

5. Repräsentation von Informationen und Funktionen

5. Repräsentation von Informationen und Funktionen

- Informationen und Funktionen sind wesentliche Elemente von Informations- und Kommunikationssysteme (IKS).

F R A G E:

Wie werden Informationen bzw. Daten dargestellt,
die digital verarbeitet werden sollen?

5. Repräsentation von Informationen und Funktionen

5.1 Darstellung von Informationen durch Daten

Grundlagen:

- Daten bestehen aus *Zeichen*.
- Ein Zeichen z ist ein Element aus einer endlichen Menge Z von (unterscheidbaren) Elementen.
- Z wird auch als *Zeichenvorrat* bezeichnet,
z.B.: $Z_1 = \{A, B, \dots, Z\}$, $Z_2 = \{0, 1, \dots, 9\}$, etc.
- Endliche Zeichenfolge, die als Einheit betrachtet wird, heißt *Wort*,
z.B. 1000.

5. Repräsentation von Informationen und Funktionen

- Digitale Darstellung benutzt den *binären Zeichenvorrat* $B = \{0,1\}$
- Ein Element b aus $B = \{0,1\}$ heißt *Binärzeichen* bzw. *Bit* (binary digit)
- Jedes Datum bestehend aus beliebigen Zeichen wird *eindeutig* auf eine Bitfolge abgebildet (Code)
- International standardisierter Code für eine mögliche Abbildung:

→ **ASCII** (American Standard Code for Information Interchange)

Der ASCII-Code

- ASCII ist eine Abbildung der Form: $A \rightarrow B^7$
- Die Menge A besteht aus den großen und kleinen Buchstaben, den Dezimalziffern und einer Reihe von Sonderzeichen.
- B^7 ist die Menge der 7-stelligen Worte über dem Alphabet $B = \{0,1\}$, d.h.
 $B^7 = B \times B \times B \times B \times B \times B \times B$
- Mächtigkeit der Menge B^7 beträgt $|B^7| = 2^7 = 128$ Elemente; d.h. A kann maximal 128 Zeichen umfassen.

5. Repräsentation von Informationen und Funktionen

ASCII Codierung

16 × 8	000	001	010	011	100	101	110	111
0000	NUL	DLE	SP	0	@	P	`	p
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	"	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EOT	DC4	\$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	'	7	G	W	g	w
1000	BS	CAN	(	8	H	X	h	x
1001	HT	EM	)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[	k	{
1100	FF	FS	,	<	L	\	l	
1101	CR	GS	-	=	M	]	m	}
1110	SO	RS	.	>	N	^	n	~
1111	SI	US	/	?	O	_	o	DEL

Der HEX-Code

- Manuelle Schreibweise langer Bitfolgen ist unhandlich.

→ Kompaktere Codierung durch **HEX-Code**:

HEX: $B^4 \Rightarrow S$ mit $S = \{0, 1, \dots, 9, A, B, \dots, F\}$, $|S|=16$

- Die Elemente von S heißen *Hexadezimalziffern*.

5. Repräsentation von Informationen und Funktionen

HEX Codierung

Bit- folge	Hex- ziffer	dez. Wert	Bit- folge	Hex- ziffer	dez. Wert
0000	0	0	1000	8	8
0001	1	1	1001	9	9
0010	2	2	1010	A	10
0011	3	3	1011	B	11
0100	4	4	1100	C	12
0101	5	5	1101	D	13
0110	6	6	1110	E	14
0111	7	7	1111	F	15

5. Repräsentation von Informationen und Funktionen

- Bei der Codierung von ASCII-Zeichen mit Hexadezimalziffern wird eine erweiterte Wortmenge B^8 verwendet:

$$A \rightarrow B^8 \rightarrow S^2$$

z.B. der Buchstabe 'A' ist codiert als $0100\ 0001_B$ oder 41_H

→ Die Indizes B und H identifizieren den Code als Bitfolgen bzw. Folgen von Hexadezimalziffern

- Weitere Beispiele für Codes:

- Extended ASCII (eASCII: $A' \Rightarrow B^8$)
- Extended Binary Coded Decimal Interchange Code (EBCDIC: $A'' \Rightarrow B^8$)

5. Repräsentation von Informationen und Funktionen

Beispiele für eASCII und EBCDIC Codierung

Zeichen	eASCII	dezimal	EBCDIC	dezimal
1	0011 0001	(49)	1111 0001	(241)
5	0011 0101	(53)	1111 0101	(245)
9	0011 1001	(57)	1111 1001	(249)
A	0100 0001	(65)	1100 0001	(193)
a	0110 0001	(97)	1000 0001	(129)
R	0101 0010	(82)	1101 1001	(217)
r	0111 0010	(114)	1001 1001	(153)
T	0101 0100	(84)	1110 0011	(227)
t	0111 0100	(116)	1010 0011	(163)
+	0010 1011	(43)	0100 1110	(78)
?	0011 1111	(63)	0110 1111	(111)

5. Repräsentation von Informationen und Funktionen

Darstellung von Zahlen

- **Dezimal:** {0, 1, 2, 3, 4, 5, 6, 7, 8, 9} und Basis 10
Beispiel: $20_{10} = 2 \cdot 10^1 + 0 \cdot 10^0$
- **Dual:** {0, 1} und Basis 2
Beispiel: $10.100_2 = 1 \cdot 2^4 + 1 \cdot 2^2$
- **Oktal:** {0, 1, 2, 3, 4, 5, 6, 7} und Basis 8
Beispiel: $24_8 = 2 \cdot 8^1 + 4 \cdot 8^0$
- **Hexadezimal:** {0, ..., 9, A, B, ..., F} und Basis 16
Beispiel: $14_{16} = 1 \cdot 16^1 + 4 \cdot 16^0$
- **Allgemein:** $Z = \text{SUM}_{i=0, \dots, n-1} z_i \cdot b^i$
mit Z als Wert einer Zahl, z_i als Ziffern und b als Basis

5. Repräsentation von Informationen und Funktionen

Beispiele für Zahlendarstellung

Dezimal	Dual	Oktal	Hexadezimal
1	00001	01	01
2	00010	02	02
3	00011	03	03
4	00100	04	04
5	00101	05	05
6	00110	06	06
7	00111	07	07
8	01000	10	10
17	10001	21	11
18	10010	22	12
19	10011	23	13
20	10100	24	14

5. Repräsentation von Informationen und Funktionen

Dezimalcode

Dezimalcode: Darstellung von Zahlen des Dezimalsystems durch ziffernweise Codierung mit ASCII unter Verwendung von $B = \{0,1\}$ (vgl. Tab. S.5).

Darstellung von Zahlen mit Dezimalcode ist unwirtschaftlich, da für die Codierung der zehn Ziffern des Dezimalsystems nur 10 von 128 ASCII-Zeichen bzw. 10 von 256 eASCII-Zeichen verwendet werden.

Dezimalcode eignet sich nur bedingt zur Durchführung arithmetischer Operationen in Rechnern.



Einführung neuer Codes!

Binärer Festpunktcode

Binärer Festpunktcode: Beruht auf der Transformation einer positiven ganzen Dezimalzahl Z in eine Zahl zur Basis 2 (duale Darstellung)

→ Die Koeffizienten c_i sind die Bits 0, 1

→ Ziffern von Z : $c_{n-1} c_{n-2} \dots c_0$

→ Wert von Z : $Z = c_{n-1} * 2^{n-1} + c_{n-2} * 2^{n-2} + \dots + c_0 * 2^0$

5. Repräsentation von Informationen und Funktionen

Vergleich:

Im Dezimalsystem dagegen wird Z als Polynom mit der Basis 10 dargestellt (dezimale Darstellung).

Die Koeffizienten z_i sind Dezimalziffern $0,1,2,\dots,9$

Ziffern von Z :

$$z_{n-1} \dots z_0$$

Wert von Z :

$$Z = z_{n-1} * 10^{n-1} + z_0 * 10^0$$

→ In den Tabellen auf S.7 und 9 lassen sich entsprechende Beispiele für $n=4$ und $n=8$ finden.

Binärer Gleitpunktcode

Eine reelle Zahl r lässt sich darstellen als $r = m * b^e$

m heißt Mantisse, e Exponent zur Basis b

- e ist eine ganze Zahl,
- m wird so gewählt, daß $b^{-1} < |m| < 1$ oder $m=0$ gilt,
- die Vorzeichen von m und e lassen sich berücksichtigen.

Reelle Zahl wird in Mantisse und Exponent zerlegt.

Binärer und dezimaler Gleitpunktcode

Binärer Gleitpunktcode:

- Basis $b = 2$,
- Mantisse und Exponent werden im binären Festpunktcode dargestellt.

Dezimaler Gleitpunktcode:

- $r = 3,14159$ wird dargestellt als $0,314159 * 10^1$ mit $m = 0,314159$, $b=10$ und $e=1$.
- m und e werden als Tupel $(314159,1)$ dargestellt.

5.2 Datenstruktur und Datenorganisation

Relationen

Häufig werden Daten durch Relationen (Teilmenge des Kartesischen Produkts von Mengen) repräsentiert.

z.B.: $Q_1 = \{\text{rot, grün, gelb, blau, weiß, schwarz}\}$

$Q_2 = \{1, 2, 3, 4, 5, 6\}$

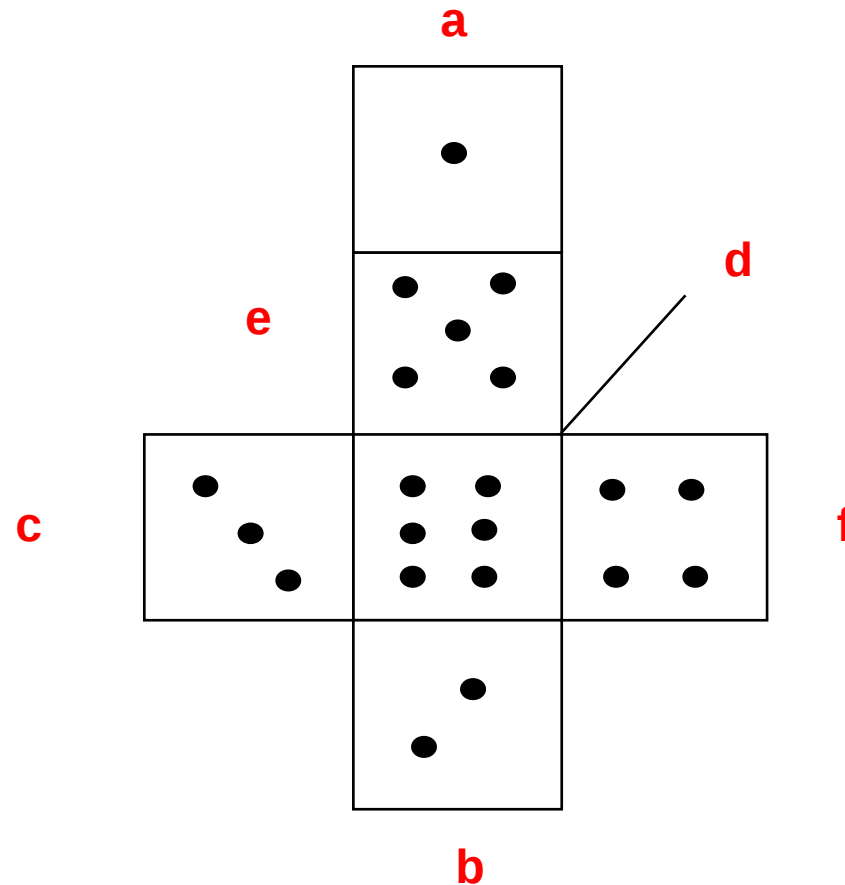
$R = Q_1 \times Q_2 = \{(\text{rot},1), (\text{rot},2), \dots (\text{rot},6), \dots, (\text{gelb},3), \dots, (\text{schwarz},6)\}$

→ Relationen lassen sich als Mengen von Tupeln (Einheiten) darstellen.

5. Repräsentation von Informationen und Funktionen

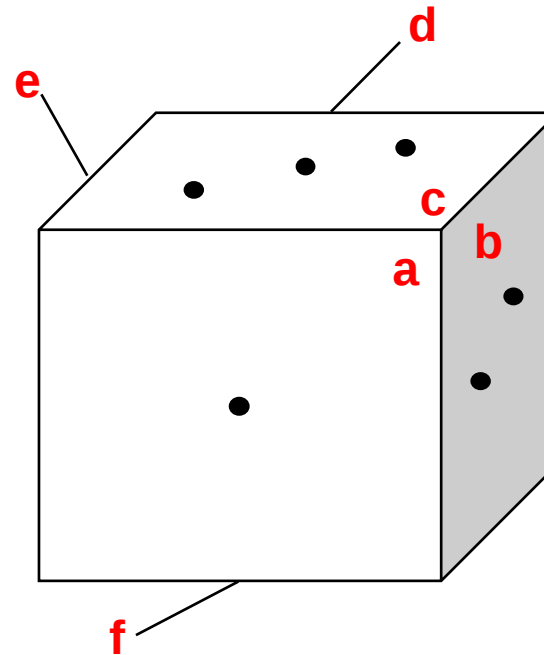
Beispiel:

Ein Würfel soll mit Hilfe von (Nachbarschafts-)Relationen beschrieben werden.



5. Repräsentation von Informationen und Funktionen

Fortsetzung Beispiel:



(x,y)-Relation:

'x grenzt an y', mit $x \neq y$ und $x, y \in S$ mit $S = \{a, b, c, d, e, f\}$

Hülle eines Würfels wird beschrieben durch die Relation:

$H = \{(a,b), (a,c), (a,e), (a,f), (b,c), (b,d), (b,f), (c,e), (c,d), (d,e), (d,f), (e,f)\}.$

5. Repräsentation von Informationen und Funktionen

Relation als Tabelle

→ Eine einfache Darstellungsform für Relationen sind Tabellen

Fortsetzung Beispiel: Relation Würfel $W = (a, b, c, d, e, f)$

Spalten: Flächen eines Würfels; Zeilen: verschiedene Würfel

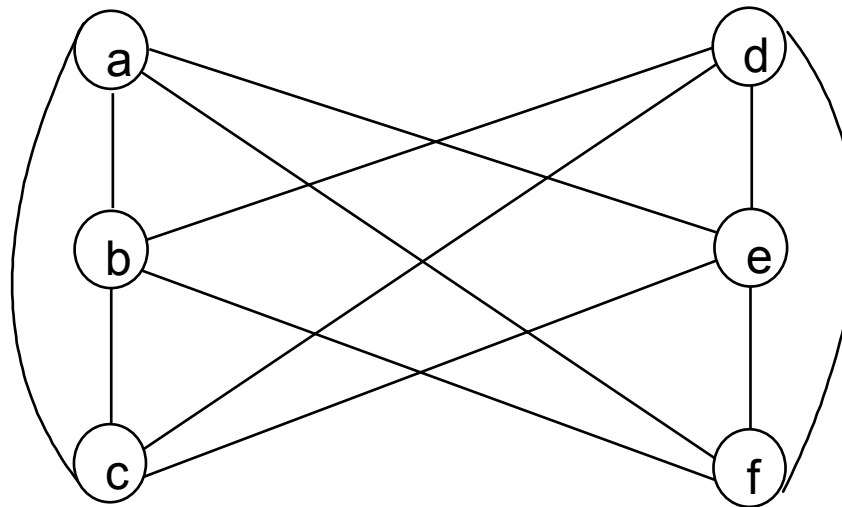
Würfel	a	b	c	d	e	f
Farbe (W1)	rot	grün	blau	gelb	schwarz	weiß
Farbe (W2)	blau	rot	gelb	schwarz	weiß	grün
Zahl (W1)	1	2	3	6	5	4
Zahl (W2)	3	6	5	4	1	2
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.

5. Repräsentation von Informationen und Funktionen

Relation als Graph

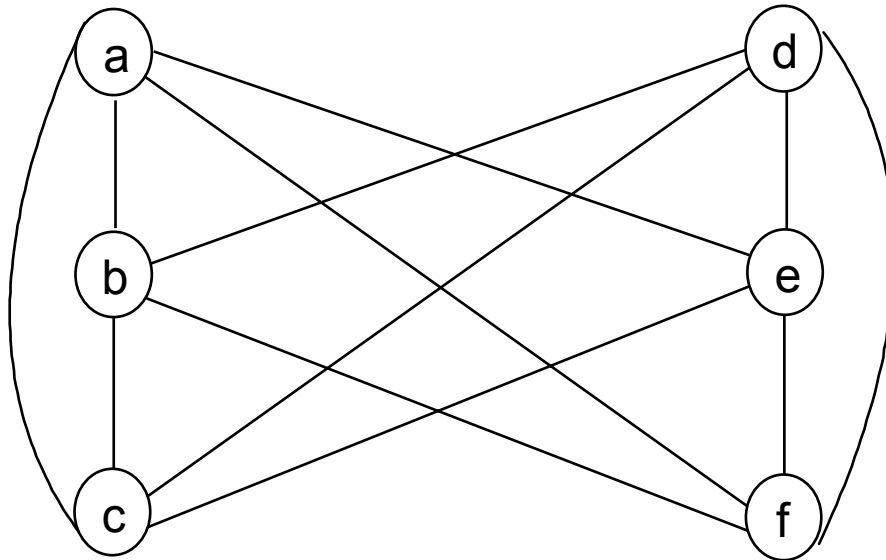
Fortsetzung Beispiel: Darstellung der Relation Würfel als Graph.

- Knoten x und y , Kanten (x,y) .
- Knoten repräsentiert Fläche und Kante eine (Nachbarschafts)-Relation zwischen Paaren von Flächen.



5. Repräsentation von Informationen und Funktionen

Nachbarschaftsrelation zwischen Paaren und Flächen (Beispiel):



	a	b	c	d	e	f
a	0	1	1	0	1	1
b	1	0	1	1	0	1
c	1	1	0	1	1	0
d				.		
e				.		
f				.		

Organisation von Listen

Elementare Zusammenhänge zwischen Daten:

- Kleinste abzubildende Einheit ist ein *Datenelement*.
- Mehrere Datenelemente bilden einen *Datensatz*.
- Mehrere Datensätze bilden eine *Datei*.
- *Schlüssel* dienen zur eindeutigen Identifikation von Daten.

Lineare Felder

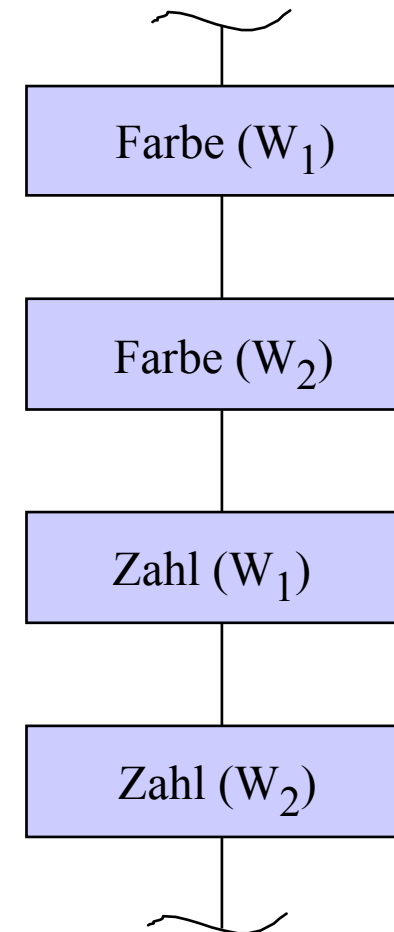
- Lineares Feld (Liste) lässt sich als Spalte oder Zeile einer Tabelle vorstellen.
- Eine Zeile hat so viele Datenelemente wie es Spalten gibt.
- Eine Spalte hat so viele Datenelemente wie es Zeilen gibt.
- Dateioperationen: *Suchen*, *Ändern*, *Einfügen* und *Löschen* eines Satzes.

5. Repräsentation von Informationen und Funktionen

Beispiel für Listen

Repräsentation von Listen:

- Jedes Datenelement einer Liste hat nur ein einziges Nachfolger-Datenelement.
- Dies läßt sich auf mehrere Arten abbilden:
 - sequenziell
 - gestreut:
 - verkettet
 - unverkettet



Sequenzielle Speicherung

- Felder werden in einer unverketteten, (sortierten) Reihenfolge abgelegt.
- Mühsames Hinzufügen (durch Verschiebung), Suchen und Löschen.
- Einfügen eines Eintrags mit dem Schlüssel cd.

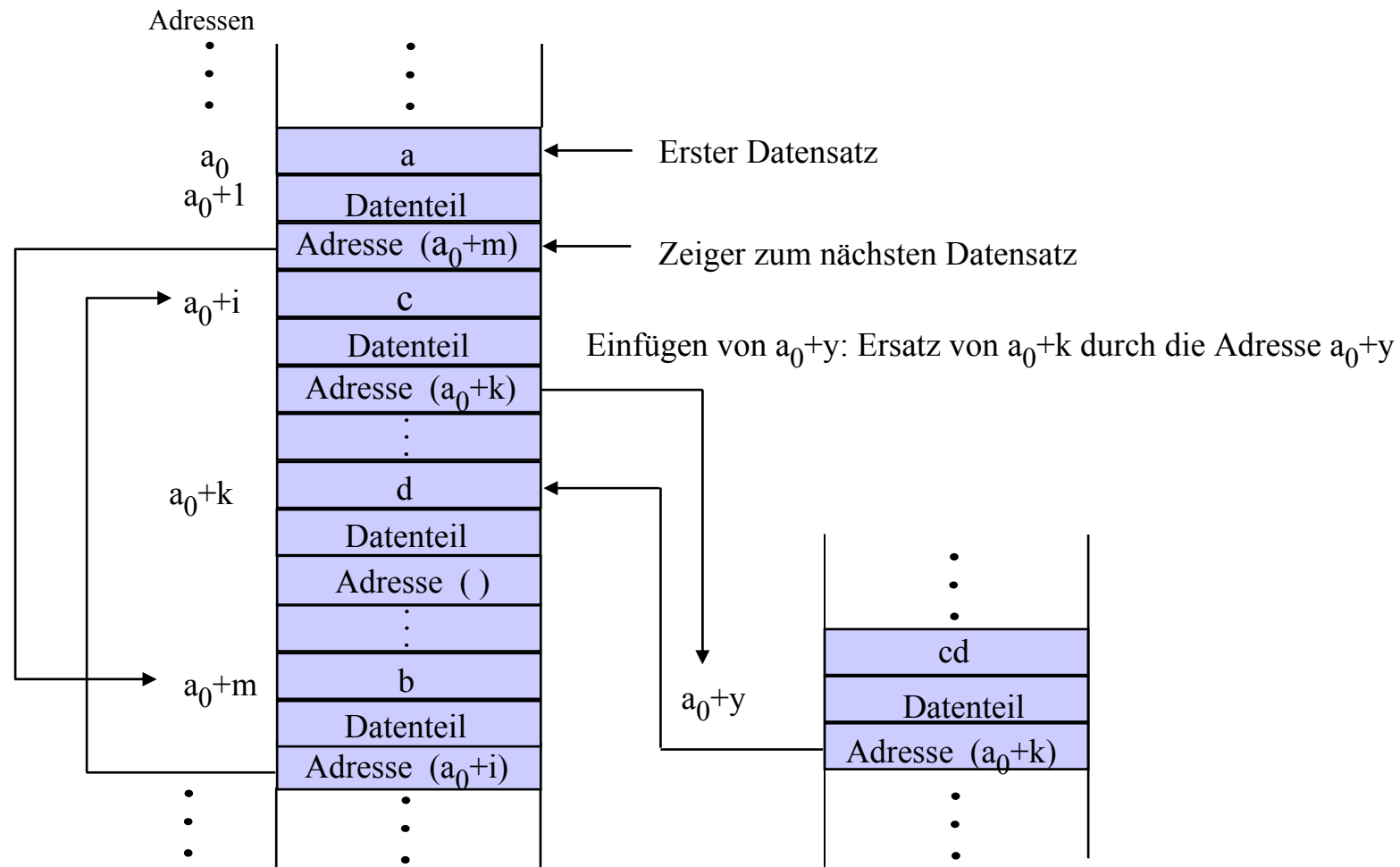
Adressen- beginn	
⋮	⋮
a_0	Schlüssel a
a_0+1	Farbe (W_1)
a_0+2	Schlüssel b
	Farbe (W_2)
a_0+4	Schlüssel c
	Zahl (W_1)
a_0+6	Schlüssel d
	Zahl (W_2)
⋮	⋮

Gestreute Speicherung mit Verkettung

- Zeigerfelder zur Anzeige des Startfeldes, eines folgenden Feldes und des Endes einer Verkettung.
- Einfaches Hinzufügen ohne eine Verschiebung von Datensätzen.

5. Repräsentation von Informationen und Funktionen

Gestreute Speicherung mit Verkettung



Gestreute Speicherung ohne Verkettung

- Errechnung einer Speicheradresse aus einem Suchschlüssel.
- Funktion, die aus einem numerischen Schlüssel (NS) eine Speicheradresse errechnet, heißt *Hash-Funktion*.

5. Repräsentation von Informationen und Funktionen

Beispiel für das Arbeiten mit Hash-Funktionen:

200 Datensätze mit den numerischen Schlüsseln 100 bis 299 zu je 10 Sätzen pro Spur sollen den Spuren 20 bis 39 einer Speicherplatte zugeordnet werden.

→ Speicherfunktion:

$(NS-100)/10 = q$ mit Rest r ; Spur-Nummer: $q+20$; Position auf Spur: $r+1$;
d.h der Datensatz mit dem numerischen Schlüssel 150 hat die Speicher-
adresse:

→ **Spur 25, Position 1**