



HOCHSCHULE KONSTANZ TECHNIK, WIRTSCHAFT UND GESTALTUNG (HTWG)  
**Fakultät Informatik**  
Rechner- und Kommunikationsnetze  
Prof. Dr. Dirk Staehle

# **Labor zur Vorlesung Kommunikationstechnik**

## **Laborübung Quellcodierung**

Prof. Dr. Dirk Staehle

**Bearbeitung in Zweier-Teams**

**Team-Mitglied 1:**

**Team-Mitglied 2:**

## 1 Einleitung

In der Vorlesung wurden verschiedene Verfahren zur Quellcodierung vorgestellt. In dieser Laborübung sollen die Verfahren in Matlab angewandt und verglichen werden.

## 2 Vorbereitung

Machen Sie sich mit den Funktionen zur Huffman-Codierung und arithmetischen Codierung vertraut, die Sie in der Matlab Hilfe für die Toolbox „Communication Systems Toolbox“ unter System Design/Source Coding finden.

## 3 Vergleich mit Vorlesung

In der Theorieübung bzw. in der Vorlesung haben Sie mehrere Zeichenketten nach den Verfahren von Shannon-Fano bzw. Huffman codiert. Implementieren Sie diese Verfahren in Matlab und vergleichen Sie die Ergebnisse mit denen aus der Vorlesung.

Implementieren dazu Sie eine Matlab-Funktion mit einer Zeichenkette als Argument, das

1. die in der Zeichenkette vorkommenden Zeichen und deren Auftrittswahrscheinlichkeiten bestimmt
2. die Entropie einer Quelle mit diesen Auftrittswahrscheinlichkeiten bestimmt
3. die Codierungsvorschrift für diese Auftrittswahrscheinlichkeiten nach Huffman und nach Shannon Fano bestimmt. Nutzen Sie dazu die Funktionen `huffmandict` und `shannonfanodict`. Letztere Funktion wird in Moodle bereitgestellt.
4. die Zeichenketten mit Shannon-Fano-Codierung, Huffman-Codierung und arithmetischer Codierung codiert  
Hinweis: Verwenden Sie zur Codierung nach der Shannnon-Fano-Codierung die Funktionen `huffmanenco` und `huffmandeco` mit der Codierungsvorschrift, die Sie mit der Funktion `shannonfanodict` erstellt haben
5. für die 3 Verfahren jeweils die Länge der Nachricht in Bits sowie die Anzahl der Redundanz-Bits ausgibt.

### Hinweise zu arithmetischer Codierung:

1. Die arithmetische Codierung erfolgt über den Befehl `arithenco(seq, counts)`, wobei `seq` die zu codierende Nachricht ist und `counts` die Anzahl der Zeichen in einem Testkorpus (z.B. der Zeichenkette) ist. Wichtig ist hierbei, dass in `counts` die Anzahl nicht 0 werden darf. Die Symbole in der Nachricht sollten also durchgängig nummeriert sein. HOCHSCHULE z.B. könnte als `seq=[1 2 3 1 4 3 1 5 6 7]` und `counts=[3 1 2 1 1 1 1]` codiert werden. Eine Nachricht (`seq`) mit den ASCII-Codes der Zeichen von Hochschule ist nicht möglich. Sie können die Symbolnummern einfach über den Befehl `ismember` (zweiter Rückgabewert) generieren.

2. Der Befehl `arithenco` liefert keine reelle Zahl aus dem Intervall  $[0,1)$  zurück sondern eine Bit-Folge. Details der Implementierung können gerne im Matlab-Code nachgesehen werden. Der Quellcode kann mit `edit arithenco` geöffnet und auch gedebuggt werden.

## 4 Vergleichsstudie

Wählen Sie nun einen Zeichenvorrat entsprechend der Datei „rfc2795“, die in der ersten Laborübung bereits analysiert wurde.

1. Codierung Sie die Datei nach den Verfahren Huffman, Shannon-Fano und Arithmetische Codierung. Vergleichen Sie die drei Verfahren hinsichtlich der Länge der codierten Datei sowie der Redundanz.
2. Führen Sie eine Studie durch, indem Sie 20 Nachrichten mit jeweils 1000 zufälligen Zeichen aus der Datei mit den drei Verfahren codieren und die mittlere Codewortlänge sowie die Redundanz vergleichen. Ist die Arithmetische Codierung immer mindestens so gut wie Huffman und Huffman immer mindestens so gut wie Shannon-Fano? Sie können zufällige Integerwerte mit der Funktion `randi` erzeugen.