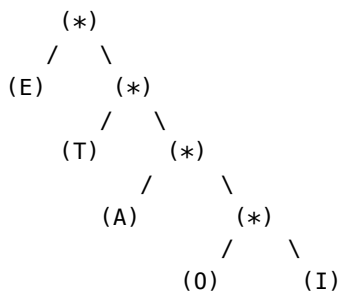


## Digitale AV-Technik - Aufgabenblatt 04

Kodieren und Dekodieren mit Huffman-Bäumen

### Aufgabe 1: Kodieren mit einem gegebenen Huffman-Baum

Gegeben sei der folgende Huffman-Baum:



- Die Buchstaben sind die Blätter des Baums, und die Häufigkeit wurde bereits in diesem Baum berücksichtigt. Nutze die linke Seite des Baums für die 0 und die rechte Seite für die 1.
1. **Erstelle eine Kodierungstabelle**, die jedem Buchstaben den entsprechenden Binärcode zuordnet.

Symbol	Code
E	
T	
A	
O	
I	

2. **Kodieren:** Kodieren Sie mit Hilfe der Kodierungstabelle den Text EAT und notieren Sie die Binärcores.

.....

.....

.....

.....

3. **Dekodieren:** Dekodieren Sie, wenn möglich, die folgenden Binärcores mit dem gegebenen Baum. Ist es für alle Binärcores möglich?

- 011
- 1010
- 00

.....

.....

.....

.....

## Aufgabe 2: Erstellen eines kanonischen Huffman-Baums

Ein kanonischer Huffman-Baum stellt eine optimierte Form des Huffman-Baums dar, bei der der Baum so umsortiert wird, dass Symbole mit den kürzesten Codes nach lexikographischer Ordnung an erster Stelle stehen. Hierbei werden keine Binärbäume benötigt – nur die Länge der Codes ist entscheidend.

Gegeben sind folgende Symbole und ihre Binärcode-Längen:

Symbol	Code-Länge
E	1
T	2
A	3
O	3

1. **Erstelle einen kanonischen Huffman-Code** für jedes der Symbole, wobei E den kürzesten Code und O den längsten Code erhalten soll. Weise dabei dem E die 0 zu und gehe dann zum nächstlängeren Code, also dem T. Erhöhe den bisherigen Code 0 um eins und hänge eine Null hinten dran. Weise diesem Symbol also den Code 10 zu. Die führende 1 sorgt dafür die Präfixeigenschaft zu erhalten. Gehe so weiter mit den verbleibenden Symbolen vor und beachte, dass sie lexikografisch sortiert werden müssen.

Symbol	Code
E	
T	
A	
O	

.....

.....

.....

.....

2. **Kodieren:** Nutze die kanonischen Codes, um das Wort ETA zu kodieren.

.....

.....

.....

.....

3. **Dekodieren:** Lassen sich die folgenden Binärcodes mit dem kanonischen Huffman-Code dekodieren?

Wenn ja, welche Nachricht ergibt sich?

- 011
- 1000
- 01

.....

.....

.....

.....

4. **Zusatzfrage:** Warum sind kanonische Huffman-Codes besonders praktisch in der Datenkompression und Speicherung?

.....

.....

.....

.....

.....

.....

.....

.....

**Aufgabe 3**

Gegeben ist die folgende Codierung einer kanonischen Huffman Codetabelle.

$(\emptyset, 2, 2), ('A', 'C', 'D', 'E')$

Wie lautet die zugehörige Codetabelle?

Symbol	Code
A	
C	
D	
E	

**Aufgabe 4**

Wie sieht die Codierung der folgenden Tabelle aus?

Symbol	Code
A	0
B	10
C	110
D	111

.....

.....

.....

.....

# Lösungen

---

Prüfe die Lösungen und korrigiere sie wenn nötig.

## Lösung Aufgabe 1

### 1. Kodierungstabelle:

Symbol	Code
E	0
T	10
A	110
O	1110
I	1111

### 2. Kodieren des Textes "EAT":

- $E = 0$ ,  $A = 110$ ,  $T = 10 \rightarrow$  **Code: 011010**

### 3. Dekodieren:

- 011  $\rightarrow$  E?  $\rightarrow$  unvollständig, da 11 kein gültiger Code ist.
- 1010  $\rightarrow$  TT
- 00  $\rightarrow$  EE

## Lösung Aufgabe 2

### 1. Kanonischer Huffman-Code:

Symbol	Code-Länge	Kanonischer Code
E	1	0
T	2	10
A	3	110
O	3	111

### 2. Kodieren des Wortes "ETA":

- $E = 0$ ,  $T = 10$ ,  $A = 110 \rightarrow$  **Code: 010110**

### 3. Dekodieren:

- 011  $\rightarrow$  E?  $\rightarrow$  unvollständig, da 11 kein gültiger Code ist.
- 1000  $\rightarrow$  TEE
- 01  $\rightarrow$  E?  $\rightarrow$  unvollständig, da 1 kein gültiger Code ist.

4. **Zusatzantwort:** Kanonische Huffman-Codes sind praktisch, da sie die Decodiertabelle stark vereinfachen. Die Decodierung erfolgt anhand der Reihenfolge und Länge der Codes, wodurch nur die Code-Längen und nicht der vollständige Baum gespeichert werden müssen, was Speicherplatz spart.

## Lösung Aufgabe 3

Um die Codetabelle zu erstellen, analysieren wir die gegebenen Informationen:

Die kanonische Huffman-Codetabelle ist in zwei Teile unterteilt:

1. **Code-Längen:** (0, 2, 2) – das bedeutet:
  - Es gibt **0 Symbole mit einer Code-Länge von 1.**
  - Es gibt **2 Symbole mit einer Code-Länge von 2.**
  - Es gibt **2 Symbole mit einer Code-Länge von 3.**
2. **Symbole:** ('A', 'C', 'D', 'E') – die Symbole, die codiert werden sollen, sind in der Reihenfolge angegeben.

### Schritte zur Erstellung der Codetabelle

1. **Verteilung der Code-Längen auf die Symbole:**
  - Da wir 0 Symbole mit einer Länge von 1 haben, gibt es keine Codes der Länge 1.
  - Die nächsten **2 Symbole** erhalten Codes mit der Länge 2.
  - Die verbleibenden **2 Symbole** erhalten Codes mit der Länge 3.
2. **Kanonische Codes zuweisen:**
  - Da dies ein **kanonischer Huffman-Code** ist, beginnen wir mit dem kürzesten Code und gehen der Reihe nach weiter, wobei wir die Binärwerte inkrementieren.
  - Der erste Code der Länge 2 beginnt bei 00, und wir erhöhen den Wert um 1 für das nächste Symbol.
3. **Kanonische Code-Zuweisung für die Symbole:**
  - **Symbole mit einer Länge von 2:**
    - A erhält 00
    - C erhält 01
  - **Symbole mit einer Länge von 3:**
    - D erhält 100
    - E erhält 101

### Ergebnis: Codetabelle

Symbol	Code
A	00
C	01
D	100
E	101

## Lösung Aufgabe 4

Um die kanonische Codierungstabelle für diese gegebene Huffman-Codierung zu ermitteln, schauen wir uns die Codes und ihre Längen an und wandeln die Tabelle in eine komprimierte Form um, die für kanonische Huffman-Codierungen typisch ist.

### Schritt-für-Schritt-Lösung

#### 1. Bestimme die Code-Längen:

- A hat eine Länge von **1**.
- B hat eine Länge von **2**.
- C hat eine Länge von **3**.
- D hat eine Länge von **3**.

#### 2. Zähle die Anzahl der Codes pro Länge:

- **1 Code** mit Länge **1** ( A ).
- **1 Code** mit Länge **2** ( B ).
- **2 Codes** mit Länge **3** ( C und D ).

#### 3. Erstelle die kanonische Kodierungsangabe:

- Für kanonische Codierungen wird häufig die Anzahl der Codes pro Länge angegeben, gefolgt von der Reihenfolge der Symbole.

Daraus ergibt sich:

(1, 1, 2), ('A', 'B', 'C', 'D')

Hierbei bedeutet:

- (1, 1, 2) : Es gibt **1 Code mit Länge 1**, **1 Code mit Länge 2**, und **2 Codes mit Länge 3**.
- ('A', 'B', 'C', 'D') : Die Reihenfolge der Symbole ist so angeordnet, dass die Kürzesten zuerst kommen, gefolgt von den Symbolen mit den längeren Codes.

## Ergebnis

Die kanonische Kodierungsangabe für die Tabelle lautet:

(1, 1, 2), ('A', 'B', 'C', 'D')