HOCHSCHULE OSNABRÜCK
UNIVERSITY OF APPLIED SCIENCES

Technische Grundlagen der Informatik

Boole‘sche Algebra

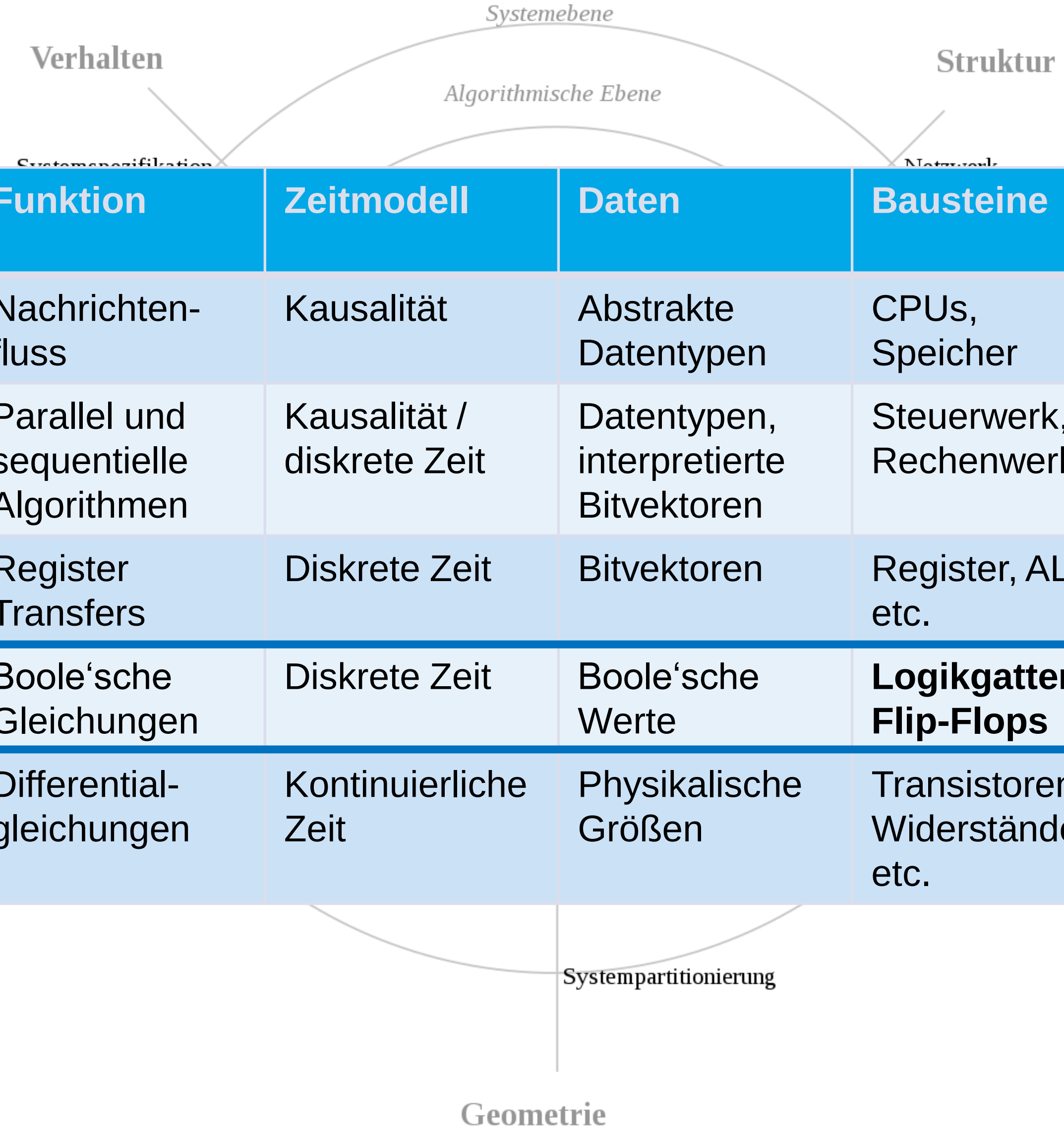
Prof. Dr.-Ing. Benjamin Weinert

Gliederung

HOCHSCHULE OSNABRÜCK
UNIVERSITY OF APPLIED SCIENCES

- Einordnung
- Schaltkreise und Wahrheitstafeln
- Boole‘sche Algebra
- Schaltalgebra
- Boole‘sche Funktionen
- Boole‘sche Ausdrücke
- Beweise durch vollständige Wahrheitstafeln
- Beweis durch Umformung Boole‘scher Ausdrücke
- Vereinfachung durch Umformung Boole‘scher Ausdrücke
- Gattersymbole
- Realisierung logischer Funktionen durch Logikgatter

Einordnung



Abstraktions-ebene	Funktion	Zeitmodell	Daten	Bausteine	Geometrie
Systemebene	Nachrichten-fluss	Kausalität	Abstrakte Datentypen	CPUs, Speicher	IC-Partitionierung
Algorithmische Ebene	Parallel und sequentielle Algorithmen	Kausalität / diskrete Zeit	Datentypen, interpretierte Bitvektoren	Steuerwerk, Rechenwerk	Floorplan
Register-Transfer-Ebene	Register Transfers	Diskrete Zeit	Bitvektoren	Register, ALUs, etc.	Makrozellen
Logikebene	Boole‘sche Gleichungen	Diskrete Zeit	Boole‘sche Werte	**Logikgatter, Flip-Flops**	Zellen
Schaltkreis-ebene	Differential-gleichungen	Kontinuierliche Zeit	Physikalische Größen	Transistoren, Widerstände, etc.	Masken-geometrie

Einordnung

- Für programmierbare Rechner existiert mit der Booleschen Algebra eine Formalisierung von Aussageverknüpfungen (=**Aussagenlogik**)

Boolesche Algebra

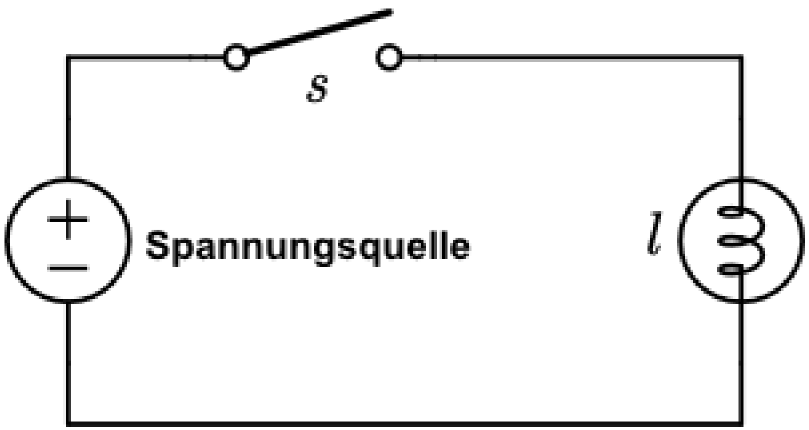
Mengenalgebra | Schaltalgebra

- Benannt nach dem englischen Mathematiker George Boole (1815-1864)
- 1937 (veröffentlicht 1938) benutzte der amerikanische Mathematiker und Elektrotechniker Claude Shannon (1916-2001) boolesche Algebren erstmals zur Beschreibung elektrischer Schaltungen
- Das Verständnis der Booleschen Algebra ist wichtig für den Bau effizienter Schaltungen zur Verarbeitung binärer Daten
  - Grundlage und mathematisches Werkzeug zum Bau von Rechnern
- In der Booleschen Algebra existieren nur zwei Wahrheitswerte:
  - 0 (falsch btw. false) und 1 (wahr btw. true)
- Beide Werte kann man als Bitwerte oder Stromzustände interpretieren

Schaltkreise und Wahrheitstafeln

HOCHSCHULE OSNABRÜCK
UNIVERSITY OF APPLIED SCIENCES

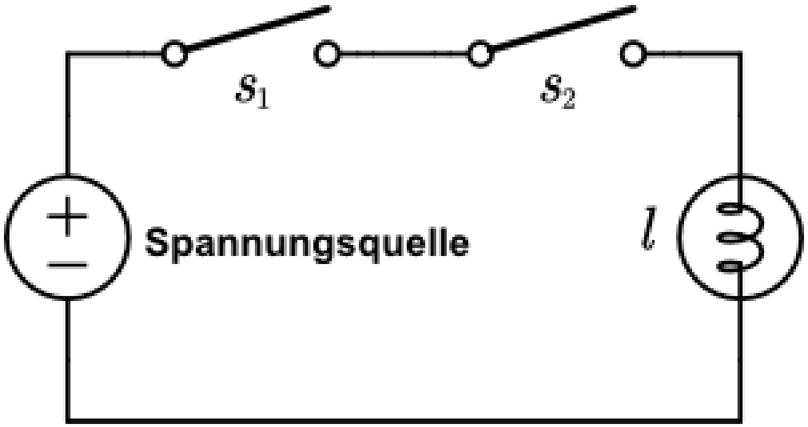
- Einfacher Stromkreis, bestehend aus
  - Spannungsquelle (z.B. Batterie)
  - Schalter s
  - Lampe l
- Schalter offen (s = 0)
  - Lampe aus (l = 0)
- Schalter geschlossen (s = 1)
  - Lampe an (l = 1)
- Darstellung als Wahrheitstafel



s	l
0	0
1	1

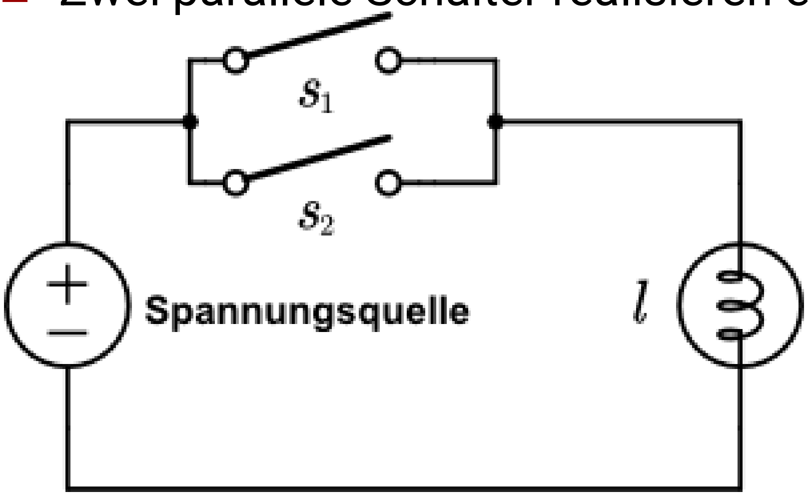
Schaltkreise und Wahrheitstafeln

- Zwei Schalter in Reihe realisieren eine logische **UND-Verknüpfung (<u>Konjunktion</u>)**



s_1	s_2	l
0	0	
0	1	
1	0	
1	1	

- Zwei parallele Schalter realisieren eine logische **ODER-Verknüfung (<u>Disjunktion</u>)**

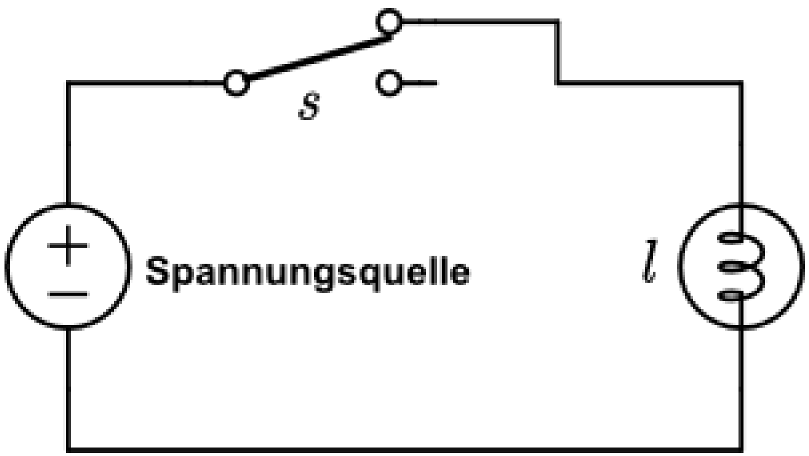


s_1	s_2	l
0	0	
0	1	
1	0	
1	1	

Schaltkreise und Wahrheitstafeln

- Schalter, bei dem im nicht-geschalteten Zustand Strom fließt
  - Logisches **Nicht (<u>Negation</u>)**

s	l
0	
1	



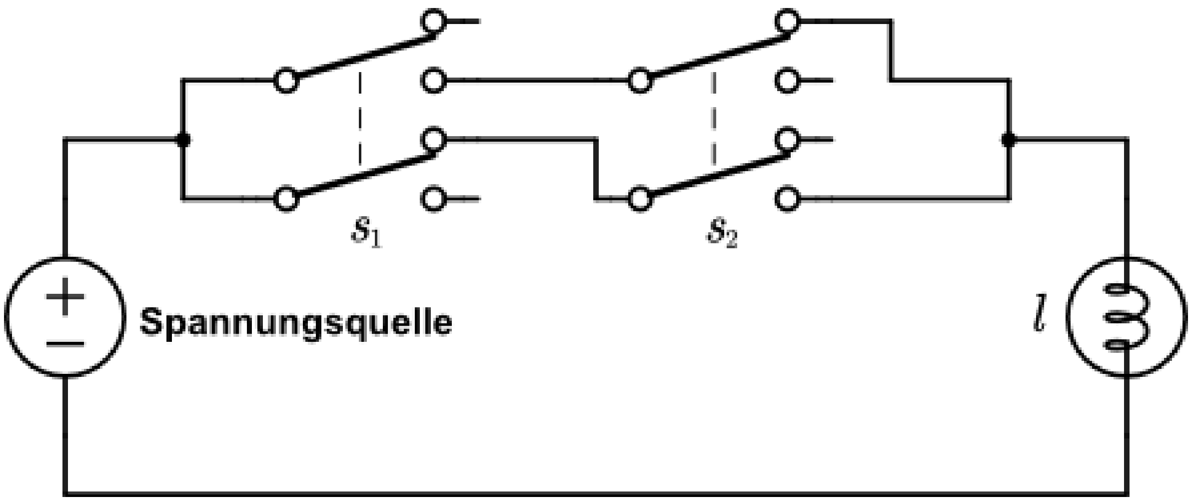
- Mit **UND**, **ODER** und **NICHT** lassen sich beliebige Schaltungen aufbauen
  - Wir verwenden im weiteren die englischen Bezeichnungen **AND**, **OR** und **NOT**

Schaltkreise und Wahrheitstafeln

- Wie könnte der Schaltkreis aussehen, der folgende Wahrheitstafel realisiert?

s_1	s_2	l
0	0	0
0	1	1
1	0	1
1	1	0

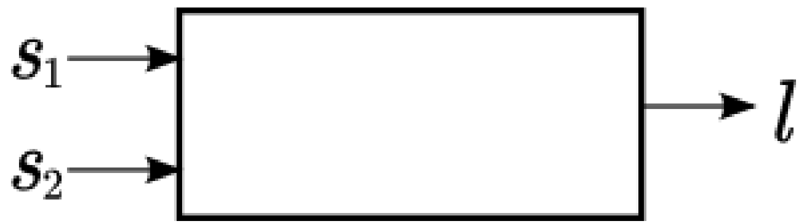
- Lösung: (s_1 = 0 AND s_2 = 1) OR (s_1 = 1 AND s_2 = 0)



Schaltkreise und Wahrheitstafeln

HOCHSCHULE OSNABRÜCK
UNIVERSITY OF APPLIED SCIENCES

- Wie viele mögliche Funktionen mit zwei Eingangsvariablen n gibt es?
- Wie viele Kombinationen der zwei Variablen kann es geben?



- Die Anzahl der Kombinationen berechnet sich aus 2^n

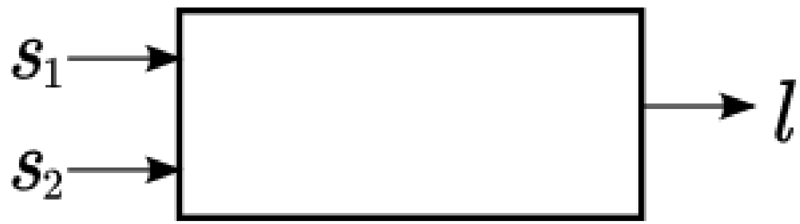
s_1	s_2	Mögliche Funktionen l_0 bis l_{15}															
0	0																
0	1																
1	0																
1	1																

- Allgemein: Bei n Eingangsvariablen gibt es $2^{(2^n)}$ mögliche Funktionen
  - Es gibt demnach 16 boolesche Funktionen

Schaltkreise und Wahrheitstafeln

HOCHSCHULE OSNABRÜCK
UNIVERSITY OF APPLIED SCIENCES

- Wie viele mögliche Funktionen mit zwei Eingangsvariablen n gibt es?
- Wie viele Kombinationen der zwei Variablen kann es geben?



- Die Anzahl der Kombinationen berechnet sich aus 2^n

s_1	s_2	Mögliche Funktionen l_0 bis l_{15}															
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

- Allgemein: Bei n Eingangsvariablen gibt es $2^{(2^n)}$ mögliche Funktionen
  - Es gibt demnach 16 boolesche Funktionen

Boole'sche Algebra

- Huntington'sche Axiome
  - Sei B eine nicht leere Menge von Elementen mit den binären Operationen
    $\bullet : B \times B \rightarrow B$
    $\circ : B \times B \rightarrow B$
  - Das Tripel $(\bullet, \circ, B)$ heißt Boole'sche Algebra, wenn für beliebige $a, b, c \in B$ folgende Axiome gelten:

	Operator $\bullet$	Operator $\circ$
Kommutativgesetz	$a \bullet b = b \bullet a$	$a \circ b = b \circ a$
Distributivgesetz	$a \bullet (b \circ c) = (a \bullet b) \circ (a \bullet c)$	$a \circ (b \bullet c) = (a \circ b) \bullet (a \circ c)$
Neutrale Elemente	$a \bullet e = a$	$a \circ n = a$
Inversion	$a \bullet a^{-1} = n$	$a \circ a^{-1} = e$

- **$e, n \in B$**
- **e** wird **Einselement** genannt; **n** wird **Nullelement** genannt

Boole‘sche Algebra

HOCHSCHULE OSNABRÜCK
UNIVERSITY OF APPLIED SCIENCES

- Ist der Körper der reellen Zahlen mit den Operationen für Multiplikation * und Addition + eine Boole‘sche Algebra ?

- **Nein**, da einige Axiome nicht gelten:

	Operator *	Operator +
Kommutativgesetz	a * b = b * a	a + b = b + a
Distributivgesetz	a * (b + c) = (a * b) + (a * c)	a + (b * c) = (a + b) * (a + c)
Neutrale Elemente	a * 1 = a	a + 0 = a
Inversion	$a * a^{-1} = 0$	$a + a^{-1} = 1$

Schaltalgebra

- Die **Schaltalgebra** ist eine **spezielle Boolesche Algebra**, die durch folgende Korrespondenztabelle definiert wird

Boolesche Algebra	Schaltalgebra
B	{0,1}
●	$\wedge$
○	$\vee$
n	0
e	1
a^{-1}	$\neg a$

Huntington'sche Axiome in der Schaltalgebra

HOCHSCHULE OSNABRÜCK
UNIVERSITY OF APPLIED SCIENCES

- Die Schaltalgebra ($\wedge$, $\vee$, B) mit
  - B = {0, 1}
  - UND-Verknüpfung $\wedge$
  - ODER-Verknüpfung $\vee$
  - Dem inversen Element, der NICHT-Operation $\neg$
- ist eine Boole'sche Algebra
- Es gelten daher laut Definition

	Operator $\wedge$	Operator $\vee$
Kommutativgesetz	$a \wedge b = b \wedge a$	$a \vee b = b \vee a$
Distributivgesetz	$a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$	$a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c)$
Neutrale Elemente	$a \wedge 1 = a$	$a \vee 0 = a$
Inversion	$a \wedge \neg a = 0$	$a \vee \neg a = 1$

- Dualitätsprinzip

Schaltalgebra

- Die Schaltalgebra besitzt folgende Grundoperationen

a	b	a ∧ b
0	0	0
0	1	0
1	0	0
1	1	1

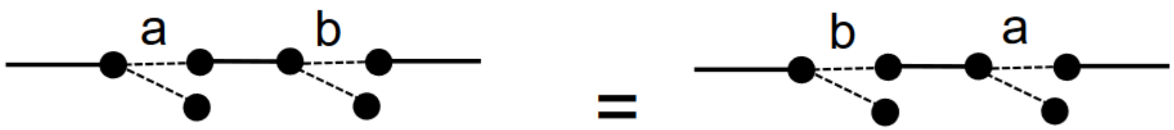
a	b	a ∨ b
0	0	0
0	1	1
1	0	1
1	1	1

a	¬a
0	1
1	0

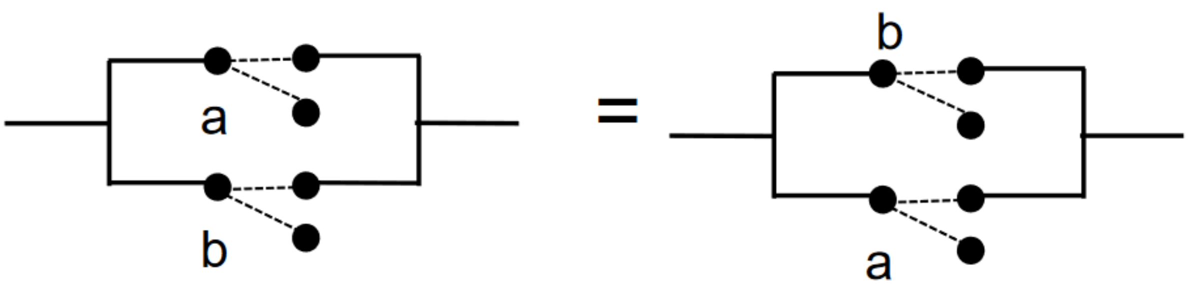
- Konjunktion Disjunktion Negation

Huntington‘sche Axiome in der Schaltalgebra

- Kommutativgesetz
  - $a \wedge b = b \wedge a$

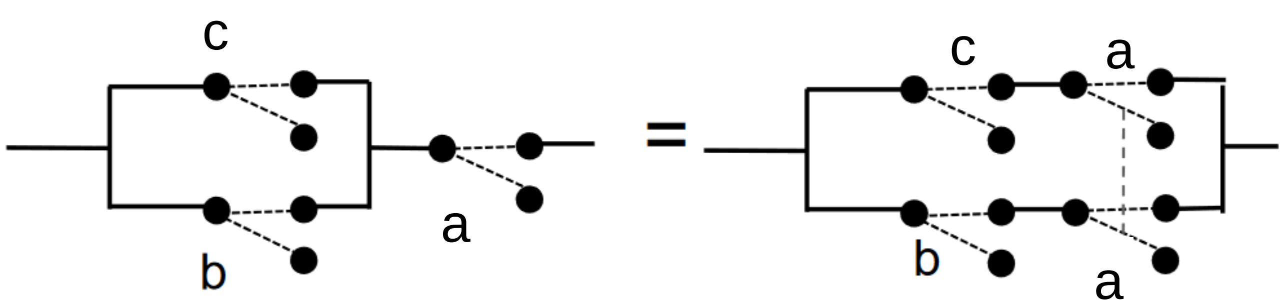


  - $a \vee b = b \vee a$

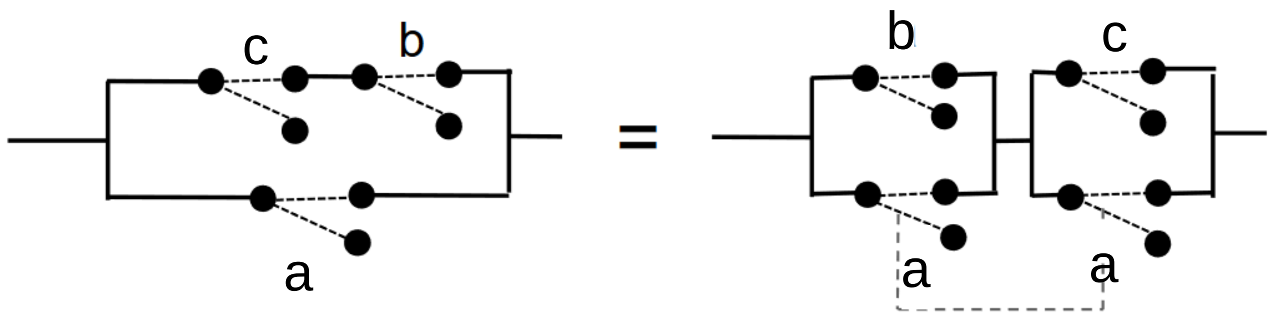


Huntington‘sche Axiome in der Schaltalgebra

- Distributivgesetz
  - a ∧ (b ∨ c) = (a ∧ b) ∨ (a ∧ c)

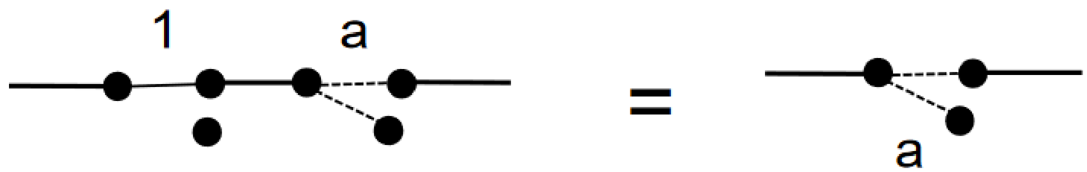


  - a ∨ (b ∧ c) = (a ∨ b) ∧ (a ∨ c)

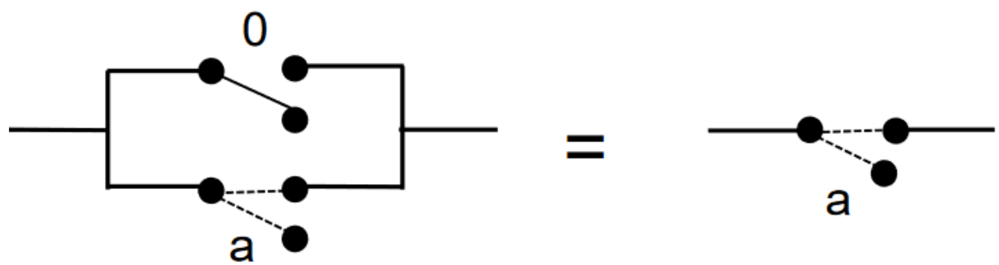


Huntington'sche Axiome in der Schaltalgebra

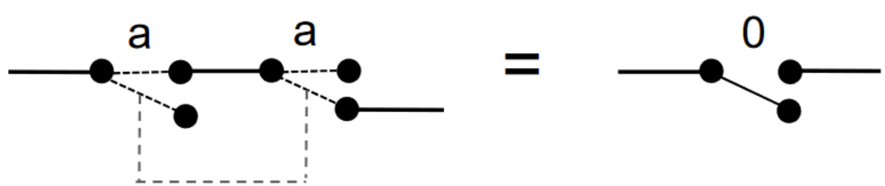
- Neutrale Elemente
  - $a \land 1 = a$



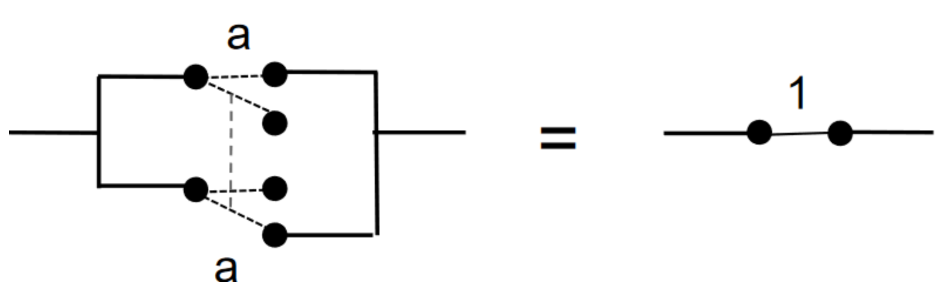
  - $a \lor 0 = a$



- Inversion
  - $a \land \neg a = 0$



  - $a \lor \neg a = 1$



Boole'sche Funktionen

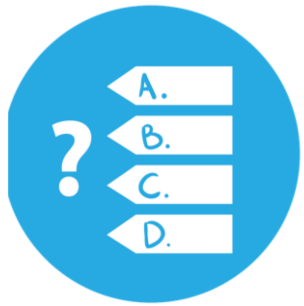
- Eine Funktion $y = f(x_1, x_2, \dots, x_{\mathbf{n}})$ mit $x_1, x_2, \dots, x_{\mathbf{n}} \in \{0, 1\}$ heißt Boole'sche Funktion der Stelligkeit **n**
- Boole'sche Funktionen werden auch Schaltfunktionen genannt
- Die **Eingabevariablen** $x_1, x_2, \dots, x_n$ werden als **freie** Variablen bezeichnet
- Die **Ausgabevariable** y wird **abhängige** Variable genannt

- Beispiele
  - Der Negationsoperator $y = \neg x_1$ ist eine Boole'sche Funktion der Stelligkeit 1
  - Der Konjunktionsoperator $y = x_1 \wedge x_2$ und der Disjunktionsoperator $y = x_1 \vee x_2$ sind Boole'sche Funktionen der Stelligkeit 2

Boole'sche Funktionen

- Boole'sche Funktionen können eindeutig durch vollständige Wahrheitstafeln beschrieben werden

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



Welche Funktion wird hier beschrieben?

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

- Diese Darstellung wird schnell sehr groß und unübersichtlich
  - → Formelschreibweise

Boole‘sche Ausdrücke

HOCHSCHULE OSNABRÜCK
UNIVERSITY OF APPLIED SCIENCES

- Die Darstellung Boole‘scher Funktionen in Formelschreibweise ist **nicht** eindeutig
- Beispiel – die Funktion XOR

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

- Verschiedene Boole‘sche Ausdrücke möglich
  - $y = (\neg x_1 \wedge x_2) \vee (x_1 \wedge \neg x_2)$
  - $y = (x_1 \vee x_2) \wedge \neg(x_1 \wedge x_2)$
  - ...
- Wie lassen sich diese Ausdrücke ineinander überführen?

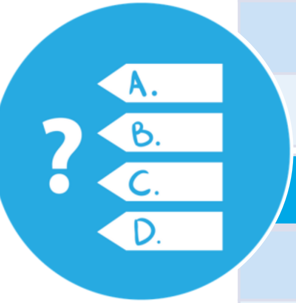
Boole‘sche Algebra
Axiome und Theoreme

#	Name	UND-Operator	ODER-Operator
1	Identität (Neutrale Elemente)	$a \land 1 = a$	$a \lor 0 = a$
2	Elimination	$a \land 0 = 0$	$a \lor 1 = 1$
3	Idempotenz	$a \land a = a$	$a \lor a = a$
4	Involution	$\neg(\neg a) = a$	
5	Inversion / Komplement	$a \land \neg a = 0$	$a \lor \neg a = 1$
6	Kommutativität	$a \land b = b \land a$	$a \lor b = b \lor a$
7	Assoziativität	$(a \land b) \land c = a \land (b \land c)$	$(a \lor b) \lor c = a \lor (b \lor c)$
8	Distributivität	$a \land (b \lor c) = (a \land b) \lor (a \land c)$	$a \lor (b \land c) = (a \lor b) \land (a \lor c)$
9	Vereinigung	$(a \lor b) \land (a \lor \neg b) = a$	$(a \land b) \lor (a \land \neg b) = a$
10	Absorption	$a \land (a \lor b) = a$	$a \lor (a \land b) = a$
11	Absorption 2	$(a \land \neg b) \lor b = a \lor b$	$(a \lor \neg b) \land b = a \land b$
12	Faktorisierung	$(a \lor b) \land (\neg a \lor c) = (a \land c) \lor (\neg a \land b)$	$(a \land b) \lor (\neg a \land c) = (a \lor c) \land (\neg a \lor b)$
13	Konsens	$(a \lor b) \land (b \lor c) \land (\neg a \lor c) = (a \lor b) \land (\neg a \lor c)$	$(a \land b) \lor (b \land c) \lor (\neg a \land c) = (a \land b) \lor (\neg a \land c)$
14	De Morgan	$\neg(a \land b \land \ldots) = \neg a \lor \neg b \lor \ldots$	$\neg(a \lor b \lor \ldots) = \neg a \land \neg b \land \ldots$

De Morgansche Gesetze

- Die beiden Gesetze von August de Morgan lauten:
  - ¬(a ∧ b) = ¬a ∨ ¬b
  - ¬(a ∨ b) = ¬a ∧ ¬b
- Die Gültigkeit kann leicht durch Wahrheitstafeln gezeigt werden

a	b	a ∧ b	¬(a ∧ b)	¬a	¬b	¬a ∨ ¬b
0	0					
0	1					
1	0					
1	1					



a	b	a ∨ b	¬(a ∨ b)	¬a	¬b	¬a ∧ ¬b
0	0					
0	1					
1	0					
1	1					

Weitere logische Operatoren

- Neben den elementaren logischen Operatoren (**AND**, **OR**, **NOT**) gibt es weitere wichtige Operatoren
  - Implikation und inverse Implikation
  - Äquivalenz und Antivalenz
  - Sheffer-Funktion (NAND) und Peirce-Funktion (NOR)
  - …
- Diese Operationen können jedoch alle unter Verwendung der elementaren Operationen realisiert werden
  - Erlauben aber eine kompaktere Formelschreibweise

Implikation und inverse Implikation

- Implikation: $y = a \rightarrow b = \neg a \lor b$
- Entspricht der Aussage: a ist eine hinreichende Bedingung für b
- Inverse Implikation: $y = a \leftarrow b = \neg b \lor a$

a	b	$a \rightarrow b$
0	0	1
0	1	1
1	0	0
1	1	1

a	b	$a \leftarrow b$
0	0	1
0	1	0
1	0	1
1	1	1

Äquivalenz und Antivalenz

- Äquivalenz: $y = a \leftrightarrow b = (\neg a \land \neg b) \lor (a \land b)$
  - **Wahr**, wenn beide Operanden den **gleichen** Wert haben
- Antivalenz: $y = a \nleftrightarrow b = (\neg a \land b) \lor (a \land \neg b)$
  - **Wahr**, wenn beide Operanden **unterschiedliche** Werte haben
  - Exklusives ODER (**XOR**)

a	b	$a \leftrightarrow b$
0	0	1
0	1	0
1	0	0
1	1	1

a	b	$a \nleftrightarrow b$
0	0	0
0	1	1
1	0	1
1	1	0

Sheffer-Funktion und Peirce-Funktion

- Sheffer-Funktion: $y = a \barwedge b = \neg(a \land b)$
  - UND-Verknüpfung mit anschließender Negation („Nicht-Und, NAND“)
- Peirce-Funktion: $y = a \veebar b = \neg(a \lor b)$
  - ODER-Verknüpfung mit anschließender Negation („Nicht-Oder, NOR“)

a	b	$a \barwedge b$
0	0	1
0	1	1
1	0	1
1	1	0

a	b	$a \veebar b$
0	0	1
0	1	0
1	0	0
1	1	0

Operatorsymbole nach DIN 66000

Operator	Bezeichnung	Beispiel	Sprechweise
¬	**Negation, NOT**	**y = ¬a**	**nicht a**
∧	**Konjunktion, AND**	**y = a ∧ b**	**a und b**
∨	**Disjunktion, OR**	**y = a ∨ b**	**a oder b**
→	Implikation	y = a → b	a impliziert b
↔	Äquivalenz	y = a ↔ b	a äquivalent b
↮	Antivalenz, XOR	y = a ↮ b	a xor b
$\bar{\wedge}$	Sheffer-Funktion, NAND	y = a $\bar{\wedge}$ b	a nand b
$\bar{\vee}$	Peirce-Funktion, NOR	y = a $\bar{\vee}$ b	a nor b

- Ein System von Operatoren, mit dem alle booleschen Funktionen dargestellt werden können, heißt **vollständiges Operatorensystem**
  - Die Operatoren ¬, ∧, ∨ bilden ein vollständiges Operatorensystem, Beispiel:
    - a ↔ b liefert das gleiche Ergebnis wie (a ∧ b) ∨ (¬a ∧ ¬b)
    - ↔ lässt sich demnach durch die Grundoperatoren ¬, ∧, ∨ ersetzen

Boolesche Funktionen

- Alle mögliche Funktionen f_0 bis f_{15} bei zwei Variablen, dargestellt durch Wahrheitstafel und algebraischen Ausdruck

		0	$a \wedge b$	$a \wedge \neg b$	a	$\neg a \wedge b$	b	$a \nLeftrightarrow b$	$a \vee b$	$a \,\bar{\vee}\, b$	$a \leftrightarrow b$	$\neg b$	$a \leftarrow b$	$\neg a$	$a \rightarrow b$	$a \,\bar{\wedge}\, b$	1
a	**b**	f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}
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
		Nullfunktion	Konjunktion	Inhibition	Identität	Inhibition	Identität	Antivalenz (XOR)	Disjunktion	Peirce-Funktion(NOR)	Äquivalenz	Negation	Inverse Implikation	Negation	Implikation	Sheffer-Funktion (NAND)	Tautologie / Einsfunktion

Beweise durch vollständige Wahrheitstafeln

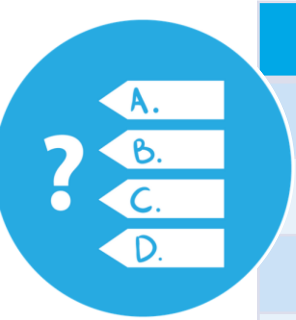
- Wie kommt man von der symbolischen Darstellung zur Wahrheitstafel
  - Durch Auswertung des symbolischen Ausdrucks

- Laut DIN gelten folgende Vorrangregeln:
  - Stärkste Bindung: $\neg$
  - Mittlere Bindung: $\wedge$, $\vee$, $\overline{\wedge}$, $\overline{\vee}$
  - Niedrigste Bindung: $\rightarrow$, $\leftrightarrow$, $\nleftrightarrow$

- Anders ausgedrückt:
  - Negation vor Konjunktion
  - Konjunktion vor Disjunktion
    - $\wedge$, $\vee$ werden linksassoziativ ausgewertet
  - Durch Klammern kann eine andere Reihenfolge der Auswertung festgelegt werden

Beweise durch vollständige Wahrheitstafeln

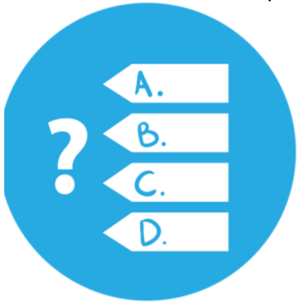
- Aussagen können leicht durch die Aufstellung vollständiger Wahrheitstafeln bewiesen werden
- Beispiel: Vereinigungstheorem
  - $(x \land y) \lor (x \land \neg y) = x$



x	y	x ∧ y	x ∧ ¬y	(x ∧ y) ∨ (x ∧ ¬y)
0	0			
0	1			
1	0			
1	1			

Beweis durch Umformung

- Für größere Funktionen ist die Verwendung von Wahrheitstafeln aufwendig
- Durch Verwendung der Axiome und Theoreme können Ausdrücke umgeformt und bewiesen werden
- Beispiel 1: Vereinigungstheorem $(x \land y) \lor (x \land \neg y) = x$
  - $(x \land y) \lor (x \land \neg y)$ Distributivität -> $a \land (b \lor c) = (a \land b) \lor (a \land c)$
  - $= x \land (y \lor \neg y)$ Komplement -> $a \lor \neg a = 1$
  - $= x \land 1$ Identität -> $a \land 1 = a$
  - x
- Beispiel 2: Absorptionstheorem $x \lor (x \land y) = x$
  - $x \lor (x \land y)$



Vereinfachung durch Umformung

- Beispiel
  - $(x \nleftrightarrow y) \vee (x \boxed{\leftrightarrow} y)$ — Auflösen
  - $= (\neg x \wedge y) \vee (x \wedge \neg y) \vee (\neg x \wedge \neg y) \vee (x \wedge y)$ — Kommutativität + Assoziativ
  - $= ((\neg x \wedge y) \vee (\neg x \wedge \neg y)) \vee ((x \wedge \neg y) \vee (x \wedge y))$ — Distributivität 2x
  - $= (\neg x \wedge (y \vee \neg y)) \vee (x \wedge (\neg y \vee y))$ — Komplement -> $a \vee \neg a = 1$
  - $= (\neg x \wedge 1) \vee (x \wedge 1)$ — Identität -> $a \wedge 1 = a$
  - $= \neg x \vee x$ — Komplement -> $a \vee \neg a = 1$
  - $= 1$
- Der Boole‘sche Ausdruck $(x \nleftrightarrow y) \vee (x \boxed{\leftrightarrow} y)$ ist also immer gleich 1
- Solche Ausdrücke heißen <u>allgemeingültig</u>

Tipps

- Beim Vereinfachen eines Ausdrucks von den innersten geklammerten Ausdrücken nach außen vorgehen
  - Eventuell Klammerpaare farblich markieren
- Immer zunächst in folgender Reihenfolge prüfen, ob sich der Ausdruck vereinfachen lässt:
  - Absorption
  - Absorption 2
  - Vereinigung
- Distributivität erst nach Prüfen aller anderen Möglichkeiten verwenden, sofern sich der Ausdruck verlängert

Literale und Monome

- Ein **Literal** ist eine **Variable** oder eine **negierte Variable**
  - a, b, ¬c, $\bar{x}$
- Ein **Monom** ist eine **UND-Verknüpfung** von Literalen
  - (a ∧ b ∧ ¬c) = $ab\bar{c}$, $\bar{f}\bar{e}\bar{d}$, $xy\bar{z}$
- Die Schaltfunktion eines Monoms kann nur dann 1 sein
  - Wenn jedes vorkommende Literal 1 ist, d.h.
    - Jede vorkommende Variable ist mit 1 belegt
    - Jede vorkommende negierte Variable ist mit 0 belegt

- Beispiel: Das Monom $\bar{x}y\bar{z}$ ist also nur dann 1, wenn x = z = 0 und y = 1

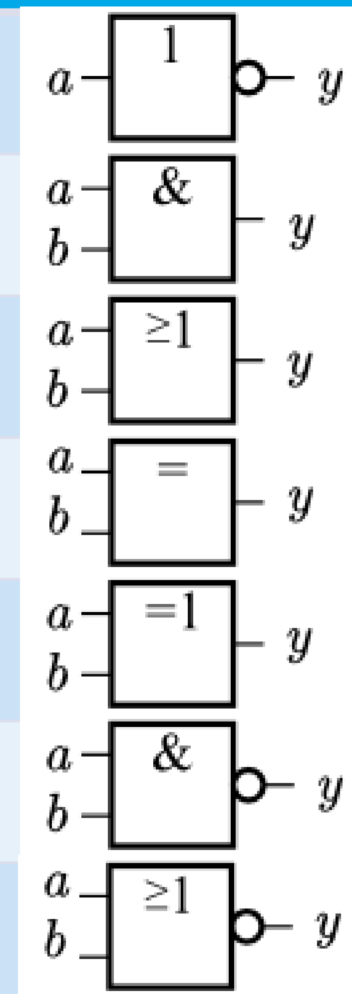
Literale und Monome

- Ein Boole'scher Ausdruck, der aus einer ODER-Verknüpfung von Monomen besteht, ist genau dann 1, wenn mindestens eines der Monome zu 1 evaluiert
- So evaluiert der Ausdruck y =$(\bar{x}y\bar{z}$ v $x\bar{y}z)$ zu 1, wenn x = z = 0 und y= 1 ODER x = z = 1 und y = 0
- Durch die ODER-Verknüpfung von Monomen kann somit eine beliebige Schaltfunktion abgebildet werden

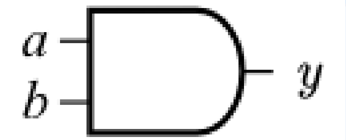
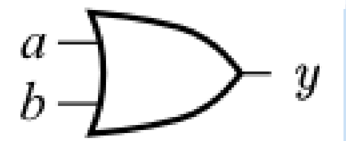
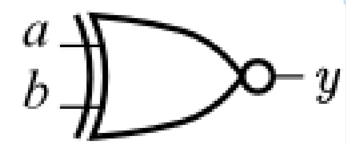
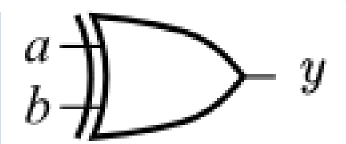
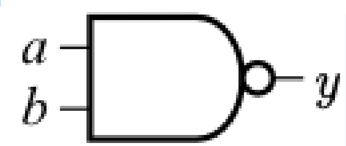
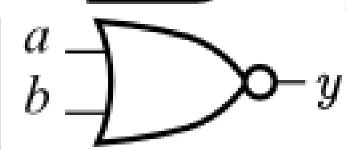
Gattersymbole nach DIN 40900

HOCHSCHULE OSNABRÜCK
UNIVERSITY OF APPLIED SCIENCES

Bezeichnung	Boole‘schr Ausdruck	Gattersymbol
Negation, NOT	$y = \neg a$	
Konjunktion, AND	$y = a \wedge b$	
Disjunktion, OR	$y = a \vee b$	
Äquivalenz	$y = a \leftrightarrow b$	
Antivalenz, XOR	$y = a \not\leftrightarrow b$	
Sheffer-Funktion, NAND	$y = a \barwedge b$	
Peirce-Funktion, NOR	$y = a \overline{\vee} b$	

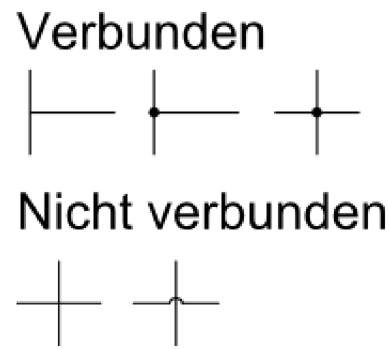


Gattersymbole US ANSI 91

Bezeichnung	Boole‘schr Ausdruck	Gattersymbol
negation, NOT	y = ¬a	
conjunction, AND	y = a Λ b	
disjunction, OR	y = a V b	
Equivalence, (E)XNOR	y = a ↔ b	
Exclusive disjunction, (E)XOR	y = a ↮ b	
NAND	y = a ⊼ b	
NOR	y = a ⊽ b	

Verwendete Notation

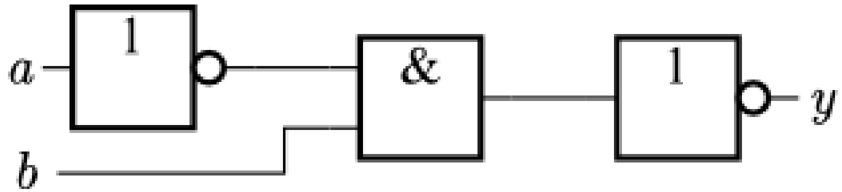
- Wir werden weitestgehend die DIN-Symbole verwenden
  - Aber die US-Symbole sind vor allem in der englischsprachigen Literatur sehr verbreitet
- Um aus mehreren Gattern Schaltnetze aufzubauen, können Ein- und Ausgänge von Gattern mit durchgezogenen rechtwinklig verlaufenden Leitungen verbunden werden

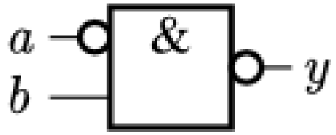


- Bei einer rechtwinkligen Verzweigung wird davon ausgegangen, dass eine Verbindung zwischen den Leitungen besteht
  - Dies kann zusätzlich mit einem ausgefüllten Punkt verdeutlicht werden
- Um verbundene Kreuzungen zu kennzeichnen, ist der ausgefüllte Punkt unbedingt erforderlich, ansonsten werden die Leitungen als nicht verbunden interpretiert
- Dass zwei sich kreuzende Leitungen nicht verbunden sind, kann zusätzlich durch einen Halbkreis verdeutlicht werden

Kompaktere Darstellung

- Sowohl Ein- als auch Ausgänge lassen sich invertieren, indem sie mit dem (nicht ausgefüllten) Negationskreis versehen werden
- Beispiel $y = \neg(\neg a \land b)$





- AND- und OR-Gatter können auch mehr als zwei Eingänge haben
- Ein OR-Gatter mit n Eingängen realisiert den Ausdruck $y = x_1 \lor x_2 \lor \ldots \lor x_n$
  Ein AND-Gatter mit n Eingängen realisiert den Ausdruck $y = x_1 \land x_2 \land \ldots \land x_n$
- Beispiel $y = (a \lor b \lor c) \land \neg d \land \neg e$