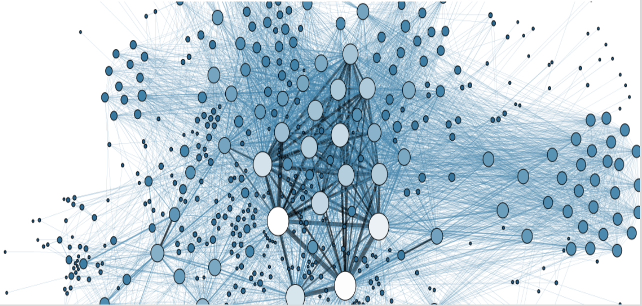
Grundbegriffe der Informatik
Tutorium 38

Übersetzungen/Kodierungen, Zweierkomplement, Huffman-Codierung

Patrick Fetzer, `uxkln@student.kit.edu` | 15.11.2018



Häufige Fehler

- Wahrheitstabellen beginnen mit fff und “zählen“ dann hoch
- fff, ffw, fwf, fww, wff, wfw, wwf, www (wie Binärzahlen)
- $B_1 \equiv B_2$ um zu zeigen, dass zwei Ausdrücke für jede Belegung gleichen Wahrheitswert haben
- $(a \wedge b) \vee b \equiv b$
- $f : \mathbb{N}_0 \times \mathbb{N}_0 \to \mathbb{N}_0$ Zeige etwas für alle $n \in \mathbb{N}_0$
- $f(n+1, n+1) = 2f(n, n) + f(n+1, n-1) + f(n-1, n+1)$
- Wie muss Induktionsanfang/voraussetzung lauten?
- IA für $n = 0$ und $n = 1$
- IV: Behauptung gelte für ein $n \in N_+$

Herführung zu Zahlendarstellungen

Wir betrachten die Alphabete $A_{dez} := \mathbb{Z}_{10}, A_{bin} := \{0, 1\}, A_{oct} := \mathbb{Z}_8$.

- Was können wir daraus machen?
- $A^*_{dez} \supset \{42, 1337, 999\}$.
- $A^*_{bin} \supset \{101010, 10100111001, 1111100111\}$.
- $A^*_{oct} \supset \{52, 2471, 1747\}$.
- Wir suchen eine Möglichkeit, diese Zahlen zu deuten.
- Aber irgendwie so, dass $42_{\in A_{dez}} \overset{Deutung}{=} 101010_{\in A_{bin}} \overset{Deutung}{=} 52_{\in A_{oct}}$.

Definition von Zahlendarstellungen

Karlsruher Institut für Technologie

Num_k

Einer Zeichenkette Z_k aus Ziffern wird mit Num_k eine eindeutige Zahl zugeordnet:

$$Num_k(\varepsilon) = 0$$

$Num_k(wx) = k \cdot Num_k(w) + num_k(x)$ mit $w \in Z_k^*$ und $x \in Z_k$.

num_k

Einer einzelnen Ziffer $x \in Z_k$ aus einem Alphabet von Ziffern Z_k wird mit $num_k(x)$ der Wert der Zahl zugewiesen.

- Wichtig: $Num_k(w) \neq num_k(w)$!
- Was ist: $num_{10}(3) = 3$, $num_{10}(7) = 7$, $num_{10}(11) =$ nicht definiert.
- Für Zahlen $\geq k$: Benutze Num_k!

Beispiel zu Zahlendarstellungen

$Num_k(\varepsilon) = 0.$
$Num_k(wx) = k \cdot Num_k(w) + num_k(x)$ mit $w \in Z_k^*$ und $x \in Z_k$.

Was ist $Num_{10}(123)$?

- $Num_{10}(123) = 10 \cdot Num_{10}(12) + num_{10}(3) =$
  $10 \cdot (Num_{10}(1) + num_{10}(2)) + num_{10}(3) =$
  $10 \cdot (num_{10}(1) + 10 \cdot num_{10}(2)) + num_{10}(3) = 10 \cdot 10 \cdot 1 + 10 \cdot 2 + 3 = 123.$

Yay?

Was ist der dezimale Zahlenwert der Binärzahl 1010? Diesmal Basis $k = 2$.

- $Num_2(1010) = 2 \cdot Num_2(101) + num_2(0) =$
  $2 \cdot (2 \cdot Num_2(10) + num_2(1) + num_2(0) =$
  $2 \cdot (2 \cdot (2 \cdot Num_2(1) + num_2(0)) + num_2(1)) + num_2(0) =$
  $2 \cdot (2 \cdot (2 \cdot num_2(1) + num_2(0)) + num_2(1)) + num_2(0) =$
  $2 \cdot (2 \cdot (2 \cdot 1 + 0) + 1) + 0) = 10.$

Yay!

Aufgaben zu Zahlendarstellungen

$Num_k(\varepsilon) = 0$.
$Num_k(wx) = k \cdot Num_k(w) + num_k(x)$ mit $w \in Z_k^*$ und $x \in Z_k$.

Übungen zu Zahlendarstellungen

Berechne den numerischen Wert der folgenden Zahlen anderer Zahlensysteme nach dem vorgestellten Schema:

- $Num_8(345)$.
- $Num_2(11001)$.
- $Num_2(1000)$.
- $Num_4(123)$.
- $Num_{16}(4DF)$. (Zusatz)

Anmerkung: Hexadezimalzahlen sind zur Basis 16 und verwenden als Ziffern (in aufsteigender Reihenfolge:
$1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F$.

Aufgaben zu Zahlendarstellungen

Lösungen:

- $Num_8(345) = 229$.
- $Num_2(11001) = 25$.
- $Num_2(1000) = 8$.
- $Num_4(123) = 27$.
- $Num_{16}(4DF) = 1247$.

Einfachere Umrechnung von Zahlendarstellungen

Es gilt:
$2(2(2(2(2\cdot 1+0)+1)+0)+1)+0 = 2^5\cdot 1+2^4\cdot 0+2^3\cdot 1+2^2\cdot 0+2^1\cdot 1+2^0\cdot 0.$
Daher, einfachere Rechenweise:
$Num_k(w) = k^0 \cdot w(0) + k^1 \cdot w(1) + k^2 \cdot w(2) + ...$
Was sind folgende Zahlen in Dezimal im Kopf gerechnet?

- $Num_2(10101) = 21.$
- $Num_2(11101) = 29.$
- $Num_2(1111111111) = 1023.$

Einfachere Umrechnung von Zahlendarstellungen

$Num_k(w) = k^0 \cdot w(0) + k^1 \cdot w(1) + k^2 \cdot w(2) + ...$

Was sind folgende Zahlen in Dezimal im Kopf gerechnet?

- $Num_{16}(A1) = 161.$
- $Num_{16}(BC) = 188.$
- $Num_{16}(14) = 20.$

- Ist Num injektiv? $Num_2(01) = Num_2(1)$
- Ist Num surjektiv? Ja

Rechnen mit *div* und *mod*

div Funktion

Die Funktion *div* dividiert ganzzahlig. (Schneidet also den Rest ab).

mod Funktion

Die Modulo Funktion *mod* gibt den Rest einer ganzzahligen Division zurück.

- 22 div 8 $= 2$ $\left(\frac{22}{8} = 2,75\right)$.
- 22 mod 8 $= 6$.

Fülle die Tabelle aus:

x	0	1	2	3	4	5	6	7	8	9	10	11	12
x *div* 4	0	0	0	0	1	1	1	1	2	2	2	2	3
x *mod* 4	0	1	2	3	0	1	2	3	0	1	2	3	0

Von Zeichen zu Zahlen zurück zu Zahlen

11101_2 ist also 29_{10}. Was ist 29_{10} in binär?

k-äre Darstellung

Die Repräsentation einer Zahl n zur Basis k lässt sich wie folgt ermitteln:

$$\mathbf{Repr}_k(n) = \begin{cases} \mathbf{repr}_k(n) & \text{falls } n < k \\ \mathbf{Repr}_k(n \operatorname{div} k) \cdot \mathbf{repr}_k(n \bmod k) & \text{falls } n \geq k \end{cases}$$

Achtung! Das $\cdot$ Symbol steht für Konkatenation, nicht für Multiplikation!

Beispiel zu *Repr$_k$*

$$\textbf{Repr}_k(n) = \begin{cases} \textbf{repr}_k(n) & \text{falls } n < k \\ \textbf{Repr}_k(n \operatorname{div} k) \cdot \textbf{repr}_k(n \bmod k) & \text{falls } n \geq k \end{cases}$$

Zum Beispiel:

$$\begin{aligned} \textbf{Repr}_2(29) &= \textbf{Repr}_2(29 \textbf{ div } 2) \cdot \textbf{repr}_2(29 \textbf{ mod } 2) \\ &= \textbf{Repr}_2(14) \cdot \textbf{repr}_2(1) \\ &= \textbf{Repr}_2(14 \textbf{ div } 2) \cdot \textbf{repr}_2(14 \textbf{ mod } 2) \cdot 1 \\ &= \textbf{Repr}_2(7) \cdot \textbf{repr}_2(0) \cdot 1 \\ &= \textbf{Repr}_2(7 \textbf{ div } 2) \cdot \textbf{repr}_2(7 \textbf{ mod } 2) \cdot 01 \\ &= \textbf{Repr}_2(3) \cdot \textbf{repr}(1) \cdot 01 \\ &= \textbf{Repr}_2(3 \textbf{ div } 2) \cdot \textbf{repr}(3 \textbf{ mod } 2) \cdot 101 \\ &= \textbf{Repr}_2(1) \cdot \textbf{repr}(1) \cdot 101 \\ &= 11101 \end{aligned}$$

Beispiel zu *Repr*$_k$

$$\mathbf{Repr}_k(n) = \begin{cases} \mathbf{repr}_k(n) & \text{falls } n < k \\ \mathbf{Repr}_k(n \operatorname{div} k) \cdot \mathbf{repr}_k(n \bmod k) & \text{falls } n \geq k \end{cases}$$

Beispiel mit Hexadezimalzahlen:

$$\begin{aligned} \mathbf{Repr}_{16}(29) &= \mathbf{Repr}_{16}(29 \ \mathbf{div} \ 16) \cdot \mathbf{repr}_{16}(29 \ \mathbf{mod} \ 16) \\ &= \mathbf{Repr}_{16}(1) \cdot \mathbf{repr}_{16}(13) \\ &= 1 \cdot D = 1D \end{aligned}$$

Übung zu $Repr_k$

$$\mathbf{Repr}_k(n) = \begin{cases} \mathbf{repr}_k(n) & \text{falls } n < k \\ \mathbf{Repr}_k(n \text{ div } k) \cdot \mathbf{repr}_k(n \bmod k) & \text{falls } n \geq k \end{cases}$$

Übung zu $Repr_k$

Berechne die Repräsentationen folgender Zahlen in gegebenen Zahlensystemen:

- $\mathbf{Repr}_2(13)$.
- $\mathbf{Repr}_4(15)$.
- $\mathbf{Repr}_{16}(268)$.

- $\mathbf{Repr}_2(13) = 1101$.
- $\mathbf{Repr}_4(15) = 33$.
- $\mathbf{Repr}_{16}(268) = 10C$.

Feste Länge von Binärzahlen

bin_ℓ

Die Funktion $\mathbf{bin}_\ell : \mathbb{Z}_{2^\ell} \to \{0,1\}^\ell$ bringt eine gegebene Binärzahl auf eine feste Länge, indem sie mit Nullen vorne aufgefüllt wird. Formell wird sie definiert als:

$$\mathbf{bin}_\ell(n) = 0^{\ell - |\mathbf{Repr}_2(n)|}\mathbf{Repr}_2(n)$$

Beispiel:

- $\mathbf{bin}_8(3) = 0^{8-|\mathbf{Repr}_2(3)|}\mathbf{Repr}_2(3) = 0^{8-|11|} \cdot 11 = 0^{8-2} \cdot 11 = 0^6 \cdot 11 = 00000011.$
- $\mathbf{bin}_{16}(3) = 0000000000000011.$

Zweierkomplement

Was ist mit negative Zahlen?

- Idee: Verwende das erste Bit, um zu speichern, ob die Zahl positiv oder negativ ist.
- Beispiel: $5 = 0101_{zkpl}$, $-5 = 1011_{zkpl}$.

Zweierkomplement Darstellung

Die Zweierkomplementdarstellung einer Zahl x mit der Länge ℓ ist wie folgt definiert:

$$\mathbf{Zkpl}_\ell(x) = \begin{cases} 0\mathbf{bin}_{\ell-1}(x) & \text{falls } x \geq 0 \\ 1\mathbf{bin}_{\ell-1}(2^{\ell-1} + x) & \text{falls } x < 0 \end{cases}$$

- Wieso $\ell - 1$?

Aufgaben zu Zweierkomplement-Darstellung

$$\mathbf{Zkpl}_\ell(x) = \begin{cases} 0\mathbf{bin}_{\ell-1}(x) & \text{falls } x \geq 0 \\ 1\mathbf{bin}_{\ell-1}(2^{\ell-1} + x) & \text{falls } x < 0 \end{cases}$$

Was sind folgende Zahlen in Zweierkomplement-Darstellung?

- $\mathbf{Zkpl}_4(3) = 0011$.
- $\mathbf{Zkpl}_4(7) = 0111$.
- $\mathbf{Zkpl}_4(-5) = 1011$.
- $\mathbf{Zkpl}_8(13) = 00001101$.
- $\mathbf{Zkpl}_8(-34) = 11011110$.
- $\mathbf{Zkpl}_8(-9) = 11110111$.

Übersetzungen

Definition der Semantikabbildung

Sei *Sem* die Menge der Bedeutungen. Ferner seien A und B Alphabete und $L_A \subseteq A^*$ und $L_B \subseteq B^*$.
Weiter sei $sem_A : L_A \to Sem$ und $sem_B : L_B \to Sem$
Dann heißt $f : L_A \to L_B$ Übersetzung , wenn gilt: für jedes $w \in L_A$ gilt $sem_A(w) = sem_B(f(w))$.

- Bedeutungserhaltende Abbildungen von Wörtern auf Wörter

Beispiel

Betrachte $Trans_{2,16} : \mathbb{Z}_{16}^* \to \mathbb{Z}_2^*$ mit $Trans_{2,16}(w) = Repr_2(Num_{16}(w))$

- $Trans_{2,16}(A3) = Repr_2(Num_{16}(A3)) = Repr_2(163) = 10100011$

Wozu Übersetzungen

- Lesbarkeit (vergleiche DF_{16} mit 11011111_2)
- Verschlüsselung
- Kompression (Informationen platzsparend aufschreiben)
- Kontextabhängige Semantiken (Deutsch $\longrightarrow$ Englisch)
- Fehlererkennung

Codierungen

Definitionen

- Codewort $f(w)$ einer Codierung $f : L_A \rightarrow L_B$
- Code: $\{f(w)|w \in L_A\} = f(L_A)$
- Codierung: **Injektive** Übersetzung
  - Ich komme immer eindeutig von einem Codewort f(w) zu *w* zurück

Bemerkung

- Was ist, wenn L_A unendlich ist (man kann nicht alle Möglichkeiten aufzählen)
- Auswege: Homomorphismen, Block-Codierungen

Homomorphismen

Definition von Homomorphismen

Seien A, B Alphabete. Dann ist $h : A^* \to B^*$ ein Homomorphismus, falls für alle $w_1, w_2 \in A^*$ gilt:

$$h(w_1 w_2) = h(w_1)h(w_2)$$

- Ein Homomorphismus ist Abbildung, die mit Konkatenation verträglich ist
- Homomorphismus ist ε-frei, wenn für jedes $x \in A : h(x) \neq \varepsilon$
- Homomorphismen lassen das leere Wort unverändert, also $h(\varepsilon) = \varepsilon$

Sei h ein Homomorphismus.

Übung zu Homomorphismen

1. $h(a) = 001$ und $h(b) = 1101$. Was ist dann $h(bba)$?

$\rightarrow$ $h(bba) = h(b)h(b)h(a) = 1101 \cdot 1101 \cdot 001 = 11011101001$

2. Sei $h(a) = 01$, $h(b) = 11$ und $h(c) = \varepsilon$. Nun sei $h(w) = 011101$. Was war w?

$\rightarrow$ *aba* oder *cabccac*, ... Allgemein:
$w \in \{c\}^* \cdot \{a\} \cdot \{c\}^* \cdot \{b\} \cdot \{c\}^* \cdot \{a\} \cdot \{c\}^*$
ε-Freiheit hat also die Eindeutigkeit zerstört!

3. Kann h aus 2 eine Codierung sein?

$\rightarrow$ Nein, da nicht injektiv!

4. Warum will man ε-freie Homomorphismen?

$\rightarrow$ Information geht sonst verloren!

5. Was heißt hier “Information geht verloren”?

$\rightarrow$ Es gibt $w_1 \neq w_2$ mit $h(w_1) = h(w_2)$

- Information kann auch anders “verloren” gehen

$\rightarrow$ z.B. $h(a) = 0, h(b) = 1, h(c) = 10$ – Wie das?

Präfixfreiheit

Gegeben ist ein Homomorphismus $h : A^* \rightarrow B^*$.
Wenn für keine zwei verschiedenen $x_1, x_2 \in A$ gilt, dass $h(x_1)$ Präfix von $h(x_2)$ ist, dann ist h präfixfrei.

Satz

Präfixfreie Codes sind injektiv.

Beispiele

- $h(a) = 01$ und $h(b) = 1101$ ist präfixfrei
- $g(a) = 01$ und $g(b) = 011$ ist nicht präfixfrei

Huffman-Codierung

- Komprimiert eine Zeichenkette
- Kodiert häufiger vorkommende Zeichen zu kürzeren Codewörter als Zeichen die seltener vorkommen.
- Vorgehensweise:
  1. Zähle Häufigkeiten aller Zeichen der Zeichenkette
  2. Schreibe alle vorkommenden Zeichen und ihre Häufigkeiten nebeneinander
  3. Wiederhole, bis der Baum fertig ist:
     - Verbinde die zwei Zeichen mit niedrigsten Häufigkeiten zu neuem Knoten über diesen
     - Dieser hat als Zahl die aufsummierte Häufigkeiten
  4. Danach: Alle linken Kanten werden mit 0 kodiert, alle rechten Kanten mit 1

Das Ergebnis ist eine Zeichenkette aus $\{0, 1\}$, die kürzer ist als die ursprüngliche Zeichenkette in binär.

Huffman-Codierung

Gegeben

- $w \in A^*$

  w = `afebfecaffdeddccefbeff`
- Anzahl der Vorkommen aller Zeichen in w ($N_x(w)$)

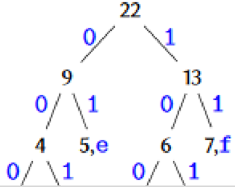
Häufigkeiten:

x	a	b	c	d	e	f
$N_x(w)$	2	2	3	3	5	7

Zwei Phasen zur Bestimmung eines

Huffman-Codes

1. Konstruieren eines "Baumes"
   - Blätter entsprechen den Zeichen
   - Kanten mit 0 und 1 beschriften



Häufigkeiten:

Übung zu Huffman Codierung

Übung

Sei $A = \{\text{a, b, c, d, e, f, g, h}\}$

- Codiere das Wort `badcfehg` mit Hilfe der Huffman-Codierung

$\rightarrow$ Mögliche Lösung: 001 100 010 011 101 000 111 110

- Wie lauten die Codewörter, wenn für das Wort w gilt: $N_a(w) = 1, N_b(w) = 2, N_c(w) = 2, N_d(w) = 8, N_e(w) = 16, N_f(w) = 32, N_g(w) = 64, N_h(w) = 128$

Mögliche Lösung:

x	a	b	c	d	e	f	g	h
h(x)	0000000	0000001	000001	00001	0001	001	01	1

- Wie lang wäre das zweite Wort ($\texttt{abbcccc d}^8\ldots\texttt{g}^{64}\texttt{h}^{128}$) mit dem ersten Code codiert?

$\rightarrow$ 741 Symbole. Also dreimal so lang wie das Original.

- Wie lang wäre das zweite Wort mit dem zweiten Code codiert?

$\rightarrow$ 501 Symbole. Also nur zweimal so lang wie das Original.

- Was fällt euch auf?

Wahr oder falsch?

Sei $h : A^* \to \mathbb{Z}_2$ eine Huffman-Codierung

- h ist ein ε-freier Homomorphismus **Wahr!**
- Häufigere Symbole werden mit langen Worten codiert, seltene mit kürzeren **Falsch!**
- Die Kompression ist am stärksten, wenn die Häufigkeiten aller Zeichen ungefähr gleich sind. **Falsch!**
- h ist präfixfrei **Wahr!**
- Es kann noch kürzere Codierungen geben **Falsch!**

Huffman-Codierung

Karlsruher Institut für Technologie

Eigenschaften

Sei A ein Alphabet und $w \in A$. Dann gilt für die Huffman-Codierung h:

- $h : A^* \to \mathbb{Z}_2$
- h ist ε-freier Homomorphismus
- h ist präfixfreier Homomorphismus
- Häufigere Symbole werden mit kurzen Worten codiert, seltene mit längeren
- Produziert kürzestmögliche Codierungen

Block-Codierung mit Huffman

- Wir betrachten nicht mehr einzelne Symbole, sondern Blöcke von fester Länge $b > 1$
- Blätter des Huffman-Baums sind jetzt *Wörter der Länge b*

Beispiel an der Tafel: Codierung von
$aab \cdot deg \cdot deg \cdot aab \cdot ole \cdot aab \cdot deg \cdot aab$.

- Alphabet $A = \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}\}$
- Text über A, der nur aus Teilwörtern der Länge 10 zusammengesetzt ist, in denen jeweils immer nur ein Symbol vorkommt
- Angenommen $\mathtt{a}^{10}$, ..., $\mathtt{d}^{10}$ kommen alle gleich häufig vor. Wie lang ist dann die Huffman-Codierung?

$\rightarrow$ Ein Fünftel, weil jeder Zehnerblock durch zwei Bits codiert wird

Wahr oder falsch?

Sei $h : A^* \to \mathbb{Z}_2$ eine Huffman-Codierung

- h ist ein ε-freier Homomorphismus **Wahr!**
- Häufigere Symbole werden mit langen Worten codiert, seltene mit kürzeren **Falsch!**
- Die Kompression ist am stärksten, wenn die Häufigkeiten aller Zeichen ungefähr gleich sind. **Falsch!**
- h ist präfixfrei **Wahr!**
- Es kann noch kürzere Codierungen geben **Falsch!**


That's all Folks!