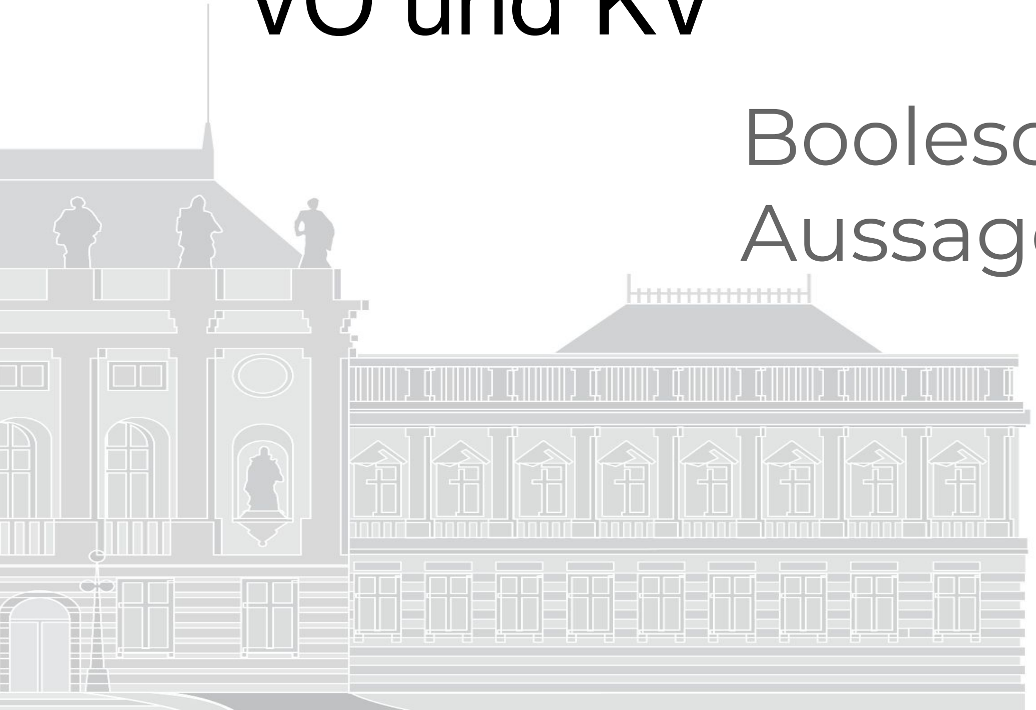
Grundlagen der Informatik
VO und KV

Boolesche Algebra und Aussagenlogik



Überblick

Aussagenlogik

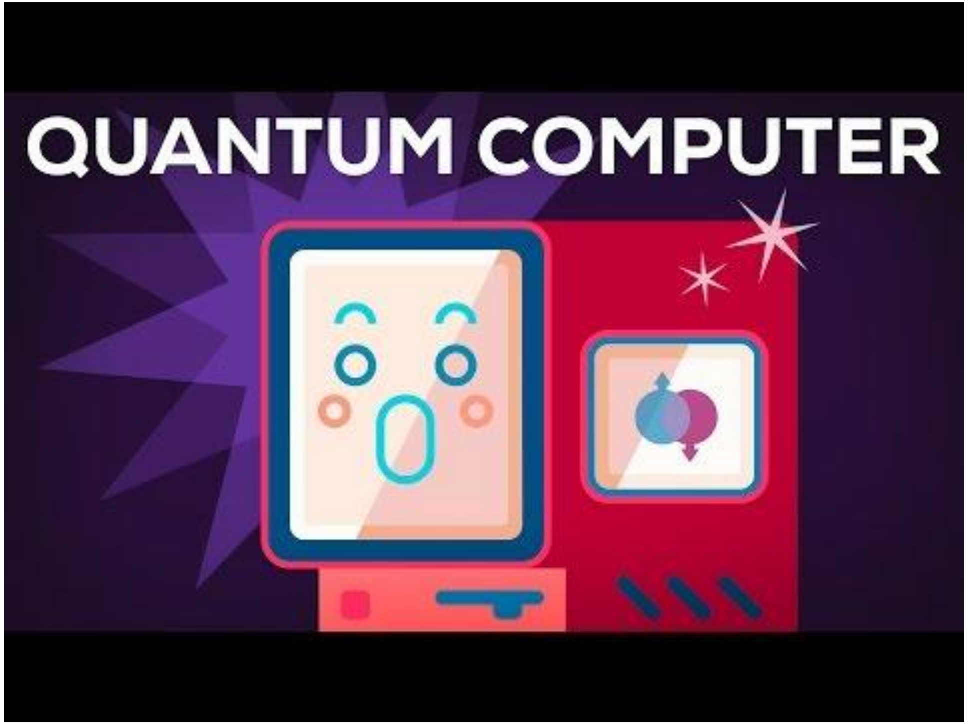
Boolesche Operatoren

Wahrheitstabellen

Logikgatter

Parsetree

Konjunktive Normalform (KNF)



(bis 02:15)

Aussagenlogik

Ist eine einfache Logik und Grundlage für weitere Logiken.
Damit lassen sich Aussagen formulieren, die wahr/falsch (1/0) sind.

Aussagen im Sinne der Logik

"Heute hat es in Graz geregnet."	*//wahr, wenn es geregnet hat*
"Die Erde ist rund"	*//wahr*
1 + 1 = 2.	*//wahr*
"Der Papst ist Präsident der USA."	*//falsch*

Keine Aussagen

"Geh lernen!"	*//Imperativ*
"Magst du Bier?"	*//Frage*
"Rot ist besser als Gelb!"	*//ein Gefühl (subjektiv)*
"Riech dich später!"	*//Floskeln… etc.*

Aussagen lassen sich mit Junktoren verknüpfen

• “Es regnet” **und** “die Straße ist nass”	**AND**	$p \land q$
• “Ich trinke Bier” **oder** “ich trinke Wein”	**OR**	$p \lor q$
• “Ich trinke gerade **nichts**.” ○ “Es ist nicht der Fall, dass ich gerade trinke”	**NOT**	$\neg p$
• **Wenn** “ich die Prüfung schaffe, **dann** “gehe ich ein Bier trinken”	**IMPLIES**	$p \rightarrow q$
• Morgen ist Freitag, **nur wenn** heute Donnerstag ist. (gestern war Donnerstag, weil heute Freitag ist)	**EQUALS**	$p \leftrightarrow q$
• **Entweder** “du isst das Eis”, **oder** “du bekommst nichts”	**XOR**	$p \oplus q$

Aus dem Wahrheitsgehalt von A und B lässt sich der Wahrheitswert der zusammengesetzten Aussage schlussfolgern.

Mit Klammern $(p \rightarrow q) \land q$ können Terme zusammengefasst werden

Implikation

- $A \rightarrow B$
- A impliziert B.
- “Wenn es regnet, dann ist die Straße nass.”

  oder

- “Dass es regnet, ist eine hinreichende Bedingung dafür, dass die Straße nass ist.”

  [Zur Frage, warum das eine hinreichende Bedingung ist – ob auf Grund eines kausalen Zusammenhangs oder auch nur rein zufällig –, nimmt die Implikation nicht Stellung.]

  oder

- “B ist eine notwendige Bedingung für A.”
- Umkehrschluss $\neg B \rightarrow \neg A$
- “Wenn die Straße nicht nass ist, regnet es nicht.”
- “Weil es nicht regnete, kann die Straße nicht nass sein.”

  [Diese Folgerung ist falsch, da die Straße auch aus anderen Gründen nass werden kann (Rohrbruch, Übung der Feuerwehr …).]

Boolesche Algebra

Menge {0,1}

Binäre Operatoren: ∧ (UND) ∨ (inklusives ODER)

Unäre Operatoren: ¬ (NEGIERUNG)

∧	0	1
0	0	0
1	0	1

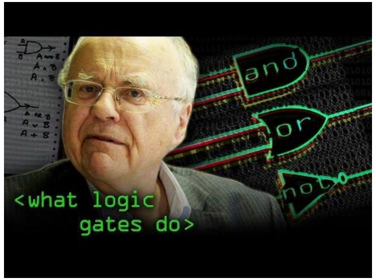
“So wie Multiplikation”
Logisches Produkt

∨	0	1
0	0	1
1	1	1

“So wie Addition”
Logisches Summe

¬	
0	1
1	0

“Konverter”



“Punkt vor Strich”

1.	Negation	NOT	$\neg$
2.	Konjunktion	AND	$\wedge$
3.	Disjunktion	OR	$\vee$
4.	Implikation	IMPLIES	$\rightarrow$

rechts orientiert:

$p \rightarrow q \rightarrow r$ heißt $p \rightarrow (q \rightarrow r)$

Hands-On

Formalisiere folgende Aussagen:

(1) Diese Hütte steht nicht an einem See oder der Schatz ist nicht in der Küche.

(2) Wenn der Baum vor der Hütte eine Ulme ist, dann ist der Schatz in der Küche.

(3) Der Baum hinter der Hütte ist eine Eiche oder der Baum vor der Hütte ist keine Ulme oder der Baum hinter der Hütte ist eine Ulme.

**(4) Die Aussage ”Der Baum vor der Hütte ist keine Ulme
und der Schatz ist nicht unter dem Fahnenmast begraben“ ist nicht wahr.**

(5) Wenn der Baum hinter der Hütte eine Eiche ist, dann ist der Schatz im Keller.

Hands-On

Formalisiere folgende Aussagen:

(1) Diese Hütte steht nicht an einem See oder der Schatz ist nicht in der Küche.

(2) Wenn der Baum vor der Hütte eine Ulme ist, dann ist der Schatz in der Küche.

(3) Der Baum hinter der Hütte ist eine Eiche oder der Baum vor der Hütte ist keine Ulme oder der Baum hinter der Hütte ist eine Ulme.

(4) Die Aussage ”Der Baum vor der Hütte ist keine Ulme und der Schatz ist nicht unter dem Fahnenmast begraben“ ist nicht wahr.

(5) Wenn der Baum hinter der Hütte eine Eiche ist, dann ist der Schatz im Keller.

Formalisierung der Schatzsuche

(1)	Diese Hütte steht nicht an einem See oder der Schatz ist nicht in der Küche.	$\neg See \lor \neg Kü$
(2)	Wenn der Baum vor der Hütte eine Ulme ist, dann ist der Schatz in der Küche.	$BvHU \Rightarrow Kü$
(3)	Der Baum hinter der Hütte ist eine Eiche oder der Baum vor der Hütte ist keine Ulme oder der Baum hinter der Hütte ist eine Ulme.	$BhHE \lor \neg BvHU \lor BhHU$
(4)	Die Aussage "Der Baum vor der Hütte ist keine Ulme und der Schatz ist nicht unter dem Fahnenmast begraben" ist nicht wahr.	$\neg(\neg BvHU \land \neg Fa)$
(5)	Wenn der Baum hinter der Hütte eine Eiche ist, dann ist der Schatz im Keller.	$BhHE \Rightarrow Ke$

Der Baum **hinter** der Hütte ist keine Ulme: $\neg BhHU$

Wie zeichne ich eine Wahrheitstabelle...

AND

p	q	p $\land$ q

Diesen Term wollen wir untersuchen

Wahrheitstabellen

AND

p	q	p ∧ q
T	T	T
T	F	F
F	T	F
F	F	F

OR

p	q	p ∨ q
T	T	T
T	F	T
F	T	T
F	F	F

XOR

p	q	p ⊕ q
T	T	F
T	F	T
F	T	T
F	F	F

NOT

p	¬p
T	F
F	T

Konjunktion

Disjunktion

“Milch oder Zucker” - Oder

Exklusives Oder

Entweder, oder

IMPLIES

p	q	$p \rightarrow q$	$\neg p \vee q$
T	T	T	
T	F	F	
F	T	T	
F	F	T	

Implikation

EQUAL

p	q	$p \leftrightarrow q$	
T	T	T	
T	F	F	
F	T	F	
F	F	T	

Äquivalenz

IMPLIES

p	q	$p \rightarrow q$	$\neg p \vee q$
T	T	T	
T	F	F	
F	T	T	
F	F	T	

Hands-On: Vervollständigen der Tabelle

EQUAL

p	q	$p \leftrightarrow q$	$(p \rightarrow q) \wedge (q \rightarrow p)$
T	T	T	
T	F	F	
F	T	F	
F	F	T	

IMPLIES

p	q	p → q	¬p v q	¬p
T	T	T	T	F
T	F	F	F	F
F	T	T	T	T
F	F	T	T	T

$p \to q \equiv \neg p \vee q$

EQUAL

p	q	p ↔ q	(p → q) ∧ (q → p)
T	T	T	T
T	F	F	F
F	T	F	F
			T

p → q	q → p
T	T
F	T
T	F
T	T

Implikation

p	q	$p \rightarrow q$
T	T	T
T	F	F
F	T	T
F	F	T

“Wenn … gilt, dann gilt …“

“Wenn schlechtes Wetter ist, dann habe ich meinen Schirm dabei”

Wann ist diese Aussage falsch?
Wann lüge ich bei dieser Aussage?

Der einzige Fall, wann diese Aussage wirklich falsch ist, ist, wenn ich sage, das Wetter ist schlecht und ich habe den Schirm nicht dabei.

Im Alltag verbinden wir mit “wenn dann”, ein “genau, dann wenn”.

“Genau dann, wenn” ist die Äquivalenz

Implikation

- Die Aussage „Wenn Feuer ausbricht, dann gibt es dort Sauerstoff“ soll durch die Implikation ausgedrückt werden.
- Aussage A: Feuer bricht aus
- Aussage B: dort gibt es Sauerstoff
- Die Aussage A ⇒ B ist richtig,
    - **wenn ein Feuer ausbricht und Sauerstoff vorhanden ist** (das Vorhandensein von Sauerstoff ist für das Ausbrechen eines Feuers notwendig, also:
- (B ist notwendig für A),
- Hingegen ist das Ausbrechen eines Feuers hinreichend für die Existenz von Sauerstoff.
- (A ist hinreichend für B).
    - **wenn kein Feuer ausbricht aber Sauerstoff vorhanden ist**,
    - **wenn kein Feuer ausbricht und kein Sauerstoff vorhanden ist**.
- Die Aussage ist nur falsch,
- - **wenn ein Feuer ausbricht und es gibt dort keinen Sauerstoff.**

https://www.matheretter.de/wiki/implikation

Aussagenlogische Formeln, die in jeder Interpretation wahr sind, nennt man **Tautologien = T**

p	**¬p**	**p ∨ ¬p**
T	F	T
F	T	T

Aussagenlogische Formeln, die in mindestens einer Interpretation wahr sind, nennt man **erfüllbar**

p	q	p ∧ q
T	T	T
T	F	F
F	T	F
F	F	F

Aussagenlogische Formeln, die in keiner Interpretation wahr sind, nennt man **Kontradiktionen = ⊥**

p	**¬p**	**p ∧ ¬p**
T	F	F
F	T	F

Aussagenlogische Formeln, in denen alle Interpretation wahr sind, nennt man **gültig**

p	**¬p**	**p ∨ ¬p**
T	F	T
F	T	T

F	Synonyme	Bedingung	$\neg F$
allgemeingültig	tautologisch (in der Aussagenlogik)	Es gibt keine Interpretation, welche die Formel nicht erfüllt.	unerfüllbar
erfüllbar	konsistent, widerspruchsfrei	Es gibt eine Interpretation, welche die Formel erfüllt.	falsifizierbar
falsifizierbar	widerlegbar	Es gibt eine Interpretation, welche die Formel widerlegt.	erfüllbar
unerfüllbar	inkonsistent, widersprüchlich	Es gibt keine Interpretation, welche die Formel erfüllen kann/könnte.	allgemeingültig

https://de.wikipedia.org/wiki/Allgemeing%C3%BCltigkeit

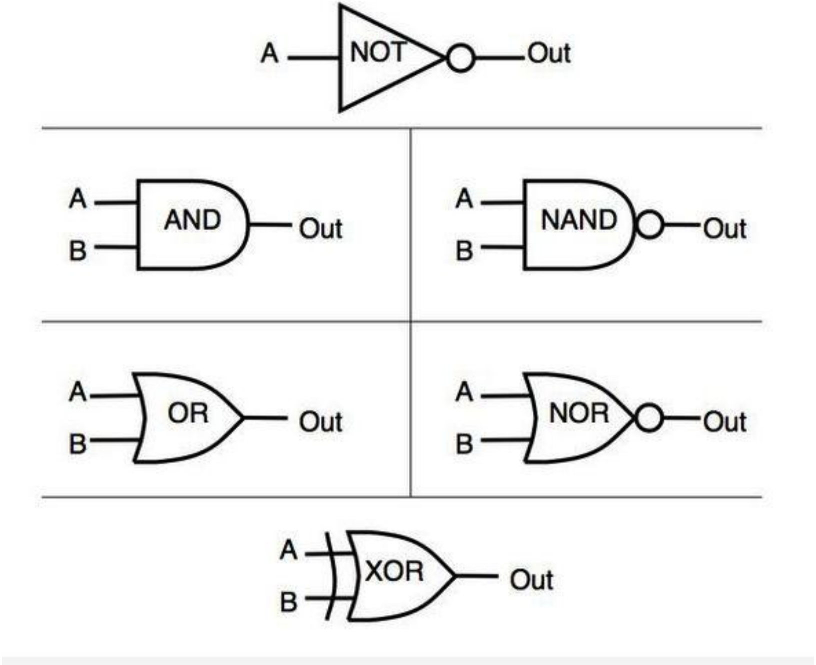
Mit XOR (exklusives ODER) wird binäre Addition umgesetzt

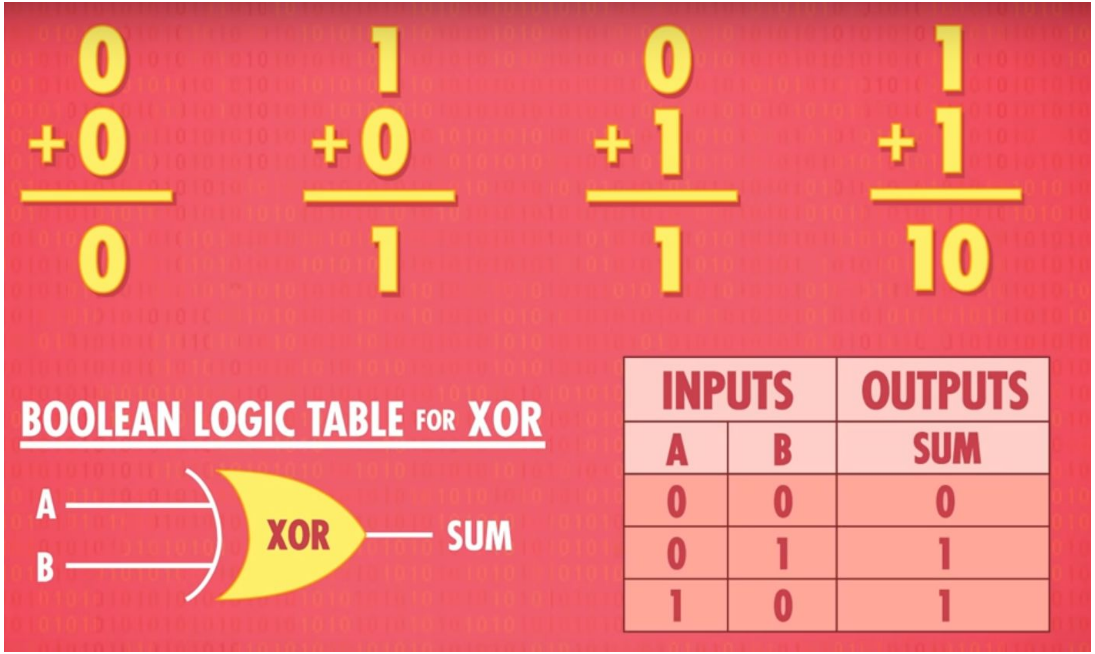
⊕	0	1
0	0	1
1	1	0

```
   10 = 2
⊕  11 = 3
  101 = 5
```

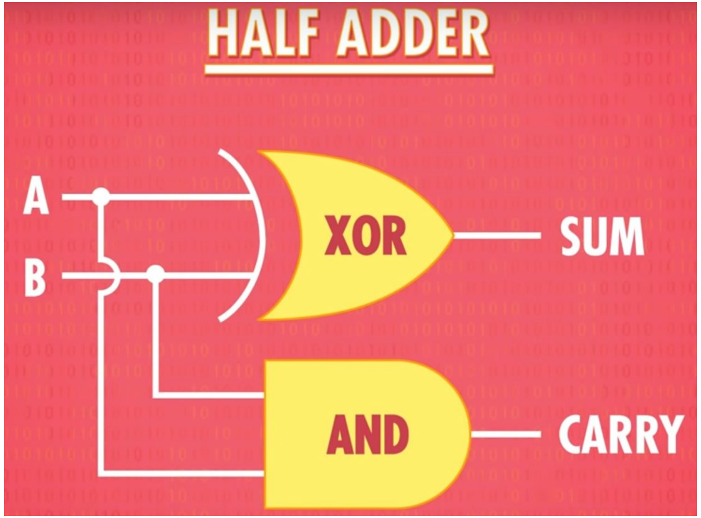
Logikgatter / Logic Gates

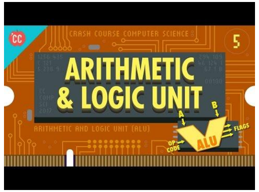
Logikgatter sind Anordnungen (elektronische Schaltung) zur Realisierung einer booleschen Funktion, die binäre Eingangssignale durch Implementierung logischer Operatoren (AND, OR, XOR, NOT) zu einem binären Ausgangssignal verarbeitet (einem logischen Ergebnis umgewandelt).





Übertrag

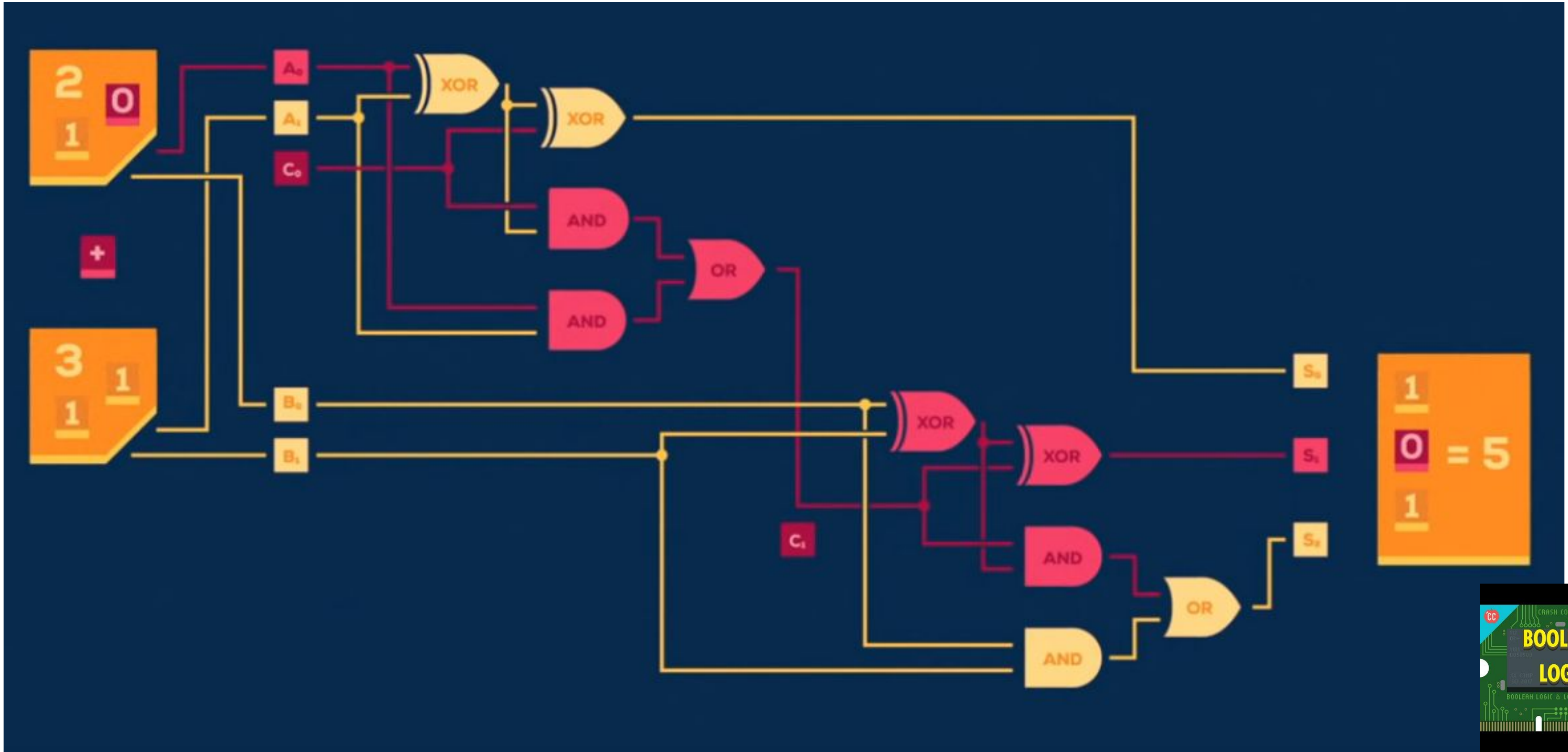


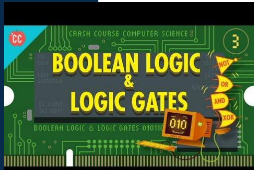


https://www.youtube.com/watch?v=1I5ZMmrOfnA

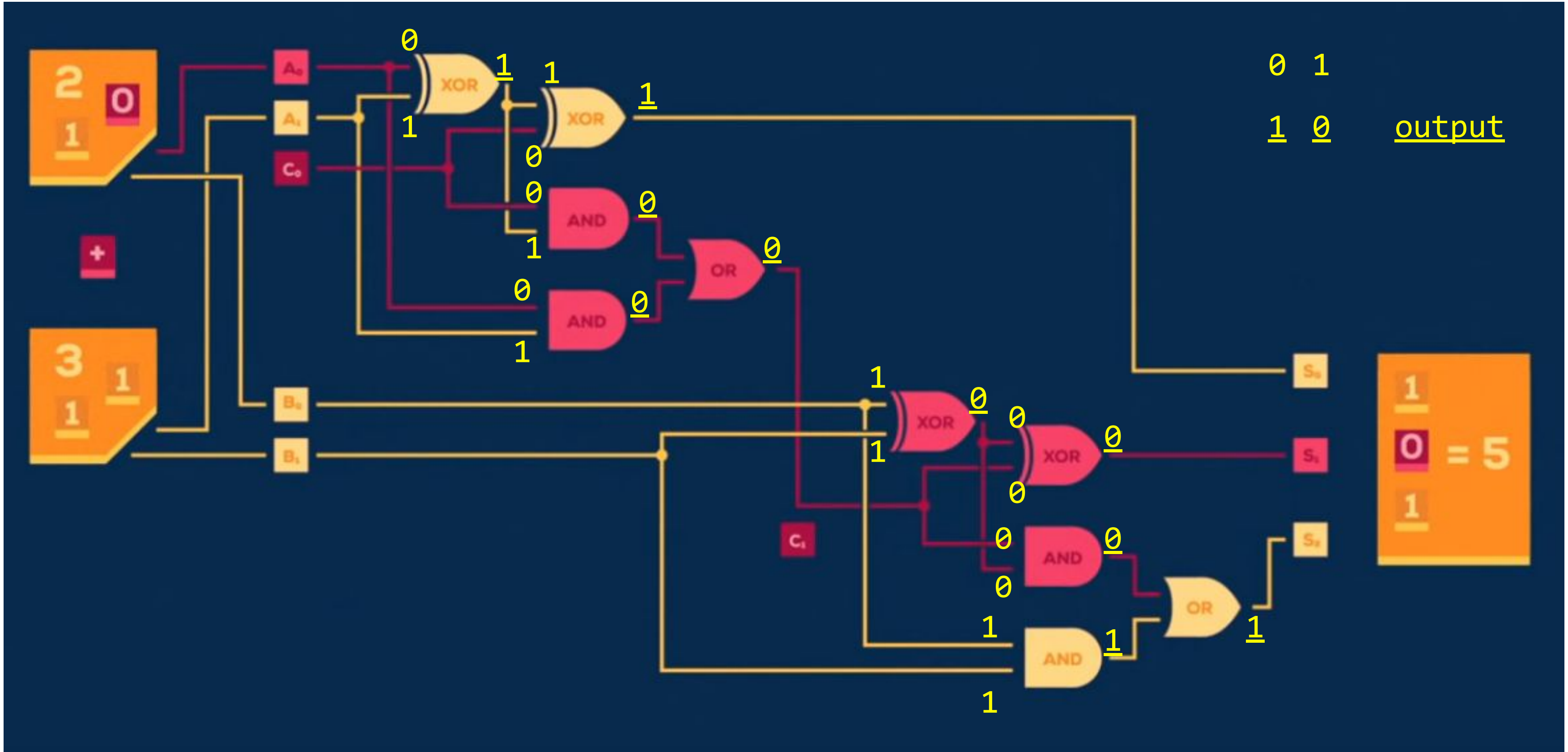
Die Verschachtelung von Logikgattern, um die Addition von 2 und 3 umzusetzen

```
  10
 +11
------
 101
```





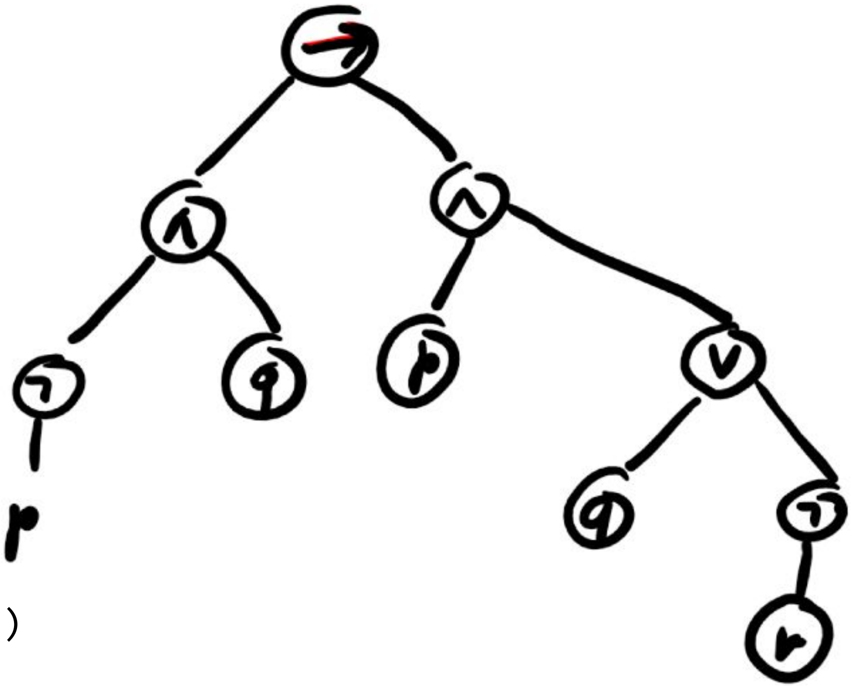
```
 10
+11
------
101
```



Parse Tree / binärer Entscheidungsbaum mit Modell

$(\neg p \land q) \Rightarrow (p \land (q \lor \neg r))$

$M\colon p = T,\ q = F,\ r = T$



Wahrheitstabelle!

WolframAlpha:
https://www.wolframalpha.com mit folgendem Input:

```
(not(p) and q) implies (p and (q or not(r)))
```

https://www.wolframalpha.com/input/?i=%28not%28p%29++AND+q%29+IMPLIES+%28+p+AND+%28q+OR+not%28r%29%29+%29:

```
(not(p)  AND q) IMPLIES ( p AND (q OR not(r)) )
```

Truth table:

p	q	r	$\neg p \wedge q \Rightarrow p \wedge (q \vee \neg r)$
T	T	T	T
T	T	F	T
T	F	T	T
T	F	F	T
F	T	T	F
F	T	F	F
F	F	T	T
F	F	F	T

M: p = **T**, q = **F**, r = **T**

Ein anderes Modell, bei dem es F wird:

M: p = **F**, q = **T**, r = **T**

Konjunktive Normalform (KNF)

Unter der konjunktiven Normalform versteht man einen aussagenlogischen Ausdruck, der aus einer beliebigen Anzahl von UND-verknüpften Klauseln gebildet wird.

Jede beliebige boolesche Funktion lässt sich als KNF darstellen. Es lässt sich auch ein allgemeiner Algorithmus angeben aus dem die KNF eines beliebigen Ausdrucks gebildet werden kann:

1. Erstelle die Wahrheitstabelle.

2. Wähle alle Konfigurationen unter denen die Formel zu falsch evaluiert.

3. Negiere alle Variablen.

4. Verbinde alle Variablen einer Konfiguration mit ODERs.

5. Verbinde alle Konfigurationen mit mehreren UNDs

Videotutorial: https://www.youtube.com/watch?v=WRYAmf864tQ

Beispiel KNF

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

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

$\wedge$

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

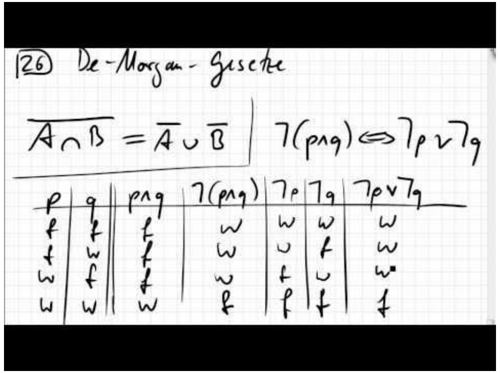
$\wedge$

$(a \vee b \vee c)$

Umformen “Rechenregeln”

Teilformel dürfen durch äquivalente Formel ersetzt werden:

A	$\equiv$	$\neg\neg A.$	Doppelte Negation
$A \Rightarrow B$	$\equiv$	$\neg A \lor B.$	Implikation
$A \Leftrightarrow B$	$\equiv$	$(A \Rightarrow B) \land (B \Rightarrow A).$	Äquivalenz
$A \lor B$	$\equiv$	$B \lor A.$	Kommutativität
$A \land B$	$\equiv$	$B \land A.$	
$A \Rightarrow B$	$\equiv$	$(\neg B) \Rightarrow (\neg A).$	Kontraposition
$\neg(A \lor B)$	$\equiv$	$(\neg A) \land (\neg B).$	De Morgan
$\neg(A \land B)$	$\equiv$	$(\neg A) \lor (\neg B).$	



https://de.wikipedia.org/wiki/Boolesche_Algebra

Boolesche Algebra in Python

https://www.onlinegdb.com/online_python_compiler

Ein *boolescher Ausdruck* ist ein Ausdruck, der entweder *wahr* oder *falsch* ist, in Python True und False genannt. Sind folgende aussagen in Python True oder False:

```
>>> "Hallo Welt" > "Hallo"
```
//true

```
>>> 4 != 5
```
//true

```
>>> 4 == 4 or 4 == 5
```
//true

```
>>> "Hallo" == "Hallo Welt"
```
//false

```
>>> not not 5 == 5
```
//true

```
>>> not 3 <= 4 and not 4 >= 5
```
//not TRUE and not FALSE
//FALSE and TRUE
//FALSE

https://www.onlinegdb.com/online_python_compiler

```
print(4 > 5)

print(4 != 5)

print(4 == 4 or 4 == 5)

print("Hallo" == "Hallo Welt")

print(not not 5 == 5)

print(not 3 >= 4 and not 4 >= 5)
```

Übungen

Wahrheitstabellen - Beispiele

1. $(\neg p \vee q) \wedge (p \rightarrow q)$
2. $(p \rightarrow q) \wedge (q \rightarrow p)$

Wahrheitstabellen - Beispiele

1.)

p	q	$\neg p$	$\neg p \lor q$	$p \rightarrow q$	$(\neg p \lor q) \land (p \rightarrow q)$
T	T	F	T	T	T
T	F	F	F	F	F
F	T	T	T	T	T
F	F	T	T	T	T

2.)

p	q	$p \rightarrow q$	$q \rightarrow p$	$(p \rightarrow q) \land (q \rightarrow p)$
T	T	T	T	T
T	F	F	T	F
F	T	T	F	F
F	F	T	T	T

Hands-On (p IMPLIES q) IMPLIES p

p	q	$(p \to q) \to p$		
T	T			
T	F			
F	T			
F	F			

Hands-On

p	q	$p \to q$	$(p \to q) \to p$
T	T	T	T
T	F	F	T
F	T	T	F
F	F	T	F

Hands-On

p	q	$(p \lor q) \leftrightarrow (\neg p \rightarrow q)$		
T	T			
T	F			
F	T			
F	F			

Hands-On (p OR q) EQUAIVALENT (not(p) IMPLIES q)

p	q	$\neg p$	$\neg p \to q$	$p \vee q$	$(p \vee q) \leftrightarrow (\neg p \to q)$
T	T	F	T	T	T
T	F	F	T	T	T
F	T	T	T	T	T
F	F	T	T	F	T

Übung

Verwende Junktoren um **folgende Sätze** so detailliert wie möglich in Aussagenlogik darzustellen:

(a) If you are a good student, then you like Logic and Computability.

(b) In the evening, Paul will study or go out with his friends, but not both

(c) If both Christopher and Sepp pass the exam, they have a party

Übung (Ass2)

Zeige mit einer **Wahrheitstabelle, ob** folgende Formel erfüllbar ist.
Eine Formel ist erfüllbar, wenn mindestens eine Belegung der Variablen eine wahre Aussage erzeugt. Erzeuge eine KNF.

$$(p \rightarrow q) \land (\neg q \lor p)$$

Übung

Zeichne einen **Parse Tree** für folgende Formel und verwende die Zuteilungen (das Modell): p = F, q = T, r = F

Ist diese Formel erfüllbar?

$$((q \to \neg p) \lor r) \to (q \land (r \to p))$$

Übung

$(p \land q) \rightarrow \neg((q \rightarrow \bot) \land (p \rightarrow \top))$

- Prüfe mit einer Wahrheitstabelle, ob diese Aussage gültig oder erfüllbar ist.
- Prüfe mit einem Parse Tree ob die Zuordnung
  $p = T$, $q = T$ wahr oder falsch ergibt. (Achte auf $\bot$ und $\top$)

Assignment 2

1.) Formalisiere jede der folgenden Aussagen durch eine aussagenlogische Formel:

1.1) *Wenn du die drei Fragen beantwortet hast, dann kannst du die Brücke des Todes überqueren.*
1.2) *Die Ritter der Kokosnuss gibt es nur zusammen mit Pferden und Kokosnüssen.*
1.3) *Entweder wird ein Gebüsch gekauft oder es wird kein Gebüsch gekauft.*

2.) Formuliere die Regel, "wenn der Hahn auf dem Misthaufen kräht, dann ändert sich das Wetter oder es bleibt so, wie es ist" als Formel der Aussagenlogik. Zeige, dass die Formel eine Tautologie ist (Wahrheitstabelle)

3.) Begründe ob folgende Formeln erfüllbar, gültig oder unerfüllbar sind. Erfüllbar bist ist eine Formel, wenn mindesten einmal "wahr" in der Wahrheitstabelle auftaucht. Gültig wenn alle "wahr" sind und unerfüllbar wenn alle "falsch" sind.

3.1) $a \rightarrow a \land b$
3.2) $(a \rightarrow b) \rightarrow (b \rightarrow a)$
3.3) $(a \land b) \rightarrow (\neg a \lor \neg b)$
3.4) $(\bot \rightarrow p) \land \neg(q \longleftrightarrow p) \land \neg((\neg p \land \neg q) \rightarrow q)$

4.) Zeichne einen Parse Tree für folgende Formel und verwende den Tree, um zu überprüfen ob die Zuordnung p = F, q =T, r = F die Formel true oder false werden lässt. Finde ein Modell, welches das Gegenteil erreicht.
$(\neg(r \longleftrightarrow q) \rightarrow \neg r) \land (\neg(r \rightarrow q) \lor (p \rightarrow q))$

5.) Gib folgende Befehle in deiner Konsole ein. Erkläre was jeweils passiert und um welche unterschiedlichen Operatoren es sich handelt.
(https://www.linux.com/tutorials/logical-ampersand-bash/)

- *echo $((2 & 3))*
- *echo $((2 && 3))*
- *echo $((5 | 5))*
- *echo $((5 || 5))*

Bonus:
Was ist die Konjunktive Normalform (KNF) des folgenden logischen Ausdrucks:
(p -> q) ∧ (¬q v p)