„Ich hoffe, es werden Ew. Hochfürstliche Durchlaucht in Gnaden vermerken, daß ich sowohl dem Gebrauche, als meinem Gemüths=Triebe zu Folge, bei dem eingetretenen neuen Jahre, auf dieses und viele folgende, Denenselben in beständiger Gesundheit alle selbst verlangende hohe Fürstliche Ersprießlichkeit zu gemeinem und Dero Lande besondern Besten, aus treuem Herzen anwünsche.“

Brief vom 2. Januar 1697 an den Herzog von Braunschweig-Wolfenbüttel

„Ich hoffe, es werden Ew. Hochfürstliche Durchlaucht in Gnaden vermerken, daß ich sowohl dem Gebrauche, als meinem Gemüths=Triebe zu Folge, bei dem eingetretenen neuen Jahre, auf dieses und viele folgende, Denenselben in beständiger Gesundheit alle selbst verlangende hohe Fürstliche Ersprießlichkeit zu gemeinem und Dero Lande besondern Besten, aus treuem Herzen anwünsche.“

Brief vom 2. Januar 1697 an den Herzog von Braunschweig-Wolfenbüttel

von Gottfried Wilhelm Leibniz (1646–1716)

Philosoph, Mathematiker, Physiker, Bibliothekar, …

Grundbegriffe der Informatik

Einheit 8: Codierungen

Thomas Worsch

KIT, Institut für Theoretische Informatik

Wintersemester 2015/2016

Von Wörtern zu Zahlen und zurück

Von einem Alphabet zum anderen

Huffman-Codierung

Wo sind wir?

Dezimaldarstellung von Zahlen

Ziffern des Alphabetes $Z_{10} = \{0, \ldots, 9\}$.

Bedeutung einzelner Ziffer x als Zahl $\mathrm{num}_{10}(x)$:

x	0	1	2	$\cdots$	8	9
$\mathrm{num}_{10}(x)$	*null*	*eins*	*zwei*	$\cdots$	*acht*	*neun*

Dezimaldarstellung von Zahlen

Ziffern des Alphabetes $Z_{10} = \{0, \ldots, 9\}$.

Bedeutung einzelner Ziffer x als Zahl $\mathrm{num}_{10}(x)$:

x	0	1	2	3	4	5	6	7	8	9
$\mathrm{num}_{10}(x)$	0	1	2	3	4	5	6	7	8	9

Bedeutung einer Ziffernfolge $x_{k-1} \cdots x_0 \in Z_{10}^*$

$$\mathrm{Num}_{10} : Z_{10}^* \to \mathbb{N}_0$$

Dezimaldarstellung von Zahlen

Ziffern des Alphabetes $Z_{10} = \{0, \ldots, 9\}$.

Bedeutung einzelner Ziffer x als Zahl $\text{num}_{10}(x)$:

x	0	1	2	3	4	5	6	7	8	9
$\text{num}_{10}(x)$	0	1	2	3	4	5	6	7	8	9

Bedeutung einer Ziffernfolge $x_{k-1} \cdots x_0 \in Z_{10}^*$

$$\text{Num}_{10} : Z_{10}^* \to \mathbb{N}_0$$

$$\text{Num}_{10}(x_{k-1} \cdots x_1 x_0)$$

Dezimaldarstellung von Zahlen

Ziffern des Alphabetes $Z_{10} = \{0, \ldots, 9\}$.

Bedeutung einzelner Ziffer x als Zahl $\mathrm{num}_{10}(x)$:

x	0	1	2	3	4	5	6	7	8	9
$\mathrm{num}_{10}(x)$	0	1	2	3	4	5	6	7	8	9

Bedeutung einer Ziffernfolge $x_{k-1} \cdots x_0 \in Z_{10}^*$

$$\mathrm{Num}_{10} : Z_{10}^* \to \mathbb{N}_0$$

$$\begin{aligned} &\mathrm{Num}_{10}(x_{k-1} \cdots x_1 x_0) \\ &= 10^{k-1} \cdot \mathrm{num}_{10}(x_{k-1}) + \cdots + 10^1 \cdot \mathrm{num}_{10}(x_1) + 10^0 \cdot \mathrm{num}_{10}(x_0) \end{aligned}$$

Dezimaldarstellung von Zahlen

Ziffern des Alphabetes $Z_{10} = \{0, \ldots, 9\}$.

Bedeutung einzelner Ziffer x als Zahl $\mathrm{num}_{10}(x)$:

x	0	1	2	3	4	5	6	7	8	9
$\mathrm{num}_{10}(x)$	0	1	2	3	4	5	6	7	8	9

Bedeutung einer Ziffernfolge $x_{k-1} \cdots x_0 \in Z_{10}^*$

$$\mathrm{Num}_{10} : Z_{10}^* \to \mathbb{N}_0$$

$$\begin{aligned} &\mathrm{Num}_{10}(x_{k-1} \cdots x_1 x_0) \\ &= 10^{k-1} \cdot \mathrm{num}_{10}(x_{k-1}) + \cdots + 10^1 \cdot \mathrm{num}_{10}(x_1) + 10^0 \cdot \mathrm{num}_{10}(x_0) \\ &= 10 \left(10^{k-2} \cdot \mathrm{num}_{10}(x_{k-1}) + \cdots + 10^0 \cdot \mathrm{num}_{10}(x_1)\right) + \mathrm{num}_{10}(x_0) \end{aligned}$$

Dezimaldarstellung von Zahlen

Ziffern des Alphabetes $Z_{10} = \{0, \ldots, 9\}$.

Bedeutung einzelner Ziffer x als Zahl $\mathrm{num}_{10}(x)$:

x	0	1	2	3	4	5	6	7	8	9
$\mathrm{num}_{10}(x)$	0	1	2	3	4	5	6	7	8	9

Bedeutung einer Ziffernfolge $x_{k-1} \cdots x_0 \in Z_{10}^*$

$$\mathrm{Num}_{10} : Z_{10}^* \to \mathbb{N}_0$$

$$\begin{aligned}
&\mathrm{Num}_{10}(x_{k-1} \cdots x_1 x_0) \\
&= 10^{k-1} \cdot \mathrm{num}_{10}(x_{k-1}) + \cdots + 10^1 \cdot \mathrm{num}_{10}(x_1) + 10^0 \cdot \mathrm{num}_{10}(x_0) \\
&= 10 \left(10^{k-2} \cdot \mathrm{num}_{10}(x_{k-1}) + \cdots + 10^0 \cdot \mathrm{num}_{10}(x_1)\right) + \mathrm{num}_{10}(x_0) \\
&= 10 \cdot \mathrm{Num}_{10}(x_{k-1} \cdots x_1) + \mathrm{num}_{10}(x_0)
\end{aligned}$$

Dezimaldarstellung von Zahlen

Ziffern des Alphabetes $Z_{10} = \{0, \ldots, 9\}$.

Bedeutung einzelner Ziffer x als Zahl $\mathrm{num}_{10}(x)$:

x	0	1	2	3	4	5	6	7	8	9
$\mathrm{num}_{10}(x)$	0	1	2	3	4	5	6	7	8	9

Bedeutung einer Ziffernfolge $x_{k-1} \cdots x_0 \in Z_{10}^*$

$$\mathrm{Num}_{10} : Z_{10}^* \to \mathbb{N}_0$$

$$\mathrm{Num}_{10}(\varepsilon) = 0$$

für jedes $w \in Z_{10}^*$, für jedes $x \in Z_{10}$:

$$\mathrm{Num}_{10}(wx) = 10 \cdot \mathrm{Num}_{10}(w) + \mathrm{num}_{10}(x)$$

Induktive Definitionen — „über die Wortlänge“

Abbildungen z. B. der Form $f : A^* \to M$

definiere induktiv für jede Wortlänge $n \in \mathbb{N}_0$

- $f(v)$ für jedes $v \in A^n$

„Definitionsanfang“: Wortlänge $n = 0$

- definiere $f(\varepsilon)$

„Definitionsschritt“: für jede Wortlänge n von n nach $n + 1$

- für jedes $v \in A^{n+1}$ definiere $f(v)$
- und benutze dabei schon $f(w)$ für $w \in A^n$
- jedes $v \in A^{n+1}$ ist von der Form wx (bzw. xw) für $w \in A^n$ und $x \in A$

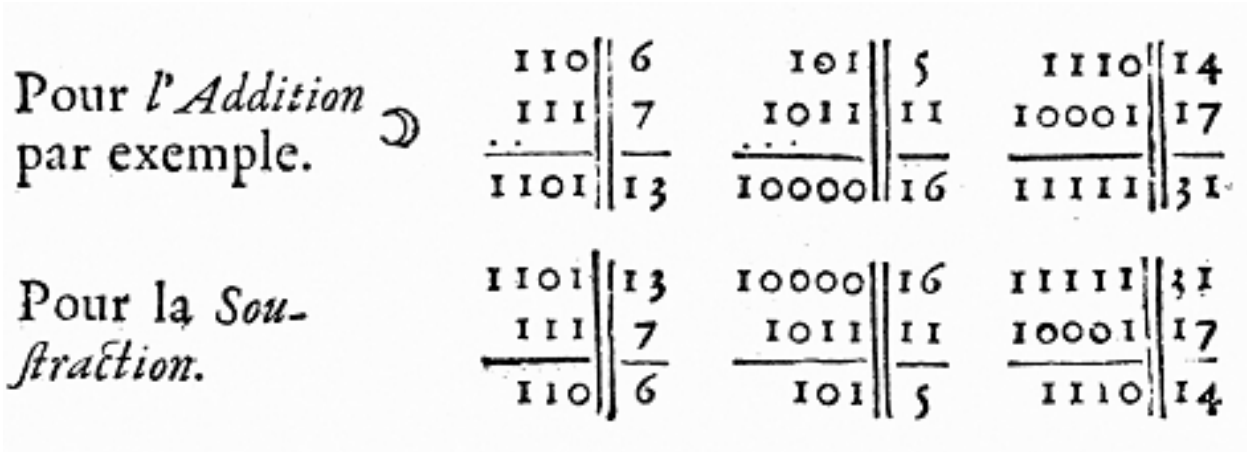
Wo sind wir?

Gottfried Wilhelm Leibniz: zwei Ziffern reichen aus!

- geboren 1. Juli 1646 in Leipzig
  gestorben am 14. November 1716 in Hannover
- baute erste Maschine, die zwei Zahlen multiplizieren konnte

Beobachtung

- mit nur zwei Ziffern 0 und 1 kann man alle nichtnegativen Zahlen darstellen und vernünftig rechnen



Bildquelle `http://commons.wikimedia.org/wiki/Image:Leibniz_binary_system_1703.png`

Binärdarstellung von Zahlen — Stellenwertsystem zur Basis 2

Ziffernmenge $Z_2 = \{0, 1\}$

definiere: $\mathrm{num}_2(0) = 0$ und $\mathrm{num}_2(1) = 1$ und
$\mathrm{Num}_2(\varepsilon) = 0$

für jedes $w \in Z_2^*$, für jedes $x \in Z_2$:

$\mathrm{Num}_2(wx) = 2 \cdot \mathrm{Num}_2(w) + \mathrm{num}_2(x)$

$$
\begin{aligned}
\mathrm{Num}_2(1101) &= 2 \cdot \mathrm{Num}_2(110) + 1 \\
&= 2 \cdot (2 \cdot \mathrm{Num}_2(11) + 0) + 1 \\
&= 2 \cdot (2 \cdot (2 \cdot \mathrm{Num}_2(1) + 1) + 0) + 1 \\
&= 2 \cdot (2 \cdot (2 \cdot (2 \cdot \mathrm{Num}_2(\varepsilon) + 1) + 1) + 0) + 1 \\
&= 2 \cdot (2 \cdot (2 \cdot (2 \cdot 0 + 1) + 1) + 0) + 1 \\
&= 2^3 \cdot 1 + 2^2 \cdot 1 + 2^1 \cdot 0 + 2^0 \cdot 1 = 13
\end{aligned}
$$

Binärdarstellung von Zahlen — Stellenwertsystem zur Basis 2

Ziffernmenge $Z_2 = \{0, 1\}$

definiere: $\mathrm{num}_2(0) = 0$ und $\mathrm{num}_2(1) = 1$ und
$\mathrm{Num}_2(\varepsilon) = 0$

für jedes $w \in Z_2^*$, für jedes $x \in Z_2$:

$\mathrm{Num}_2(wx) = 2 \cdot \mathrm{Num}_2(w) + \mathrm{num}_2(x)$

$$
\begin{aligned}
\mathrm{Num}_2(1101) &= 2 \cdot \mathrm{Num}_2(110) + 1 \\
&= 2 \cdot (2 \cdot \mathrm{Num}_2(11) + 0) + 1 \\
&= 2 \cdot (2 \cdot (2 \cdot \mathrm{Num}_2(1) + 1) + 0) + 1 \\
&= 2 \cdot (2 \cdot (2 \cdot (2 \cdot \mathrm{Num}_2(\varepsilon) + 1) + 1) + 0) + 1 \\
&= 2 \cdot (2 \cdot (2 \cdot (2 \cdot 0 + 1) + 1) + 0) + 1 \\
&= 2^3 \cdot 1 + 2^2 \cdot 1 + 2^1 \cdot 0 + 2^0 \cdot 1 = 13
\end{aligned}
$$

Binärdarstellung von Zahlen — Stellenwertsystem zur Basis 2

Ziffernmenge $Z_2 = \{0, 1\}$

definiere: $\mathrm{num}_2(0) = 0$ und $\mathrm{num}_2(1) = 1$ und
$\mathrm{Num}_2(\varepsilon) = 0$

für jedes $w \in Z_2^*$, für jedes $x \in Z_2$:

$\mathrm{Num}_2(wx) = 2 \cdot \mathrm{Num}_2(w) + \mathrm{num}_2(x)$

$$
\begin{aligned}
\mathrm{Num}_2(1101) &= 2 \cdot \mathrm{Num}_2(110) + 1 \\
&= 2 \cdot (2 \cdot \mathrm{Num}_2(11) + 0) + 1 \\
&= 2 \cdot (2 \cdot (2 \cdot \mathrm{Num}_2(1) + 1) + 0) + 1 \\
&= 2 \cdot (2 \cdot (2 \cdot (2 \cdot \mathrm{Num}_2(\varepsilon) + 1) + 1) + 0) + 1 \\
&= 2 \cdot (2 \cdot (2 \cdot (2 \cdot 0 + 1) + 1) + 0) + 1 \\
&= 2^3 \cdot 1 + 2^2 \cdot 1 + 2^1 \cdot 0 + 2^0 \cdot 1 = 13
\end{aligned}
$$

Hexadezimaldarstellung oder Sedezimaldarstellung

Ziffern $Z_{16} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F\}$.

x	0	1	2	3	4	5	6	7
$\mathrm{num}_{16}(x)$	0	1	2	3	4	5	6	7
x	8	9	A	B	C	D	E	F
$\mathrm{num}_{16}(x)$	8	9	10	11	12	13	14	15

Zuordnung von Wörtern zu Zahlen gegeben durch

$$\mathrm{Num}_{16}(\varepsilon) = 0$$

für jedes $w \in Z_{16}^*$, für jedes $x \in Z_{16}$:

$$\mathrm{Num}_{16}(wx) = 16 \cdot \mathrm{Num}_{16}(w) + \mathrm{num}_{16}(x)$$

$$\mathrm{Num}_{16}(20A) = 2 \cdot 16^2 + 0 \cdot 16^1 + 10 \cdot 16^0 = 522$$

Hexadezimaldarstellung oder Sedezimaldarstellung

Ziffern $Z_{16} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F\}$.

x	0	1	2	3	4	5	6	7
$\text{num}_{16}(x)$	0	1	2	3	4	5	6	7
x	8	9	A	B	C	D	E	F
$\text{num}_{16}(x)$	8	9	10	11	12	13	14	15

Zuordnung von Wörtern zu Zahlen gegeben durch

$$\text{Num}_{16}(\varepsilon) = 0$$

für jedes $w \in Z_{16}^*$, für jedes $x \in Z_{16}$:

$$\text{Num}_{16}(wx) = 16 \cdot \text{Num}_{16}(w) + \text{num}_{16}(x)$$

$$\text{Num}_{16}(20A) = 2 \cdot 16^2 + 0 \cdot 16^1 + 10 \cdot 16^0 = 522$$

Hexadezimaldarstellung oder Sedezimaldarstellung

Ziffern $Z_{16} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F\}$.

x	0	1	2	3	4	5	6	7
$\text{num}_{16}(x)$	0	1	2	3	4	5	6	7
x	8	9	A	B	C	D	E	F
$\text{num}_{16}(x)$	8	9	10	11	12	13	14	15

Zuordnung von Wörtern zu Zahlen gegeben durch

$$\text{Num}_{16}(\varepsilon) = 0$$

für jedes $w \in Z_{16}^*$, für jedes $x \in Z_{16}$:

$$\text{Num}_{16}(wx) = 16 \cdot \text{Num}_{16}(w) + \text{num}_{16}(x)$$

$$\text{Num}_{16}(20A) = 2 \cdot 16^2 + 0 \cdot 16^1 + 10 \cdot 16^0 = 522$$

Ein kleines Problem

die Alphabete Z_2, Z_3, usw. sind nicht disjunkt

Darstellungen mehrdeutig

$\mathrm{Num}_2(\mathtt{111})$	die Zahl sieben
$\mathrm{Num}_8(\mathtt{111})$	die Zahl dreiundsiebzig
$\mathrm{Num}_{10}(\mathtt{111})$	die Zahl einhundertelf
$\mathrm{Num}_{16}(\mathtt{111})$	die Zahl zweihundertdreiundsiebzig

in manchen Programmiersprachen

- Präfix `0b` für Darstellungen zur Basis 2
- Präfix `0o` für Darstellungen zur Basis 8
- Präfix `0x` für Darstellungen zur Basis 16

Wo sind wir?

Noch offene Frage: Ist jede Zahl darstellbar?

eben: zu jedem Wort dargestellte Zahl definierbar
(und „berechenbar“)

nun: zu jeder Zahl eine Darstellung definierbar
(und „berechenbar“)

k-äre Darstellung von Zahlen

Es sei $k \in \mathbb{N}_0$ mit $k \geq 2$.

Alphabet Z_k mit k Ziffern

- Bedeutung: die Zahlen in $\mathbb{Z}_k$
- für $i \in \mathbb{Z}_k$ sei $\text{repr}_k(i)$ das entsprechende Zeichen
- repr_k ist also gerade die Umkehrfunktion zu num_k

gesehen $\text{Num}_k : Z_k^* \to \mathbb{N}_0$

gesucht $\text{Repr}_k : \mathbb{N}_0 \to Z_k^*$

- Repräsentation von $n \in \mathbb{N}_0$ als $w \in Z_k^*$ mit $\text{Num}_k(w) = n$
- für die naheliegende Definition von Num_k

Redeweisen

- *binäre* Darstellung falls $k = 2$
- *ternäre* Darstellung falls $k = 3$

Operationen **div** und **mod**

es sei $x \in \mathbb{N}_0$ und $y \in \mathbb{N}_+$

x **mod** y

- Rest der ganzzahligen Division von x durch y
- $0 \leq x$ **mod** $y < y$

x **div** y

- Ergebnis der ganzzahligen Division von x durch y

für jedes $x \in \mathbb{N}_0$ und jedes $y \in \mathbb{N}_+$ gilt

$$x = y \cdot (x \textbf{ div } y) + (x \textbf{ mod } y)$$

Beispiele

- 6 **div** $2 = 3$ und 6 **mod** $2 = 0$
- 7 **div** $2 = 3$ und 7 **mod** $2 = 1$
- 8 **div** $2 = 4$ und 8 **mod** $2 = 0$

k-äre Darstellung von Zahlen (2)

Es sei $k \in \mathbb{N}_0$ und $k \geq 2$.

$$\mathrm{Repr}_k : \mathbb{N}_0 \to Z_k$$

$$n \mapsto \begin{cases} \mathrm{repr}_k(n) & \text{falls } n < k \\ \mathrm{Repr}_k(n \textbf{ div } k) \cdot \mathrm{repr}_k(n \textbf{ mod } k) & \text{falls } n \geq k \end{cases}$$

kann man auch so schreiben

$$n \mapsto \begin{cases} \mathrm{repr}_k(n \textbf{ mod } k) & \text{falls } n < k \\ \mathrm{Repr}_k(n \textbf{ div } k) \cdot \mathrm{repr}_k(n \textbf{ mod } k) & \text{falls } n \geq k \end{cases}$$

Ist das eine Definition?

Die Definition von Repr_k ist sinnvoll. (1)

$$\mathrm{Repr}_k : \mathbb{N}_0 \to Z_k$$

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

Behauptung: Für jedes $n \in \mathbb{N}_0$ ist $\mathrm{Repr}_k(n)$ definiert.

Die Definition von Repr_k ist sinnvoll. (1)

$$\mathrm{Repr}_k : \mathbb{N}_0 \to Z_k$$

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

Behauptung: Für jedes $n \in \mathbb{N}_0$ ist $\mathrm{Repr}_k(n)$ definiert.

Behauptung: Für jedes $m \in \mathbb{N}_+$ gilt:
Für jedes $n \in \mathbb{N}_0$ mit $n < k^m$ ist $\mathrm{Repr}_k(n)$ definiert.

Beweis durch vollständige Induktion

Die Definition von Repr_k ist sinnvoll. (2)

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

Behauptung: Für jedes $m \in \mathbb{N}_+$ gilt:
Für jedes $n \in \mathbb{N}_0$ mit $n < k^m$ ist $\mathrm{Repr}_k(n)$ definiert.

Ind.anfang: $m = 1$, also $n < k$: ✓

Ind.Schritt: sei $m \in \mathbb{N}_+$ und gelte:
Ind.vor. für jedes $n \in \mathbb{N}_0$ mit $n < k^m$ ist $\mathrm{Repr}_k(n)$ definiert.

zeige: für jedes $n \in \mathbb{N}_0$ mit $n < k^{m+1}$ ist $\mathrm{Repr}_k(n)$ definiert.

- falls sogar $n < k^m$: ✓ wegen I.V.
- falls $k^m \leq n < k^{m+1}$: dann $n \textbf{ div } k < k^m$ ($k \geq 2$!), also nach I.V. $\mathrm{Repr}_k(n \textbf{ div } k)$ definiert, also auch $\mathrm{Repr}_k(n)$

Num_k ist linksinvers zu Repr_k (1)

Lemma. Für jedes $n \in \mathbb{N}_0$ ist

$$\mathrm{Num}_k(\mathrm{Repr}_k(n)) = n$$

- „umgekehrt“ im allgemeinen $\mathrm{Repr}_k(\mathrm{Num}_k(w)) \neq w$ weil „überflüssige“ führende Nullen wegfallen.

Beweis: ähnliche vollständige Induktion wie eben

Num_k ist linksinvers zu Repr_k (2)

Lemma. Für jedes $n \in \mathbb{N}_0$ ist $\mathrm{Num}_k(\mathrm{Repr}_k(n)) = n$.

Ind.anfang: für jedes $n \in \mathbb{N}_0$ mit $n < k$: ✓

Ind.schritt: Es sei $m \in \mathbb{N}_+$. Unter der
Ind.vor.: für jedes $n \in \mathbb{N}_0$ mit $n < k^m$ ist $\mathrm{Num}_k(\mathrm{Repr}_k(n)) = n$
zeige: für jedes $n \in \mathbb{N}_0$ mit $n < k^{m+1}$ ist $\mathrm{Num}_k(\mathrm{Repr}_k(n)) = n$.

- falls sogar $n < k^m$: ✓ wegen I.V.
- falls $k^m \leq n < k^{m+1}$: dann $n \operatorname{\mathbf{div}} k < k^m$, also nach I.V. $\mathrm{Num}_k(\mathrm{Repr}_k(n \operatorname{\mathbf{div}} k)) = n \operatorname{\mathbf{div}} k$, also auch

$$\begin{aligned}
\mathrm{Num}_k(\mathrm{Repr}_k(n)) &= \mathrm{Num}_k(\mathrm{Repr}_k(k(n \operatorname{\mathbf{div}} k) + (n \operatorname{\mathbf{mod}} k))) \\
&= \mathrm{Num}_k(\mathrm{Repr}_k(n \operatorname{\mathbf{div}} k) \cdot \mathrm{repr}_k(n \operatorname{\mathbf{mod}} k)) \\
&= k \cdot \mathrm{Num}_k(\mathrm{Repr}_k(n \operatorname{\mathbf{div}} k)) + \mathrm{num}_k(\mathrm{repr}_k(n \operatorname{\mathbf{mod}} k)) \\
\text{nach I.V.} \qquad &= k(n \operatorname{\mathbf{div}} k) + (n \operatorname{\mathbf{mod}} k) = n
\end{aligned}$$

Num_k ist linksinvers zu Repr_k (2)

Lemma. Für jedes $n \in \mathbb{N}_0$ ist $\mathrm{Num}_k(\mathrm{Repr}_k(n)) = n$.

Ind.anfang: für jedes $n \in \mathbb{N}_0$ mit $n < k$: ✓

Ind.schritt: Es sei $m \in \mathbb{N}_+$. Unter der
Ind.vor.: für jedes $n \in \mathbb{N}_0$ mit $n < k^m$ ist $\mathrm{Num}_k(\mathrm{Repr}_k(n)) = n$
zeige: für jedes $n \in \mathbb{N}_0$ mit $n < k^{m+1}$ ist $\mathrm{Num}_k(\mathrm{Repr}_k(n)) = n$.

- falls sogar $n < k^m$: ✓ wegen I.V.
- falls $k^m \leq n < k^{m+1}$: dann $n \operatorname{\mathbf{div}} k < k^m$, also
  nach I.V. $\mathrm{Num}_k(\mathrm{Repr}_k(n \operatorname{\mathbf{div}} k)) = n \operatorname{\mathbf{div}} k$, also auch

$$\begin{aligned}
\mathrm{Num}_k(\mathrm{Repr}_k(n)) &= \mathrm{Num}_k(\mathrm{Repr}_k(k(n \operatorname{\mathbf{div}} k) + (n \operatorname{\mathbf{mod}} k))) \\
&= \mathrm{Num}_k(\mathrm{Repr}_k(n \operatorname{\mathbf{div}} k) \cdot \mathrm{repr}_k(n \operatorname{\mathbf{mod}} k)) \\
&= k \cdot \mathrm{Num}_k(\mathrm{Repr}_k(n \operatorname{\mathbf{div}} k)) + \mathrm{num}_k(\mathrm{repr}_k(n \operatorname{\mathbf{mod}} k)) \\
\text{nach I.V.} \qquad &= k(n \operatorname{\mathbf{div}} k) + (n \operatorname{\mathbf{mod}} k) = n
\end{aligned}$$

Num_k ist linksinvers zu Repr_k (2)

Lemma. Für jedes $n \in \mathbb{N}_0$ ist $\mathrm{Num}_k(\mathrm{Repr}_k(n)) = n$.

Ind.anfang: für jedes $n \in \mathbb{N}_0$ mit $n < k$: ✓

Ind.schritt: Es sei $m \in \mathbb{N}_+$. Unter der
Ind.vor.: für jedes $n \in \mathbb{N}_0$ mit $n < k^m$ ist $\mathrm{Num}_k(\mathrm{Repr}_k(n)) = n$
zeige: für jedes $n \in \mathbb{N}_0$ mit $n < k^{m+1}$ ist $\mathrm{Num}_k(\mathrm{Repr}_k(n)) = n$.

- falls sogar $n < k^m$: ✓ wegen I.V.
- falls $k^m \leq n < k^{m+1}$: dann $n \operatorname{\mathbf{div}} k < k^m$, also
  nach I.V. $\mathrm{Num}_k(\mathrm{Repr}_k(n \operatorname{\mathbf{div}} k)) = n \operatorname{\mathbf{div}} k$, also auch

$$\begin{aligned}
\mathrm{Num}_k(\mathrm{Repr}_k(n)) &= \mathrm{Num}_k(\mathrm{Repr}_k(k(n \operatorname{\mathbf{div}} k) + (n \operatorname{\mathbf{mod}} k))) \\
&= \mathrm{Num}_k(\mathrm{Repr}_k(n \operatorname{\mathbf{div}} k) \cdot \mathrm{repr}_k(n \operatorname{\mathbf{mod}} k)) \\
&= k \cdot \mathrm{Num}_k(\mathrm{Repr}_k(n \operatorname{\mathbf{div}} k)) + \mathrm{num}_k(\mathrm{repr}_k(n \operatorname{\mathbf{mod}} k)) \\
\text{nach I.V.} \qquad &= k(n \operatorname{\mathbf{div}} k) + (n \operatorname{\mathbf{mod}} k) = n
\end{aligned}$$

Num_k ist linksinvers zu Repr_k (2)

Lemma. Für jedes $n \in \mathbb{N}_0$ ist $\mathrm{Num}_k(\mathrm{Repr}_k(n)) = n$.

Ind.anfang: für jedes $n \in \mathbb{N}_0$ mit $n < k$: ✓

Ind.schritt: Es sei $m \in \mathbb{N}_+$. Unter der
Ind.vor.: für jedes $n \in \mathbb{N}_0$ mit $n < k^m$ ist $\mathrm{Num}_k(\mathrm{Repr}_k(n)) = n$
zeige: für jedes $n \in \mathbb{N}_0$ mit $n < k^{m+1}$ ist $\mathrm{Num}_k(\mathrm{Repr}_k(n)) = n$.

- falls sogar $n < k^m$: ✓ wegen I.V.
- falls $k^m \leq n < k^{m+1}$: dann $n \operatorname{\mathbf{div}} k < k^m$, also
  nach I.V. $\mathrm{Num}_k(\mathrm{Repr}_k(n \operatorname{\mathbf{div}} k)) = n \operatorname{\mathbf{div}} k$, also auch

$$\begin{aligned}
\mathrm{Num}_k(\mathrm{Repr}_k(n)) &= \mathrm{Num}_k(\mathrm{Repr}_k(k(n \operatorname{\mathbf{div}} k) + (n \operatorname{\mathbf{mod}} k))) \\
&= \mathrm{Num}_k(\mathrm{Repr}_k(n \operatorname{\mathbf{div}} k) \cdot \mathrm{repr}_k(n \operatorname{\mathbf{mod}} k)) \\
&= k \cdot \mathrm{Num}_k(\mathrm{Repr}_k(n \operatorname{\mathbf{div}} k)) + \mathrm{num}_k(\mathrm{repr}_k(n \operatorname{\mathbf{mod}} k)) \\
\text{nach I.V.} \qquad &= k(n \operatorname{\mathbf{div}} k) + (n \operatorname{\mathbf{mod}} k) = n
\end{aligned}$$

Num_k ist linksinvers zu Repr_k (2)

Lemma. Für jedes $n \in \mathbb{N}_0$ ist $\mathrm{Num}_k(\mathrm{Repr}_k(n)) = n$.

Ind.anfang: für jedes $n \in \mathbb{N}_0$ mit $n < k$: ✓

Ind.schritt: Es sei $m \in \mathbb{N}_+$. Unter der
Ind.vor.: für jedes $n \in \mathbb{N}_0$ mit $n < k^m$ ist $\mathrm{Num}_k(\mathrm{Repr}_k(n)) = n$
zeige: für jedes $n \in \mathbb{N}_0$ mit $n < k^{m+1}$ ist $\mathrm{Num}_k(\mathrm{Repr}_k(n)) = n$.

- falls sogar $n < k^m$: ✓ wegen I.V.
- falls $k^m \leq n < k^{m+1}$: dann $n \operatorname{\mathbf{div}} k < k^m$, also nach I.V. $\mathrm{Num}_k(\mathrm{Repr}_k(n \operatorname{\mathbf{div}} k)) = n \operatorname{\mathbf{div}} k$, also auch

$$\begin{aligned}
\mathrm{Num}_k(\mathrm{Repr}_k(n)) &= \mathrm{Num}_k(\mathrm{Repr}_k(k(n \operatorname{\mathbf{div}} k) + (n \operatorname{\mathbf{mod}} k))) \\
&= \mathrm{Num}_k(\mathrm{Repr}_k(n \operatorname{\mathbf{div}} k) \cdot \mathrm{repr}_k(n \operatorname{\mathbf{mod}} k)) \\
&= k \cdot \mathrm{Num}_k(\mathrm{Repr}_k(n \operatorname{\mathbf{div}} k)) + \mathrm{num}_k(\mathrm{repr}_k(n \operatorname{\mathbf{mod}} k)) \\
\text{nach I.V.} \qquad &= k(n \operatorname{\mathbf{div}} k) + (n \operatorname{\mathbf{mod}} k) = n
\end{aligned}$$

Num_k ist linksinvers zu Repr_k (2)

Lemma. Für jedes $n \in \mathbb{N}_0$ ist $\mathrm{Num}_k(\mathrm{Repr}_k(n)) = n$.

Ind.anfang: für jedes $n \in \mathbb{N}_0$ mit $n < k$: ✓

Ind.schritt: Es sei $m \in \mathbb{N}_+$. Unter der
Ind.vor.: für jedes $n \in \mathbb{N}_0$ mit $n < k^m$ ist $\mathrm{Num}_k(\mathrm{Repr}_k(n)) = n$
zeige: für jedes $n \in \mathbb{N}_0$ mit $n < k^{m+1}$ ist $\mathrm{Num}_k(\mathrm{Repr}_k(n)) = n$.

- falls sogar $n < k^m$: ✓ wegen I.V.
- falls $k^m \leq n < k^{m+1}$: dann $n \operatorname{\mathbf{div}} k < k^m$, also
  nach I.V. $\mathrm{Num}_k(\mathrm{Repr}_k(n \operatorname{\mathbf{div}} k)) = n \operatorname{\mathbf{div}} k$, also auch

$$\begin{aligned}
\mathrm{Num}_k(\mathrm{Repr}_k(n)) &= \mathrm{Num}_k(\mathrm{Repr}_k(k(n \operatorname{\mathbf{div}} k) + (n \operatorname{\mathbf{mod}} k))) \\
&= \mathrm{Num}_k(\mathrm{Repr}_k(n \operatorname{\mathbf{div}} k) \cdot \mathrm{repr}_k(n \operatorname{\mathbf{mod}} k)) \\
&= k \cdot \mathrm{Num}_k(\mathrm{Repr}_k(n \operatorname{\mathbf{div}} k)) + \mathrm{num}_k(\mathrm{repr}_k(n \operatorname{\mathbf{mod}} k)) \\
\text{nach I.V.} \qquad &= k(n \operatorname{\mathbf{div}} k) + (n \operatorname{\mathbf{mod}} k) = n
\end{aligned}$$

Unübliche Methode für negative Zahlen — mit der Ziffer „meins“

Ziffernmenge $Z = \{\mathtt{I}, \mathtt{0}, \mathtt{1}\}$

definiere num : $Z \to \mathbb{Z}$

$$\begin{aligned} \mathtt{I} &\mapsto -1 \\ \mathtt{0} &\mapsto 0 \\ \mathtt{1} &\mapsto 1 \end{aligned}$$

definiere Num : $Z^* \to \mathbb{Z}$ „wie üblich“:

$$\mathrm{Num}(\varepsilon) = 0$$

$$\forall w \in Z^* \; \forall x \in Z : \mathrm{Num}(wx) = 3 \cdot \mathrm{Num}(w) + \mathrm{num}(x)$$

dann z. B.

- $\mathrm{Num}(\mathtt{I01}) = -3^2 + 0 + 3^0 = -8$
- $\mathrm{Num}(\mathtt{1I01}) = +3^3 - 3^2 + 0 + 3^0 = 19$

Unübliche Methode für negative Zahlen — Negation und Addition/Subtraktion sind ganz einfach

definere inv : $Z^* \to Z^*$

- $\mathrm{inv}(\mathtt{1}) = \mathtt{\overline{1}}$, $\mathrm{inv}(\mathtt{\overline{1}}) = \mathtt{1}$ und $\mathrm{inv}(\mathtt{0}) = \mathtt{0}$.
- $\mathrm{inv}(\varepsilon) = \varepsilon$ und für jedes $w \in Z^*$: für jedes $x \in Z$: $\mathrm{inv}(wx) = \mathrm{inv}(w)\mathrm{inv}(x)$

also z. B. $\mathrm{inv}(\mathtt{1\overline{1}01}) = \mathtt{\overline{1}10\overline{1}}$

dann gilt für alle $w \in Z^*$: $\mathrm{Num}(\mathrm{inv}(w)) = -\mathrm{Num}(w)$.

Rechnen in $\mathbb{Z}_k$ (1)

Es sei $k \in \mathbb{N}_0$ mit $k \geq 2$.

binäre Operationen auf $\mathbb{Z}_k$; für jedes $x, y \in \mathbb{Z}_k$ sei

$$x +_k y = (x + y) \ \textbf{mod} \ k$$
$$x -_k y = (x - y) \ \textbf{mod} \ k$$

- üblicherweise schreibt man + statt $+_k$ (und − statt $-_k$)

Lemma. Für jedes $x, y \in \mathbb{N}_0$ ist

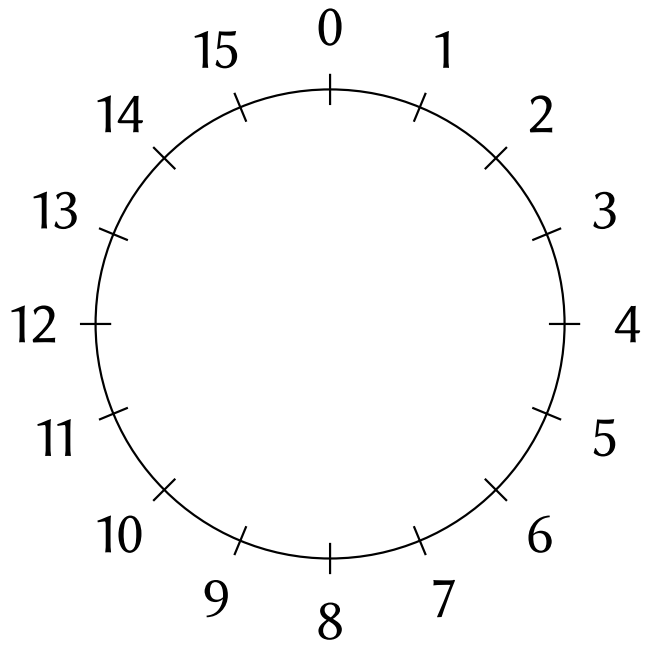
$$(x \pm y) \ \textbf{mod} \ k = (x \ \textbf{mod} \ k) \pm_k (y \ \textbf{mod} \ k) \ .$$

Beweis: sei $x = kq_x + r_x$ und $y = kq_y + r_y$ mit $r_x, r_y \in \mathbb{Z}_k$:

$$\begin{aligned}(x \pm y) \ \textbf{mod} \ k &= (kq_x \pm kq_y + r_x \pm r_y) \ \textbf{mod} \ k \\ &= (r_x \pm r_y) \ \textbf{mod} \ k = r_x \pm_k r_y\end{aligned}$$

Rechnen in $\mathbb{Z}_k$ (2) — graphisch: Zahlen als Pfeile

kreisförmige Darstellung von $\mathbb{Z}_{16}$

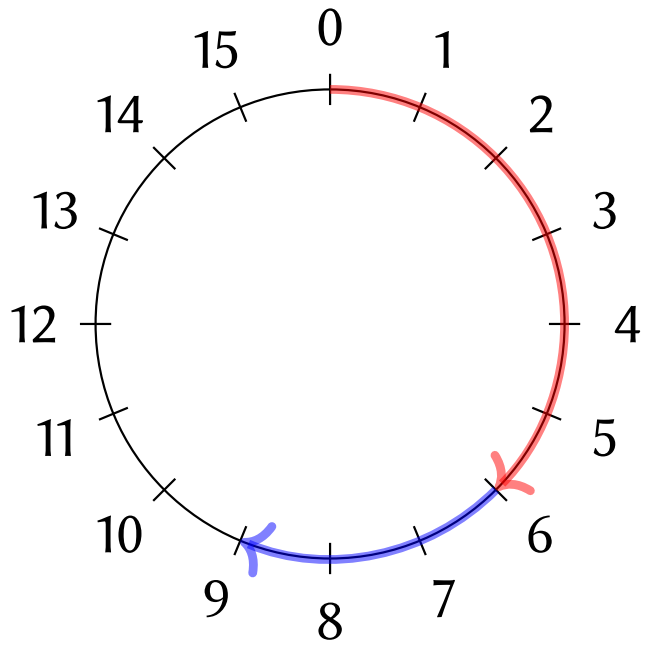


Addition $x +_k y$

- Pfeile für x und y „hintereinander setzen“ (Anfang an Spitze)

Rechnen in $\mathbb{Z}_k$ (2) — graphisch: Zahlen als Pfeile

kreisförmige Darstellung von $\mathbb{Z}_{16}$



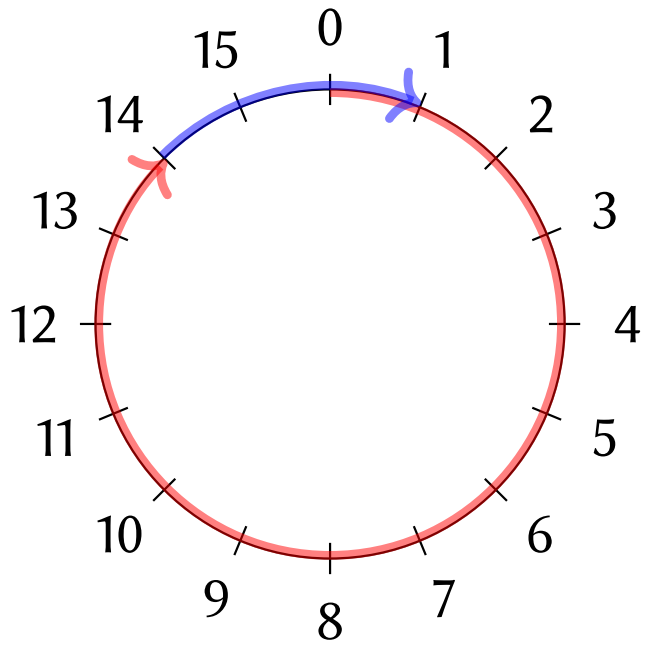
$6 +_{16} 3 = 9$

Addition $x +_k y$

- Pfeile für x und y „hintereinander setzen“ (Anfang an Spitze)

Rechnen in $\mathbb{Z}_k$ (2) — graphisch: Zahlen als Pfeile

kreisförmige Darstellung von $\mathbb{Z}_{16}$



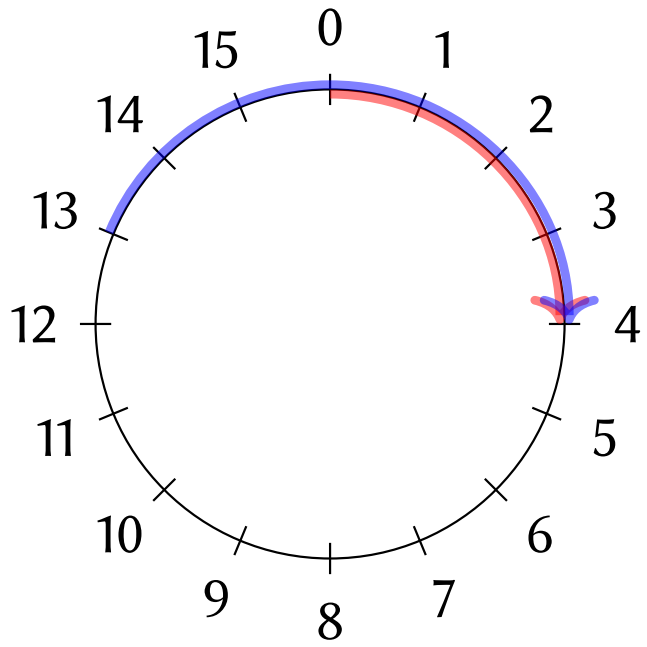
$14 +_{16} 3 = 1$

Addition $x +_k y$

- Pfeile für x und y „hintereinander setzen“ (Anfang an Spitze)

Rechnen in $\mathbb{Z}_k$ (2) — graphisch: Zahlen als Pfeile

kreisförmige Darstellung von $\mathbb{Z}_{16}$



$4 -_{16} 7 = 13$

Addition $x +_k y$

- Pfeile für x und y „hintereinander setzen“ (Anfang an Spitze)

Subtraktion $x -_k y$:

- Pfeile für x und y „nebeneinander setzen“ (Spitze an Spitze)

Zahldarstellungen mit beschränkter fester Länge

in Rechnern üblich

- Mima benutzt 24 Bits (siehe Kapitel 10)

Länge $\ell \in \mathbb{N}_+$, $\ell \geq 2$

(zu) einfache Idee:

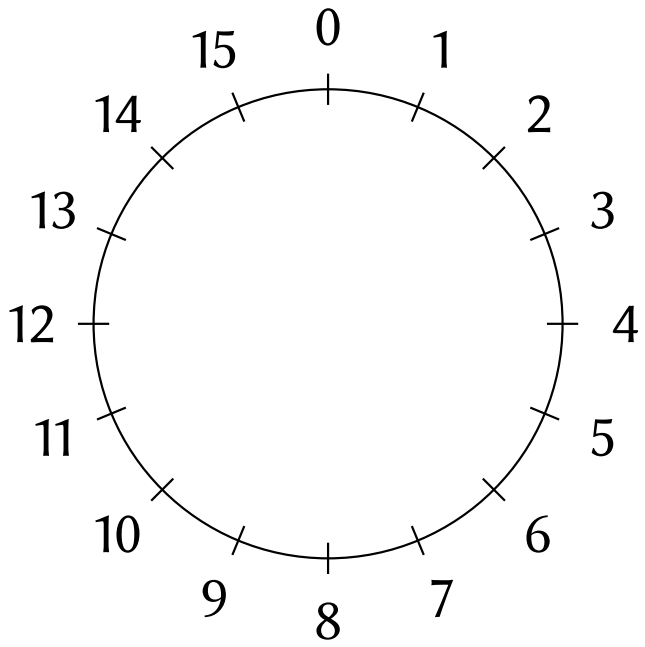
- $\mathrm{bin}_\ell : \mathbb{Z}_{2^\ell} \to \{0,1\}^\ell$

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

- $|\mathrm{bin}_\ell(n)| = \ell$ und $\mathrm{Num}_2(\mathrm{bin}_\ell(n)) = n$

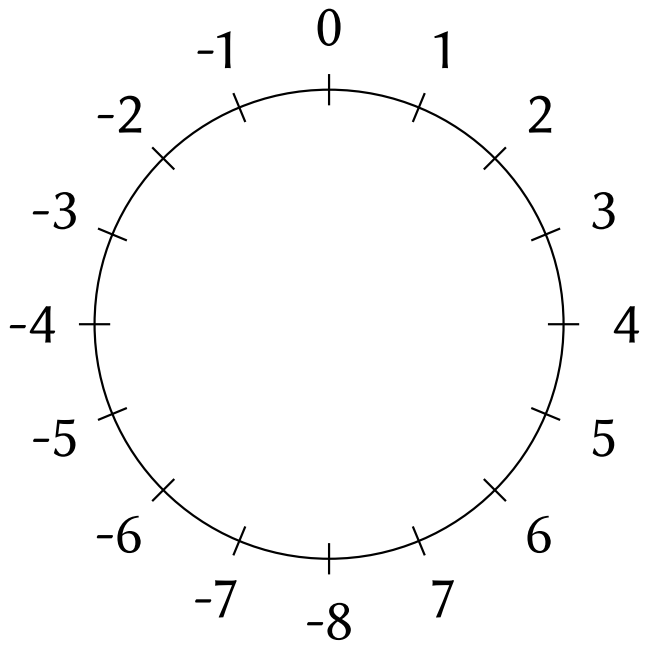
Negative Zahlen — Pfeile in die entgegengesetzte Richtung

kreisförmige Darstellung von $\mathbb{Z}_{16}$



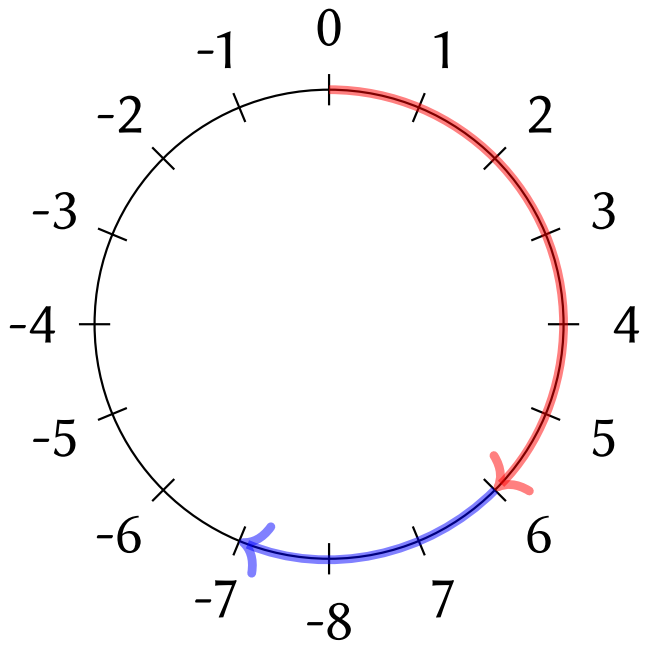
Negative Zahlen — Pfeile in die entgegengesetzte Richtung

kreisförmige Darstellung von $\mathbb{K}_4$ (Definition gleich)



Negative Zahlen — Pfeile in die entgegengesetzte Richtung

kreisförmige Darstellung von $\mathbb{K}_4$ (Definition gleich)



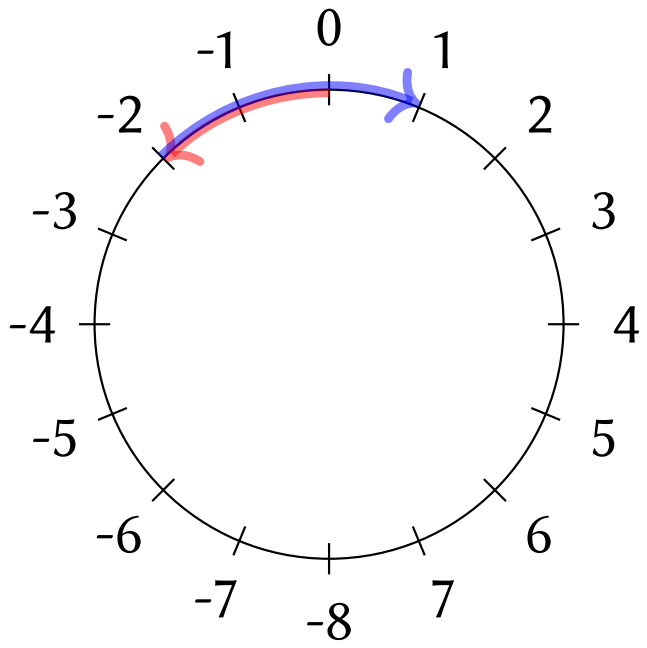
6 „+“ $3 = -7$

Addition x „+“ y (und auch Subtraktion)

- analog zum Fall $\mathbb{Z}_k$

Negative Zahlen — Pfeile in die entgegengesetzte Richtung

kreisförmige Darstellung von $\mathbb{K}_4$ (Definition gleich)



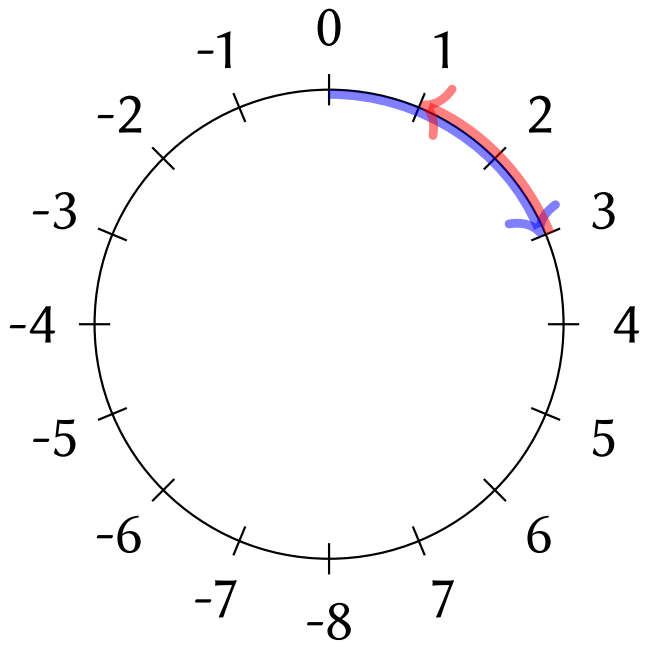
-2 „+“ $3 = 1$

Addition x „+“ y (und auch Subtraktion)

- analog zum Fall $\mathbb{Z}_k$

Negative Zahlen — Pfeile in die entgegengesetzte Richtung

kreisförmige Darstellung von $\mathbb{K}_4$ (Definition gleich)



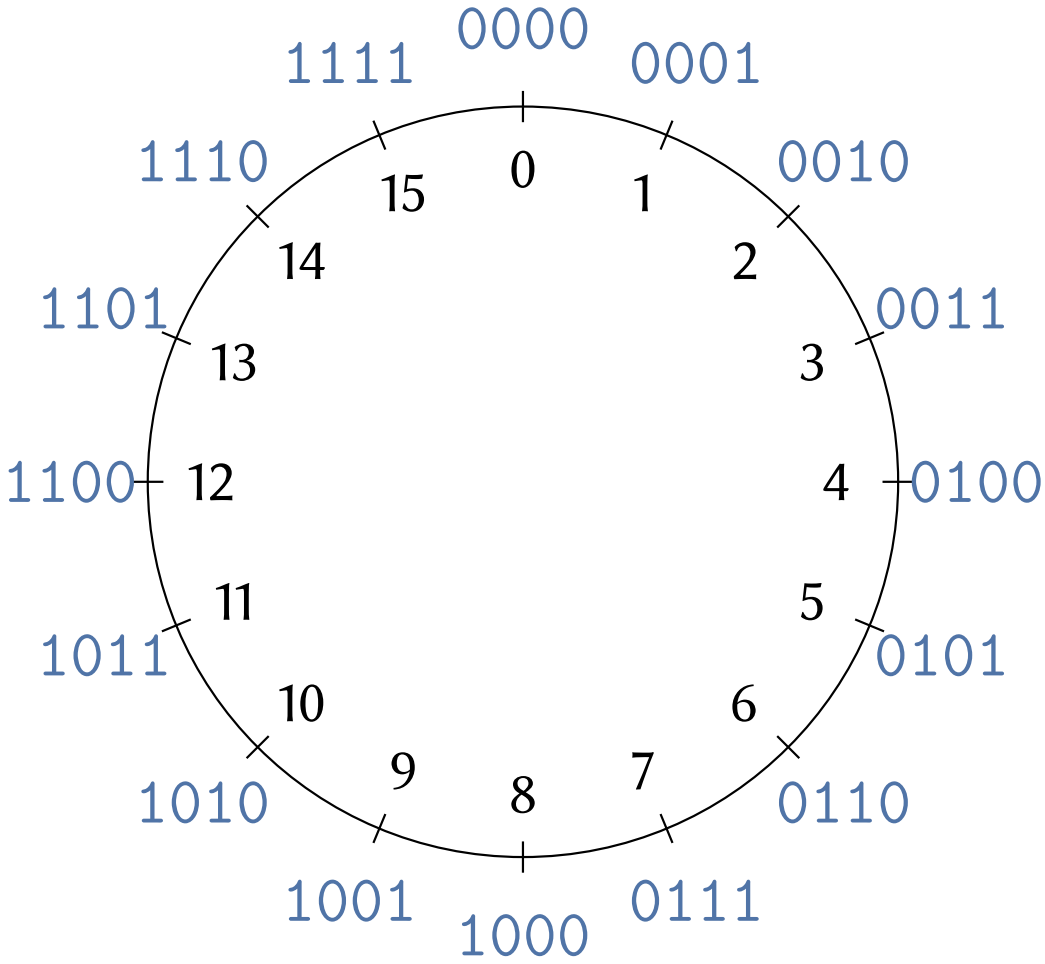
3 „+“ $-2 = 1$

Addition x „+“ y (und auch Subtraktion)

- analog zum Fall $\mathbb{Z}_k$

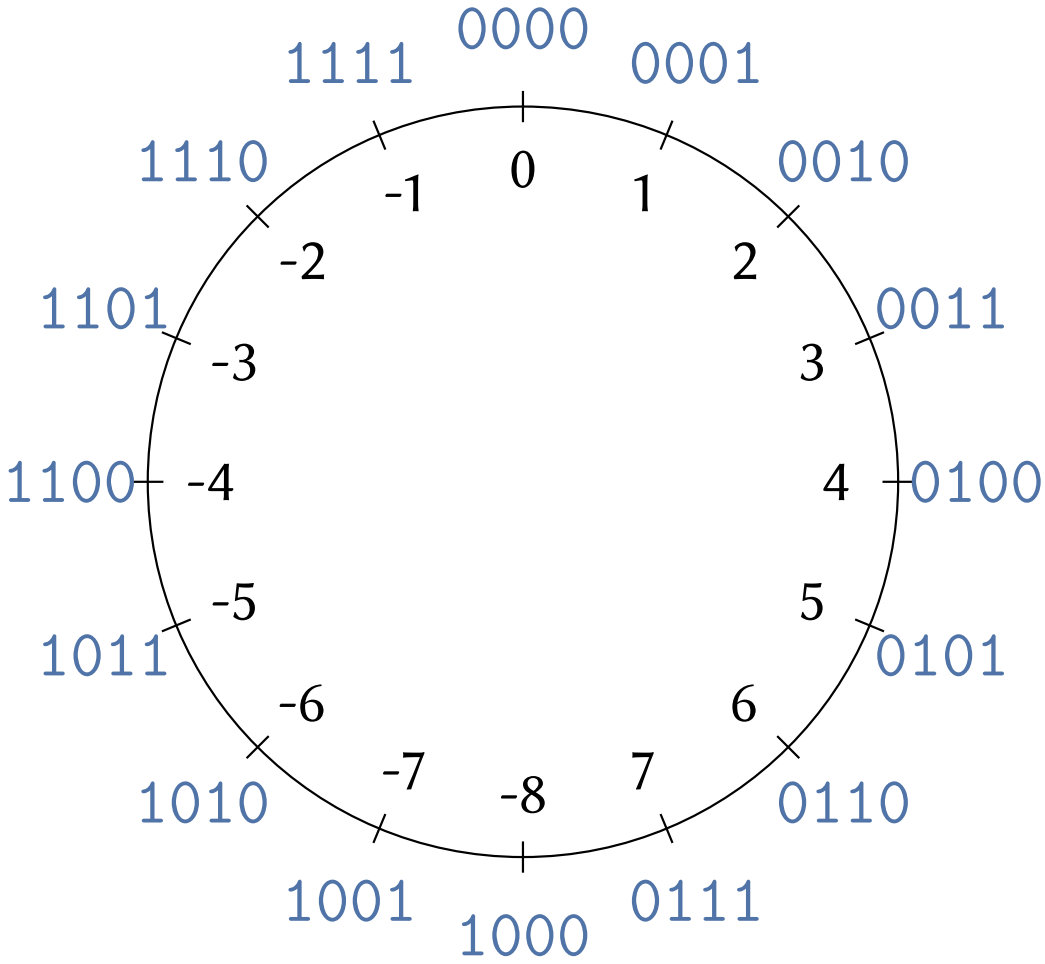
Darstellung auch negativer Zahlen

$\mathbb{Z}_{16}$ in Binärdarstellung



Darstellung auch negativer Zahlen

$\mathbb{K}_4$ in Zweierkomplementdarstellung



Zweierkomplement-Darstellung — für negative und nichtnegative Zahlen

für die Zahlen in

$$\mathbb{K}_\ell = \{x \in \mathbb{Z} \mid -2^{\ell-1} \leq x \leq 2^{\ell-1} - 1\}$$

- $\mathbb{K}_2 = \{-2, -1, 0, 1\}$
- $\mathbb{K}_8 = \{-128, -127, \ldots, -1, 0, 1, \ldots, 127\}$

$\mathsf{Zkpl}_\ell \colon \mathbb{K}_\ell \to \{0,1\}^\ell$

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

äquivalent

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

Das ist wichtig

Das sollten Sie mitnehmen:

- Umwandlungen zwischen Zahlen und Wörtern
- schon Leibniz kannte die Binärdarstellung
- Zweierkomplement-Darstellung

Das sollten Sie üben:

- selber Zahlen verschieden repräsentieren
- Definitionen auch in Randfällen ausprobieren

Von Wörtern zu Zahlen und zurück

Von einem Alphabet zum anderen

Huffman-Codierung

Wo sind wir?

Von Hexadezimal- zu Binärdarstellung

betrachte $\mathrm{Trans}_{2,16} : Z_{16}^* \to Z_2^*$ mit

- $\mathrm{Trans}_{2,16}(w) = \mathrm{Repr}_2(\mathrm{Num}_{16}(w))$

Beispiel

$$\begin{aligned}\mathrm{Trans}_{2,16}(\texttt{A3}) &= \mathrm{Repr}_2(\mathrm{Num}_{16}(\texttt{A3})) \\ &= \mathrm{Repr}_2(163) = \texttt{10100011}\end{aligned}$$

wesentlicher Punkt:

- A3 und 10100011 haben *die gleiche Bedeutung*
- nämlich die Zahl einhundertdreiundsechzig

So etwas wollen wir eine Übersetzung nennen.

Übersetzungen — bedeutungserhaltende Abbildungen von Wörtern auf Wörter

Wörter aus Sprache $L_A \subseteq A^*$ haben meist Bedeutung.

Menge Sem von Bedeutungen

- Zahlen
- Ausführungen von Java-Programmen
- ...

Übersetzungen — bedeutungserhaltende Abbildungen von Wörtern auf Wörter

Wörter aus Sprache $L_A \subseteq A^*$ haben meist Bedeutung.

Menge Sem von Bedeutungen

- Zahlen
- Ausführungen von Java-Programmen
- ...

Gegeben:

- Alphabete A und B
- $L_A \subseteq A^*$ und $L_B \subseteq B^*$
- Abbildungen $\text{sem}_A : L_A \to \text{Sem}$ und $\text{sem}_B : L_B \to \text{Sem}$

$f : L_A \to L_B$ heiße eine Übersetzung wenn

$$\text{für jedes } w \in L_A : \text{sem}_A(w) = \text{sem}_B(f(w))$$

$\mathrm{Trans}_{2,16}$ — eine Übersetzung

$\mathrm{Trans}_{2,16} = \mathrm{Repr}_2 \circ \mathrm{Num}_{16}$.

einfacher Fall: $L_A = A^* = Z_2^*$ und $L_B = B^* = Z_{16}^*$.

Bedeutungsfunktionen: $\mathrm{sem}_A = \mathrm{Num}_{16}$ und $\mathrm{sem}_B = \mathrm{Num}_2$

Nachrechnen, dass $\mathrm{Trans}_{2,16}$ eine Übersetzung ist:

$$\begin{aligned}\mathrm{sem}_B(f(w)) &= \mathrm{Num}_2(\mathrm{Trans}_{2,16}(w)) \\ &= \mathrm{Num}_2(\mathrm{Repr}_2(\mathrm{Num}_{16}(w))) \\ &= \mathrm{Num}_{16}(w) \\ &= \mathrm{sem}_A(w)\end{aligned}$$

Wozu Übersetzungen

Wozu Übersetzungen

Legalität: nur bestimmter Zeichensatz erlaubt

Lesbarkeit: vergleiche `A3` mit `10100011`

Verschlüsselung: nicht verbesserte Lesbarkeit, sondern

- gar keine Lesbarkeit … für Außenstehende
- siehe Vorlesungen über Kryptographie

Kompression: Übersetzungen können kürzer sein

- und zwar *ohne* größeres Alphabet
- in diesem Kapitel: Huffman-Codes

Fehlererkennung und Fehlerkorrektur:

- durch Übersetzung Text länger so, dass
- u. U. Fehlererkennung oder gar Fehlerkorrektur möglich
- siehe Vorlesungen über Codierungstheorie

Codierungen — injektive Übersetzungen

$\mathrm{sem}_A(w) = \mathrm{sem}_B(f(w))$ manchmal kein Problem

- Verschlüsselung, manche Kompressionsverfahren

wenn f injektiv

- von $f(x)$ *eindeutig* zurück zu x
- dann sem_B *definierbar* durch $\mathrm{sem}_B(f(x)) = \mathrm{sem}_A(x)$

Codierung: Übersetzung mit injektivem f

Codewörter: die $f(w)$

Code: $\{f(w) \mid w \in L_A\}$

Wie spezifiziert man eine Übersetzung?

wenn L_A unendlich

- kann man nicht alle Möglichkeiten aufzählen …

Auswege:

- Homomorphismen
- Block-Codierungen

Wo sind wir?

Von Wörtern zu Zahlen und zurück

Von einem Alphabet zum anderen

- Übersetzung von Zahldarstellungen
- Homomorphismen
- Beispiel Unicode: UTF-8

Huffman-Codierung

Homomorphismen —
mit Konkatenation verträgliche Abbildungen

gegeben Abbildung $h : A^* \to B^*$

- für Alphabete A und B

h Homomorphismus, falls für jedes $w_1, w_2 \in A^*$ gilt:

$$h(w_1 w_2) = h(w_1)h(w_2) \ .$$

Beispiel:

- es sei $h(\mathtt{a}) = \mathtt{10}$ und $h(\mathtt{b}) = \mathtt{001}$
- dann $h(\mathtt{bab}) = h(\mathtt{ba}) \cdot h(\mathtt{b}) = h(\mathtt{b}) \cdot h(\mathtt{a}) \cdot h(\mathtt{b})$
  $= \mathtt{001} \cdot \mathtt{10} \cdot \mathtt{001}$

Homomorphismus ε-frei, wenn für jedes $x \in A : h(x) \neq \varepsilon$.

Homomorphismen lassen das leere Wort unverändert

Homomorphismus $h : A^* \to B^*$,
also für jedes $w_1, w_2 \in A^*$:

$$h(w_1 w_2) = h(w_1)h(w_2) \ .$$

kurze Überlegung

- $h(\varepsilon) = h(\varepsilon\varepsilon) = h(\varepsilon)h(\varepsilon)$
- also $|h(\varepsilon)| = |h(\varepsilon)| + |h(\varepsilon)|$
- also $|h(\varepsilon)| = 0$
- also $h(\varepsilon) = \varepsilon$

Homomorphismen — die Bilder einzelner Symbole legen alles fest

Lemma. Es seien A und B Alphabete und $h : A^* \to B^*$ und $g : A^* \to B^*$ Homomorphismen.
Wenn für jedes $x \in A$ gilt, dass $h(x) = g(x)$ ist,
dann gilt für jedes $w \in \mathcal{A}^*$, dass $h(w) = g(w)$ ist.

Beweis durch vollständige Induktion über die Wortlänge

Induktionsanfang: $w = \varepsilon$

- $h(\varepsilon) = \varepsilon = g(\varepsilon)$

Induktionsschritt: Es seien $w \in A^*$ und $x \in A$
und es gelte die
Induktionsvoraussetzung $h(w) = g(w)$.

Homomorphismen — die Bilder einzelner Symbole legen alles fest (2)

Induktionsschritt: Es seien $w \in A^*$ und $x \in A$
und es gelte die
Induktionsvoraussetzung $h(w) = g(w)$.

dann gilt auch

$$
\begin{aligned}
h(wx) &= h(w)h(x) && \text{da } h \text{ Homomorphismus} \\
&= g(w)h(x) && \text{Induktionsvoraussetzung} \\
&= g(w)g(x) && \text{Voraussetzung des Lemmas} \\
&= g(wx) && \text{da } g \text{ Homomorphismus}
\end{aligned}
$$

Homomorphismen — so legen die Bilder einzelner Symbole alles fest

Es seien A und B Alphabete und $f : A \to B^*$.

definiere $f^{**} : A^* \to B^*$ vermöge

$$f^{**}(\varepsilon) = \varepsilon$$

$$\text{für jedes } w \in A^*, \text{für jedes } x \in A : f^{**}(wx) = f^{**}(w)f(x)$$

Lemma. f^{**} ist ein Homomorphismus.

- Beweis durch vollständige Induktion …

f^{**} heißt der durch f induzierte Homomorphismus

Präfixfreie Codes

gegeben Homomorphismus $h : A^* \to B^*$

Frage: Ist h Codierung, also injektiv?

im allgemeinen nicht ganz einfach zu sehen

einfacher Spezialfall: h ist präfixfrei

Das bedeutet: für *keine zwei verschiedene* Symbole $x_1, x_2 \in A$ gilt: $h(x_1)$ ist ein Präfix von $h(x_2)$.

gleich: präfixfreie Codes *sind* injektiv

Präfixfreie Codes: Decodierung

Problem: nicht alle $w \in B^*$ sind Codewort

- d. h. h ist im allgemeinen nicht surjektiv

damit Decodierung u trotzdem total

- definiere $u : B^* \to (A \cup \{\bot\})^*$.
- wenn $w \in B^*$ nicht Codewort,
  dann soll in $u(w)$ das Symbol $\bot$ vorkommen
  - lies: „undefiniert“

Beispiel

- Homomorphismus $h : \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}^* \to \{\mathtt{0}, \mathtt{1}\}^*$ mit
  $h(\mathtt{a}) = \mathtt{1},\ h(\mathtt{b}) = \mathtt{01},\ h(\mathtt{c}) = \mathtt{001}$ (präfixfrei)
- $u : \{\mathtt{0}, \mathtt{1}\}^* \to \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \bot\}^*$
- es soll gelten:
  - $u(\mathtt{001}) = \mathtt{c}$
  - $u(\mathtt{0101}) = \mathtt{bb}$
  - $u(\mathtt{0}) = \bot$ o. ä.

Präfixfreie Codes: Decodierung (2)

wir schreiben mal hin (und diskutieren das anschließend):

$$u(w) = \begin{cases} \varepsilon, & \text{falls } w = \varepsilon \\ \mathtt{a}u(w'), & \text{falls } w = \mathtt{1}w' \\ \mathtt{b}u(w'), & \text{falls } w = \mathtt{01}w' \\ \mathtt{c}u(w'), & \text{falls } w = \mathtt{001}w' \\ \bot, & \text{sonst} \end{cases}$$

sei $w = \mathtt{100101} = h(\mathtt{acb})$; wir rechnen:

$$\begin{aligned} u(\mathtt{100101}) &= \mathtt{a}u(\mathtt{00101}) \\ &= \mathtt{ac}u(\mathtt{01}) \\ &= \mathtt{acb}u(\varepsilon) \\ &= \mathtt{acb} \end{aligned}$$

Präfixfreie Codes: Decodierung (3)

$$u(w) = \begin{cases} \varepsilon & \text{falls } w = \varepsilon \\ \mathtt{a}u(w') & \text{falls } w = \mathtt{1}w' \\ \mathtt{b}u(w') & \text{falls } w = \mathtt{01}w' \\ \mathtt{c}u(w') & \text{falls } w = \mathtt{001}w' \\ \bot & \text{sonst} \end{cases}$$

$$u(\mathtt{100101}) = \mathtt{a}u(\mathtt{00101}) = \mathtt{ac}u(\mathtt{01}) = \mathtt{acb}u(\varepsilon) = \mathtt{acb}$$

Warum hat das geklappt?

Präfixfreie Codes: Decodierung (3)

$$u(w) = \begin{cases} \varepsilon & \text{falls } w = \varepsilon \\ \mathtt{a}u(w') & \text{falls } w = \mathtt{1}w' \\ \mathtt{b}u(w') & \text{falls } w = \mathtt{01}w' \\ \mathtt{c}u(w') & \text{falls } w = \mathtt{001}w' \\ \bot & \text{sonst} \end{cases}$$

$$u(\mathtt{100101}) = \mathtt{a}u(\mathtt{00101}) = \mathtt{ac}u(\mathtt{01}) = \mathtt{acb}u(\varepsilon) = \mathtt{acb}$$

Warum hat das geklappt?

In jedem Schritt war klar,
welche Zeile der Definition von u anzuwenden war …

Präfixfreie Codes: Decodierung (4)

Man spricht hier von Wohldefiniertheit

Problem, wenn Funktionswert potenziell „auf mehreren Wegen“ festgelegt

dann klar machen:

- entweder nur ein Weg „gangbar“
- oder auf allen Wegen gleicher Funktionswert

präfixfreien Code kann man so decodieren:

$$u(w) = \begin{cases} \varepsilon & \text{falls } w = \varepsilon \\ x\,u(w') & \text{falls } w = h(x)w' \text{ für ein } x \in A \\ \bot & \text{sonst} \end{cases}$$

„so einfach“ geht das nur für *präfixfreie* Codes

Wo sind wir?

Von Wörtern zu Zahlen und zurück

Von einem Alphabet zum anderen

- Übersetzung von Zahldarstellungen
- Homomorphismen
- Beispiel Unicode: UTF-8

Huffman-Codierung

UTF-8 Codierung von Unicode — ein Homomorphismus

Codierung einzelner Zeichen: kommt gleich

Wörter werden zeichenweise codiert.

UTF-8 ist präfixfrei

UTF-8 — Auszug aus RFC 3629

Char. number range (hexadecimal)	UTF-8 octet sequence
0000 0000 - 0000 007F	0*xxxxxxx*
0000 0080 - 0000 07FF	110*xxxxx* 10*xxxxxx*
0000 0800 - 0000 FFFF	1110*xxxx* 10*xxxxxx* 10*xxxxxx*
0001 0000 - 0010 FFFF	11110*xxx* 10*xxxxxx* 10*xxxxxx* 10*xxxxxx*

präfixfrei

Determine the number of octets required …

Prepare the high-order bits of the octets …

Fill in the bits marked x …

- Start by putting the lowest-order bit of the character number in the lowest-order position of the last octet of the sequence,
- then …
- When last octet filled in, move on to the next to last octet, …

Beispiel: UTF-8 Codierung des Integralzeichens $\int$

Unicode Codepoint 0x222B

benutze also die Zeile

Char. number range	UTF-8 octet seq.
0000 0800 - 0000 FFFF	1110*xxxx* 10*xxxxxx* 10*xxxxxx*

0x222B in Bits 0010 0010 0010 1011

also 0010 0010 0010 1011 = 0010 001000 101011

also UTF-8 Codierung 11100010 10001000 10101011

Das ist wichtig

Das sollten Sie mitnehmen:

- Übersetzungen sind in verschiedenen Situationen nützlich
- Homomorphismen
- Codes
- UTF-8

Das sollten Sie üben:

- Homomorphismen anwenden
- Decodieren
- Zeichen in UTF-8 codieren

Von Wörtern zu Zahlen und zurück

Von einem Alphabet zum anderen

Huffman-Codierung

Huffman-Codierung — ein Überblick

gegeben: Alphabet A und Wort $w \in A^*$

Eigenschaften der zu w gehörigen Huffman-Codierung

- eine Abbildung $h : A^* \to Z_2^*$,
- ε-freier Homomorphismus
- h typischerweise auf w angewendet

Bestandteil z. B. von Kompressionsverfahren gzip, bzip2

bei Huffman-Codierungen werden

- häufigere Symbole durch kürzere Wörter codiert und
- seltenere Symbole durch längere

Wo sind wir?

Von Wörtern zu Zahlen und zurück

Von einem Alphabet zum anderen

Huffman-Codierung

- Algorithmus zur Berechnung von Huffman-Codes
- Weiteres zu Huffman-Codes

Voraussetzungen

Gegeben

- $w \in A^*$ und
- die Anzahlen $N_x(w)$ der Vorkommen aller $x \in A$ in w
- o. B. d. A. alle $N_x(w) > 0$

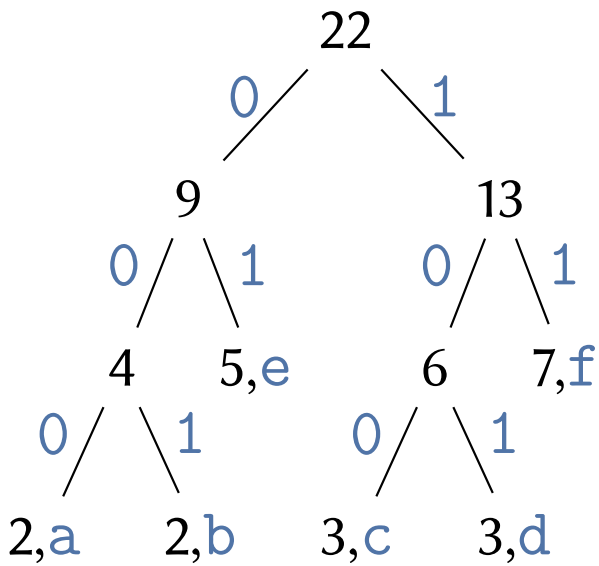
Bestimmung eines Huffman-Codes in zwei Phasen

1. Konstruktion eines „Baumes“
   - Blätter entsprechen den $x \in A$ und
   - Kanten mit 0 und 1 beschriftet
2. Ablesen der Codes aus dem Baum (Pfadbeschriftungen)

Algorithmus für Huffman-Codes

Beispiel: $w =$ `afebfecaffdeddccefbeff`

Baum am Ende:



Homomorphismus (präfixfrei!)

x	a	b	c	d	e	f
h(x)	000	001	100	101	01	11

Konstruktion des Huffman-Baumes (1)

zu jedem Zeitpunkt

- Menge M_i
  „zu betrachtender Symbolmengen mit ihren Häufigkeiten“
- ebenso große Menge schon konstruierter Teilbäume

Initialisierung:

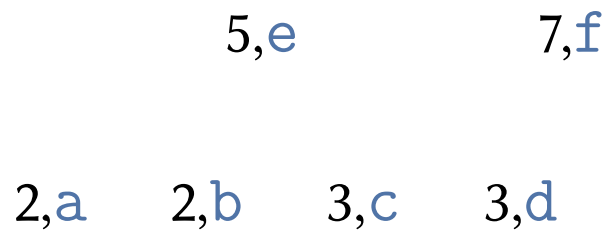
- M_0 ist die Menge aller $\{(N_x(w), \{x\})\}$ für $x \in A$,
- Als Anfang für die Konstruktion des Baumes zeichnet man für jedes Symbol einen Knoten mit Markierung $(N_x(w), \{x\})$.

Beispiel

$$M_0 = \{\ (2, \{\texttt{a}\}),\ (2, \{\texttt{b}\}),\ (3, \{\texttt{c}\}),\ (3, \{\texttt{d}\}),\ (5, \{\texttt{e}\}),\ (7, \{\texttt{f}\})\ \}$$

Konstruktion des Huffman-Baumes (2)

Anfang im Beispiel:



Konstruktion des Huffman-Baumes (3)

Iterationsschritt des Algorithmus:

- Solange M_i noch mindestens zwei Elemente enthält, bestimme M_{i+1} wie folgt:
  - wähle Paare (k_1, X_1) und (k_2, X_2) mit kleinsten Häufigkeiten
  - ersetze diese Paare durch $(k_1 + k_2, X_1 \cup X_2)$

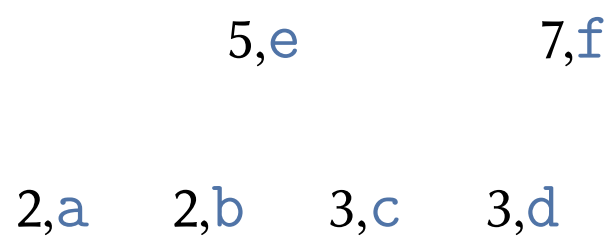
$$M_{i+1} = M_i \setminus \{\ (k_1, X_1),\ (k_2, X_2)\ \} \cup \{\ (k_1 + k_2, X_1 \cup X_2)\ \}$$

- im Graphen
  - neuer Knoten
  - markiert mit Häufigkeit $k_1 + k_2$ und
  - Kanten zu den Knoten für (k_1, X_1) und (k_2, X_2)

Konstruktion des Huffman-Baumes (4)

Beispiel

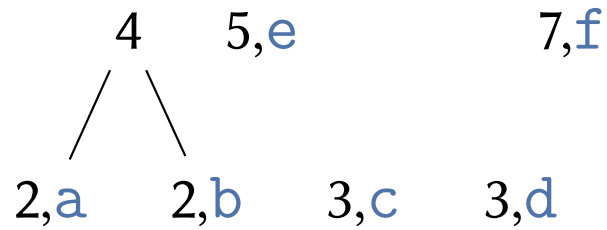
$$M_0 = \{\ (2, \{\texttt{a}\}),\ (2, \{\texttt{b}\}),\ (3, \{\texttt{c}\}),\ (3, \{\texttt{d}\}),\ (5, \{\texttt{e}\}),\ (7, \{\texttt{f}\})\ \}$$



Konstruktion des Huffman-Baumes (4)

Beispiel

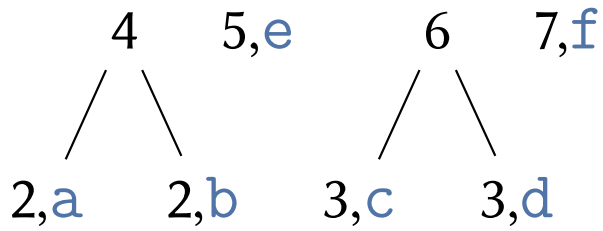
$$M_1 = \{\ (4, \{\texttt{a}, \texttt{b}\}),\ (3, \{\texttt{c}\}),\ (3, \{\texttt{d}\}),\ (5, \{\texttt{e}\}),\ (7, \{\texttt{f}\})\ \}$$



Konstruktion des Huffman-Baumes (5)

Beispiel

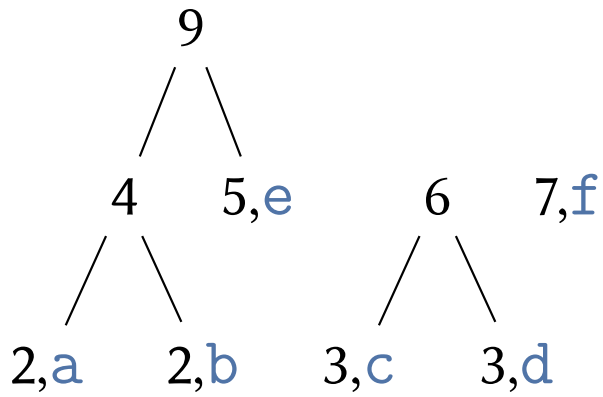
$$M_2 = \{\ (4, \{\texttt{a}, \texttt{b}\}),\ (6, \{\texttt{c}, \texttt{d}\}),\ (5, \{\texttt{e}\}),\ (7, \{\texttt{f}\})\ \}$$



Konstruktion des Huffman-Baumes (6)

Beispiel

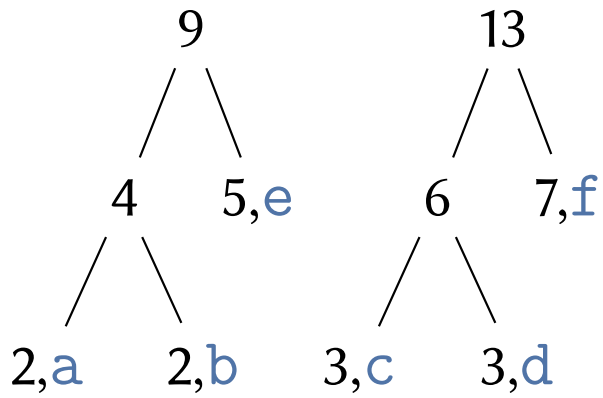
$$M_3 = \{\ (9, \{\texttt{a}, \texttt{b}, \texttt{e}\}),\ (6, \{\texttt{c}, \texttt{d}\}),\ (7, \{\texttt{f}\})\ \}$$



Konstruktion des Huffman-Baumes (7)

Beispiel

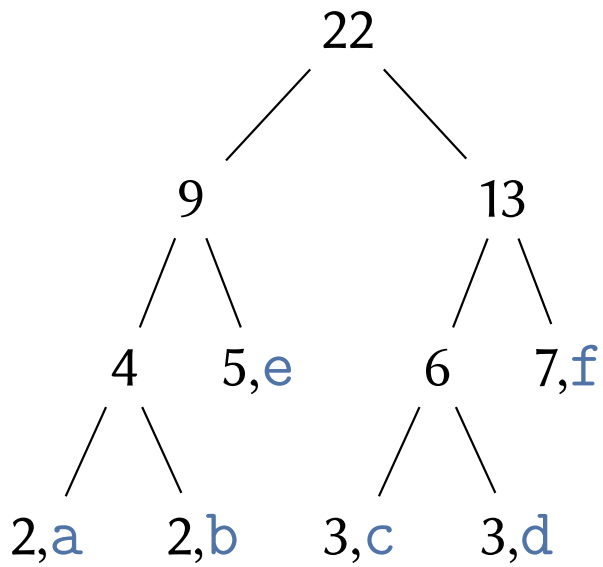
$$M_4 = \{\ (9, \{\texttt{a}, \texttt{b}, \texttt{e}\}),\ (13, \{\texttt{c}, \texttt{d}, \texttt{f}\})\ \}$$



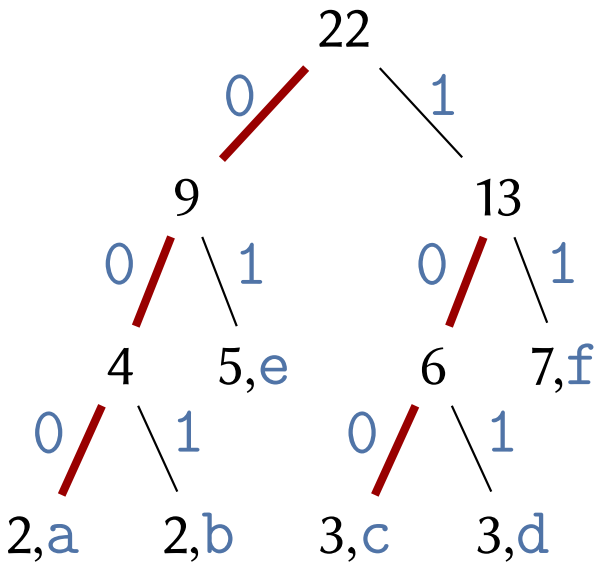
Konstruktion des Huffman-Baumes (8)

Beispiel

$$M_5 = \{\ (22, \{\texttt{a}, \texttt{b}, \texttt{c}, \texttt{d}, \texttt{e}, \texttt{f}\})\ \}$$



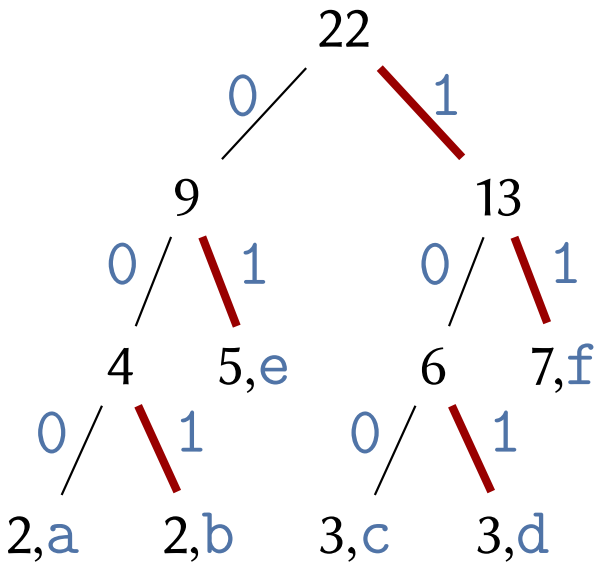
Beschriftung der Kanten



nach links führende Kanten
mit 0 beschriftet

nach rechts führende Kanten
mit 1 beschriftet

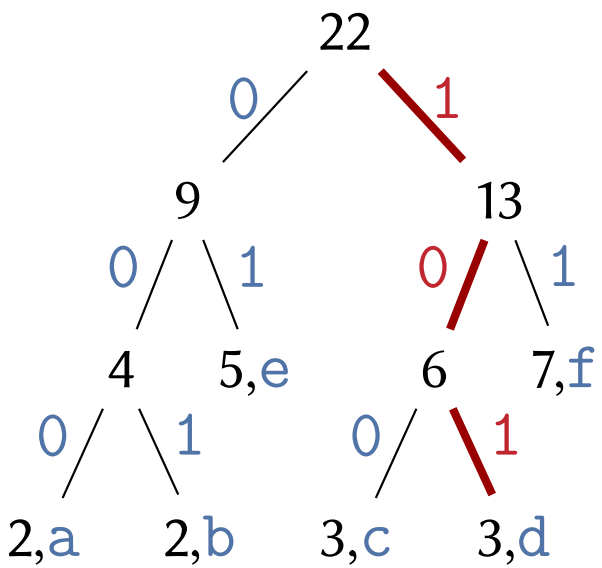
Beschriftung der Kanten



nach links führende Kanten
mit 0 beschriftet

nach rechts führende Kanten
mit 1 beschriftet

Ablesen der Codierungen



gehe auf kürzestem Weg

- von der Wurzel des Baumes
- zu dem Blatt für x

konkateniere der Reihe nach

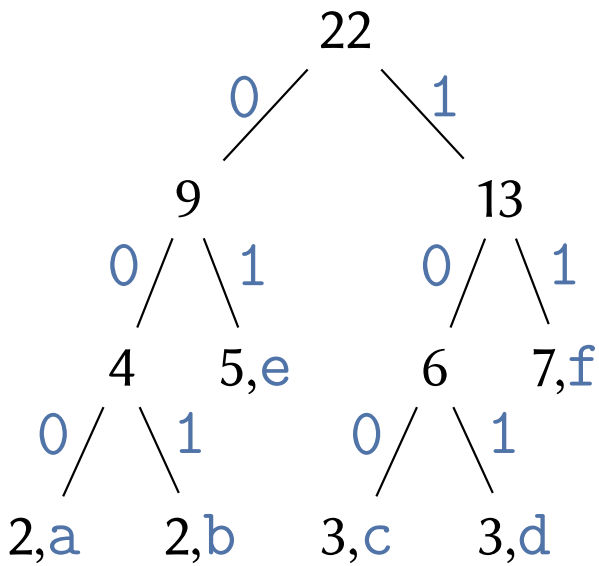
- alle Symbole an den
- Kanten auf diesem Weg

$h(\mathtt{d}) = \mathtt{101}$

Algorithmus für Huffman-Codes

Beispiel: $w =$ `afebfecaffdeddccefbeff`

Baum am Ende:



Homomorphismus (präfixfrei!)

x	a	b	c	d	e	f
h(x)	000	001	100	101	01	11

Wo sind wir?

Von Wörtern zu Zahlen und zurück

Von einem Alphabet zum anderen

Huffman-Codierung

- Algorithmus zur Berechnung von Huffman-Codes
- Weiteres zu Huffman-Codes

Eigenschaften von Huffman-Codes

Huffman-Code nicht eindeutig

- im allgemeinen mehrere Möglichkeiten,
  welche zwei Knoten vereinigt werden
- im Baum links und rechts vertauschbar

aber alle sind „gleich gut“:

- Unter allen präfixfreien Codes
  führen Huffman-Codes zu kürzesten Codierungen
  des Wortes, für das die Huffman-Codierung konstruiert wurde.

Block-Codierungen

Verallgemeinerung des obigen Verfahrens:

- Betrachte nicht Häufigkeiten einzelner Symbole,
- sondern für Teilwörter einer festen Länge $b > 1$.
- einziger Unterschied: an den Blättern des Huffman-Baumes stehen Wörter der Länge b.

So etwas nennt man eine Block-Codierung.

- Statt $h(x)$ für $x \in A$ festzulegen,
- legt man $h(w)$ für alle Blöcke $w \in A^b$ fest, und
- erweitert dies zu einer Funktion $h : (A^b)^* \to B^*$.

Das ist wichtig

Das sollten Sie mitnehmen:

- Huffman-Codierung liefert kürzest mögliche präfixfreie Codes
- „Algorithmus“ zur Bestimmung des Huffman-Baumes

Das sollten Sie üben:

- Huffman-Codes berechnen