Technische Universität Berlin Sommersemester 06
Dr. S. Pauli
M. Wagner
www.math.tu-berlin.de/~kant/Codierungstheorie

2. Übung Codierungstheorie

1. Aufgabe **ISBN-Code** (4 Punkte)

Mit dem ISBN-Code (=**I**nternational **S**tandard **B**ook **N**umber) werden Bücher von den Verlagen mit Codeworten der Länge 10 über dem Alphabet $A = \{0, 1, \ldots, 9, X\}$ versehen, wobei X für die Zahl 10 steht. Dabei sind die ersten 9 Ziffern aus $A \backslash \{X\}$. Die letzte Ziffer ist eine Kontrollziffer und berechnet sich durch

$$\sum_{i=1}^{10} (11 - i) \cdot a_i \equiv 0 \mod 11.$$

wenn $a = (a_1, \ldots, a_{10})$ ein Element des ISBN-Codes ist. Zeige, dass der ISBN-Code die Vertauschung zweier beliebiger Ziffern erkennt.

2. Aufgabe **Perfekte Codes** (4 Punkte)

(a) Sei $A = \{0, 1\}$ und $C =$

$$\left\{ (c_1, \ldots, c_7) \mid c_i \in A, \quad \begin{array}{ccccccccc} c_1 & + & c_4 & + & c_6 & + & c_7 & = & 0 \\ c_2 & + & c_4 & + & c_5 & + & c_7 & = & 0 \\ c_3 & + & c_5 & + & c_6 & + & c_7 & = & 0 \end{array} \right\}.$$

Zeige, dass C ein perfekter Code ist.

(b) Sei C ein perfekter Code und $|C| > 1$. Dann ist die Minimaldistanz von C ungerade.

3. Aufgabe **MDS-Codes** (2 Punkte)

(a) Beweise oder widerlege: Jeder MDS-Code über einem endlichen Alphabet ist ein perfekter Code.

(b) Ist jeder perfekter Code ein MDS-Code?

4. Aufgabe **Programmieraufgabe** **(6 Punkte)**

Das Alphabet $A = \{0, 1\}$ sei vorgelegt. Schreibe in KASH3 einen Algorithmus der den Widerholungscode darstellt. Als Eingabe erhält der Algorithmus ein $n \in \mathbb{N}$, den Code $C \subseteq A^n$ und die Anzahl der Wiederholungen l wobei l ein Teiler von n ist. Zuerst soll der Algorithmus den Source codieren. In einem zweiten Teil soll der Algorithmus eine über den Kanal übertragene Nachricht erhalten und decodieren wenn möglich.

Der Algorithmus besteht aus zwei Teilen: Der erste Teil ist der Teilalgorithmus **EncodeRepetition** und der zweite **DecodeRepetition**. Auf der Homegage der Codierungstheorie liegt eine KASH3-Datei mit dem Namen **NoisyChannel**. Dies ist ein Algorithmus der eine Fehlerwahrscheinlichkeit $0 < p < 0.5$ und den Output von **EncodeRepetition** als Input bekommt. Der Output vom Algorithmus **NoisyChannel** soll der Input von Algorithmus **DecodeRepetition** sein.

In der Datei **RepetitionCode** findet ihr einige Vorgaben und Tipps. Der Algorithmus **NoisyChannel** erwartet als Output von **DecodeRepetition** eine Liste und gibt auch wieder eine Liste aus.