

CODIERUNGSTHEORIE - PRAKTIKA 1

Wombacher Sascha

26. November 2015

1. Generierung des Huffman-Baums
2. Zeichencodierung
3. Rekonstruktion des Huffman-Baums

GENERIERUNG DES HUFFMAN-BAUMS

GENERIERUNG DES HUFFMAN-BAUMS - WAHRSCHEINLICHKEIT EINES ZEICHEN

- für jedes Zeichen z der Eingabezeichenkette wird die Häufigkeit (Anzahl) errechnet
- Anschließend wird diese Anzahl durch die Eingabelänge Normiert (Intervall zwischen $(0, 1]$)

GENERIERUNG DES HUFFMAN-BAUMS - BAUMGENERIERUNG

- Erstelle je ein Node pro Zeichen
- Sortiere alle Nodes aufsteigend anhand ihrer Wahrscheinlichkeit
- Verbinde die Zwei Nodes mit den geringsten Wahrscheinlichkeiten
- Füge das entstehende Node in die Zeichenliste ein (Wahrscheinlichkeiten = Summe der einzel Nodes)
- Stelle die Sortierung wieder her (*insertion sort* - Ansatz)
- Wiederhole Schritt 3-5 bis nur noch ein Node existiert (=> Tree-Root)

GENERIERUNG DES HUFFMAN-BAUMS - SPEICHERUNG

für die Weiterverarbeitung ist das Speichern des Huffman-Baums ein wichtiger Bestandteil

- Ersten 32Bit beschreiben die Anzahl der Zeichen im Baum (Blätter)
- Die Folgenden 8Bit beschreiben die Länge eines Zeichens in Byte
(Bsp.: char = 1, int32 = 4)
- Nehme ein Zeichen aus Huffman-Baum
- Schreibe die gesamte Länge des Zeichen, Codierung in die Folgenden 8Bit
- Schreibe das Zeichen in die Nächsten Bits
- Schreibe die Codierung des Zeichen in die folge Bits
- Wiederhole Schritt 3-6 für jedes Zeichen im Huffman-Baum

ZEICHENCODIERUNG

- Generiere pro Zeichen die jeweilige Codierung (von Root gesehen: leftNode = 1, rightNode = 0)
- Lese ein Zeichen z des Input-Strings
- Finde die Codierung für z
- Füge die Codierung dem BitWriter hinzu
- Wiederhole Schritte 2-4 für alle Zeichen des Strings
- Flush für den BitWriter

REKUNSTRUKTION DES HUFFMAN-BAUMS

.

content...

VIELEN DANK FÜR DIE AUFMERKSAMKEIT!
