Lösungen Kapiteltests

Kapitel 1

1. (K1) Der 3-Kreis-Hammingcode braucht weniger Sendeaufwand.

2. (K2) Wir wissen (vom vorletzten Unterkapitel): Der *n*-Bit-Repetitionscode kann $\frac{n-1}{2}$ Fehler korrigieren (sofern *n* ungerade ist) .
   Also kann der 7-Bit-Repetitionscode 3 Fehler korrigieren. Passieren jedoch 4 oder mehr Fehler, dann überwiegt die Anzahl der invertierten Bits und es wird falsch decodiert.
   Es müssen also 4 (oder mehr) Fehler auftreten, damit falsch decodiert wird.

3. (K2) Der 3-Kreis-Hamming-Code kann einen Fehler korrigieren. Passieren jedoch zwei oder mehr Fehler, ist die Fehlerkorrektur nicht mehr gewährleistet.
   Es müssen also zwei (oder mehr) Fehler auftreten, damit falsch decodiert wird.

4. (K2) Die Codierung lautet 0000000 (7 Nullen).

5a). (K3) Unsere Ereignisraum beinhaltet alle 3-Tupel bestehend aus *R* und *F*. (*R* = das entsprechende Bit wurde richtig gesendet, *F* = das entsprechende Bit wurde fehlerhaft gesendet). Betrachten wir ein Tupel *t* und bezeichnen seine Anzahl *R* mit *r* und seine Anzahl *F* mit *f*. Es gilt:
$Prob(t) = Prob(\text{ein Bit wird richtig gesendet})^r \cdot Prob(\text{ein Bit wird fehlerhaft gesendet})^f$
$= 0.8^r \cdot 0.2^f$

Die "guten Fälle" sind nun alle Übertragungen in denen höchstens ein Fehler passiert – sprich: Die Tupel, die höchstens ein *F* enthalten.
Es gilt:

- $Prob(\text{kein Fehler}) = Prob(RRR) = 0.8^3 = 0.512$
- Wir bezeichnen mit t_i das Tupel, welches an der Stelle *i* ein *F* und an allen andern Stellen ein *R* hat. Es gilt:

$Prob(t_i) = 0.8^2 \cdot 0.2 = 0.128$
Es gibt nun 3 Möglichkeiten, die Position für das *F* zu wählen. Also
$Prob(\text{genau ein Fehler}) = 3 \cdot 0.128 = 0.384$

Daraus schliessen wir:
***Prob*(richtig decodiert) = *Prob*(kein Fehler) + *Prob*(genau ein Fehler)**
= 0.512 + 0.384
= 0.896

5b) (K3) Unser Ereignisraum beinhaltet nun alle 2^5 5-Tupel bestehend aus *R* und *F*. (Die Bedeutung von *R* und *F* ist nicht mehr erwähnenswert.)

Betrachten wir ein Tupel t und bezeichnen seine Anzahl R mit r und seine Anzahl F mit f.
Es gilt:
$Prob(t) = 0.8^r \cdot 0.2^f$
Der 5-Bit-Repetitionscode kann 2 Fehler korrigieren; passieren jedoch mehr als 3 Fehler, wird falsch decodiert.
Die "guten Fälle" sind deshalb alle Uebertragungen bei denen höchstens 2 Fehler passieren, bzw. alle Tupel, die höchstens zwei F besitzen.
Es gilt:

- $Prob$(kein Fehler) = Prob(RRRRR) = $0.8^5 = 0.32768$

- Wir bezeichnen mit t_i das Tupel, welches an der Stelle i ein F und an allen andern Stellen ein R hat. Es gilt:
  $Prob(t_i) = 0.8^4 \cdot 0.2 = 0.08192$
  Es gibt nun 5 Möglichkeiten, die Position für das F zu wählen. Also
  $Prob$(genau ein Fehler) = $5 \cdot 0.08192 = 0.4096$
  Wir bezeichnen mit t_{jk} das Tupel, welches an den Stelle j und k ein F und an allen andern Stellen ein R hat. Es gilt:
  $Prob(t_{jk}) = 0.8^3 \cdot 0.2^2 = 0.02048$
  Es gibt nun $\binom{5}{2} = 10$ Möglichkeiten, diese zwei Stellen zu wählen. Also
  $Prob$(genau zwei Fehler) = $10 \cdot 0.02048 = 0.2048$

  Zu guter Letzt:
  ***Prob*(richtig decodiert)**
  = *Prob*(kein Fehler) + *Prob*(genau ein Fehler) + *Prob*(genau zwei Fehler)
  = 0. 32768 + 0.4096 + 0.2048
  = 0.94208 (gerundet)

5c) (K2) Aus a) und b) wissen wir:
Die Wahrscheinlichkeit, dass eine 1-Bit-Nachricht im 3-Bit-Repetitionscode richtig decodiert wird, beträgt 0.896

- Die Wahrscheinlichkeit, dass eine 1-Bit-Nachricht im 5-Bit-Repetitionscode richtig decodiert wird, beträgt 0.94208

Damit die Mitteilung (bestehend aus 4 Bits) richtig decodiert wird, müssen alle 4 1-Bit-Mitteilungen richtig decodiert werden. Die Wahrscheinlichkeit dafür beträgt
***Prob*(Mitteilung richtig decodiert) = $0.896^2 \cdot 0.94208^2$ = 0.71** (gerundet)

6. (K3) Wir berechnen zuerst die Wahrscheinlichkeit, dass ein 7-Bit-Block (= Codierung einer 4-Bit-Mitteilung) richtig decodiert wird. Wir arbeiten dazu wieder mit dem altbekannten Ereignisraum – er enthält diesmal die 2^7 7-Bit-Tupeln bestehend aus R und F. Die "guten Fälle" sind alle Tupel, die höchstens 1 F enthalten. Es gilt:

- $Prob$(kein Fehler) = $Prob(RRRRRRR) = 0.8^7 = 0.2097$
- Mit t_i bezeichnen wir das Tupel, welches an der Stelle i ein F und an allen andern Stellen ein R hat. Es gilt:
  $Prob(t_i) = 0.8^6 \cdot 0.2$

Es gibt nun 7 Möglichkeiten, die Stelle für das F zu wählen. Also:
$Prob$(es passiert genau ein Fehler) = $7 \cdot Prob(t_i) = 7 \cdot 0.8^6 \cdot 0.2 = 0.3670$

Und daraus können wir schliessen:
***Prob*(richtig decodiert) = *Prob*(kein Fehler) + *Prob*(genau ein Fehler)**
= 0.2097+ 0.3670= 0.58

Nun möchten wir jedoch nicht nur eine Nachricht à 4 Bits, sondern gleich zwei davon schicken. Wir möchten die Wahrscheinlichkeit berechnen, dass beide richtig decodiert werden (also die erste UND die zweite). Es gilt also
***Prob*(beide Nachrichten richtig decodiert) = *Prob*(eine Nachricht richtig decodiert)^2**
= 0.33

Kapitel 2

1. (K2) Der Code kann einen Fehler korrigieren (der Minimalabstand beträgt 3).

2a) (K2) Der Minimalabstand in diesem Code beträgt 3. Er kann daher einen Fehler korrigieren. Es müssen also mindestens 2 Fehler passieren, damit falsch decodiert wird.

2b) (K3) Bei der folgenden (fehlerhaften) Uebertragung führen zwei Fehler zu einer falschen Decodierung:

$w_{ges} = 010000 \quad \rightarrow \quad w_{empf} = 010011 \rightarrow \quad w_{dec} = 011011$

3) (K2) Einfach so: $C = \{0000000, 1111111\}$.

4) (K2) Dieser Code hat Minimalgewicht 3. Er kann deshalb einen Fehler korrigieren.

5) (K3) Wir müssen lediglich zeigen, dass die Summe von zwei Bitsequenzen mit geradem Gewicht selbst wieder gerades Gewicht hat. Wir nehmen zwei (n-Bit)-Sequenzen mit geradem Gewicht und nennen sie v_1 und v_2. Mit w_1 und w_2 bezeichnen wir ihre Gewichte. Den Abstand zwischen v_1 und v_2 (sprich die Anzahl Stellen, an denen sie sich unterscheiden) bezeichnen wir mit r.
Es gilt:

- Es gibt $w_1 - r$ Stellen, an denen v_1 eine '1' hat, und v_2 eine '0'.
- Es gibt $w_2 - r$ Stellen, an denen v_2 eine '1' hat und v_1 eine '1'.

v_1 und v_2 unterscheiden sich also an $(w_1 - r) + (w_2 - r) = w_1 + w_2 - 2r$ Stellen.
Zur Erinnerung: das Gewicht der Summe von zwei Codewörtern entspricht genau deren Abstand.
$v_1 + v_2$ hat also Gewicht $w_1 + w_2 - 2r$ und das ist gerade!