*Technische Universität Berlin*


# Technische Grundlagen der Informatik 1

# Digitale Systeme

Technische Universität Berlin
Fakultät IV • Informatik und Elektrotechnik
Institut für Technische Informatik
FG Rechnertechnologie
Franklinstraße 28/29 • D-10587 Berlin

# Vorwort

Wir haben uns in diesem Semester entschlossen ein Skript anzubieten, das mit etwas Glück diesen Namen auch verdient. Auch wenn der Inhalt des Skripts nahezu deckungsgleich mit den Folien der Vorlesung ist, so erhoffen wir uns von einem Skript dennoch Vorteile gegenüber dem Abdrucken der Folien (Handouts). Zum einen ist es natürlich kompakter, zum anderen hoffen wir, dass es auch etwas lesbarer ist und dadurch vielleicht eher einen Nachschlagecharakter hat. Beim Neugestalten haben wir auch versucht die Abbildungen aneinander anzugleichen und im Aussehen konsistenter zu machen, sodass ein höherer Wiedererkennungswert zwischen den einzelnen technischen Darstellungen in den unterschiedlichen Kapiteln gegeben ist. Bedanken wollen wir uns vor allem auch bei den Tutoren, die uns beim Umsetzen des Vorhabens unter die Arme gegriffen haben:

Sven-Garrit Czarnian
Carsten Dülsen
Thomas Faber
Matthias Hartmann
Evren Kücükbayraktar
Marcus Schubert
Jan Wetter

Wir hoffen, dass die Arbeit nicht umsonst war und wenigstens ein paar Studierenden gefällt.

Carsten Gremzow
Nico Moser

# Inhaltsverzeichnis

# 1 Zweiwertige Logik

## 1.1 Grundbegriffe der Logik

### 1.1.1 Aussagen

Jede wörtlich formulierte Behauptung ist eine Aussage (z.B. “3 + 5 = 8”,“er läuft”). Von solchen Sätzen lässt sich im Allgemeinen bestimmen, ob sie zutreffen oder nicht, d.h. ob sie wahr oder falsch sind. Es lässt sich somit der Wahrheitswert einer Aussage bestimmen.

**Definition 1.** *Ein Satz, dem eindeutig ein Wahrheitswert zugeordnet werden kann, heißt Aussage. Ist A eine Aussage, dann ist W(A) der Wahrheitswert der Aussage mit: W(A) := wahr (falsch), falls Aussage A zutrifft (falls Aussage A nicht zutrifft).*

#### Verknüpfungen von Aussagen

Mehrere Aussagen lassen sich zu einer neuen Aussage zusammenfassen – auch dafür lässt sich ein Wahrheitswert finden. Diese Zusammenfassung von Einzelaussagen wird umgangssprachlich mit bestimmten Wörtern gebildet.

**Beispiel 1.** *Verknüpfungen von Aussagen*
*5 ist eine Primzahl und 10 ist durch 5 teilbar.*
*Ist a durch 4 teilbar, dann ist a auch durch 2 teilbar.*

Allgemein für A1 und A2 als zwei Aussagen gilt:

| | |
|---|---|
| A1 und A2 | Konjunktion |
| A1 oder A2 | (einschließendes) Oder, Disjunktion |
| entweder A1 oder A2 | ausschließendes Oder, Antivalenz |
| wenn A1 dann A2 | Folgerung, Implikation |
| A1 genau dann, wenn A2 | Äquivalenz |

Die Bindewörter verknüpfen Aussagen. Jeder dieser aus Aussagen zusammengesetzte Satz ist selbst wieder eine Aussage. Wahrheitswerte der Einzelaussagen stehen dem Wahrheitswert der zusammengesetzten Aussagen gegenüber.

### Wahrheitswert der Verknüpfung

Aussagenlogik betrachtet unter Vernachlässigung des Inhalts der Aussagen nur die Wahrheitswerte. Für entsprechende Verknüpfungen zwischen den Wahrheitswerten soll gelten:

*Verknüpfung der Wahrheitswerte der Einzelaussagen =*
*Wahrheitswert der zusammengesetzten Aussage*

Ist z. B. "A1 und A2" eine zusammengesetzte Aussage, dann soll gelten:

$$\mathsf{W(A1)\ und\ W(A2)\ =\ W(A1\ und\ A2)}$$

Diese Gleichung muss für alle möglichen Kombinationen der Wahrheitswerte von A1 und A2 gelten.

### Wertetabelle, Wahrheitstabelle

Die Aussage "A1 und A2" ist nur dann eine wahre Aussage, wenn beide Einzelaussagen A1 und A2 zutreffen. Für die entsprechende Verknüpfung lässt sich eine Wertetabelle bzw. Wahrheitstabelle aufstellen. Für die Konjunktion (Symbol: $\wedge$) gilt folgende Wertetabelle:

| W(A1) | W(A2) | W(A1) $\wedge$ W(A2) |
|---|---|---|
| falsch | falsch | falsch |
| falsch | wahr | falsch |
| wahr | falsch | falsch |
| wahr | wahr | wahr |

### Vereinfachung

Wahr und falsch wird durch w und f oder durch 1 und 0 dargestellt. w und f sind innerhalb der Aussagenlogik üblich, 1 und 0 in der formalen Logik. Für die Wertetabelle der Konjunktion oder UND-Verknüpfung gilt dann:

| W(A1) | W(A2) | W(A1) $\wedge$ W(A2) |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

### Logische Operationen

Entsprechend der Wertetabelle für die Konjunktion, lässt sich eine Tabelle für alle anderen zusammengesetzten Aussagen aufstellen. Diese Verknüpfungen werden auch logische Operationen genannt. Entsprechend der Anzahl der Wahrheitswerte oder Operanden, die miteinander verknüpft werden, spricht man von ein-, zwei- oder allgemein n-stelligen Operationen. Die Konjunktion ist ein Beispiel für eine zweistellige Operation.

#### Negation

Aussagen lassen sich auch verneinen. Dementsprechend gibt es in der Aussagenlogik die einstellige Operation der Negation (Symbol: $\neg$).

#### Erkenntnis

Die Aussagenlogik untersucht die formalen Zusammenhänge zwischen Aussagen. Diese können entweder wahr oder falsch sein.

### 1.1.2 Grundverknüpfungen

#### Logische Operationen

Die Grundverknüpfungen werden als logische Operationen durch Symbole dargestellt. Für die veränderlichen Operanden wird der Begriff der logischen Variablen eingeführt.

**Definition 2.** *Ein Zeichen, welches anstelle eines Elementes aus* $\{0, 1\}$ *steht, heißt logische Variable. Eine Zuordnung konkreter Werte zu den Variablen heißt Belegung der Variablen.*

#### Schaltalgebra

Eine Sonderform der Booleschen Algebra ist die Schaltalgebra. Mit ihrer Hilfe lassen sich Digitalschaltungen berechnen und vereinfachen. Die Schaltalgebra kennt Variablen und Konstanten. Es gibt nur zwei mögliche Konstanten: 0 und 1. Die Konstanten entsprechen den logischen Zuständen 0 und 1. Eine Variable kann entweder den Wert 0 oder 1 annehmen. Jede Größe, mit Wert 0 oder Wert 1, stellt eine Variable dar. Ausdrücke wie $(a \wedge b)$, die z. B. aus zwei Variablen bestehen, können wie eine Variable behandelt werden, denn ihr Wert kann ebenfalls nur 0 oder 1 sein. Eine Variable der Schaltalgebra ist also eine binäre Größe.

| Operation | Symbolik | Bindewörter | Wahrheitstabelle |
|---|---|---|---|
| Negation | $\bar{a}$ | nicht a | a \| $\bar{a}$<br>0 \| 1<br>1 \| 0 |
| Konjunktion | $a \wedge b, a \cdot b$ | a und b | a b \| $a \wedge b$<br>0 0 \| 0<br>0 1 \| 0<br>1 0 \| 0<br>1 1 \| 1 |
| Disjunktion | $a \vee b, a + b$ | a oder b | a b \| $a \vee b$<br>0 0 \| 0<br>0 1 \| 1<br>1 0 \| 1<br>1 1 \| 1 |
| Äquivalenz | $a \equiv b, a \leftrightarrow b$ | a äquivalent b | a b \| $a \equiv b$<br>0 0 \| 1<br>0 1 \| 0<br>1 0 \| 0<br>1 1 \| 1 |
| Antivalenz | $a \not\equiv b, a \oplus b$<br>exklusives oder | a antivalent b | a b \| $a \not\equiv b$<br>0 0 \| 0<br>0 1 \| 1<br>1 0 \| 1<br>1 1 \| 0 |
| Implikation | $a \rightarrow b$ | a impliziert b | a b \| $a \rightarrow b$<br>0 0 \| 1<br>0 1 \| 1<br>1 0 \| 0<br>1 1 \| 1 |
| Sheffer-Funktion (NAND) | $\overline{a \wedge b}, \overline{a \cdot b}, a \mid b$<br>NAND | a und b nicht | a b \| $\overline{a \wedge b}$<br>0 0 \| 1<br>0 1 \| 1<br>1 0 \| 1<br>1 1 \| 0 |
| Peirce-Funktion (NOR) | $\overline{a \vee b}, \overline{a + b}, a \downarrow b$ | weder a noch b | a b \| $\overline{a \vee b}$<br>0 0 \| 1<br>0 1 \| 0<br>1 0 \| 0<br>1 1 \| 0 |

### 2-stellige Operationen

Neben den Grundverknüpfungen sind weitere Verknüpfungen zwischen a und b denkbar. In einer Wertetabelle für zwei Operanden kann jede beliebige Kombination von

| a | b | | $a < b$ | | a | | b | $a \not\equiv b$ | $a > b$ | $\overline{a > b}$ | $a \equiv b$ | $\overline{b}$ | | $\overline{a}$ | $a \rightarrow b$ | $\overline{a < b}$ | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |

Tabelle 1.1: mögliche Kombinationen 2-stelliger logischer Operationen

0-1-Werten eingesetzt werden, was eine zweistellige Operation ergibt. Nicht alle diese Operationen haben einen Namen oder eine symbolische Bezeichnung. Es ergeben sich $2^{2^2} = 16$ Möglichkeiten (siehe Tabelle 1.1).

#### 1-stellige Operationen

Für einstellige Operationen, wie z.B. die Negation ergeben sich $2^{2^1} = 4$ Möglichkeiten:

| a | | a | $\overline{a}$ | |
|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 |

#### 0-stellige Operationen

Konstanten werden als nullstellige Operationen bezeichnet. Es existieren $2^{2^0} = 2$ verschiedene Anordnungen von 0 und 1. Die Konstanten haben somit die Werte 0 (Kontradiktion/nie) oder 1 (Tautologie/immer).

### 1.1.3 Ausdrücke

#### n-stellige Operationen

Verknüpfungen von nur zwei Operanden sind im Allgemeinen nicht ausreichend und bei mehrstelligen Operationen wächst die Zahl der möglichen Verknüpfungen rasch an. Bei n-stelligen Operationen ergeben sich $2^n$ verschiedene Möglichkeiten für die Belegung der Variablen. Diesen Belegungen können wieder $2^{2^n}$ 0-1-Kombinationen als Ergebnisse zugeordnet werden. Mehr als zweistellige Operationen lassen sich durch 1- und 2-stellige Operationen ausdrücken. Die Grundverknüpfungen genügen, um sämtliche weiteren logischen Zusammenhänge zu beschreiben. Dazu kombiniert man verschiedene Operationen, was zum Begriff des Ausdrucks führt.

**Definition 3.** *Eine Verknüpfung von endlich vielen Konstanten und logischen Variablen mittels Grundverknüpfungen heißt ein Ausdruck. Die Reihenfolge, in der Operationen anzuwenden sind, wird durch Klammern bestimmt. Ausdrücke die nur aus den Operationen $\overline{\ }$ (Negation), $\wedge$ (Konjunktion) und $\vee$ (Disjunktion) zusammengesetzt sind, heißen boolesche Ausdrücke.*

**Beispiel 2.** *Ausdrücke*
$\overline{a}$
$(a \wedge b) \vee 1$
$(a \rightarrow b) \wedge (c \downarrow a)$
$((((a \vee \overline{b} \leftrightarrow c) \oplus a) \vee c)$
$((a \wedge b) \vee (\overline{a} \wedge \overline{b}))$
$(((\overline{a \wedge b}) \vee (b \wedge \overline{c})) \wedge 1$
*kein Ausdruck ist:*
$a \rightarrow b \oplus c$
*weil die Reihenfolge der Operationen nicht eindeutig ist*

### Ermittlung des Wahrheitswerts eines Ausdrucks

Die aktuellen Variablenwerte werden in den Ausdruck eingesetzt und schrittweise der Klammerung entsprechend abgearbeitet. Dabei werden jeweils die Grundverknüpfungen durch den Wahrheitswert ersetzt, der durch die zugehörige Wertetabelle gegeben ist.

**Beispiel 3.** *Mit* $a = 1$, $b = 0$, $c = 1$ *gilt für den Wahrheitswert des Ausdrucks* $(a \rightarrow b) \wedge (c \downarrow a)$ *:*

$$(a \rightarrow b) \wedge (c \downarrow a) \;=\; (1 \rightarrow 0) \wedge (1 \downarrow 1) \;=\; 0 \wedge 0 \;=\; 0$$

Soll der Ausdruck vollständig durch eine Wertetabelle beschrieben werden, ist es meist sinnvoll, auch Zwischenschritte mit in der Tabelle zu notieren.

**Beispiel 4.** *Die Verknüpfung* $(a \rightarrow b) \wedge (c \downarrow a)$ *soll durch eine Wertetabelle vollständig beschrieben werden:*

| a | b | c | $(a \rightarrow b)$ | $(c \downarrow a)$ | $(a \rightarrow b) \wedge (c \downarrow a)$ |
|---|---|---|---|---|---|
| *0* | *0* | *0* | *1* | *1* | *1* |
| *0* | *0* | *1* | *1* | *0* | *0* |
| *0* | *1* | *0* | *1* | *1* | *1* |
| *0* | *1* | *1* | *1* | *0* | *0* |
| *1* | *0* | *0* | *0* | *0* | *0* |
| *1* | *0* | *1* | *0* | *0* | *0* |
| *1* | *1* | *0* | *1* | *0* | *0* |
| *1* | *1* | *1* | *1* | *0* | *0* |

### Äquivalente Ausdrücke

**Definition 4.** *Zwei Ausdrücke A1, A2 heißen äquivalent (A1 = A2) genau dann, wenn sie bei gleicher Belegung gemeinsamer Variablen stets gleiche Wahrheitswerte annehmen. A1 = A2 heißt dann eine logische Gleichung.*

Dabei ist nicht gefordert, dass beide Ausdrücke genau dieselben Variablen enthalten.

**Beispiel 5.** *Seien* $\mathsf{a}, \mathsf{b}, \mathsf{c}$ *logische Variable.*
*Behauptung:* $\mathsf{a} \vee (\mathsf{b} \wedge \overline{\mathsf{b}}) = \mathsf{a} \vee (\mathsf{c} \wedge \overline{\mathsf{c}})$
*Beweis (durch Wertetabelle):*

| a | b | c | $(\mathsf{a} \vee (\mathsf{b} \wedge \overline{\mathsf{b}}))$ | $\mathsf{a} \vee (\mathsf{c} \wedge \overline{\mathsf{c}})$ |
|---|---|---|---|---|
| *0* | *0* | *0* | *0* | *0* |
| *0* | *0* | *1* | *0* | *0* |
| *0* | *1* | *0* | *0* | *0* |
| *0* | *1* | *1* | *0* | *0* |
| *1* | *0* | *0* | *1* | *1* |
| *1* | *0* | *1* | *1* | *1* |
| *1* | *1* | *0* | *1* | *1* |
| *1* | *1* | *1* | *1* | *1* |

*Beide Ausdrücke hängen weder von b noch von c ab, und gerade dann gilt auch nur die Äquivalenz unter derartigen Ausdrücken. Solche überflüssigen (redundante oder fiktive) Variablen können eliminiert werden.*

### Vorrangregeln

Bei komplexeren Ausdrücken kann die Zahl der Klammern stark zunehmen. Zur Vermeidung zahlreicher Klammern werden unter den wichtigsten Grundverknüpfungen, Vorrangregeln eingeführt. Das geschieht ähnlich wie unter den arithmetischen Rechenarten („Punktrechnung vor Strichrechnung", etc.).

**Definition 5.** *Bei der Ausführung von Operationen in logischen Ausdrücken gelten folgende Prioritäten: Geklammerte Verknüpfungen haben Vorrang vor Negation* ($\overline{\ }$) *hat Vorrang vor Konjunktion* ($\wedge$) *hat Vorrang vor allen anderen Operationen. Reicht das Negationszeichen* ($\overline{\ }$) *über mehrere Variable oder Konstante, gelten diese als geklammert und die Negation gilt für den Klammerausdruck. Bei mehreren aufeinanderfolgenden zweistelligen Grundverknüpfungen gleicher Vorrangstufe, also bei Hintereinanderausführung gleicher Verknüpfungen, wird von links nach rechts abgearbeitet. Das Konjunktionszeichen kann weggelassen werden, wenn keine Verwechslungen mit anderen Variablen möglich sind.*

**Beispiel 6.** *Seien* $\mathsf{a}, \mathsf{b}, \mathsf{c}, \mathsf{d}$ *logische Variablen.*
$\mathsf{a} \vee (\mathsf{b} \wedge \mathsf{c}) = \mathsf{a} \vee \mathsf{b} \wedge \mathsf{c} = \mathsf{a} \vee \mathsf{bc}$,
$(\mathsf{a} \wedge \overline{\mathsf{b}}) \vee (\overline{\mathsf{c} \wedge \mathsf{d}}) = \mathsf{a}\overline{\mathsf{b}} \vee \overline{\mathsf{c}}\,\overline{\mathsf{d}}$,
$\mathsf{a} \to \mathsf{b} \to \mathsf{c} = (\mathsf{a} \to \mathsf{b}) \to \mathsf{c}$,
$(\mathsf{a} \vee \mathsf{b}) \wedge (\mathsf{a} \vee \mathsf{c}) = (\mathsf{a} \vee \mathsf{b})(\mathsf{a} \vee \mathsf{c})$,
*aber:*
$\overline{\mathsf{a}} \wedge \overline{\mathsf{b}} = \overline{\mathsf{a}}\ \overline{\mathsf{b}} \neq \overline{\mathsf{a}}\overline{\mathsf{b}}$

### 1.1.4 Umformung logischer Ausdrücke

#### Regeln zur Vereinfachung und Umformung gegebener Ausdrücke

Es wird sich auf boolesche Ausdrücke beschränkt, weil es für diese Ausdrücke eine entsprechende algebraische Struktur gibt, nämlich die Boolesche Algebra. Andererseits lassen sich alle anderen Operationen als boolesche Ausdrücke schreiben. Beweisen lassen sich diese Regeln in einfachster Weise durch Vergleich der Wertetabellen der Ausdrücke auf beiden Seiten der Gleichung.

**Satz 1.** *Seien* $\mathsf{a}, \mathsf{b}, \mathsf{c}$ *logische Variablen, dann gelten folgende Regeln:*

| *Name* | *Gesetzmäßigkeit* |
|---|---|
| *Negation der Negation* | $\overline{\overline{\mathsf{a}}} = \mathsf{a}$ |
| *Kommutativgesetz* | $\mathsf{a} \wedge \mathsf{b} \wedge \mathsf{c} = \mathsf{c} \wedge \mathsf{a} \wedge \mathsf{b}$<br>$\mathsf{a} \vee \mathsf{b} \vee \mathsf{c} = \mathsf{c} \vee \mathsf{a} \vee \mathsf{b}$ |
| *Assoziativgesetz* | $\mathsf{a} \wedge (\mathsf{b} \wedge \mathsf{c}) = (\mathsf{a} \wedge \mathsf{b}) \wedge \mathsf{c}$<br>$\mathsf{a} \vee (\mathsf{b} \vee \mathsf{c}) = (\mathsf{a} \vee \mathsf{b}) \vee \mathsf{c}$ |
| *Distributivgesetz* | $\mathsf{a} \wedge (\mathsf{b} \vee \mathsf{c}) = (\mathsf{a} \wedge \mathsf{b}) \vee (\mathsf{a} \wedge \mathsf{c})$<br>$\mathsf{a} \vee (\mathsf{b} \wedge \mathsf{c}) = (\mathsf{a} \vee \mathsf{b}) \wedge (\mathsf{a} \vee \mathsf{c})$ |
| *Idempotenzgesetz* | $\mathsf{a} \wedge \mathsf{a} = \mathsf{a}$<br>$\mathsf{a} \vee \mathsf{a} = \mathsf{a}$ |
| *Komplementgesetz* | $\mathsf{a} \wedge \overline{\mathsf{a}} = 0$<br>$\mathsf{a} \vee \overline{\mathsf{a}} = 1$ |
| *0-1-Gesetz* | $\mathsf{a} \wedge 1 = \mathsf{a}$<br>$\mathsf{a} \wedge 0 = 0$<br>$\mathsf{a} \vee 1 = 1$<br>$\mathsf{a} \vee 0 = \mathsf{a}$ |
| *Absorptionsgesetze* | $\mathsf{a} \wedge (\mathsf{a} \vee \mathsf{b}) = \mathsf{a}$<br>$\mathsf{a} \vee (\mathsf{a} \wedge \mathsf{b}) = \mathsf{a}$<br>$(\mathsf{a} \wedge \mathsf{b}) \vee (\mathsf{a} \wedge \overline{\mathsf{b}}) = \mathsf{a}$<br>$(\mathsf{a} \vee \overline{\mathsf{b}}) \wedge \mathsf{b} = \mathsf{a} \wedge \mathsf{b}$<br>$(\mathsf{a} \wedge \overline{\mathsf{b}}) \vee \mathsf{b} = \mathsf{a} \vee \mathsf{b}$<br>$(\mathsf{a} \vee \mathsf{b}) \wedge (\mathsf{a} \vee \overline{\mathsf{b}}) = \mathsf{a}$ |
| *De Morgan'sche Regeln* | $\overline{\mathsf{a} \wedge \mathsf{b}} = \overline{\mathsf{a}} \vee \overline{\mathsf{b}}$<br>$\overline{\mathsf{a} \vee \mathsf{b}} = \overline{\mathsf{a}} \wedge \overline{\mathsf{b}}$ |

**Beweis 1.** *Der Beweis erfolgt durch Wertetabelle. Exemplarisch soll er hierfür das vorletzte Absorptionsgesetz durchgeführt werden.*
$(a \wedge \overline{b}) \vee b = a \vee b$

| a | b | $a \wedge \overline{b}$ | $(a \wedge \overline{b}) \vee b$ | $a \vee b$ |
|---|---|---|---|---|
| *0* | *0* | *0* | *0* | *0* |
| *0* | *1* | *0* | *1* | *1* |
| *1* | *0* | *1* | *1* | *1* |
| *1* | *1* | *0* | *1* | *1* |

*Die Spalten für* $(a \wedge \overline{b}) \vee b$ *und* $a \vee b$ *stimmen überein (was zu zeigen war).*

#### Vereinfachung komplexer boolescher Ausdrücke

Die angeführten Regeln erlauben, komplizierte boolesche Ausdrücke häufig zu vereinfachen.

**Beispiel 7.**
$$\begin{aligned}
&(\overline{a} \vee b)\overline{a\ b}((\overline{a} \vee b)(\overline{a} \vee \overline{b}) \vee (ca \vee \overline{c})) \textit{ De Morgan'sche Regel}\\
&= (\overline{a} \vee b)(\overline{a} \vee \overline{b})((\overline{a} \vee b)(\overline{a} \vee \overline{b}) \vee (ca \vee \overline{c})) \textit{ Absorptionsgesetz}\\
&= \overline{a}(\overline{a} \vee (ca \vee \overline{c})) \textit{ Absorptionsgesetz}\\
&= \overline{a}
\end{aligned}$$

**Beispiel 8.**
$$\begin{aligned}
&x \vee yz \vee (\overline{x} \wedge \overline{yz})\\
&= yz \vee x \vee (\overline{x} \wedge \overline{y}z)\\
&= yz \vee (x \vee \overline{x}) \wedge (x \vee \overline{yz})\\
&= yz \vee 1 \wedge (x \vee \overline{yz})\\
&= yz \vee (x \vee \overline{yz})\\
&= yz \vee \overline{yz} \vee x\\
&= 1 \vee x\\
&= 1
\end{aligned}$$

Logische Gleichungen lassen sich in analoger Weise beweisen, denn bei mehr als 3 Variablen wird der Beweis mittels Wertetabelle recht aufwändig.

### 1.1.5 Formal wahre, formal falsche Ausdrücke

Ein häufig beschrittener Weg zur Vereinfachung von Ausdrücken ist die Suche nach "Teilausdrücken", die immer logisch 0 oder immer 1 sind, um anschließend die 0-1-Gesetze anzuwenden.

**Definition 6.** *Ein Ausdruck heißt formal wahrer Ausdruck, wenn er für jede Belegung den Wahrheitswert 1 liefert. Ein Ausdruck heißt formal falsch, wenn er für jede Belegung den Wahrheitswert 0 liefert.*

**Beispiel 9.**

| *Formal wahre Ausdrücke:* | *Formal falsche Ausdrücke:* |
|---|---|
| $a \vee \overline{a} = 1$ | $a \wedge \overline{a} = 0$ |
| $ab \vee a\overline{b} \vee \overline{a}b \vee \overline{a}\overline{b} = 1$ | $\overline{(ab \vee a\overline{b} \vee \overline{a}b \vee \overline{a}\overline{b})} = 0$ |
| $(a \rightarrow b) \vee (b \rightarrow a) = 1$ | $a \oplus a = 0$ |
| $(a \leftrightarrow b) \vee (\overline{a} \leftrightarrow b) = 1$ | $a \wedge b \wedge c \wedge 0 = 0$ |
| $(a \wedge b) \vee 1 = 1$ | |

### Tautologie, Kontradiktion

Bei formal wahren/falschen Ausdrücken handelt es sich in der Aussagenlogik um Aussagen, die immer zutreffen müssen (formal wahr) oder nie zutreffen können (formal falsch). Die Behauptung "n=m oder n $\neq$ m" ($a \vee \overline{a}$) trifft immer zu, unabhängig von den Werten der Einzelaussagen "(n = m)" und "(n $\neq$ m)". Immer falsch dagegen ist die Aussage "n=m und n $\neq$ m" ($a \wedge \overline{a}$). Also schon aus der Form des Ausdrucks lässt sich der Wahrheitswert bestimmen. Solche Ausdrücke werden auch Tautologien oder Allgemeingültigkeiten genannt, wenn sie formal wahr sind, bzw. Kontradiktionen im Falle formal falscher Ausdrücke.

### Einfache Sätze zur Umformungen von Ausdrücken

Der folgende Satz ist eine Verallgemeinerung der De Morgan'schen Regel.

**Satz 2.** *Ein boolescher Ausdruck lässt sich negieren, indem man alle Operatoren $\vee$ durch $\wedge$ und alle $\wedge$ durch $\vee$ ersetzt und die Konstanten und Variablen negiert. Die Klammerungen bleiben dabei erhalten.*

Mit Hilfe de DeMorgan'schen Regeln lässt sich die Gültigkeit gut veranschaulichen:

**Beispiel 10.**
$$\overline{(a_1 \wedge a_2 \wedge a_3 \wedge ...a_n)} = \overline{a_1} \vee \overline{a_2} \vee \overline{a_3} \vee ... \vee \overline{a_n}$$
$$\overline{(a_1 \vee a_2 \vee a_3 \vee ... \vee a_n)} = \overline{a_1} \wedge \overline{a_2} \wedge \overline{a_3} \wedge ... \wedge \overline{a_n}$$
$$\overline{((a_1a_2 \vee \overline{a_2}a_3)(a_4 \vee \overline{a_5}))} = (\overline{a_1} \vee \overline{a_2})(a_2 \vee \overline{a_3}) \vee \overline{a_4}a_5$$
$$\overline{((1 \vee a_1a_2) \wedge ((\overline{a_1} \wedge 1) \vee 0)} = (0 \wedge (\overline{a_1} \vee \overline{a_2})) \vee ((a_1 \vee 0) \wedge 1)$$
$$= 0(\overline{a_1} \vee \overline{a_2}) \vee \overline{(a_1 \vee 0)1}$$
$$(\overline{a_1}\ \overline{a_2} \vee \overline{a_1}a_2 \vee a_1\overline{a_2} \vee a_1a_2) = (a_1 \vee a_2)(a_1 \vee \overline{a_2})(\overline{a_1} \vee a_2)(\overline{a_1} \vee \overline{a_2})$$

**Beweis 2.**
$$\overline{(a_1a_2a_3...a_n)} = \overline{((a_1a_2a_3...a_{n-1})a_n)} \text{ } \textit{(wegen Vorrangr.)}$$
$$= \overline{(a_1a_2...a_{n-1})} \vee \overline{a_n} \text{ } \textit{(De Morgansche Regel)}$$
$$= \overline{((a_1...a_{n-2})a_{n-1})} \vee \overline{a_n}$$
$$= \overline{a_1} \vee \overline{a_2} \vee \overline{a_3} \vee ... \vee \overline{a_n}$$

**Satz 3.** *Eine Konjunktion $a_1 \wedge a_2 \wedge a_3 \wedge .... \wedge a_n$ ist für eine Belegung $(a_1, a_2, ..., a_n)$ falsch genau dann, wenn es (mindestens) ein i gibt mit $a_i = 0, i \in 1, 2, ..., n$. Eine Disjunktion $a_1 \vee a_2 \vee ... \vee a_n$ ist für eine Belegung $(a_1, a_2, ..., a_n)$ richtig genau dann, wenn es (mindestens) ein i gibt mit $a_i = 1, i \in 1, 2, ..., n$.*

Der Satz kann natürlich auch umgekehrt formuliert werden:

**Satz 4.** *Eine Konjunktion* $a_1 \wedge a_2 \wedge .. \wedge a_n$ *ist genau dann wahr, wenn alle Variablen* $a_i$ *mit 1 belegt sind. Eine Disjunktion* $a_1 \vee a_2 \vee .. \vee a_n$ *ist genau dann falsch, wenn alle Variablen* $a_i$ *mit 0 belegt sind.*

Diese Aussagen werden später bei der Behandlung der Normalformen benötigt.

# 1.2 Logische Funktionen und Funktionsbündel

## 1.2.1 Logische Funktionen

### Begriff der logischen Funktionen

Ein logischer Ausdruck beschreibt eine Zuordnung zwischen einer Belegung der Variablen und den Wahrheitswerten 0 und 1. Das Ergebnis ist eine Funktion der (unabhängigen) Variablen.

**Definition 7.** *n-stellige logische Funktion*
*Sei* $D \subset \{0,1\}^n, n \in \mathbb{N}$ *und* $(a_1, a_2, ..., a_n) \in D$ *eine Belegung.* $\{0,1\}^n$ *ist Menge aller* $n-$ Tupel $(a_1, a_2, ..., a_n)$ *mit* $a_i \in \{0,1)\}$ *für alle* $i = 1, 2, ...., n$ *Eine Zuordnungsvorschrift* $f : D \to \{0,1\}$ *mit* $(a_1, \ldots, a_n) \mapsto f(a_1, a_2, \ldots, a_n)$ *heißt n-stellige logische Funktion oder logische Funktion genau dann, wenn f eindeutig ist, d.h. jeder Belegung* $(a_1, a_2, ..., a_n) \in D$ *wird genau ein Wert* $f(a_1, a_2, ..., a_n) \in \{0,1\}$ *zugeordnet.* $f(a_1, a_2, ..., a_n)$ *heißt Funktionswert, D heißt Definitionsbereich, die Elemente aus D heißen Argumente der Funktion. Ist* $D = \{0,1\}^n$*, so heißt f vollständig definiert. Ist* $D \subset \{0,1\}^n$*, so heißt f partiell definiert.*

### Partiell definierte logische Funktionen

Eine logische Funktion muss also nicht für jede Belegung ihrer Variablen definiert sein ($D \subset \{0,1\}^n$, partiell definierte logische Funktionen), was später im Zusammenhang mit Don't-care-Bedingungen in der Schaltalgebra interessant wird. Eine logische Funktion kann praktisch durch eine Wertetabelle (auch Funktionstabelle genannt) oder durch einen logischen Ausdruck angegeben werden.

**Beispiel 11.** *Vollständig definierte Funktionen*

| $x_1$ | $x_2$ | $g(x_1, x_2)$ |
|---|---|---|
| *0* | *0* | *0* |
| *0* | *1* | *1* |
| *1* | *0* | *0* |
| *1* | *1* | *0* |

$f(a, b, c) = \overline{a \vee b} \to c$

Durch logische Ausdrücke können nur vollständig definierte Funktionen angegeben werden.

### Funktionsgleichung

Eine Funktion kann auch durch eine abhängige Variable definiert werden, die selbst in einer Funktion als Variable auftritt. Auf diese Art lassen sich logische Funktionen verschachteln oder verknüpfen.

**Beispiel 12.** *Funktionsgleichung*
$\mathsf{y} = \mathsf{f}(\mathsf{a}, \mathsf{b})$
$\mathsf{z} = \mathsf{g}(\mathsf{a}, \mathsf{c}, \mathsf{y})$
$= \mathsf{g}(\mathsf{a}, \mathsf{c}, \mathsf{f}(\mathsf{a}, \mathsf{b}))$

Man spricht hierbei auch von Funktionsgleichungen. Diese werden im Zusammenhang mit der Analyse und Synthese von größeren Schaltnetzen zur besseren Übersicht benutzt.

### Dezimales Äquivalent

Unter der Voraussetzung, nur vollständig definierte Funktionen zu betrachten, gilt, dass es $2^{2^n}$ verschiedene n-stellige Funktionen gibt. Wegen der hohen Anzahl hat man Kurzschreibweisen für diese Funktionen eingeführt. Es wird der Begriff des dezimalen Äquivalents einer Belegung (oder eines logischen Vektors) eingeführt.

**Definition 8.** *Dezimales Äquivalent*
*Sei* $(\mathsf{a}_1, \ldots, \mathsf{a}_n)$ *eine Folge logischer Konstanten.* $\delta(\mathsf{a}_1\mathsf{a}_2 \ldots \mathsf{a}_n) = \mathsf{i} \in \mathbb{N}_0$ *heißt das dezimale Äquivalent zu* $(\mathsf{a}_1, \mathsf{a}_2, \ldots, \mathsf{a}_n)$*, wobei*
$\mathsf{i} = \sum_{\mathsf{j}=0}^{\mathsf{n}-1} \mathsf{b}_\mathsf{j} 2^\mathsf{j}$ und $\mathsf{b}_\mathsf{j} = \begin{matrix} 1 \; falls \; a_{n-j} = 1 \\ 0 \; falls \; a_{n-j} = 0 \end{matrix}$ mit $\mathsf{b}_\mathsf{j} \in \mathbb{N}_0$
$\mathbb{N}_0$ : *Menge der natürlichen Zahlen, einschließlich der Null*

Die Definition sagt, dass die Folge logischer Werte so wie sie dasteht als Dualzahl $\mathsf{b}_{n-1}\mathsf{b}_{n-2} \ldots \mathsf{b}_0$ aufgefasst wird und als natürliche Zahl im Dezimalsystem wiedergegeben wird. Umweg über $\mathsf{b}_\mathsf{j}$ ist nötig, um Konfusionen zwischen den Wahrheitswerten 0, 1 und den natürlichen Zahlen 0, 1 zu vermeiden. Die lexikographische Anordnung der Belegungen logischer Variablen bedeutet nur, die Belegungen als Dualzahlen zu lesen und sie dann in aufsteigender Reihenfolge zu schreiben (siehe Kapitel 2).

**Beispiel 13.** *Dezimale Äquivalente*
$\delta(000) = 0$
$\delta(0010) = 2$
$\delta(10010) = 18$

## 1.2.2 Vollständige Systeme

### Darstellung der Grundverknüpfungen

Eine logische Funktion kann durch verschiedene zueinander äquivalente Ausdrücke angegeben werden.

**Satz 5.** *Äquivalenzen zu Grundverknüpfungen*
*Seien* $x_1, x_2$ *logische Variable, dann gilt:*
$f(x_1, x_2) = x_1 \wedge x_2 = \overline{(\overline{x_1} \vee \overline{x_2})}$
$f(x_1, x_2) = x_1 \oplus x_2 = \overline{x_1}\, x_2 \vee x_1\, \overline{x_2} = \overline{(x_1 \vee \overline{x_2})} \vee \overline{(\overline{x_1} \vee x_2)}$
$f(x_1, x_2) = x_1 \vee x_2 = \overline{(\overline{x_1} \wedge \overline{x_2})}$
$f(x_1, x_2) = \overline{x_1 \vee x_2} = \overline{x_1} \wedge \overline{x_2}$
$f(x_1, x_2) = x_1 \leftrightarrow x_2 = \overline{x_1}\,\overline{x_2} \vee x_1\, x_2 = \overline{(x_1 \vee x_2)} \vee \overline{(\overline{x_1} \vee \overline{x_2})}$
$f(x_1, x_2) = x_1 \rightarrow x_2 = \overline{(x_1 \wedge \overline{x_2})} = \overline{x_1} \vee x_2$
$f(x_1, x_2) = \overline{x_1 \wedge x_2} = \overline{x_1} \vee \overline{x_2}$

Beweisen lassen sich diese Behauptungen durch Wertetabellen und die logischen Gesetze. Die Gleichungen zeigen, dass sich alle Grundverknüpfungen durch $\wedge, \vee, ^-$ ausdrücken lassen, dass sogar $\vee$ und $^-$ genügen, um alle Grundverknüpfungen darzustellen.

**Satz 6.** *Boolescher Ausdruck*
*Seien* $a_1, a_2, \ldots, a_n$ *logische Variable. Jede vollständig definierte logische Funktion* $g(a_1, a_2, \ldots, a_n)$ *lässt sich als logischer Ausdruck, der nur aus den Grundverknüpfungen* $\wedge, \vee, ^-$ *zusammengesetzt ist, d.h. als boolescher Ausdruck angeben.*

**Definition 9.** *Eine Menge logischer Operationen* $(O_1, O_2, \ldots, O_k)$ *heißt vollständiges System, wenn sich jede vollständig definierte logische Funktion nur aus diesen Operationen bilden lässt.*

Durch $(\wedge, \vee, ^-)$ sind alle booleschen Ausdrücke erfasst. Mit booleschen Ausdrücken lassen sich also alle logischen Zusammenhänge beschreiben. Es gibt natürlich noch weitere vollständige Systeme. Um Vollständigkeit zu beweisen genügt es zu zeigen, dass mit den betrachteten Operationen die Grundverknüpfungen $\wedge, \vee, ^-$ ausgedrückt werden können.

**Satz 7.** *Vollständige Systeme*
$(\wedge, ^-)$ *und* $(\vee, ^-)$ *sind vollständige Systeme.*

**Beweis 3.** *Seien* $a, b$ *logische Variablen.*
$a \vee b$ *bzw.* $a \wedge b$ *lassen sich mit* $\wedge$ *und* $^-$ *bzw.* $\vee$ *und* $^-$ *angeben.*
$(\wedge, ^-)$*:* $a \vee b = \overline{\overline{a} \wedge \overline{b}}$ *(De Morgansche Regel)*
$(\vee, ^-)$*:* $a \wedge b = \overline{\overline{a} \vee \overline{b}}$ *(De Morgansche Regel)*

**Satz 8.** *NOR und NAND bilden jeweils ein vollständiges System.*

Mit einer einzigen Verknüpfung lassen sich alle vollständig definierten Funktionen ausdrücken, was die technische Realisierung vereinfacht. Es muss prinzipiell nur ein Verknüpfungsglied gebaut werden. Allerdings erhöht sich die Anzahl der Verknüpfungsglieder dadurch in der Regel. Dennoch genießt das System $(\wedge, \vee, ^-)$ eine gewisse Vorzugsstellung, weil es über diese Verknüpfungen eine algebraische Struktur gibt.

### 1.2.3 Funktionsbündel

In der Praxis sind häufig Funktionsbündel zu betrachten. Es handelt sich dabei um eine (endliche) Menge logischer Funktionen, die das gleiche Argument haben.

**Beispiel 14.**
*Funktionsbündel werden z.B. bei der Umcodierung eines Binärcodes in einen anderen, wie es z.B. bei der Umformung des BCD-Codes in den 1-aus-10-Code der Fall ist, benötigt. Zur Beschreibung der Ausgänge werden zehn verschiedene Funktionen benötigt, die vier gemeinsame Variablen an den Eingängen haben. z. B.*
$y_1 = x_1x_2x_3 \vee x_1x_4$
$y_2 = x_1\overline{x_4} \vee x_1x_2x_3$

**Definition 10.** *Eine Zusammenfassung $\sigma$ logischer Funktionen $f_1, f_2, \ldots, f_m$, die das gleiche Argument ($a_1, \ldots, a_n$) haben ($m, n \in \mathbb{N}$), heißt ein Funktionsbündel oder eine Vektorfunktion.*
*Schreibweise:*
$\sigma(a_1, a_2, \ldots, a_n) = (y_1, y_2, \ldots, y_m)$

$$\text{mit} \quad \begin{array}{l} y_1 = f_1(a_1, \ldots, a_n) \\ y_2 = f_2(a_1, \ldots, a_n) \\ \ldots \\ y_m = f_m(a_1, \ldots, a_n) \end{array}$$

#### Darstellung von Funktionsbündeln

Die Darstellung von Funktionsbündeln kann durch eine Anzahl m verschiedener logischer Ausdrücke gegeben sein oder durch eine Wertetabelle.

**Beispiel 15.** *Funktionsbündel* $\sigma(a, b, c) = (x, y, z, t)$

| a | b | c | x | y | z | t |
|---|---|---|---|---|---|---|
| *0* | *0* | *0* | *1* | *1* | *1* | *0* |
| *0* | *0* | *1* | *1* | *1* | *1* | *1* |
| *0* | *1* | *0* | *1* | *0* | *1* | *0* |
| *0* | *1* | *1* | *0* | *0* | *1* | *0* |
| *1* | *0* | *0* | *0* | *0* | *0* | *1* |
| *1* | *0* | *1* | *1* | *0* | *0* | *0* |
| *1* | *1* | *0* | *1* | *0* | *0* | *0* |
| *1* | *1* | *1* | *1* | *1* | *0* | *1* |

#### Anwendung

Durch das Zusammenfassen mehrerer Funktionen zu einem Funktionsbündel kann beispielsweise der Hardwareaufwand minimiert werden. Gibt es in mehreren der beteiligten Funktionen (logischen Ausdrücken) gemeinsame Teile, kann dieser Teil von mehreren Funktionen gleichzeitig benutzt werden. Dies ermöglicht eine Einsparung von Hardware

und Verlustleistung bei der technischen Realisierung. Bisher existiert kein allgemeiner Algorithmus zur effizienten Berechnung optimaler Lösungen von mehrstufigen Schaltnetzen.

## 1.3 Das Karnaugh-Veitch-Diagramm

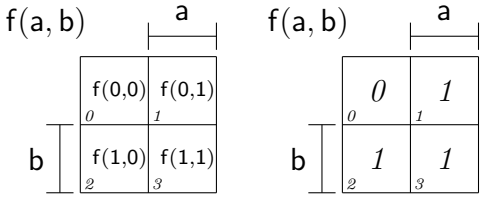
### Graphische Darstellung

Zur Veranschaulichung logischer Funktionen eignen sich graphische Darstellungen bzw. Diagramme. Eine gute Verbildlichung ist das Karnaugh-Veitch-Diagramm (KV-Diagramm). Diese grafische Form ist vorteilhaft bei der Behandlung der Normalformen, die daraus zum Teil direkt abgelesen werden können. Das KV-Diagramm ist eine Matrix von Wahrheitswerten. Jedem Element der Matrix ist eindeutig eine Belegung der Variablen der Funktionen zugeordnet, die am Rand des Schemas eingetragen werden.

**Beispiel 16.** *KV-Diagramm mit zwei Variablen*

| $\delta(ba)$ | b | a | f(a, b) |
|---|---|---|---|
| *0* | *0* | *0* | *0* |
| *1* | *0* | *1* | *1* |
| *2* | *1* | *0* | *1* |
| *3* | *1* | *1* | *1* |

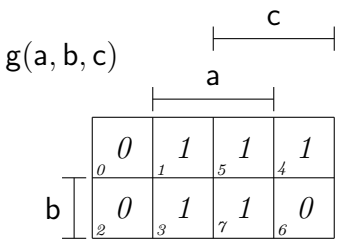
$f(a, b) = a \vee b$




**Beispiel 17.** *KV-Diagramm mit drei Variablen*

| $\delta(cba)$ | c | b | a | g(a, b, c) |
|---|---|---|---|---|
| *0* | *0* | *0* | *0* | *0* |
| *1* | *0* | *0* | *1* | *1* |
| *2* | *0* | *1* | *0* | *0* |
| *3* | *0* | *1* | *1* | *1* |
| *4* | *1* | *0* | *0* | *1* |
| *5* | *1* | *0* | *1* | *1* |
| *6* | *1* | *1* | *0* | *0* |
| *7* | *1* | *1* | *1* | *1* |

$g(a, b, c) = ab \vee a\overline{b} \vee \overline{a}\overline{b}c$




### Anordnung bei mehr als einer Variable pro Zeile /Spalte

Bei mehr als einer Variable pro Spalte oder Zeile kann man die Bezeichnung auch als Vektor der Eingangvariablen betrachten (im Beispiel a und c bei den Spalten). Die Belegungen der Vektoren werden nicht lexikographisch (d.h. entsprechend dem Dualcode) angeordnet, sondern entsprechend dem Graycode ($[a, c]_{Spalte} \Rightarrow [0, 0]_0, [0, 1]_1, [1, 1]_2, [1, 0]_3$) ähnlich wie der Dualcode ist auch der Graycode eine Codierung mit 0 und 1. Der Graycode hat aber die Eigenschaft, sich in der Darstellung von benachbarten Zahlen nur in

einer Stelle zu unterscheiden (Hammingdistanz gleich 1). In den Zeilen bzw. Spalten stehen nur Belegungen nebeneinander, die sich nur in einer Stelle unterscheiden. Das gilt sogar zwischen dem ersten und letzten Element einer Zeile bzw. Spalte. Man kann Anfang und Ende einer Zeile oder Spalte als benachbart ansehen. Gerade diese systematische Anordnung ermöglicht die Benutzung dieser Diagramme zur Vereinfachung logischer Ausdrücke (Minimierung). Mit Zunahme der Variablenzahl werden die Matrizen abwechselnd nach unten und rechts "umgeklappt", wodurch sich die Zahl ihrer Elemente jeweils verdoppelt.

### 1-Muster oder 0-Muster

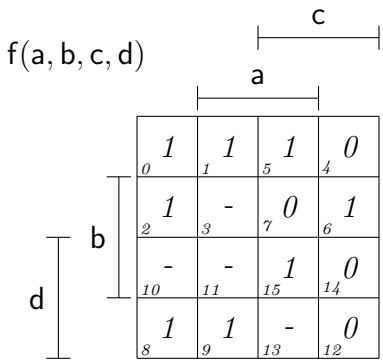
Um Wahrheitswerte eindeutig in ein KV-Diagramm einzutragen, genügt es, entweder nur die 1-Werte oder nur die 0-Werte einzutragen. Es entsteht ein sogn. 1-Muster oder 0-Muster. Vielfach schraffiert man auch nur die Kästchen, die eine 1 tragen.

### Übertragung einer Funktion in ein KV-Diagramm

Eine Funktion kann durch Wertetabelle oder logischen Ausdruck angegeben werden. Liegt die Funktion als Wertetabelle vor, ist die Übertragung in das KV-Diagramm einfach. Man sucht zu jeder Belegung die entsprechende Stelle und trägt den Wahrheitswert der Funktion ein. Bei unvollständig definierten Funktionen, werden in den entsprechenden Stellen "-" ("don't-care") eingetragen

**Beispiel 18.** *Nicht vollständig definierte Funktion*

| $\delta$(cba) | d | c | b | a | f(a, b, c, d) |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 |
| 2 | 0 | 0 | 1 | 0 | 1 |
| 3 | 0 | 0 | 1 | 1 | - |
| 4 | 0 | 1 | 0 | 0 | 0 |
| 5 | 0 | 1 | 0 | 1 | 1 |
| 6 | 0 | 1 | 1 | 0 | 1 |
| 7 | 0 | 1 | 1 | 1 | 0 |
| 8 | 1 | 0 | 0 | 0 | 1 |
| 9 | 1 | 0 | 0 | 1 | 1 |
| 10 | 1 | 0 | 1 | 0 | - |
| 11 | 1 | 0 | 1 | 1 | - |
| 12 | 1 | 1 | 0 | 0 | 0 |
| 13 | 1 | 1 | 0 | 1 | - |
| 14 | 1 | 1 | 1 | 0 | 0 |
| 15 | 1 | 1 | 1 | 1 | 1 |

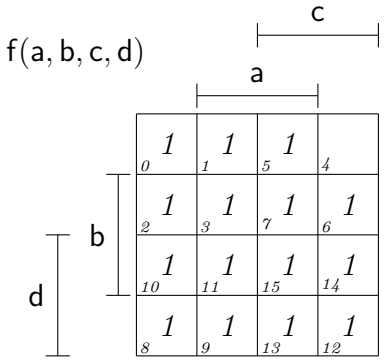


### Vorgehensweise bei der Funktionsbeschreibung

Sind boolesche Ausdrücke gegeben, um eine Funktion zu beschreiben, lässt sich jeder Wert einzeln bestimmen und eintragen. Prinzipiell ist jeder boolesche Ausdruck entweder als eine Disjunktion oder als eine Konjunktion anzusehen, abgesehen von einer Negation des gesamten Ausdrucks. Eine Disjunktion ist immer dann wahr, wenn mindestens eine der Komponenten, aus der die Disjunktion besteht, gleich 1 ist. Somit können für die einzelnen Komponenten einer Disjunktion die 1-Werte nacheinander eingetragen werden.

**Beispiel 19.** *Disjunktive Funktion*

$f(a, b, c, d) = a + b + \bar{c} + c \cdot d$




Eine Konjunktion ist immer dann falsch, wenn mindestens eine der Komponenten der Konjunktion gleich 0 ist. Analog zu Disjunktionen werden nur die 0-Werte zu jeder einzelnen Komponente eingetragen.

**Beispiel 20.** *Konjunktive Funktion*

$g(a, b, c, d) = a \cdot b \cdot \bar{c} \cdot (c + d)$