

Blatt 2

(Abgabe am 12.5.2011)

Aufgabe 1 (Arithmetische Kodierung)

Hinweis: Bei dieser Aufgabe kann es hilfreich sein, eine erweiterte Implementation ihrer arithmetischen Kodierung von Blatt 1 zu verwenden.

Betrachten Sie folgendes Wahrscheinlichkeitsmodell für eine Quelle mit drei Buchstaben a, b, c : $P(a) = 0,2$ und $P(b) = 0,3$ und $P(c) = 0,5$.

1. Bestimmen Sie den reelwertigen Stellvertreter des Wortes $aacbca$.
2. Decodieren Sie ein Wort der Länge 10 mit dem Stellvertreter $0,63215699$.
3. Geben Sie für beide Kodierungen der ersten beiden Aufgabenteile die Binärcodes an.

Aufgabe 2 (Arithmetische Kodierung)

Betrachten Sie die arithmetische Kodierung. Das Problem bei einer praktischen Implementierung sind die Fließkommaoperationen bei der Berechnung der Intervallgrenzen und des Stellvertreters. Um zu große Fehler zu vermeiden, wird die Skalierungstechnik eingesetzt.

Geben Sie ein Beispiel für ein Alphabet Σ , eine Wahrscheinlichkeitsverteilung P auf Σ und Folgen von Eingabesymbolen an, so dass sich bei der arithmetischen Kodierung mit Skalierung die Kodierung erst mit dem letzten Eingabesymbol feststellt.

Aufgabe 3 (Move to Front)

Die *Move to Front*-Transformation funktioniert folgenderweise: Die Eingabelemente werden in einer Liste gehalten, welche zu Beginn eine vorgegebene Reihenfolge hat. Wird nun ein Zeichen eingelesen, so ist seine Kodierung sein Index in der Liste. Gleichzeitig wird das Zeichen an die erste Stelle der Liste gerückt.

1. Implementieren Sie die *Move to Front*-Transformation, geben sie direkt den Index aus.
2. Warum erreicht man durch das Verfahren in 1 keine Kompression? Unter welchen Umständen ist die *Move to Front*-Transformation sinnvoll? Wie läßt sich die *Move-to-Front*-Transformation ausnutzen, um bessere Kompressionsraten zu erreichen?
Überlegen Sie sich einen einfachen Code für die Indizes, der es unter günstigen Umständen erlaubt, eine Datenkompression durch die *Move to Front*-Transformation zu erreichen.
3. Implementieren Sie Ihre Lösung aus Teil 2. Um die Programmierung zu vereinfachen, können Sie die „Binärausgabe“ durch Ausgeben von 0 und 1 simulieren. Geben Sie ein Beispiel an, bei dem Sie eine Datenkompression erreichen.

Aufgabe 4 (Shannon-Kodierung)

Implementieren Sie Routinen zur Kodierung und Dekodierung mittels der Shannon-Kodierung. Zur Vereinfachung der Implementierung dürfen Sie die Binärausgabe als 0,1-String wie in Aufgabe 3 kodieren.

Allgemeine Hinweise zu den Programmieraufgaben

Ist bei einer Aufgabe die Implementation eines Kodier- bzw. Dekodieralgorithmus gefragt, so ist automatisch auch das entsprechende Pendant gefragt. Das hilft Ihnen auch beim Entwickeln Ihrer Lösungen, da Sie gleich testen können ob Ihre Ergebnisse Sinn machen.