Id} (1/(1/18)) = \text{Id}(18) = 4,170 \text{ bit}$$

*Die Einheit der Information wird Bit genannt. Sie ist aber nicht identisch mit der Stelle einer Binärzahl.

Häufigkeit und Wahrscheinlichkeit IV



- ◆ Daraus lässt sich die Entropie H einer beliebigen Nachricht A berechnen (Nach Definition von Claude Shannon ist das der gewichtete Mittelwert der Informationsgehalte aller vorkommenden Nachrichten):

$$H(A) = - \sum_{i=1}^n P(A_i) \text{Id} (P(A_i)) \text{ [Bits]}$$

- ◆ In unserem Beispiel also:

$$\begin{aligned} H(A) &= 0,500 \cdot 1 \\ &+ 0,333 \cdot 1,585 \\ &+ 0,111 \cdot 3,170 \\ &+ 0,056 \cdot 4,170 \\ &= 1,612 \text{ Bits} \end{aligned}$$

- ◆ Das ist die Zielgröße für einen *optimalen* Code.
(Wir haben 30 Bits für 18 Nachrichten erreicht: Das sind durchschnittlich 1,667 Bits pro Nachricht)

UTF-8 I (UCS Transformation Format)



Bits	Last Code Point	Byte 1
7	U+007F	0xxxxxxx

ASCII Kompatibilität

Bits	Last Code Point	Byte 1	Byte 2	Byte 4	Byte 4	Byte 5	Byte 6
7	U+007F	0xxxxxxx					
11	U+07FF	110xxxxx	10xxxxxx				
16	U+FFFF	1110xxxx	10xxxxxx	10xxxxxx			
21	U+1FFFFF	11110xxx	10xxxxxx	10xxxxxx	10xxxxxx		
26	U+3FFFFFFF	111110xx	10xxxxxx	10xxxxxx	10xxxxxx	10xxxxxx	
31	U+7FFFFFFF	1111110x	10xxxxxx	10xxxxxx	10xxxxxx	10xxxxxx	10xxxxxx

UTF-8 II



Code point range	Binary code point	UTF-8 bytes	Example
U+0000 to U+007F	0xxxxxxx	0xxxxxxx	character '\$' = code point U+0024 = 00100100 → 00100100 → hexadecimal 24
U+0080 to U+07FF	00000yyy yyxxxxxx	110yyyyy 10xxxxxx	character 'ç' = code point U+00A2 = 00000000 10100010 → 11000010 10100010 → hexadecimal C2 A2
U+0800 to U+FFFF	zzzzyyyy yyxxxxxx	1110zzzz 10yyyyyy 10xxxxxx	character '€' = code point U+20AC = 00100000 10101100 → 11100010 10000010 10101100 → hexadecimal E2 82 AC
U+010000 to U+10FFFF	000wwwzz zzzzyyyy yyxxxxxx	11110www 10zzzzzz 10yyyyyy 10xxxxxx	character '𐀀' = code point U+024B62 = 00000010 01001011 01100010 → 11110000 10100100 10101101 10100010 → hexadecimal F0 A4 AD A2

Quelle: Wikipedia



- Was versteht man unter der Fano-Bedingung?
- Was zeichnet einen optimalen bzw. effizienten Code aus?
- Welche Vorteile haben Multibyte-Character-Codierungen gegenüber Single-Byte-CC, und welche Zeichencodierungen sind Ihnen bisher begegnet?