10. Übung **Informatik A** **WS 07/08**

Klaus Kriegel **Abgabe: 15.01.2008, 12:00 Uhr**

Aufgabe 1: **Minimalabstand** (2 + 4 Punkte)

Einer der bekanntesten nichtbinren Blockcodes ist der sogenannte Triple–Check–Code über dem Alphabet $Q = \{0, 1, 2\}$. Die zugehörige Codierung ist die Abbildung

$$\varphi : Q^3 \longrightarrow Q^6 \quad \text{mit} \quad \varphi(a_1, a_2, a_3) = (a_1, a_2, a_3, q_1, q_2, q_3)$$

wobei die Prüfziffern q_1, q_2, q_3 ähnlich wie Paritätsbits aber modulo 3 bestimmt werden:

$q_1 = (a_2 + a_3) \texttt{ mod } 3$ $\quad$ $q_2 = (a_1 + a_3) \texttt{ mod } 3$ $\quad$ $q_3 = (a_1 + a_2) \texttt{ mod } 3$

a) Bestimmen Sie den Minimalabstand des Codes $C = Im(\varphi)$ und begründen Sie Ihre Antwort.

b) Entscheiden Sie für die folgenden vier 6–Tupel, ob es ein Codewort mit Hammingabstand ≤ 1 gibt, und begründen Sie Ihre Antwort:

$\vec{t} = (2, 2, 1, 2, 0, 0)$, $\vec{u} = (1, 1, 2, 0, 1, 1)$, $\vec{v} = (0, 2, 0, 2, 1, 0)$, $\vec{w} = (1, 2, 2, 0, 2, 1)$

Aufgabe 2: **Suffixcodes** (2 + 2 Punkte)

In der Vorlesung wurden ausschließlich Präfixcodes als Beispiele für eindeutig decodierbare Codes besprochen, aber auch Suffixcodes – das sind Codes bei denen kein Codewort $c(a)$ Suffix (Endstück) eines anderen Codeworts $c(b)$ ist – sind eindeutig decodierbar.

a) Geben Sie eine Begründung dafür, dass Suffixcodes eindeutig decodierbar sind, indem Sie die Konstruktion eines Codebaums und einen Decodierungsalgorithmus beschreiben. (vier bis fünf Sätze sollten reichen, wenn man es auf den Punkt bringt.)

b) Realisieren Sie Ihre Konstruktion für den folgenden Suffixcode

$$C = \{101, 100, 110, 0011, 1001, 0001, 1010, 000, 111, 1011\}$$

wobei die Codewörter in der gegebenen Reihenfolge, die Ziffern $0, 1, \ldots, 9$ codieren und decodieren Sie den String 101110000111010101.

Aufgabe 3: **eindeutige Decodierbarkeit** (4 Punkte)

Untersuchen Sie die folgenden Codes auf eindeutige Decodierbarkeit. Begründen Sie positive Antworten und geben Sie bei negativen Anworten entsprechende Gegenbeispiele an.

$$\begin{aligned} C_1 &= \{01, 10, 0100, 1101\} \\ C_2 &= \{00, 0010, 011, 100, 1111\} \\ C_3 &= \{010, 101, 0101\} \end{aligned}$$

Aufgabe 4: **Listeninduktion** (5 Punkte)

Die Funktionen `average, magic :: Float -> Float -> Float` sind wie folgt definiert:

```
average x y  = (x + y)/2     -- (1)
magic x y    = x + 0.5 * y   -- (2)
```

Beweisen Sie mit vollständiger Induktion, dass für beliebige Float-Listen xs die Aufrufe `2* foldr average 0 xs` und `foldr magic 0 xs` den gleichen Wert berechnen.
Die Zeilennummerierung in der Definition kann zur Begründung der Schritte herangezogen werden.