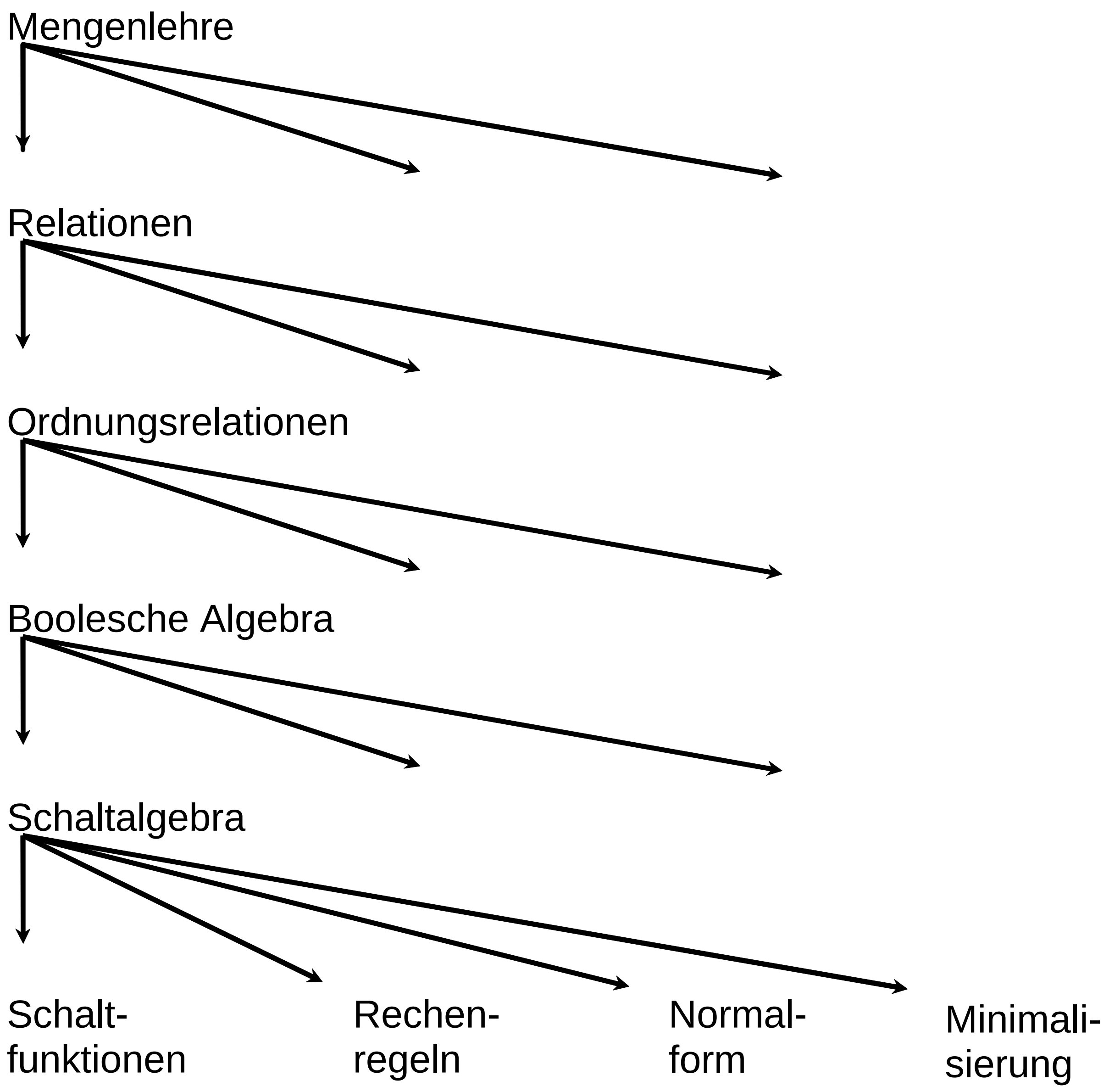
3. SCHALTALGEBRA UND SCHALTFUNKTIONEN

3.1 Einordnung der Schaltalgebra



Schaltalgebra als algebraisches Werkzeug zur Analyse von Schaltkreisen.

Abstraktion vom elektronischen Schaltkreis/Hardwareebene.

Die Schaltalgebra ist ein Modell der Booleschen Algebra über der Menge B = {0, 1}. Sie gestattet es, logische Zusammenhänge rechnerisch zu erfassen.

Die Schaltalgebra ist durch folgende Eigenschaften charakterisiert:

(a) Es existiert: B:= {**0**,**1**}

mit : Nullelement: 0
Einselement: 1

(b) Es existieren folgende Verknüpfungen:

Operation	+	**Disjunktion**,	ODER,	$\vee$
Operation	*	**Konjunktion**,	UND,	$\wedge$
Operation	-	**Negation**,	NICHT,	-

(c) Es gelten die Gesetze der Booleschen Algebra.

Operatoren der Schaltalgebra:

Eingänge		Ausgänge		
a	b	a*b	a+b	$\overline{a}$
0	0	0	0	1
0	1	0	1	1
1	0	0	1	0
1	1	1	1	0

<u>Anmerkungen:</u> Die Negation einer Variablen wird auch geschrieben als: $\neg$a, /a oder a', selten: na

Mehrwertige Logik ist ohne große praktische Anwendungen.

3.2 Eigenschaften der Schaltalgebra

Weil die Schaltalgebra eine Boolesche Algebra ist, gelten für beliebige Elemente a, b, c $\in$ B folgende Axiome:

(1)	$a+b=b+a$ $a*b=b*a$	Kommutativität
(2)	$0+a=a$ $1*a=a$	Nullelement 0 bzw. Einselement 1 bzgl. + bzw. * existiert (neutrale Elemente)
(3)	$(a+b)*c=(a*c)+(b*c)$ $(a*b)+c=(a+c)*(b+c)$	Distributivität einer Operation bzgl. der anderen.
(4)	$a+\overline{a}=1$	Zu jedem Element $a\in B$ existiert ein komplementäres Element $\overline{a}\in B$.

Es gilt die Abgeschlossenheit, d. h., wenn a, b $\in$ B, dann sind auch a+b und a*b $\in$ B.

Satz:

Für alle Elemente a, b, c einer Booleschen Algebra (B; +, *) gilt:

(a) $a+1=1$ Neutrale Elemente

$a*0=0$

(b) $a+a=a$ Idempotenzgesetz

$a*a=a$

(c1) $a+(a*b)=a$ Absorptionsgesetz

$a*(a+b)=a$

(c2) $\bar{a}+(a+b)=1$

$\bar{a}*(a*b)=0$

(d) $a+(b+c)=(a+b)+c$ Assoziativgesetz

$a*(b*c)=(a*b)*c$

(e1) Für jedes a aus B existiert genau ein $\bar{a}$ aus B (Eindeutigkeit des Komplements).

Insbesondere gilt: $\overline{(\bar{a})}=a$.

(e2) $\bar{1}=0$

$\bar{0}=1$

(f) $\overline{(a+b)}=\bar{a}*\bar{b}$ De Morgansches Gesetz

$\overline{(a*b)}=\bar{a}+\bar{b}$

(Folgt aus der Eindeutigkeit des Komplements.)

Verallgemeinerung des Gesetzes von De Morgan:

$$\overline{a * b * c * d \ldots} = \overline{a} + \overline{b} + \overline{c} + \overline{d} \ldots$$

$$\overline{a + b + c + d \ldots} = \overline{a} * \overline{b} * \overline{c} * \overline{d} \ldots$$

(Kann durch schrittweises Umformen gezeigt werden.)

Satz (Dualitätsprinzip)

Zu jeder Aussage, die sich aus den vier Axiomen der Booleschen Algebra herleiten lässt, existiert eine duale Aussage, die dadurch entsteht, dass man die Operationen + und * und gleichzeitig die neutralen Elemente 0 und 1 vertauscht.

Beweis: folgt unmittelbar aus Symmetrie der Axiome bzgl. +, * und 0, 1.

Beispiel:

f_1: UND-Verknüpfung, Konjunktion, logisches Produkt, Durchschnitt, $a \wedge b$, $a * b$, $a \cap b$, ab.

verbal: zufrieden = warm ***und*** gegessen.

f_7: ODER-Verknüpfung, Disjunktion, logische Summe, Vereinigung, a v b, a + b, $a \cup b$.

verbal: hat Kind(er) = hat Sohn ***oder*** Tochter

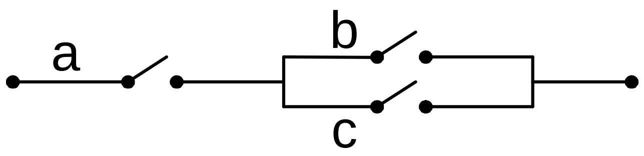
Duale Aussagen:

unzufrieden = kalt ***oder*** hungrig.

kinder***los*** = ***keinen*** Sohn ***und keine*** Tochter.

Beispielweise Interpretation der Schaltalgebra durch Schaltelemente mit binärem Verhalten:

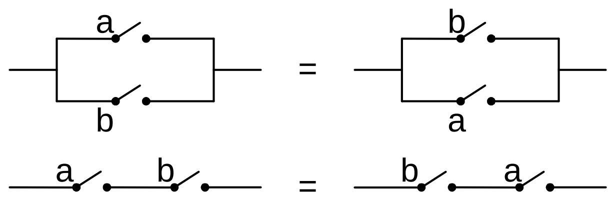
z.B.: 0 : Schalter geöffnet
1 : Schalter geschlossen



<u>Axiome (1) - (4) der Booleschen Algebra sind gültig:</u>

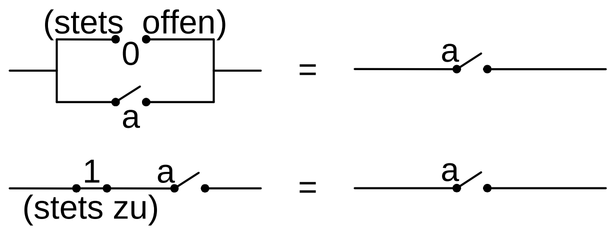
(I) $a \vee b = b \vee a$

$ab = ba$



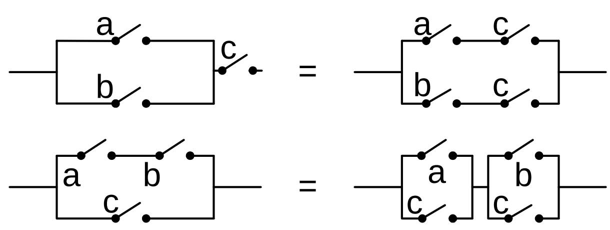
(II) $0 \vee a = a$

$1a = a$



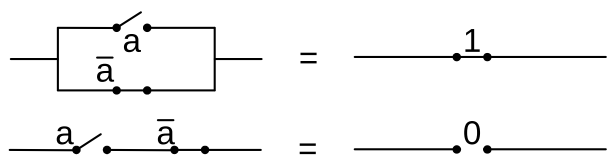
(III) $(a \vee b)c = ac \vee bc$

$(ab) \vee c = (a \vee c)(b \vee c)$



(IV) $a \vee \bar{a} = 1$

$a\bar{a} = 0$



- Schaltungen mit binären Schaltelementen können mit Boolescher Algebra beschrieben werden.
- Sätze der Booleschen Algebra sind gültig $\rightarrow$ <u>Rechenregeln</u>!
- Verschiedene Ausdrücke beschreiben oft dieselbe Funktion.

3.3 Schaltfunktionen

3.3.1 Definition und Wahrheitstafeln

Schaltvariablen sind Variablen, die nur endlich viele Werte annehmen können, hier die Werte 0 oder 1.

Eine binäre ***Schaltvariable*** **v** kann nur zwei Werte annehmen:

v = 0 oder **v = 1**.

Schaltfunktionen sind eindeutige Zuordnungsvorschriften, die jeder Wertekombination von (Schalt-)Variablen einen Wert zuordnen.

Eine binäre ***Schaltfunktion*** nimmt zwei Werte an in Abhängigkeit von den Schaltvariablen v_i:

$$f : \{0,1\}^N \to \{0,1\}$$

$$\mathbf{z = f(v_1, v_2, \ldots, v_N); \quad z, v_i \in \{0,1\}.}$$

Anwendungen:

In der Aussagenlogik:

- Auswertung von komplizierten Ausdrücken,
- Vereinfachung von Ausdrücken,
- Entscheidung zwischen *Wahr* und *Falsch*.

Elektronik:

- Schalter mit Zustand “EIN“ / “AUS“,
- Analyse von Schaltungen (“Schalt~“),
- Optimieren von Schaltungen,
- guter Störabstand in zweiwertigen Systemen.

Binäre Zahlencodierung:

- zweiwertige Speicherzellen,
- interne Zahlendarstellung in Digitalrechnern,
- Zurückführen der Arithmetik auf Schaltfunktionen.

Die Zuordnungsvorschrift kann in Form von Funktionsgleichungen oder Wertetabellen (Funktionstabelle, Wahrheitstafel) angegeben werden.

Funktionstabelle: (Beispiel)

a	b	c	f(a, b, c)
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Folgender Boolescher Ausdruck ist gleichwertig:

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

Festlegung durch Funktionstabelle

v_N	v_{N-1}	• • •	v_2	v_1	f_0	f_1	• • •	f_{max-1}*
0	0	• • •	0	0	0	0	• • •	1
0	0		0	1	0	0		1
0	0		1	0	0	0		1
⋮	⋮		⋮	⋮	⋮	⋮		⋮
1	1		1	0	0	0		1
1	1	• • •	1	1	0	1	• • •	1

* $\max = 2^{2^N}$

- Für jede Schaltfunktion f_i gibt es 2^N Argumentenkombinationen.

- Für N binäre Schaltvariablen v_i existieren maximal 2^{2^N} mögliche, verschiedene N-stellige Schaltfunktionen.

3.3.2 Ein- und zweistellige Schaltfunktionen

Einstellige Schaltfunktionen

	v	0	1
Nullfunktion	f_0	0	0
Identität	f_1	0	1
Komplement	f_2	1	0
Einsfunktion	f_3	1	1

Zweistellige Schaltfunktionen (Prinzip)

		a = 0 b = 0	a = 1 b = 0	a = 0 b = 1	a = 1 b = 1
f_0		0	0	0	0
f_1	UND	0	0	0	1
f_2		0	0	1	0
	⋮				
f_7	ODER	0	1	1	1
	⋮				
f_{14}		1	1	1	0
f_{15}		1	1	1	1

Alle 16 zweistelligen Schaltfunktionen

Funktion und Darstellung der Verknüpfung	Funktionswert für a=0, b=0	a=1, b=0	a=0, b=1	a=1, b=1	Andere Darstellungsart	Bezeichnung	Andere Bezeichnung
$Y_0 = 0$	0	0	0	0	0	Nullfunktion	Null
$Y_1 = a \wedge b$	0	0	0	1	$a \cdot b$, $a \,\&\, b$	Konjunktion	UND-Verknüpfung
$Y_2 = b \nrightarrow a = \bar{a} \wedge b$	0	0	1	0		Inhibition	Ausschluss, Nicht - Wenn - Dann
$Y_3 = b$	0	0	1	1		Identität	Tautologie
$Y_4 = a \nrightarrow b = a \wedge \bar{b}$	0	1	0	0		Inhibition	Ausschluss, Nicht - Wenn - Dann
$Y_5 = a$	0	1	0	1		Identität	Tautologie
$Y_6 = a \not\equiv b = (a \wedge \bar{b}) \vee (\bar{a} \wedge b)$	0	1	1	0	$a \oplus b$	Antivalenz	Disvalenz; Exclusives Oder
$Y_7 = a \vee b$	0	1	1	1	$a + b$	Disjunktion	ODER-Verknüpfung, Inklusiv Oder
$Y_8 = Y_7 = a \overline{\vee} b$ $= \overline{a \vee b}$ $= \bar{a} \wedge \bar{b}$	1	0	0	0	$a \downarrow b$	Peirce-Funktion	NOR-Verknüpfung (Not OR) Weder - Noch
$Y_9 = \overline{Y}_6 = a \equiv b = (\bar{a} \vee b) \wedge (a \vee \bar{b})$ $= (a \wedge b) \vee (\bar{a} \wedge \bar{b})$	1	0	0	1	$a \leftrightarrow b$	Äquivalenz	Valenz Genau dann wenn
$Y_{10} = \overline{Y}_5 = \bar{a}$	1	0	1	0	$\neg a$, a'	Negation	Inversion, Nicht
$Y_{11} = \overline{Y}_4 = a \rightarrow b$ $= \bar{a} \vee b$	1	0	1	1	$a \supset b$	Implikation	Wenn - Dann
$Y_{12} = \overline{Y}_3 = \bar{b}$	1	1	0	0	$\neg b$, b'	Negation	Inversion, Nicht
$Y_{13} = \overline{Y}_2 = b \rightarrow a$ $= \bar{b} \vee a$	1	1	0	1	$b \supset a$	Implikation	Wenn - Dann
$Y_{14} = Y_1 = a \overline{\wedge} b$ $= \overline{a \wedge b} = \bar{a} \vee \bar{b}$	1	1	1	0	$a \mid b$	Sheffer-Funktion	NAND-Verknüpfung (Not AND)
$Y_{15} = \overline{Y}_0 = 1$	1	1	1	1		Einsfunktion	Einheit

Auswertung einer Schaltfunktion

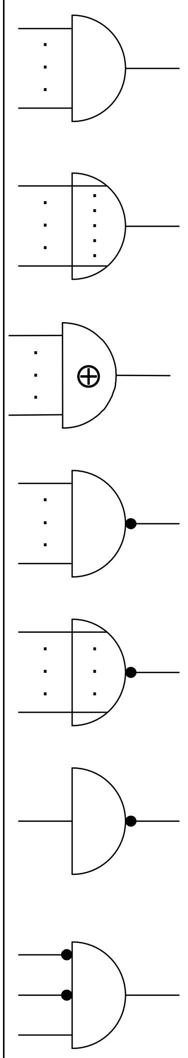
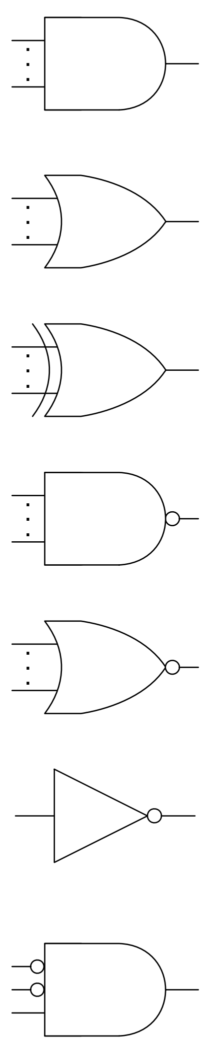
Mit Hilfe einer Wahrheitstafel für $\overline{A}B \vee A\overline{B}$:

A	B	$\overline{A}B$	$A\overline{B}$	$\overline{A}B \vee A\overline{B}$	$A \neq B$
0	0	0	0	0	0
1	0	0	1	1	1
0	1	1	0	1	1
1	1	0	0	0	0

$\overline{A}B \vee A\overline{B}$ entspricht also $A \neq B$.

→ Antivalenz, Exklusiv-ODER (XOR)

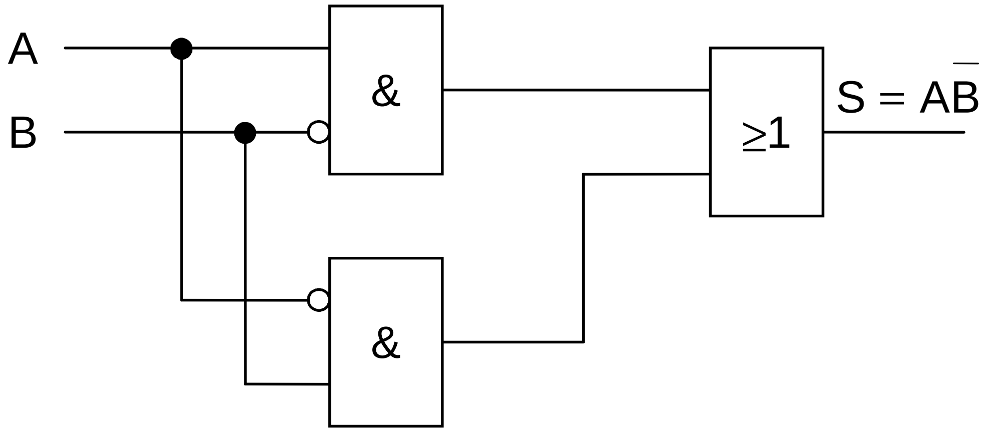
3.3.3 Graphische Darstellung durch Schaltsymbole

freie Symbole	Schaltsymbole nach DIN 40 700 Teil 14, seit 1976	Schaltsymbole nach DIN 40 700 Teil 14, bis 1976	amerikanische Symbole	logische Darstellung
UND	&			$x_1 \wedge \ldots \wedge x_n$
ODER	≥ 1			$x_1 \vee \ldots \vee x_n$
Anti-valenz	= 1	⊕		$x_1 \neq \ldots \neq x_n$
NAND	&			$\overline{x_1 \wedge x_2 \wedge \ldots \wedge x_n}$
NOR	≥ 1			$\overline{x_1 \vee x_2 \vee \ldots \vee x_n}$
Negation	1			$\bar{x}_1$
x_1, x_2, x_3: $\bar{x}_1 \wedge \bar{x}_2 \wedge x_3$	&			$\bar{x}_1 \wedge \bar{x}_2 \wedge x_3$

Hinweis: Invertierungskringel am Ein- und Ausgang möglich

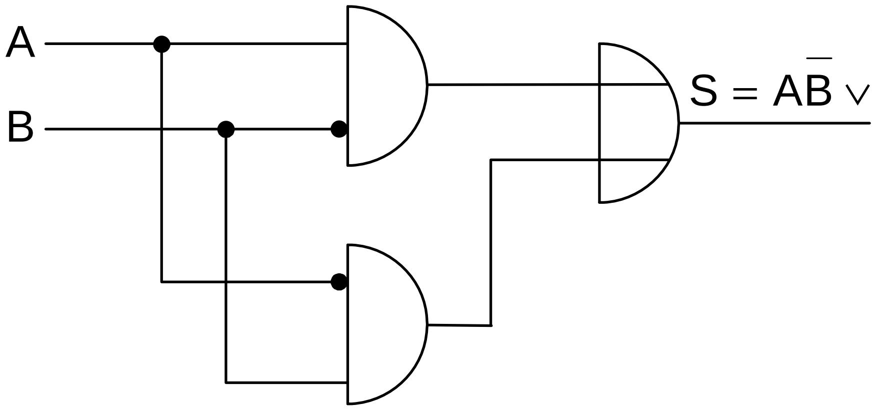
<u>Beispiel:</u> Antivalenz, Exklusiv-ODER

neuere DIN-Norm:



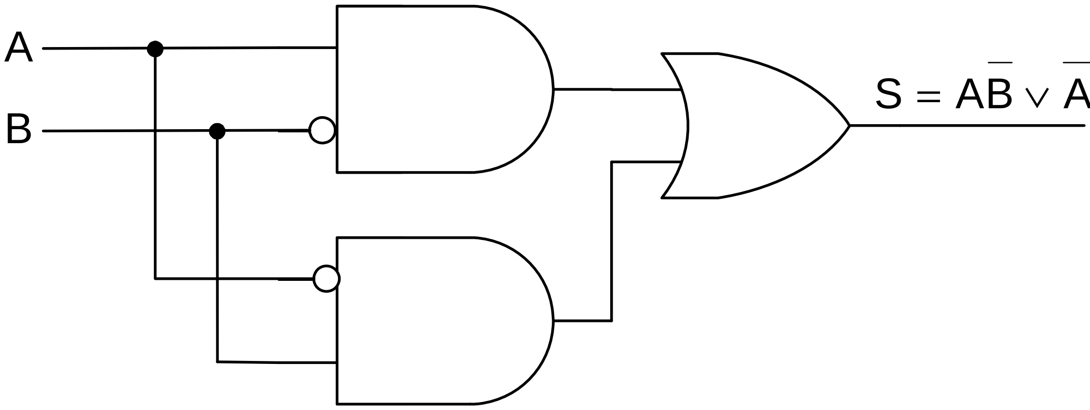
$S = A\overline{B} \vee \overline{A}B = A \oplus B$

alte DIN-Norm:



$S = A\overline{B} \vee \overline{A}B = A \oplus B$

amerikanische Norm:



$S = A\overline{B} \vee \overline{A}B = A \oplus B$

3.4 Umformung von Schaltfunktionen

Satz:

Zu jeder Schaltfunktion existieren beliebig viele Boolesche Ausdrücke, die sie darstellen.

Beweisidee:

Es existiert ein Ausdruck, der die Schaltfunktion beschreibt. Mit Hilfe der Rechenregeln (z.B. De Morgan´sche Gesetze) lassen sich Schaltfunktionen in andere Formen mit gleichem Schaltverhalten umformen, z.B. auch erweitern.

Ein Boolescher Ausdruck kann mit Hilfe der Regeln der Schaltalgebra also auch in eine bestimmte Beschreibungsform gebracht werden.

Die Darstellung ausschließlich durch NAND- bzw. NOR-Funktionen hat dabei historisch eine besondere Bedeutung.

- Mit der **Sheffer**-Funktion "|" (NAND) lässt sich jede beliebige Schaltfunktion darstellen.

    → Sie ist eine vollständige Verknüpfungsbasis.
    - Junktorensystem -

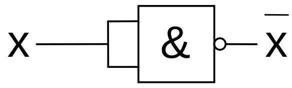
    D.h., jede beliebige Schaltfunktion kann ausschließlich mit NAND-Gattern realisiert werden.
    (Umformnung nach de Morgan)

- Das gleiche gilt für die **Peirce**-Funktion "↓" (NOR).

Sheffer-Funktion:

$$\text{Negation}: \quad f(x) = \overline{x}$$

$$= \overline{(x\ \ )}$$



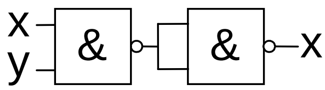
$$= x \mid x$$

$$\text{UND}: \quad f(x,y) = x$$

$$= (x\ \ ) \vee (x\ \ )$$

$$= \overline{\overline{(x\ \ ) \vee (x\ \ )}}$$

$$= \overline{\overline{(x\ \ )}}\ \overline{\overline{(x\ \ )}}$$



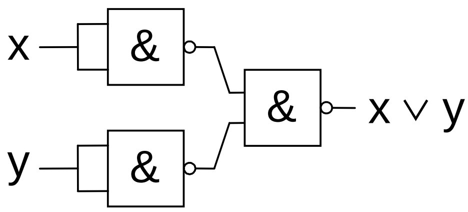
$$= \overline{(x \mid y)\ (x \mid y)}$$

$$= (x \mid y) \mid (x \mid y)$$

$$\text{ODER}: \quad f(x,y) = x \vee y$$

$$= \overline{\overline{x \vee y}}$$

$$= \overline{\overline{x}\ \overline{y}}$$



$$= \overline{x} \mid \overline{y}$$

$$= (x \mid x) \mid (y \mid y)$$

Und umgekehrt.

Praktische Bedeutung:
Sämtliche logische Verknüpfungsschaltungen in einer digitalen Rechenanlage lassen sich mit einem einzigem Bausteintyp (NOR-Gatter oder NAND-Gatter) realisieren.

Das war in der Vergangenheit essentiell für die Realisierung von Rechensystemen, als vorwiegend nur NAND- bzw. NOR-Gatter als Integrierte Schaltkreise (ICs) für die Implementierung zur Verfügung standen.

3.5 Boolesche Normalformen

Weil eine Schaltfunktion durch beliebig viele Boolesche Ausdrücke beschrieben werden kann, drängt sich unter anderem zu Vergleichszwecken oder für standardisierte Algorithmen eine Normalform auf, die eine Schaltfunktion in standardisierter Gleichungsform beschreibt:

Notation: $\mathbf{x_1 + x_2}$ sei $\mathbf{x_1}$ ODER $\mathbf{x_2}$

$\mathbf{x_1\ x_2}$, $\mathbf{x_1 x_2}$ sei $\mathbf{x_1}$ UND $\mathbf{x_2}$

Ein ***Minterm*** (Vollkonjunktion) ist eine UND-Verknüpfung, die <u>alle</u> Schaltvariablen jeweils einmal enthält, wobei diese negiert oder nicht negiert vorkommen können:

$$\overset{\dots}{x}_N \cdot \overset{\dots\dots}{x}_{N-1} \cdot \ldots \cdot \overset{\dots}{x}_2 \cdot \overset{\dots}{x}_1$$

<u>Anmerkung:</u> Jeder Minterm hat <u>nur bei einer</u> Kombination der Schaltvariablen den Wert 1, bei allen anderen 0.

Bei N Schaltvariablen gibt es 2^N verschiedene Minterme.

Disjunktive kanonische Normalform (DKN, DKNF)

<u>Vorgehensweise:</u>

Ausgehend von einer Wertetabelle (pro Zeile der Wertetabelle ein Minterm)

x_3	x_2	x_1	y
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

$\downarrow i$

Schaltfunktion aufstellen, die nur die Minterme mit dem Beitrag $y = 1$ einschließt und all diese Minterme entsprechend der Wertetabelle ODER-verknüpft (Disjunktion), also hier:

$$\overline{x}_3\overline{x}_2\overline{x}_1 + \overline{x}_3 x_2 \overline{x}_1 + \overline{x}_3 x_2 x_1 + x_3 x_2 \overline{x}_1$$

Die folgende Darstellung ist zwar äquivalent, aber ***nicht kanonisch***:

$$\overline{x}_3\overline{x}_1 + \overline{x}_3 x_2 x_1 + x_3 x_2 \overline{x}_1$$

Man spricht dann von der **disjunktiven Normalform** (DNF).

Abkürzende Schreibweise:

Die Nummerierung der Minterme erfolgt entsprechend der binären Codierung des unteren Indexwertes. Der obere Index bezeichnet die Anzahl der Variablen im Term.

z.B.: $$Min_{17}^{5} = x_5 \cdot \bar{x}_4 \cdot \bar{x}_3 \cdot \bar{x}_2 \cdot x_1 = \bigwedge_{k=1}^{5} b_k \quad ; \quad b_k \in \{x_k, \bar{x}_k\}$$

(Term 17 bei 5 Variablen)

Zusammensetzen der **DKN** aus Mintermen mit ODER-Verknüpfung:

$$\bigvee_{i=0}^{i=2^N-1} \mathrm{Min}_i^N \cdot y_i \quad ; \quad y_i \in \{0,1\}$$

$$= y_0 \cdot Min_0^N \vee y_1 \cdot Min_1^N \vee \ldots \vee y_{2^N-1} \cdot Min_{2^N-1}^N$$

Beachte: Auch hier geht Punktrechnung vor Strichrechnung, also Konjunktion vor Disjunktion.

Oft wird auch die Produkt-Summen-Darstellung verwendet:

$$\sum_{i=0}^{2^N-1} Min_i^N \cdot y_i = \sum_{i=0}^{2^N-1} \prod_{j=1}^{N} x_j \cdot y_i$$

$$= \sum_i Min_i^N$$

Konjunktive kanonische Normalform (KKN)

Analog zur disjunktiven kanonischen Normalform lässt sich eine äquivalente konjunktive kanonische Normalform angeben.

<u>Vorgehensweise:</u>

x_3	x_2	x_1	y
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

$\downarrow i$

Ausgehend von einer Wertetabelle Schaltfunktion aufstellen, die nur die **Maxterme** mit dem Beitrag y = 0 einschließt und all dies Maxterme UND-verknüpft (Konjunktion).

Die *Maxterme* (Volldisjunktionen) entstehen (analog zu Mintermen) durch ODER-Verknüpfung der zugehörigen in nichtnegierter Form für 0 und negierter Form für 1.

KKN hier:

$$(x_3 + x_2 + \overline{x}_1) \cdot (\overline{x}_3 + x_2 + x_1) \cdot (\overline{x}_3 + x_2 + \overline{x}_1) \cdot (\overline{x}_3 + \overline{x}_2 + \overline{x}_1)$$

Jeder Maxterm kann für genau eine Zeile eine 0 erzwingen.

Für alle anderen Zeilen außer seiner eigenen ist er 1.

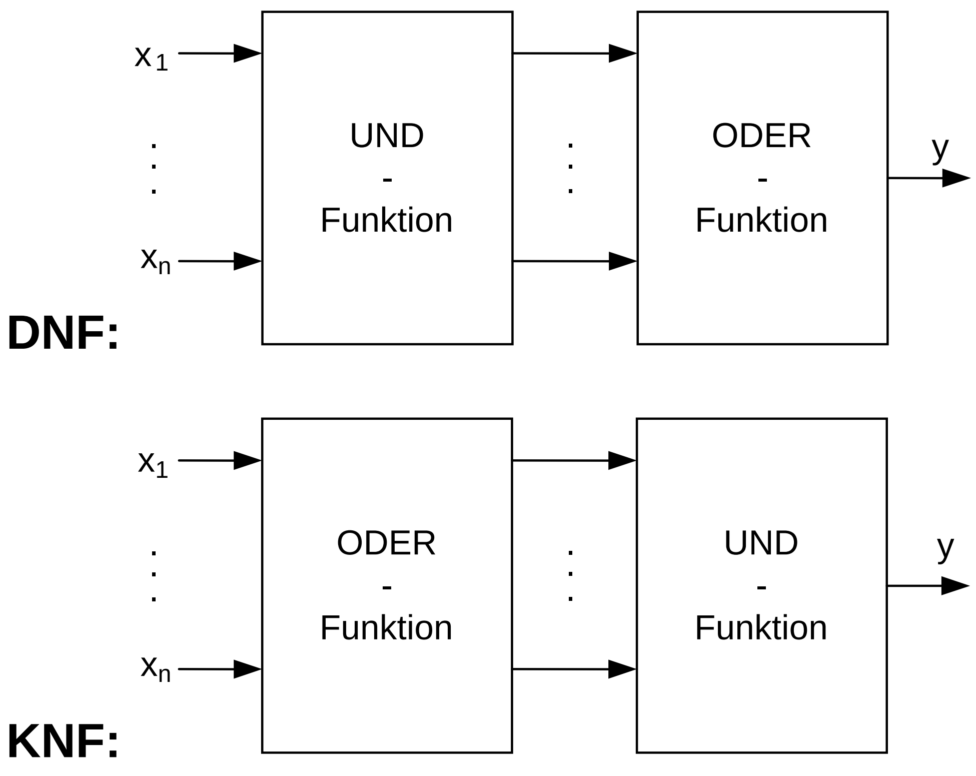
Die UND-Verknüpfung (Konjunktion) der Maxterme liefert dann eine eindeutige Darstellung der Schaltfunktion, die *konjunktive kanonische Normalform (KKN).*

Die *konjunktive Normalform KNF* ist i. Allg. nicht eindeutig.

Zusammenhang disjunktive und konjunktive Normalform

Die disjunktive und konjunktive Normalformen spielen eine besondere Rolle bei systematischen Verfahren zum Vereinfachen von Schaltfunktionen und für die zweistufige Schaltungsrealisierung, denn sie bestehen aus einer *Dekodierstufe* und einer nachgeschalteten *Codierstufe*.

DNF und KNF haben also beide eine zweistufige Realisierung:



Beziehung zwischen Mintermen und Maxtermen

$$K_i = \overline{D_i}$$

$$D_i = \overline{K_i}$$

Beispiel:

#	a	b	Minterme (Vollkonjunktion)	Maxterme (Volldisjunktion)
0	0	0	$K_0 = \overline{a} \cdot \overline{b}$	$D_0 = a + b$
1	0	1	$K_1 = \overline{a} \cdot b$	$D_1 = a + \overline{b}$
2	1	0	$K_2 = a \cdot \overline{b}$	$D_2 = \overline{a} + b$
3	1	1	$K_3 = a \cdot b$	$D_3 = \overline{a} + \overline{b}$

DNF und KNF beschreiben die selbe Schaltfunktion.

→ Die DNF enthalte K_i mit $i \in K$.

Dann enthält die KNF alle D_j

mit $j \in \{0, \ldots, 2^n-1 \setminus K\}$.

D.h., die Indizes der Maxterme sind die Indizes, die nicht bei den Mintermen enthalten sind, und umgekehrt.

→ Wenn die KNF angegeben ist, dann kann die DNF direkt angegeben werden, und umgekehrt.

Abkürzende Schreibweise von Mintermen und Maxtermen

x_3	x_2	x_1	y	Term
0	0	0	1	$m_0 = x'_3 \cdot x'_2 \cdot x'_1$
0	0	1	0	$M_1 = x_3 + x_2 + x'_1$
0	1	0	1	$m_2 = x'_3 \cdot x_2 \cdot x'_1$
0	1	1	1	$m_3 = x'_3 \cdot x_2 \cdot x_1$
1	0	0	0	$M_4 = x'_3 + x_2 + x_1$
1	0	1	0	$M_5 = x'_3 + x_2 + x'_1$
1	1	0	1	$m_6 = x_3 \cdot x_2 \cdot x'_1$
1	1	1	0	$M_7 = x'_3 + x'_2 + x'_1$

Kanonische Schreibweise:

DNF: $y = m_0 + m_2 + m_3 + m_6$

→ *Summe von Produkten*

KNF: $y = M_1 \cdot M_4 \cdot M_5 \cdot M_7$

→ *Produkt von Summen*

Kurzschreibweise:

$$y = \sum m(0,2,3,6)$$

bzw. $$y = \prod M(1,4,5,7)$$

Beachte: Kurzschreibweise für

- Negation: x'_i
- M_i für Maxterme
- m_i für Minterme

Umwandlung bei der zweistufigen Darstellung

1. Kanonische DNF in kanonische KNF

   Übertrage Mintermkurzschreibweise in Maxtermkurzschreibweise,
   ersetze die Mintermindizes durch die nicht aufgetretenen Indizes

   z.B.: $y = \sum m(0,2,3,6) = \prod M(1,4,5,7)$

2. Kanonische KNF in kanonische DNF

   Übertrage Maxtermkurzschreibweise in Mintermkurzschreibweise,
   ersetze die Maxtermindizes durch die nicht aufgetretenen Indizes

   z.B.: $y = \prod M(1,4,5,7) = \sum m(0,2,3,6)$

3. Kanonische DNF (KNF) von y in kanonische DNF (KNF) von y´

   Liste in der Mintermkurzschreibweise alle nicht aufgetretenen Indizes auf

   z.B.: $y = \sum m(0,2,3,6) \rightarrow y' = \sum m\,(1,4,5,7)$

   $= \prod M(1,4,5,7) \rightarrow \quad = \prod M\,(0,2,3,6)$

Resümee

Die zweistufige Logik (DNF, KNF) hat den Vorteil, dass

- systematisch arbeitende Verfahren gut anwendbar sind

  → nächstes Kapitel

- die sie realisierende Schaltung i.d.R. die Vorteile der kurzen Laufzeit und der kostengünstigen Implementierungsvarianten hat.

Gegenüberstellung der mathematischen und der technischen Begriffe/Realisierung:

Mathematische Begriffe	**Technische Begriffe**
binäre Variable	binäres Signal
unabhängige Variable	Eingangssignal
abhängige Variable	Ausgangssignal
Wert	Zustand
Boolesche Funktion	Schaltnetz
Verknüpfung	Gatter
Disjunktion	ODER-Gatter
Konjunktion	UND-Gatter
Negation	Invertierung
Identität	Leitungsverbindung