Zahlendarstellung, Stellenwertsystem

Unterscheide zwischen

- dem *Zahlenwert* selbst („Zweiundzwanzig“) und
- der *Darstellung* (z.B. 22_{10} aber auch 10110_2 oder gar XXII).

Wir verwenden das *Stellenwertsystem* bezüglich einer *Basis* b.

- Darstellung als geordnete Reihung der Länge n von Ziffern
- Mögliche Werte für Ziffern: $0, \dots, b-1$
- Zusätzlich: der i–ten Stelle vor dem Komma wird der Wert b^i zugeordnet ($i = 0, \dots, n-1$)
- Multipliziere Ziffern mit jeweiligem Stellenwert und addiere auf

Dies ergibt für eine n-stellige Zahl den Zahlenwert

$$x = (x_{n-1}x_{n-2}\dots x_1x_0)_b = x_{n-1}b^{n-1} + x_{n-2}b^{n-2} + \dots + x_1b^1 + x_0b^0$$

Der Stellenwert der i–ten Stelle nach dem Komma ist b^{-i} (mit $i = 1, \dots, m$ bei m Stellen nach dem Komma) also

$$y = (0, y_1y_2\dots y_m)_b = y_1b^{-1} + y_2b^{-2} + \dots + y_mb^{-m}$$

Beispiel:

$$111,01_2 = 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 + 0 \cdot 2^{-1} + 1 \cdot 2^{-2} = 7,25$$
$$201,22_3 = 2 \cdot 3^2 + 0 \cdot 3^1 + 1 \cdot 3^0 + 2 \cdot 3^{-1} + 2 \cdot 3^{-2} = 19,\overline{8}$$

Umrechen zwischen verschiedenen Basen

Gegeben: Darstellung einer Zahl zur Basis b_1

Gesucht: Darstellung dieser Zahl zur Basis $b_2 \neq b_1$

Rechnen im Quellsystem

Betrachte Vor– und Nachkommastellen getrennt; für die Vorkommastellen:

1. Teile Zahl durch Basis des Zielsystems
2. Notiere ganzahligen Anteil des Ergebnisses und den Rest
3. Wiederhole die bisherigen Schritte mit dem ganzzahligen Anteil des Ergebnisses solange bis dieser 0 ist
4. Die „Reste“ bilden von unten nach oben gelesen die Ziffern der Darstellung (des Vorkomma–Anteils) im Zielsystem

Für die Nachkommastellen:

1. Multipliziere mit Basis des Zielsystems
2. Notiere Vorkomma–Anteil des Multiplikationsergebnisses
3. Wiederhole die bisherigen Schritte mit dem Nachkomma–Anteil des Multiplikationsergebnisses solange bis dieser 0 ist
4. Die in 2. notierten Ergebnisse bilden von oben nach unten gelesen die Ziffern der Darstellung (des Nachkomma–Anteils) im Zielsystem

Beispiel: Stelle 2012_3 im Fünfersystem dar. Für die Ziffen und die Basis des Fünfersystems gilt:

$$0_5 \triangleq 0_3 \quad 1_5 \triangleq 1_3 \quad 2_5 \triangleq 2_3 \quad 3_5 \triangleq 10_3 \quad 4_5 \triangleq 11_3 \quad \text{Basis } 5 \triangleq 12_3$$

$$\left.\begin{array}{rlrlrl} 2012_3 & : 12_3 = & 102_3 & \text{Rest} & 11_3 & \rightsquigarrow 4_5 \\ 102_3 & : 12_3 = & 2_3 & \text{Rest} & 1_3 & \rightsquigarrow 1_5 \\ 2_3 & : 12_3 = & 0_3 & \text{Rest} & 2_3 & \rightsquigarrow 2_5 \end{array}\right\} \Rightarrow 2012_3 \triangleq 214_5$$

Rechnen im Zielsystem

Verwende das HORNER*schema*.

Betrachte dazu Vor– und Nachkommastellen erneut getrennt; für die Ziffern vor dem Komma von links nach rechts:

$$\text{neues Ergebnis} = \text{altes Ergebnis} \cdot \text{Basis} + \text{Ziffer}$$

Für die Ziffern nach dem Komma von rechts nach links:

$$\text{neues Ergebnis} = \text{altes Ergebnis} : \text{Basis} + \text{Ziffer}$$

Wobei im ersten Schritt für das alte Ergebnis jeweils 0 zu wählen ist. Beachte außerdem, dass bei den Nachkommastellen die 0 vor dem Komma trotzdem als letzte Ziffer zu berücksichtigen ist.

Beispiel: Stelle 312_4 im Zweiersystem dar. Für die Ziffern und die Basis des Vierersystems gilt:

$$0_4 \triangleq 0_2 \quad 1_4 \triangleq 1_2 \quad 2_4 \triangleq 10_2 \quad 3_4 \triangleq 11_2 \quad \text{Basis } 4 \triangleq 100_2$$

Quellziffer	3_4	1_4	2_4
Altes Erg. · Basis	–	$11_2 \cdot 100_2 = 1100_2$	$1101_2 \cdot 100_2 = 110100_2$
Neues Erg.	11_2	$1100_2 + 1_2 = 1101_2$	$110100_2 + 10_2 = \underline{110110_2}$

Binärsystem, Binäres Rechnen

Addition, Subtraktion, Multiplikation und Division von positiven Binärzahlen ohne Komplikationen durchführbar.

Negative Zahlen bereiten Probleme. Zunächst: Darstellung von negativen Zahlen

1. *Einerkomplement:* Invertiere jedes Bit

$$z = 0100110 \quad \rightsquigarrow \quad -z = 1011001$$

2. *Zweierkomplement (ZK):* Addiere 1 zum Einerkomplement

$$z = 0100110 \quad \rightsquigarrow \quad -z = 1011010$$

   Zweimaliges Bilden des ZKs liefert ursprüngliche Zahl

Damit gelingt die Addition von negativen Zahlen:

- Wähle Darstellung fester Breite so, dass das vorderste Bit nicht von der eigentlichen Zahlendarstellung belegt werden kann (*Vorzeichenbit*); nicht benötigte Bits werden auf 0 gesetzt.
- Wandle negative Zahlen ins Zweierkomplement um; das Vorzeichenbit wird dabei immer 1 – daher der Name: bei positiven Zahlen 0, bei negativen Zahlen 1
- Addiere dann ganz normal und ignoriere dabei den *Überlauf* (d.h. alles das, was über die fest vereinbarte Breite hinausgeht)
- Am Vorzeichenbit des Ergebnisses sieht man dann, ob dieses positiv oder negativ ist. Ist es negativ, so liegt es auch automatisch als Zweierkomplement vor

Beispiele: (wir verwenden eine feste Darstellung der Breite 6 - dabei wird das sechste Bit nicht zur Darstellung benötigt)

- Berechne $9 + 12$. Es ist $9_{10} \triangleq 001001_2$ und $12_{10} \triangleq 001100_2$

$$\begin{array}{ccccccc} & 0 & 0 & 1 & 0 & 0 & 1 \\ + & 0 & 0 & 1 & 1 & 0 & 0 \\ & & 1 & & & & \\ \hline & 0 & 1 & 0 & 1 & 0 & 1 \end{array}$$

  Das sechste Bit (das Vorzeichenbit) ist 0, das Ergebnis damit positiv und $010101_2 \triangleq 21_{10}$.

- Berechne $(-7)+(-13)$. Es ist $7_{10} \triangleq 000111_2$ und $13_{10} \triangleq 001101_2$. Die Zweierkomplemente sind $-7_{10} \triangleq 111001_{\bar{2}}$ sowie $-13_{10} \triangleq 110011_{\bar{2}}$

$$\begin{array}{ccccccc} & 1 & 1 & 1 & 0 & 0 & 1 \\ + & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & & & 1 & 1 & \\ \hline (1) & 1 & 0 & 1 & 1 & 0 & 0 \end{array}$$

  Das Vorzeichenbit ist 1, das Ergebnis also negativ und $101100_{\bar{2}} \triangleq -010100_2 \triangleq -20_{10}$.

- Berechne $8 + (-17)$. Es ist $8 \triangleq 001000_2$ und $-17 \triangleq 101111_{\bar{2}}$

$$\begin{array}{ccccccc} & 0 & 0 & 1 & 0 & 0 & 0 \\ + & 1 & 0 & 1 & 1 & 1 & 1 \\ & & 1 & & & & \\ \hline & 1 & 1 & 0 & 1 & 1 & 1 \end{array}$$

  Das Vorzeichenbit ist 1, das Ergebnis also negativ und $110111_{\bar{2}} \triangleq -001001_2 \triangleq -9_{10}$.

- Berechne $25 + (-18)$. Es ist $25 \triangleq 011001_2$ und $-18 \triangleq 101110_2$

$$
\begin{array}{ccccccc}
 & 0 & 1 & 1 & 0 & 0 & 1 \\
+ & 1 & 0 & 1 & 1 & 1 & 0 \\
1 & 1 & 1 & & & & \\
\hline
(1) & 0 & 0 & 0 & 1 & 1 & 1
\end{array}
$$

Das Vorzeichenbit ist 0, das Ergebnis also positiv und $0000111_2 \triangleq 7_{10}$.

Codierung

Zwei (von vielen) Möglichkeiten zur Codierung $c : \Sigma \to \{0,1\}^*$:

1. Nummeriere Zeichen aus Σ durch und stelle $z \in \Sigma$ durch binäre Darstellung seiner Nummer dar (feste Länge nötig).
2. Huffman–Verfahren

Huffman–Verfahren

1. Ordne Zeichen nach Wahrscheinlichkeit des Auftretens
2. Fasse die beiden Zeichen mit der kleinsten Wahrscheinlichkeit zu neuem Zeichen zusammen und ordne diesem die Summe der Wahrscheinlichkeiten zu
3. Wiederhole dies bis nur noch ein Zeichen übrig ist
4. Konstruiere Codebaum
5. Codierung eines Zeichens wird durch den Pfad von der Wurzel bis zu diesem Zeichen festgelegt

Shannonsche Informationstheorie

In aller Kürze:

$\Sigma = \{z_1, \ldots, z_m\}$; Informationsgehalt eines Zeichens:

$$H = -\sum_{i=1}^{m} p_i \cdot \operatorname{ld} p_i \quad p_i = p(z_i) \;\; \text{(rel. Häufigkeit von } z_i\text{)}$$