8. Aufgabenblatt vom Mittwoch, den 29. Juli 2009 zur Vorlesung

# Proinformatik

Marco Block

Mittwoch, den 29. Juli 2009

1. **lookupTable** In der Vorlesung blieb die Implementierung der Funktion `lookupTable` offen. Schreiben Sie die Funktion.

2. **frequencyMerge** In der Vorlesung blieb die Implementierung der Funktion `frequencyMerge` offen. Implementieren Sie die Funktion

3. **Von Hand** Vollziehen Sie von Hand die Berechnungen von `makeTree [('b', 2), ('a', 2),('t',3),('e',4)]`. Was ist das Codewort für „beat“?

4. **Suffixcodes** In der Vorlesung haben sie gelernt, was ein optimaler Präfixcode ist. Ein Suffixcode ist ein Code, bei dem keine zwei Codewörter den gleichen Suffix haben (im Präfixcode fangen sie unterschiedlich an, im Suffixcode hören sie unterschiedlich auf).

   Ein optimaler Suffixcode hat, wie ein optimaler Präfixcode, eine minimale mittlere Codewortlänge. Sie ist definiert als

   $$\sum_i p_i * |c(w_i)| \text{ mit } p_i = \text{Wahrscheinlichkeit von } w_i$$

   Überlegen sich sich einen guten Algorithmus, um aus einer Liste von $[(w_1, p_1), \ldots, (w_n, p_n)]$ einen optimalen Suffixcode $[(c_1, w_1), \ldots, (c_n, w_n)]$ zu konstruieren.

   Sie dürfen alles als gegeben betrachten, was sie bisher in der Vorlesung gelernt haben.

   (Dies war eine Klausuraufgabe)

5. **Suchbäume** In der Vorlesung haben wir uns überlegt, dass es die Effizienz verbessert, wenn man in den Knoten eines Suchbaums die Größe des Teilbaums mit ihnen als Wurzel speichert. Natürlich muss man diese Zahlen beim Einfügen und Löschen anpassen. Schreiben Sie eine geeignete Funktion `einfuegenWert`.