Klausurrelevanter Stoff – Informatik A:

Boolesche Terme/Funktionen

Boolesche Gesetze

anwenden können..

Boolesche Funktionen

- **Wertetabelle**
- **DNF/KNF**
- Vereinfachungen der obigen
- **Bsp. Für n-stellige: PARITÄT!**

Schaltnetze

- **Transformation: DNF/Boolscher Term ⇔ Schaltnetz**
- Schaltsymbole nach US-Norm

Haskell

Listenfunktionen

map, filter, zip, unzip, concat, fold
Listcomprehension: [functionVon x | x <- ausListe, prädikat x]
Unendliche Listen

Induktionen

Über Listenlänge: Induktionsanfang xs=[] oder xs=[x] (je nach Funktion)

- ➔ Induktionsvorraussetzung: $\exists n \forall xs : length(xs) \leq n$ gilt:
- ➔ Induktionsbehauptung: p (x:xs) = q (x:xs)
- ➔ Umformen (links -> rechts, umgekehrt od. beides zum gleichen Ausdruck)
- ➔ Dann Behauptung wahr.

Definitionen von Teilfunktionen verwenden und angeben!

Argebraische Datentypen:

- **Konstruktor (unär, binär...)**
- **Rekursise Def.: Bäume**
- **Anwendung von Funktionen darauf**
  (Bsp.: anteil Mix x y z = (anteil x)+(anteil y)+(anteil z)
- Instanziierung (Eq; Ord) durch Definition ein spezifischen compare-Funktion

Codierungstheorie

Stichwörter: Warum? Informationsrate/Kompression; Fehlerkorrektur; Abbildung in maschienentaugliche Form

Blockcodes

- Ersetze Zeichen eines Alphabets A durch Strings *S* aus {0,1}*, wobei |*S*|=ceiling(log|A|) nach beliebiger Codierungsvorschrift $c:A \rightarrow \{0,1\}^*$.
- Vorteil: Schnelle Zuordnung, Simple Speicherung der De-/Codierungsliste als Tabelle, einfache Fehlerkorrektur (Parität, Verdopplung etc.)
- Nachteil: Evtl. große Wortlängen, da keine Berücksichtigung der Häufigkeit der Buchstaben

Präfixcodes

- Sind Codes, deren Codewörter verschiedene Länge haben nicht Präfix eines anderen Codeworts sind.
- Vorteil: Kompression, da Berücksichtigung der Zeichenhäufigkeit
- Speicherung im Codebaum, dabei Codewort für Buchstabe *x* ist Weg von *r* zu Blatt *x* (Kanten mit 0,1 markiert).
- **Bsp:Huffman**
  Gegeben: Alphabet A mit zugehöriger Wahrscheinlichkeits-Abbildung $p:A\rightarrow[0,1]$ (wie oft kommt Buchstabe *a* aus *A* in einem Text *T* vor) evtl. Abgabe auch in Prozent (p*100%).
  Vorgehen: Sortiere Zeichen aufsteigend entsprechend *p*. Ersetze die beiden kleinsten Zeichen x und *y* durch einen Binärbaum mit Blättern *x* und *y* und schreibe an die Wurzel *r* gemeinsame Wahrscheinlichkeit: $p(r)=p(x)+p(y)$. Die Kanten werden mit 0 und 1 kodiert. Und von vorn..

Bäume

Ein Baum ist allgemein ein verbundener, azyklischer und ungerichteter Graph, also eine Menge von Knoten, die Elemente enthalten (können), in einer Vater-Kind-Beziehung mit folgenden Eigenschaften:

T hat einen ausgezeichneten Knoten r genannt Wurzel.
Jeder Knoten v ausser r hat einen Vater u.

Bezeichnungen:

Höhe eines Baums: max{TiefeBlätter} (=> Blatt hat Höhe 0!)
Tiefe eines Knotens: #Knoten auf dem Weg von *r* zu diesem Knoten (=> Wurzel hat Tiefe 0!)
Gewicht einen (Teil-)Baumes: #Knoten (Baum nur mit Wurzel => Gewicht 1)

Eigenschaften:

Binärbäume: balanciert: Jeder Teilbaum unter Knoten k hat gleiches Gewicht
#innereKnoten <= #Blätter
Höhe der Wurzel <= log#Blätter
PreOrderTraverse: Besuche erst Vater, dann linkes Kind und dann rechtes Kind.
InOrderTraverse: Besuche erst linkes Kind, dann Vater und dann rechtes Kind.
PostOrderTraverse: Besuche erst linkes Kind, dann rechtes Kind und dann Vater.