Boolesche Algebra

George Boole und seine Algebra mit nur zwei Werten

George Boole war ein englischer Mathematiker, der sich Mitte des 19. Jahrhunderts mit der formalen Sicht digitaler Strukturen beschäftigte.

Dabei entwickelte er die nach ihm benannte *boolesche Algebra*. In der booleschen Algebra existieren nur zwei Werte: 0 (*falsch* bzw. *false*) und 1 (*wahr* bzw. *true*).

Das Verständnis der booleschen Algebra ist wichtig für die Konstruktion und den Bau von effizienten Strukturen und Schaltungen zur Verarbeitung binärer Größen und bildet damit die Grundlage für die heutige Rechner-Hardware.

Boolesche Algebra
Operatoren

Hat man zwei Variablen a, $b \in B$, so lassen sich drei Operatoren der booleschen Algebra definieren, wobei sich für die gleichen Operationen unterschiedliche Schreibweisen (Symbole) eingebürgert haben.

- OR-Operator (logische Summe); geschrieben als + oder $\vee$.

a	b	a OR b
0	0	0
0	1	1
1	0	1
1	1	1

- Das Ergebnis einer OR-Operation ist 1, falls mindestens eine der Variablen in $a \vee b$ den Wert 1 besitzt.

Boolesche Algebra
Operatoren

– Hinsichtlich der Aussagenlogik gilt: Eine mit dem OR-Operator gebildete Gesamtbedingung ist bereits wahr, wenn nur eine der beiden mit $\vee$ bzw. + verknüpften Einzelbedingungen wahr ist.

Es ist Weihnachten $\vee$ Es schneit $\vee$ Sie haben Schnupfen

Diese Gesamtbedingung ist bereits erfüllt, wenn mindestens eine Einzelbedingung wahr ist. Hat man Schnupfen und es ist ein heißer Sommertag,dann ist die Gesamtbedingung erfüllt.

alter $<$ 6 $+$ alter $>=$ 67

Wenn `alter` kleiner als 6 oder aber größer als oder gleich 67 ist, dann ist diese Gesamtbedingung wahr, andernfalls falsch. Diese Verknüpfung ist also sowohl für noch nicht schulpflichtige Kinder als auch für Personen im Rentenalter erfüllt.

Boolesche Algebra
Operatoren

– AND-Operator (logisches Produkt); geschrieben als * oder $\wedge$ oder leer.

– Das Ergebnis einer AND-Operation ist 1, falls beide Variablen in $a \wedge b$ den Wert 1 besitzen.

– Hinsichtlich der Aussagenlogik gilt: Zwei mit dem AND-Operator ($\wedge$ oder *) verknüpfte Einzelbedingungen ergeben nur dann eine wahre Gesamtbedingung, wenn beide Einzelbedingungen wahr sind, ansonsten ist die Gesamtbedingung falsch.

a	b	a AND b
0	0	0
0	1	0
1	0	0
1	1	1

Es ist Weihnachten $\wedge$ Es schneit $\wedge$ Sie haben Schnupfen

Diese Gesamtbedingung ist nur dann erfüllt, wenn alle Einzelbedingungen wahr sind. Hat man z. B. keinen Schnupfen, dann ist auch die Gesamtbedingung nicht erfüllt, selbst wenn es Weihnachten ist und es schneit.

Boolesche Algebra
Operatoren

– NOT-Operator (Invertierung); geschrieben als $-a$, $\neg a$ oder $\overline{a}$

– Das Ergebnis einer NOT-Operation ist 1, falls die entsprechende Variable den Wert 0 besitzt. Besitzt die Variable den Wert 1, so ist das Ergebnis der Wert 0.

a	NOT a
0	1
1	0

$\neg$ **(alter >= 6** $\wedge$ **alter < 67)**

Diese Verknüpfung deckt wie das Beispiel beim OR-Operator sowohl alle noch nicht schulpflichtigen Kinder als auch Personen im Rentenalter ab, denn diese Bedingung bedeutet: Alle Personen, deren Alter **nicht** im Intervall [6,67[liegt.

Boolesche Algebra
Operatoren

– Im vorherigen Beispiel wurde eines der *De Morgan'schen Gesetze* angewendet:

$$-a + -b = -(a * b) \quad \text{bzw.} \quad \overline{a} + \overline{b} = \overline{a * b}$$

$$-a * -b = -(a + b) \quad \text{bzw.} \quad \overline{a} * \overline{b} = \overline{a + b}$$

Es wurde also die Bedingung aufgestellt:

$\overline{\text{alter} < 6} \wedge \overline{\text{alter} >= 67}$ was bedeutet:

„nicht jünger als 6 und nicht schon 67"

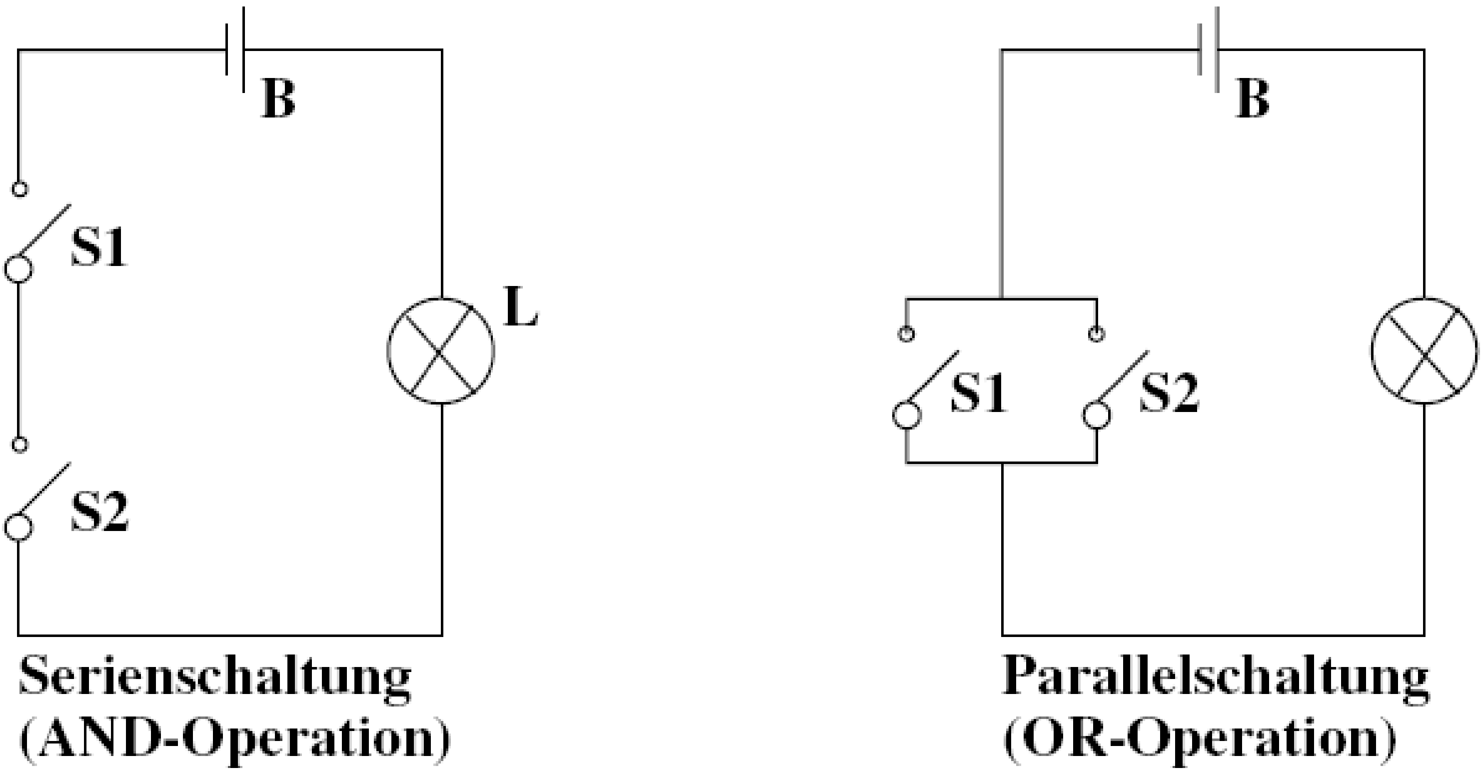
– Durch Anwendung des De Morgan'schen Gesetzes erhält man dann: $\overline{\overline{\text{alter} < 6} + \overline{\text{alter} >= 67}} = \overline{\text{alter} >= 6 + \text{alter} < 67}$

– Eine boolesche Menge B zusammen mit diesen drei Operatoren wird als boolesche Algebra (B, AND, OR, NOT) oder (B, $+$, $*$, $\neg$) oder auch B($\wedge$, $\vee$, $-$) bezeichnet.

Boolesche Algebra
Boolesche Schaltungen

- Die zuvor vorgestellten booleschen Operatoren OR und AND kann man auch durch so genannte boolesche Schaltungen verdeutlichen.
  - Man nimmt einen einfachen Stromkreis her, der aus einer Batterie B, einer Lampe L und einem Schalter S besteht. Ist ein Schalter offen, so bedeutet dies den Wert 0, während ein geschlossener Schalter für den Wert 1 steht.
  - Die Lampe hat nur dann den Wert 1, also leuchtet
    - bei der *Serienschaltung*: wenn *beide Schalter den Wert 1 haben*, also geschlossen sind,
    - bei der *Parallelschaltung*: wenn *mindestens ein Schalter den Wert 1 hat*, also geschlossen ist.

Boolesche Algebra
Boolesche Schaltungen



Serien- und Parallelschaltung für den AND- und OR-Operator

Boolesche Algebra
Axiome

- In einer booleschen Algebra gelten verschiedene Gesetze (Axiome), die zur Manipulation logischer Gleichungen hilfreich sind.
  – Da * höhere Priorität als + hat, wird * oft auch weggelassen, um die Lesbarkeit zu erhöhen.

Boolesche Algebra
Axiome

Kommutativgesetze	a * b = b * a	$ab = ba$
	a + b = b + a	
Assoziativgesetze	a * (b * c) = (a * b) * c	$a(bc) = (ab)c$
	a + (b + c) = (a + b) + c	
Distributivgesetze	a * (b + c) = (a * b) + (a * c)	$a(b + c) = ab + ac$
	a + (b * c) = (a + b) * (a + c)	$a + bc = (a + b)(a + c)$
Identitätsgesetze	a * 1 = a	$a1 = a$
	a + 0 = a	
Null-/Einsgesetze	a * 0 = 0	$a0 = 0$
	a + 1 = 1	
Komplementärgesetze	a * -a = 0	$a\overline{a} = 0$
	a + -a = 1	$a + \overline{a} = 1$

Boolesche Algebra
Axiome

Idempotenzgesetze	a * a = a a + a = a	$aa = a$
Verschmelzungsgesetze	a * (a + b) = a a + (a * b) = a	$a(a + b) = a$ $a + ab = a$
De Morgan'sche Gesetze	-(a * b) = -a + -b -(a + b) = -a * -b	$\overline{ab} = \overline{a} + \overline{b}$ $\overline{a + b} = \overline{a}\,\overline{b}$
Doppeltes Negationsgesetz	-(-a) = a	$\overline{\overline{a}} = a$

Boolesche Algebra
Axiome

Wahrheitstabelle zu den *De Morgan'schen Gesetzen*

a	b	ab	$\overline{ab}$	$\overline{a}$	$\overline{b}$	$\overline{a}+\overline{b}$	a+b	$\overline{a+b}$	$\overline{a}$	$\overline{b}$	$\overline{a}\,\overline{b}$
		$\overline{ab}=\overline{a}+\overline{b}$					$\overline{a+b}=\overline{a}\,\overline{b}$				
0	0	0	**1**	1	1	**1**	0	**1**	1	1	**1**
0	1	0	**1**	1	0	**1**	1	**0**	1	0	**0**
1	0	0	**1**	0	1	**1**	1	**0**	0	1	**0**
1	1	1	**0**	0	0	**0**	1	**0**	0	0	**0**

Boolesche Algebra
Funktionen

Ausgehend von den betrachteten drei Operatoren wollen wir allgemein definieren, was eine boolesche Algebra ist:

Eine Funktion $f\colon B^n \to B$ heißt n-stellige boolesche Funktion.

– Eine solche Funktion bildet die Menge aller möglichen n-Tupel aus {0,1} auf {0,1} ab und kann in Form einer Tabelle mit 2^{2^n} Werten dargestellt werden.

n = 1: ergibt vier einstellige boolesche Funktionen

$$f(x) = 0\,, \quad f(x) = 1\,, \quad f(x) = x\,, \quad f(x) = -x$$

Boolesche Algebra
Funktionen

n = 2: ergibt 16 zweistellige boolesche Funktionen

b=	0	1	0	1	Term	Bezeichnung	Sprech- bzw.
a=	0	0	1	1			Schreibweise
f0	0	0	0	0	0	Nullfunktion	
f1	**0**	**0**	**0**	**1**	ab	**Konjunktion**	**a AND b**
f2	**0**	**0**	**1**	**0**	$a\overline{b}$	**1. Differenz**	**a AND NOT b**
f3	0	0	1	1	a	1. Identität	
f4	**0**	**1**	**0**	**0**	$\overline{a}b$	**2. Differenz**	**NOT a AND b**
f5	0	1	0	1	b	2. Identität	
f6	**0**	**1**	**1**	**0**	$\overline{a}b + a\overline{b}$	**Antivalenz**	**a XOR b**
f7	**0**	**1**	**1**	**1**	$a + b$	**Disjunktion**	**a OR b**
f8	**1**	**0**	**0**	**0**	$\overline{a + b}$	**Negatdisjunktion**	**a NOR b**

Boolesche Algebra
Funktionen

n = 2: ergibt 16 zweistellige boolesche Funktionen

f9	**1**	**0**	**0**	**1**	$(\overline{a}+b)(a+\overline{b})$	**Äquivalenz**	**$a \leftrightarrow b$**
f10	1	0	1	0	$\overline{b}$	2. Negation	NOT b
f11	**1**	**0**	**1**	**1**	$a+\overline{b}$	**2. Implikation**	**$b \rightarrow a$: Wenn b, dann a**
f12	1	1	0	0	$\overline{a}$	1. Negation	NOT a
f13	**1**	**1**	**0**	**1**	$\overline{a}+b$	**1. Implikation**	**$a \rightarrow b$ Wenn a, dann b**
f14	**1**	**1**	**1**	**0**	$\overline{ab}$	**Negatkonjunktion**	**a NAND b**
f15	1	1	1	1	1	Einsfunktion	

Boolesche Algebra
Funktionen

– Für die einzelnen Funktionen gilt dabei Folgendes:
  - *Null- bzw. einstellige Funktionen sind nicht von Interesse* Die Funktionen f0, f3, f5, f10, f12 und f15 sind uninteressant, da es gewissermaßen auf zwei Argumente aufgeblasene null- bzw. einstellige Funktionen sind.
  - *Einfache Funktionen mit den Operatoren AND, OR und NOT*

		f1	**f2**	**f4**	**f7**
a	b	ab	$a\overline{b}$	$\overline{a}b$	$a+b$
0	0	0	0	0	0
0	1	0	0	1	1
1	0	0	1	0	1
1	1	1	0	0	1

Boolesche Algebra
Funktionen

- *Andere Funktionen und ihre Nachbildung mit AND, OR und NOT*
  Für die restlichen Funktionen wird in der letzten Zeile noch gezeigt, wie diese sich mittels der Operatoren AND, OR und NOT realisieren lassen.

		f6	f8	f9	f11	f13	f14
a	b	a XOR b	a NOR b	$a \leftrightarrow b$	$b \rightarrow a$	$a \rightarrow b$	a NAND b
0	0	0	1	1	1	1	1
0	1	1	0	0	0	1	1
1	0	1	0	0	1	0	1
1	1	0	0	1	1	1	0
		$\overline{a}b + a\overline{b}$	$\overline{a + b}$	$(\overline{a} + b)(a + \overline{b})$	$a + \overline{b}$	$\overline{a} + b$	$\overline{ab}$

Boolesche Algebra

Funktionen

Es lassen sich nun folgende Sätze aufstellen:

1. *Satz: Alle zweistelligen booleschen Funktionen können mit Hilfe der Negation (–), der Konjunktion (*) und der Disjunktion (+) dargestellt werden.*
2. *Satz: Alle zweistelligen booleschen Funktionen können entweder mit Hilfe der Negation und der Konjunktion, oder mit Hilfe der Negation und der Disjunktion dargestellt werden.*
3. *Satz: Alle zweistelligen booleschen Funktionen können entweder mit Hilfe der NAND-Verknüpfung oder mit Hilfe der NOR-Verknüpfung dargestellt werden.*

Der letzte Satz hat zur Folge, dass alle Schaltnetze aus Gattern einer einzigen Art, wie z.B. nur aus NAND-Gattern, aufgebaut werden können.

Boolesche Algebra
Funktionen

n-stellige boolesche Funktionen

– Allgemein gilt, dass es für jedes beliebige $n \in \mathbb{N}$ mit $n \geq 1$ genau 2^{2^n} n-stellige boolesche Funktionen gibt. Zudem kann gezeigt werden, dass alle höherstelligen booleschen Funktionen (n≥3) durch Verknüpfungen 2-stelliger boolescher Funktionen aufgebaut werden können.