Karlsruher Institut für Technologie

Tutorium 42, #4

Max Göckel– *uzkns@student.kit.edu*

Institut für Theoretische Informatik - Grundbegriffe der Informatik

Übungsblatt häufige Fehler

1.3.: *"Geben sie [...] möglichst genaue untere und obere Schranken an."*

- Möglichst genaue Schranke heißt:
  - Für obere Schranken der kleinste Wert der über der Kardinalität liegt
  - Für untere Schranken der größte wert der unter der Kardinalität liegt

1.4.: *"Geben Sie eine injektive Abbildung g: B $\rightarrow$ A an."*

- A, B bel., d.h. konkrete Mengen wie $\{\mathbb{N}, \{1, 2, 3, 4\}\}$ sind nicht erlaubt.
- So allgemein wie möglich, am besten mit bel. n, m

1.5.: *"Beweisen oder widerlegen Sie..."*

- Beweis = Mathematischer Beweis zB mit Äquivalenzumformungen oder *Sei x $\in$ A bel.*
- Begründung, Schaubild reichen nicht aus
- Gegenbeweis immer erst mit Gegenbeispiel probieren (spart Zeit & Platz)

Fakultät für Informatik
Institut für Theoretische Informatik

Num_k

Wenn w ein Wort ist, das eine Zahl zur Basis k darstellt, so ist $Num_k(w)$ die Darstellung dieses Wortes/dieser Zahl im Dezimalsystem.

- Definition der Funktion mit w=w'·x.
  - $Num_k(\epsilon) = 0$
  - $Num_k(w'x) = k \cdot Num_k(w') + num_k(x)$ mit $x \in \mathbb{Z}_k$ und $w' \in \mathbb{Z}_k^*$
  - $num_k(x) = x$

Num_k: Beispiele

Sei w_1=5 in der Basis 8

- $Num_8(5) \Leftrightarrow Num_8(\epsilon) + num_8(5) \Leftrightarrow 5$

Sei w_2=234 in der Basis 9

- $Num_9(234) \Leftrightarrow Num_9(23) + num_9(4) \Leftrightarrow Num_9(2) + num_9(3) + num_9(4) \Leftrightarrow Num_9(\epsilon) + num_9(2) + num_9(3) + num_9(4) = 234$

Fakultät für Informatik
Institut für Theoretische Informatik

Num_k: **Aufgabe**

Sei w_2=B66 in der Basis 16

- Gib die Repräsentation der Zahl im Dezimal mithilfe der Num_k-Funktion an.

Fakultät für Informatik
Institut für Theoretische Informatik

Num_k: Lösung

Sei w_2=B66 in der Basis 16

- $Num_{16}(B66) \Leftrightarrow Num_{16}(B6) + num_{16}(6) \Leftrightarrow$
  $Num_{16}(B) + num_1 6(6) + num_{16}(6) \Leftrightarrow$
  $Num_{16}(\epsilon) + num_{16}(B) + num_{16}(6) + num_{16}(6) \Leftrightarrow$ 1166

$Repr_k$

$Repr_k$ ist die Darstellung einer Dezimalzahl in der Basis k.

- Definition:
  - n < k: $repr_k(x)$
  - n $\geq$ k: $Repr_k(xdivk) \cdot repr_k(nmodk)$

Fakultät für Informatik
Institut für Theoretische Informatik

$Repr_k$: Beispiel

Mit w=29 und k=3.

- $Repr_3(29) \Leftrightarrow Repr_3(29d3) \cdot repr_3(29m3) \Leftrightarrow Repr_3(9)2 \Leftrightarrow Repr_3(9d3) \cdot repr_3(9m3)2 \Leftrightarrow Repr_3(3)02 \Leftrightarrow Repr_3(1)repr_3(0)02 \Leftrightarrow repr_3(1)002 \Leftrightarrow 1002$

$Repr_k$: Aufgabe

Karlsruher Institut für Technologie

w=53 und k=5.

- Stelle w mithilfe der $Repr_k$-Funktion dar.

$Repr_k$: **Lösung**

w=53 und k=5.

- $Repr_5(10) \cdot 3 \Leftrightarrow Repr_5(2) \cdot 03 \Leftrightarrow 203$

Zweierkomplement

Zweierkomplementdarstellung ($\mathbb{K}_n$) ist die übliche Darstellung von ganzen Zahlen im PC.
Zahlen werden gleichmäßig auf den positiven und negativen Bereich verteilt. Mit der 0 ist im positiven Bereich eine Zahl weniger als im negativen.

Definition

- $\mathbb{K}_n = \{x \in \mathbb{Z} | -2^{n-1} \leq x \geq 2^{n-1} - 1\}$

- $\mathbb{K}_7 = \{-64, -63, ..., ..., 62, 63\}$

$Zkpl_k$

Umrechung eriner n-stelligen Binärzahl ins Zweierkomplement mit $Zkpl_n$.

Vorgehen bei negativen Zahlen:

- Nullen durch Einsen erstzen und Einsen durch Nullen (Invertieren)
- 1 dazu addieren
- Schauen ob die Zahl vorher positiv oder negativ war, entsprechend das erste Bit anpassen

Bei positiven Zahlen:

- Nichts tun, evtl. auf richtige Länge mit 0'len auffüllen

$Zkpl_k$: Beispiel

Beispielzahl ist -01101011_2

Vorgehen:

- Invertieren: 10010100
- +1: 10010101

$Zkpl_k$: Aufgaben

Rechnet die fogenden Zahlen mit Zwischenschritten um:

- 5 zuerst in binär, denn in ZK_4
- -100101_2 in ZK_9
- 11110_{ZK} zurück in binär

Fakultät für Informatik
Institut für Theoretische Informatik

$Zkpl_k$: Lösung

Rechnet die fogenden Zahlen mit Zwischenschritten um:

- $5 = 101_2 = 0101_{ZK}$
- $-100101_2 = 111011011_{ZK}$
- $11110_{ZK} = -10_2$

Homomorphismen

A,B so bildet eine Abbildung $h : A \rightarrow B^*$ einen Buchstaben von A uf ein Wort aus B ab.
ein Homorphismus $h^{**} : A^* \rightarrow B^*$ macht das selbe mit einem ganzen Wort aus A, wobei jedes Zeichen einzeln abgebildet wird.
Ist ein Homomorphismus präfixfrei so existiert eine Umkehrfunktion die aus dem Wort aus B* wider das Wort in A* macht

Definition

- $\forall w \in A^* : \forall x \in A : h^{**}(wx) = h^{**}(w) \cdot h(x)$

Homomorphismen: Beispiel

Sei $h(a) = 101, H(b) = 0, h(c) = 1$

- $h^{**}(a) = h(a) = 101$
- $h^{**}(cbc) = h(c) \cdot h(b) \cdot h(c) = 101$

Huffman-Codierung

Die Huffman-Codierung ist ein präfixfreier Homomorphismus $A^* \to \mathbb{Z}_2{}^*$, der häufigen Zeichen eine möglichst kurze Abbildung gibt.

Sei $N_w(x)$ die Anzahl der Vorkommnisse des Buchstaben x in w und "$N_w(x), x$" die Beschriftung des Blattes in einem Baum

- Verbinde die zwei "kleinsten" Werte
- Schreibe $N_w(x_1) + N_w(x_2)$ in den neuen Knoten
- So lange wiederholen, bis der Baum komplett ist

Huffman: Beispiel

w = bcccabdd, so sind $N_w(a) = 1$, $N_w(b) = 2$, $N_w(c) = 3$, $N_w(d) = 2$ die Häufigkeiten in w.

Huffman: Aufgabe

Stelle das folgende Wort in einem Huffman-Baum dar und erstelle das Codewort:

w = ededddbcfbedaccb

Huffman: Frage

Karlsruher Institut für Technologie

Ist ein Huffman-Baum eindeutig?

-

Kann man ein Huffman-Wort wieder decodieren?

-

Huffman: Antwort

Ist ein Huffman-Baum eindeutig?

- Nein, da bei mehreren gleich-häufig vorkommenden Wörern man sich aussuchen kann, welche Kante man zuerst verbindet.

Kann man ein Huffman-Wort wieder decodieren?

- Ja, dafür benötigt man aber den Baum.

Huffman-Codierung

Kommen viele Teilwörter in einem Wort w häufig vor, so macht es Sinn die Codierung mit kurzen Wörtern anstelle von einzelnen Buchtaben durchzuführen.
Ansonsten bleibt das Vorgehen gleich, die Suche nach den besten Teilwörter n kann aber etwas Zeit in Anspruch nehmen.

Beispielwort: $w = abcabccbdabc = abc \cdot abc \cdot cbd \cdot abc$

- h(abc) = 0, h(cbd) = 1
- $h^{**}(w) = 0010$
- $|w| = 12, |h^{**}(w)| = 4$, Platzeinsparung um den Faktor 3