5. Aussagenlogik und Schaltalgebra

- **Aussageformen und Aussagenlogik**
- **Boolesche Terme und Boolesche Funktionen**
- **Boolesche Algebra**
- **Schaltalgebra**
- **Schaltnetze und Schaltwerke**

Aussagen

- Information oft in Aussagen enthalten ("Die Ampel zeigt rot", "Das Wetter ist schön")
- Aber: Der Begriff der Aussage bedarf einer genaueren Definition.
- Beispiele für unscharfe Definition:
  - Eine Aussage ist ein sprachliches Gebilde, von dem es sinnvoll ist zu sagen, es sei wahr oder falsch (Aristoteles).
    -> Problem auf die Definition von "sinnvoll" verlagert.
  - Dem Satz "Die Aussage dieses Satzes ist falsch" läßt sich kein Wahrheitswert zuordnen.
- **Ausweg**: Formale Sprache mit der ein zwar eingeschränktes aber konsistentes System von Aussageformen konstruiert werden kann.

•**Beispiel**: Aussagenlogik.

Aussagenlogik I

- Einfaches formales System zur Repräsentation von Aussagen.
- Zweiwertige Logik (Jede Aussage ist entweder wahr oder falsch)
- Jede Aussage als Wort über einem Zeichenvorrat M = {Z,O} formuliert, wobei Z die Menge der binären Aussagevariablen (Typ Boolean) und O die Menge der Junktoren (Operatoren) ist, Bsp.:

$$Z = \{a, b, c, \cdots\}, \quad O = \{\neg, \wedge, \vee\}$$

- Die Junktoren sind:

Negation	$\neg$ (nicht)
Disjunktion	$\vee$ (oder)
Konjunktion	$\wedge$ (und)

Die Operatoren sind nach ihrer Bindungsstärke geordnet. Die Ordnung (Priorisierung) ist:
nicht vor ***und*** vor ***oder***

Aussagenlogik II

- Die Aussagen sind als *wohldefinierte* Wörter über diesem Zeichenvorrat formuliert. Diese heißen auch Formeln. Die Syntax der Aussagen ist durch folgende induktive Vorschrift gegeben:
  - **Basis**: Jede Aussagenvariable ist eine (atomare) Formel.
  - **Induktion:** Wenn A und B Formeln sind, dann sind auch

$$(A), \neg A, A \wedge B \text{ und } A \vee B$$

  Formeln.
  - **Vollständigkeit:** Alle Formeln werden durch die wiederholte Anwendung vorstehender Regeln erzeugt.

Wir bezeichnen nicht-atomare Formeln mit Großbuchstaben. Eine Formel heißt n-stellig, wenn sie n binäre Variable enthält. Beispiele:

$A(a)$: $a \vee \neg a$ (einstellig)

$B(a,b)$: $(a \vee b) \wedge \neg a$ (zweistellig)

$C(a,b,c)$: $a \vee c \wedge \neg b$ (dreistellig)

Interpretation einer Formel

❑ **Boolesche Ausdrücke sind nur eine syntaktische Konstruktion**

⇨Bedeutung erhält ein Boolescher Ausdruck erst, wenn den Variablen Wahrheitswerte zugeordnet werden. Diese sind aus der Menge {true, false} oder {wahr,falsch } oder {0,1}.

=> Belegung der Variablen.

❑ **Interpretation (Auswertung) einer Formel bzw. eines Ausdruckes**

⇨Belegung der binären Variablen eines Booleschen Ausdrucks mit Wahrheitswerten

⇨Zuweisung des Wahrheitswertes der Formel

❑ Beide Schritte der Interpretation werden durch eine Wahrheitstabelle realisiert. Eine Wahrheitstabelle legt zunächst die Semantik der Junktoren fest. Für die bisher eingeführten Junktoren und die zusätzlichen Junktoren der Implikation (=>) und Äquivalenz gilt

A B	NICHT A	A UND B	A ODER B	A => B	A⇔B
W W	F	W	W	W	W
W F	F	F	W	F	F
F W	W	F	W	W	F
F F	W	F	F	W	W

.

Die Belegung mit Wahrheitswerten führt eine Semantik ein. Zwei Formeln, die für alle Belegungen immer den gleichen Wahrheitswert liefern, heißen semantisch äquivalent.

Gültigkeit einer Formel

Def.: Die Interpretation, die eine Formel P zu einer wahren Aussage macht, heißt ein Modell der Formel. Es gilt:

• Eine Formel heißt erfüllbar (konsistent), wenn sie mindestens ein Modell hat. Beispiel: $a \wedge b$ Modell ist (die Interpretation mit) Belegung a =1 und b=1.

• Eine Formel heißt nicht erfüllbar (kontradiktorisch), wenn sie kein Modell hat. Beispiel: $a \wedge \neg a$

• Eine Formel heißt eine Tautologie, wenn sie für alle Belegungen wahr ist (Alle Belegungen liefern ein Modell der Formel). Beispiel: $a \vee \neg a$

Äquivalenz von Formeln

$\neg\neg A \Leftrightarrow A$	**Doppelte Negation**
$A\wedge B \Leftrightarrow B\wedge A$	**Kommutativität der Konjunktion**
$A\vee B \Leftrightarrow B\vee A$	**Kommutativität der Disjunktion**
$A\wedge (B\wedge C) \Leftrightarrow (A\wedge B)\wedge C$	**Assoziativität (analog Disjunktion)**
$A\wedge (B\vee C) \Leftrightarrow (A\wedge B)\vee (A\wedge C)$	**Distributivität UND über ODER**
$A\vee (B\wedge C) \Leftrightarrow (A\vee B)\wedge (A\vee C)$	**Distributivität ODER über UND**
$(A \Rightarrow B) \Leftrightarrow (\neg A\vee B)$	**Implikation**
$\neg(A\wedge B) \Leftrightarrow \neg A\vee \neg B$ **$\neg(A\vee B) \Leftrightarrow \neg A \wedge \neg B$**	**De Morgan**
$A \wedge (A\vee B) \Leftrightarrow A$ **$A \vee (A \wedge B) \Leftrightarrow A$**	**Neutralität**

Operatorensysteme

Def.: Ein vollständiges Operatorensystem erlaubt die Darstellung beliebiger aussagenlogischer Formeln mit dem gewählten Satz von Operatoren.

Boolesche Basis	$\{\neg, \wedge, \vee\}$
De – Morgan – Basis	$\{\neg, \wedge\}$ oder $\{\neg, \vee\}$
Frege – Basis	$\{\neg, \Rightarrow\}$
NAND – Basis	$\{\overline{\wedge}\}$
NOR – Basis	$\{\overline{\vee}\}$

Boolesche Funktionen

Def.: Es sei ein n-Tupel von binären Variablen (x_1, x_2,...,x_n) gegeben. Eine <u>n-stellige Boolesche Funktion</u> ordnet jeder Belegung der Variablen x_1, x_2,...,x_n mit den Wahrheitswerten „wahr“ oder „falsch“ genau einen Wahrheitswert zu.

f: {wahr,falsch}^{n} -> {wahr,falsch} **oder** $f : B^n \to B$ **mit** $B = \{0,1\}$

Es gibt genau 2^n verschiedene Belegungen der Variablen einer n – stelligen Booleschen Funktion.

Die Anzahl der n – stelligen Booleschen Funktionen ist 2^{2^n}.

Bew: Über Funktionstabelle

Zweistellige Boolesche Funktionen

Benennung der Verknüpfung	Funktionswert $y = f(x_1, x_2)$		Schreibweise mit den Zeichen $\wedge \vee -$	Bemerkung
	x_1 =	0 1 0 1		
	x_2 =	0 0 1 1		
Null	y_0 =	0 0 0 0	0	Null
UND–Verknüpfung	y_1 =	0 0 0 1	$x_1 \wedge x_2$	x_1 UND x_2
Inhibition	y_2 =	0 0 1 0	$\overline{x}_1 \wedge x_2$	
Transfer	y_3 =	0 0 1 1	x_2	
Inhibition	y_4 =	0 1 0 0	$x_1 \wedge \overline{x}_2$	
Transfer	y_5 =	0 1 0 1	x_1	
Antivalenz	y_6 =	0 1 1 0	$(x_1 \wedge \overline{x}_2) \vee (\overline{x}_1 \wedge x_2)$	Exclusiv–ODER
ODER–Verknüpfung	y_7 =	0 1 1 1	$x_1 \vee x_2$	x_1 ODER x_2
NOR–Verknüpfung	y_8 =	1 0 0 0	$\overline{x_1 \vee x_2}$	NICHT–ODER
Äquivalenz	y_9 =	1 0 0 1	$(x_1 \wedge x_2) \vee (\overline{x}_1 \wedge \overline{x}_2)$	
Komplement	y_{10} =	1 0 1 0	$\overline{x}_1$	
Implikation	y_{11} =	1 0 1 1	$\overline{x}_1 \vee x_2$	
Komplement	y_{12} =	1 1 0 0	$\overline{x}_2$	
Implikation	y_{13} =	1 1 0 1	$x_1 \vee \overline{x}_2$	
NAND–Verknüpfung	y_{14} =	1 1 1 0	$\overline{x_1 \wedge x_2}$	NICHT–UND
Eins	y_{15} =	1 1 1 1	1	Eins

Operatorensysteme

Def.: Ein vollständiges Operatorensystem erlaubt die Darstellung beliebiger Boolescher Funktionen mit einer beschränkten Anzahl von Operatoren

❑ Beispiele für vollständige Operatorensystem:

Operatorensystem	**Negation**	**Konjunktion**	**Disjunktion**
$(\wedge, \vee, \bar{\ })$	$\bar{x}_1$	$x_1 \wedge x_2$	$x_1 \vee x_2$
$(\wedge, \bar{\ })$	$\bar{x}_1$	$x_1 \wedge x_2$	$\overline{\bar{x}_1 \wedge \bar{x}_2}$
$(\vee, \bar{\ })$	$\bar{x}_1$	$\overline{\bar{x}_1 \vee \bar{x}_2}$	$x_1 \vee x_2$
$(\overline{\wedge})$	$x_1 \overline{\wedge} x_1$	$(x_1 \overline{\wedge} x_2) \overline{\wedge} (x_1 \overline{\wedge} x_2)$	$(x_1 \overline{\wedge} x_1) \overline{\wedge} (x_2 \overline{\wedge} x_2)$
$(\overline{\vee})$	$x_1 \overline{\vee} x_1$	$(x_1 \overline{\vee} x_1) \overline{\vee} (x_2 \overline{\vee} x_2)$	$(x_1 \overline{\vee} x_2) \overline{\vee} (x_1 \overline{\vee} x_2)$
$(\wedge, \oplus)$	$x_1 \oplus 1$	$x_1 \wedge x_2$	$x_1 \wedge x_2 \oplus x_1 \oplus x_2$
$(\vee, \equiv)$	$x_1 \equiv 0$	$(x_1 \vee x_2) \equiv x_1 \equiv x_2$	$x_1 \vee x_2$

❑ Zum Wahrheitswert einer Aussage gelangt man durch rekursives Auswerten der Booleschen Funktionen in einem Ausdruck unter Beachtung der Prioritäten der Operatoren, d.h.
Negation vor Konjunktion, Konjunktion vor Disjunktion. Klammerung beachten!

Boolesche Algebra

❑ George Boole (1815-1864): **Algebra der Logik (Boolesche Algebra)**

Def.: Eine <u>Boolesche Algebra</u> ist eine algebraische Struktur ($V; ◊, \#$), bestehend aus

a) einer Menge *V* mit mindestens zwei Elementen

b) den zweistelligen Verknüpfungen

$◊ : V \times V \rightarrow V$ (Boolesches Produkt)

$\# : V \times V \rightarrow V$ (Boolesche Summe),

die den Huntingtonschen Axiomen (s.u.) genügen:

Nachbemerkung: Eine algebraische Struktur ist eine Menge mit auf ihr definierten Verknüpfungen, deren Ergebnisse wieder in dieser Menge liegen.

Die Huntingtonschen Axiome

- ❑ Kommutativgesetze:
  $\boldsymbol{a \lozenge b = b \lozenge a}$
  $\boldsymbol{a \# b = b \# a}$
- ❑ Distributivgesetze:
  $\boldsymbol{a \lozenge (b \# c) = (a \lozenge b) \# (a \lozenge c)}$
  $\boldsymbol{a \# (b \lozenge c) = (a \# b) \lozenge (a \# c)}$
- ❑ Neutrale Elemente:
  Es existieren zwei Elemente *e*, $n \in V$, so daß gilt:
  $\boldsymbol{a \lozenge e = a}$ (*e* wird <u>Einselement</u> genannt)
  $\boldsymbol{a \# n = a}$ (*n* wird <u>Nullelement</u> genannt)
- ❑ Inverse Elemente:
  Für alle $a \in V$ gibt es ein *a*, so daß gilt:
  $\boldsymbol{a \lozenge a = n}$
  $\boldsymbol{a \# a = e}$

Weitere Sätze

- ❑ Aus den vier Huntingtonschen Axiomen lassen sich weitere Sätze ableiten

  - ⇨ Assoziativgesetze
    $(\mathbf{x_1} \lozenge \mathbf{x_2}) \lozenge \mathbf{x_3} = \mathbf{x_1} \lozenge (\mathbf{x_2} \lozenge \mathbf{x_3})$
    $(\mathbf{x_1} \# \mathbf{x_2}) \# \mathbf{x_3} = \mathbf{x_1} \# (\mathbf{x_2} \# \mathbf{x_3})$

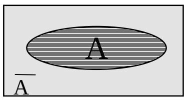
  - ⇨ Idempotenzgesetze
    $(\mathbf{x_1} \lozenge \mathbf{x_1}) = \mathbf{x_1}$ $\qquad$ $(\mathbf{x_1} \# \mathbf{x_1}) = \mathbf{x_1}$

  - ⇨ Absorptionsgesetze
    $\mathbf{x_1} \lozenge (\mathbf{x_1} \# \mathbf{x_2}) = \mathbf{x_1}$ $\qquad$ $\mathbf{x_1} \# (\mathbf{x_1} \lozenge \mathbf{x_2}) = \mathbf{x_1}$

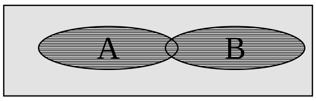
  - ⇨ DeMorgan-Gesetze
    $\overline{\mathbf{x_1} \lozenge \mathbf{x_2}} = \overline{\mathbf{x_1}} \# \overline{\mathbf{x_2}}$ $\qquad$ $\overline{\mathbf{x_1} \# \mathbf{x_2}} = \overline{\mathbf{x_1}} \lozenge \overline{\mathbf{x_2}}$

Mengenalgebra als Modell der Booleschen Algebra

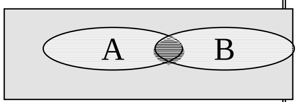
Boolesche Algebra	Mengenalgebra	
V	*P(T)*	**Potenzmenge einer Grundmenge T**
#	$\cup$	**Vereinigung**
◊	$\cap$	**Schnitt**
n	$\varnothing$	**Leere Menge**
e	*T*	**Grundmenge**
$\overline{a}$	$\overline{\mathrm{A}}$	**Komplementmenge von A**



Komplement



A∪B



A∩B

Beispiel Mengenalgebra

- Grundmenge

  T={⌨,🖱,💻}

- Potenzmenge

P(T)= {∅,{⌨},{🖱},{💻},{⌨,🖱},{⌨,💻},{🖱,💻},{⌨,🖱,💻}}

- Für alle A, B, C ∈ T gilt:
  - ⇨ Abgeschlossenheit
    A∪B ∈ P(T) **A∩B ∈ P(T)**
  - ⇨ Kommutativgesetze
    A ∩ B = B ∩ A **A ∪ B = B ∪ A**
  - ⇨ Distributivgesetze
    A ∩ (B ∪ C) = (A ∩ B) ∪ (A ∩ C)
    A ∪ (B ∩ C) = (A ∪ B) ∩ (A ∪ C)
  - ⇨ Neutrale Elemente
    A ∩ T = A **A ∪ ∅ = A**
  - ⇨ Inverse Elemente
    A ∩ $\overline{\mathbf{A}}$ = ∅ **A ∪ $\overline{\mathbf{A}}$ = T**

Schaltalgebra

Die wesentlichen Schaltungen im Computer gehorchen den Gesetzen der Booleschen Algebra. Diese ist aufgebaut durch die binäre Wertemenge *B,* auf der (bis zu) drei Verknüpfungen definiert werden und welche die neutralen Elemente *0* und *1* enthält.
Die logischen Operatoren schreiben sich in diesem Kontext meist als

- $\overline{A}$ oder A' für die Negation von A
- $A \bullet B$ oder AB für die Konjunktion von A und B
- $A + B$ für die Disjunktion von A und B

Definition Schaltfunktion: Eine Schaltfunktion wird definiert durch eine Abbildung, die einen n-stelligen binären Input in einen m-stelligen binären Ouput wandelt.

$$f_{Schalt}: \ \{0,1\}^n \longrightarrow \{0,1\}^m$$

Gatter

Jede Schaltfunktion f_{Schalt} : $\{0,1\}^n$ -> $\{0,1\}^m$ kann durch die Kombination von m n-stelligen Booleschen Funktionen realisiert werden.

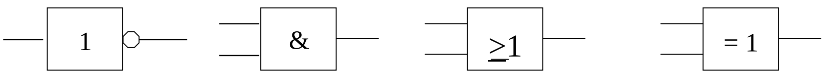
Alle Booleschen Funktionen können auf die Kombination von ein- und zweistelligen BFen zurückgeführt werden, s. Operatorenbasen. Die entsprechenden Schaltelemente heißen Gatter.
Def. Gatter: Ein Gatter ist eine nicht weiter zerlegbare Funktionseinheit, die eine elementare Boolesche Funktionen als physikalisches Bauelement realisiert.

Es gilt: Jedes Schaltnetz läßt sich durch die Kombination einer Grundmenge von Gattern (Bausteinsatz) realisieren.

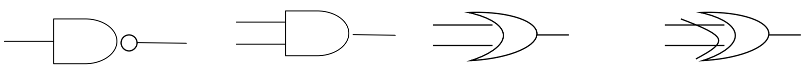
Wichtige Gatter:
- **NOT, AND, OR (Boolescher Satz)**
- **XOR (Exklusiv-Oder)**
- **NAND (Realisierung der negierten Konjunktion, auch für mehrere Variable)**
- **NOR (Realisierung der negierten Disjunktion, auch für mehrere Variable)**

❑ Zwei Möglichkeiten der symbolischen Darstellung der wichtigsten Gatter:

(tegierte Funktionen wie NAtD und NOR durch zusätzlichen Kreis am Ausgang symbolisiert, vgl. NOT.)



Amerikanische Symbole:

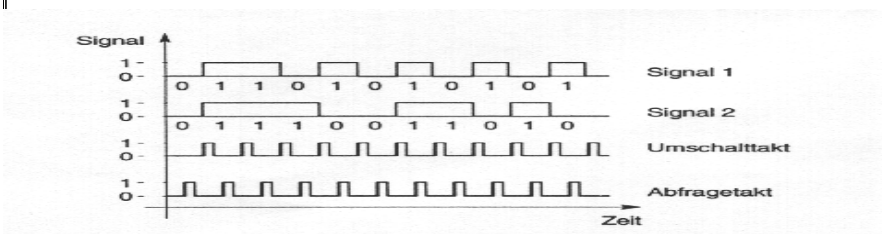


Logische Darstellung: Seien E_1 und E_2 die Inputs, A der Output.

$A = \overline{E}$	$A = E_1 \wedge E_2$	$A = E_1 \vee E_2$	**A=** $E_1 \oplus E_2$
tICHT-Gatter (tegation)	UtD-Gatter (Konjunktion)	ODER-Gatter (Disjunktion)	XOR Gatter

Schaltkreise

Taktgesteuerte binäre Signale: nahezu rechteckige Signalverläufe werden mit Hilfe von Schaltern bzw. Schaltkreisen erzeugt.

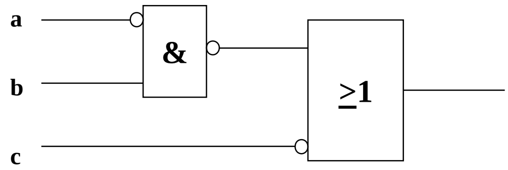


❑ **Schalter:** Element, das nur zwei Zustände annehmen kann, unabhängig vom physikalischen Aufbau (z.B. ein/aus).

❑ **Schaltkreis:** Realisierung eines Schalters mit einer bestimmten Technik (z.B. Transistoren, Dioden)

❑ Es gibt zwei Arten von Schaltungen mit gekoppelten Schaltern, die **Schaltnetze** (ohne Speicherverhalten) und die **Schaltwerke** (mit Speicherverhalten).

Schaltnetz



Repräsentation des Booleschen Ausdrucks

$$\neg(\neg a \wedge b) \vee \neg c$$

als Schaltung.

Schaltwerke

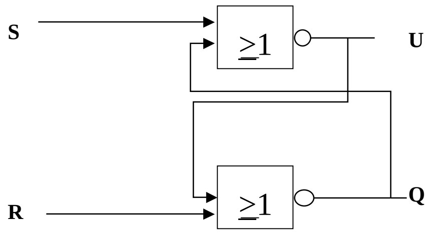
Durch Rückkoppelungen wird ein Speicherverhalten erzielt. Aktueller Output ist von aktuellen und evtl. auch von früheren Inputs abhängig.
Beispiel Flip-Flop: Abbildung nächste Folie !

Wertetabelle für ein ungetaktetes RS-Flipflop: Welches Ausgangssignal $\mathbf{U_{t+1}}$ sich ergibt, hängt vom Signal $\mathbf{U_t}$ ab, das im vorhergehenden Schritt am Ausgang anlag. Denn durch die Rückkopplung hat dieses die Eingangsbelegung mit beeinflußt.

S	R	Q_t	Q_{t+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	unzul.
1	1	1	unzul.

Ungetaktetes RS Flipflop

S steht für set, R für reset. In den erlaubten Zuständen ist immer $Q = \neg U$. Allgemein gilt im Schritt t
$U_{t+1} = \neg(S_t \vee Q_t)$, $Q_{t+1} = \neg(R_t \vee U_t)$



Stabile Zustände falls $Q_{t+1} = Q_t$ und $U_{t+1} = U_t$ gilt, also falls (beachte $Q = \neg U$)

- **R=S=0 und Q beliebig 0 oder 1**
- **S=1, R=0 und Q=1**
- **R=1, S=0 und Q=0**