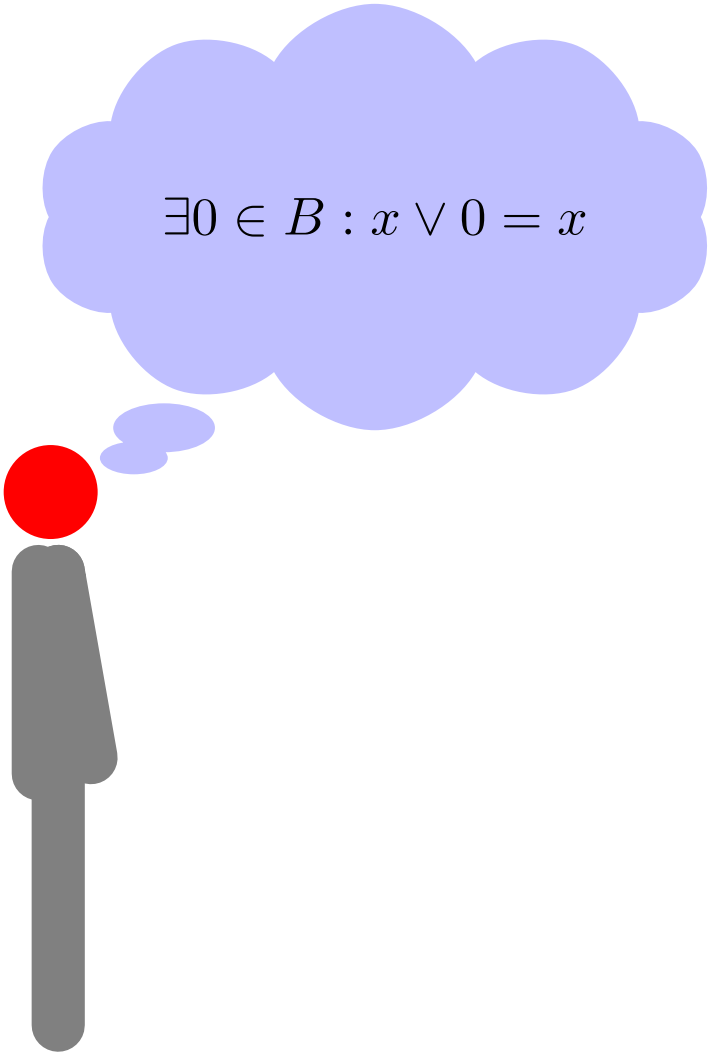
Aufgabensammlung - Teil 1



1 Vorwort

Die Aufgabensammlung dient der Vertiefung ihrer Fach- und Methodenkompetenz der Grundlagen der Technischen Informatik. Die Aufgabensammlung umfasst 3 Teilgebiete der Grundlagen digitaler Systeme:

1. Boolesche Algebra und Realisierung logischer Grundfunktionen
2. Minimierung Boolescher Funktionen und einfache Codierung
3. Zahlendarstellung, Zahlensysteme und einfache Arithmetik

Die Aufgabensammlung ist selbstständig zu bearbeiten. Bilden Sie Lerngruppen. tauschen Sie sich mit ihren Kommilitonen aus.

Die Aufgabensammlung ist derart strukturiert, dass jedes Teilgebiet eine kurzen theoretische Einführung vorsieht. Jedoch sind die theoretischen Grundlagen - neben den Unterlagen aus der Vorlesung - in der spezifischen Fachliteratur zu studieren. Achten Sie auf die Literaturhinweise!

Im Gegensatz zu den Laborübungen, ist die Bearbeitung der Aufgabensammlung nicht verpflichtend und wird von dem Dozenten nicht ausgewertet. Zur Vor- und Nachbereitung der Inhalte des Moduls sollten Sie jedoch diese in ihrer Zeit- und Aufwandsplanung vorsehen. Hier bleibt mir nur ein Zitat:

> „Das Wort "Lernen„ geht auf die gotische Bezeichnung für „ich weiß“ (lais) und das indogermanische Wort für "gehen„ (lis) zurück (Wasserzieher, 1974). Die Herkunft des Wortes deutet bereits darauf hin, dass Lernen ein Prozess ist, bei dem man einen Weg zurücklegt und dabei zu Wissen gelangt. Mielke (2001, 11)“ (Quelle: `http://www.lern-psychologie.de/common/lernen.htm`)

2 Einführung in die Digitaltechnik

Mit der rasanten Entwicklung der Digitalelektronik wurde es möglich, die Konzepte der diskreten Signalverarbeitung auf kleinster Fläche zu realisieren. Die Einsatzgebiete der Digitalelektronik sind vielfältig und reichen von regelungstechnischen Anwendungen, der Protokollverarbeitung bis hin zu komplexen Signalprozessoren.

Eine fundamentale Aussage zur Evolution der Halbleitertechnik traf Gordon Moore. Das „Mooresche Gesetz“ (engl. *Moore's Law*) sagt aus, dass sich die Komplexität Integrierter Schaltkreise (***I**ntegrated **C**ircuits*, IC) mit minimalen Komponentenkosten etwa alle zwei Jahre verdoppelt. Unter Komplexität verstand Moore, der das Gesetz 1965 formulierte, die Anzahl der Schaltkreiskomponenten auf einem Computerchip. In einer Rede aus dem Jahr 1975 korrigierte er seine ursprüngliche Angabe auf eine Verdoppelung alle zwei Jahre. Nach wie vor hat das „mooresche Gesetz“ Gültigkeit, jedoch wird eine Sättigung der Steigerungsrate vorausgesagt. Der Gesetzmäßigkeit in der steigenden Dichte von Transistoren pro Flächeneinheit wurde eine Begrenzung der Gültigkeit innerhalb der nächsten 10 bis 15 Jahre prognostiziert. Dann, so die Annahmen, ist eine fundamentale Grenze der Integrationsdichte erreicht. In vielen Literaturstellen wird das „mooresche Gesetz“ zitiert und als Argumentationshilfe für die technische Entwicklung der letzten Dekaden herangezogen. Im streng wissenschaftlichen Sinne handelt es sich jedoch nicht um eine physikalische Gesetzmäßigkeit, sondern vielmehr um eine fundierte Prognose. Zu Bemerken gilt, dass jede Prognose eine gewisse Unschärfe besitzt. Bei welchem Technologieknoten letztendlich eine Grenze erreicht sein wird, bestimmt die Entwicklung der nächsten Jahre. Denn neben den technischen und physikalischen Grenzen sind auch die immensen Kosten in der Entwicklung und Produktion moderner Sub-Micron Halbleitertechnologien als Faktor in der weiteren technischen Evolution zu berücksichtigen. Charakteristiken mikroelektronischer Schaltungen sind die minimale Strukturabmessungen, die Anzahl von Transistoren/Chip, die Chipfläche, Schaltzeiten und Grenzfrequenzen sowie der Leistungsverbrauch. Für interessierte Leser sei der Link `https://www.halbleiter.org/` empfohlen.

2.1 Digitaltechnik

Die Vorteile der digitalen Signalverarbeitung liegen in der meist einfacheren Realisierung im Vergleich zur analogen Signalverarbeitung. Mikrokontroller haben zumeist ein schon integriertes analoges Front-End (AFE) zur Digitalsierung von analogen Signalen. Diese Signale können von einem Sensorelement, einem Mikrofon oder sonstigen Signalquellen stammen. Der Mikrokontroller kann mit der Hochsprache C programmiert und das Signal weiter verarbeitet werden. Mikrokontroller werden mit unterschiedlichsten Bitbreiten (8-Bit, 32-Bit und 64-Bit Datenbus) angeboten. Zusätzliche Funktionseinheiten wie USB-Schnittstelle, JTAG-Interface, Speicher und Analog/Digital- (ADC) und Digital/Analogwandler (DAC) ermöglichen ein breites Anwendungsspektrum. Für die Programmierung der Mikrokontroller wird überwiegend die Hochsprache C eingesetzt. Eigene Entwicklungsumgebungen der Hersteller unterstützen den Entwickler.

Eine weitere Komponente für den digitalen Schaltkreisentwurf sind FPGAs (*Field Programmable Gate Arrays)*. Ähnlich den Mikrokontrollern können digitale Funktionen mittels hardware-nahen Hochsprachen wie VHDL und Verilog-HDL in den Baustein integriert werden. Zusätzliche Funktionen und Datenschnittstellen ermöglichen auch hier ein weites Anwendungsgebiet.

Vorteile der Digitaltechnik:

- Digitale Signale lassen sich einfacher und weniger störanfällig übertragen. Sie unterliegen keiner Fehlerfortpflanzung.
- Digitale Signale lassen sich leicht kodieren (dekodieren) und sind somit gut zur Datenübertragung auch für weite Strecken geeignet.
- Digitale Signale lassen sich leicht konstruieren. Sie lassen sich einfach in Rechnern, Mikroprozessoren sowie Gatterbausteinen verarbeiten und speichern.
- Miniaturisierung elektronischer Bauelemente und höhere Leistungsfähigkeit der Bauelemente (Speicherkapazität, Rechnerleistung und Transistorenzahl) führt zu zunehmender Digitalisierung.

Nachteile der Digitaltechnik:

- Digitale Systeme sind langsamer als analoge Systeme. In der Hochfrequenztechnik dominiert die Analogtechnik.
- Die Anzahl der benötigten Schaltungsbestandteile ist um ein Vielfaches höher als bei analogen Systemen (wird durch eine hohe Integrationsdichte auf entsprechende Chips kompensiert).
- Informationsverlust bei der Umwandlung analoger in digitaler Informationen. Mathematisch kann dieses Phänomen als Rundungsfehler bezeichnet werden, welcher aufgrund der stets begrenzten Anzahl von Stellen immer auftritt.
- Analoge Anzeigen sind anschaulicher und schneller zu erfassen, weil Proportionen dargestellt werden. (Vergleich: Wertetabelle (digital) und Balkendiagramm (analog)).

2.2 Vergleich zwischen Aussagenlogik und Mengenalgebra

Beim Aufbau der Mengenlehre ist offenbart worden, dass in der Mengenalgebra Gesetze gelten, zu denen es in der Ausagenlogik entsprechende Gesetze gibt. Zum Beispiel ist das Kommutativitätsgesetz bei der Bildung der Schnittmenge ($A \cap B = B \cap A$) auf das Kommutativgesetz für den Junktor $\wedge$ in der Aussagenlogik abgestützt worden $X \wedge Y \equiv Y \wedge X$. Offenbar gibt es formale Gemeinsamkeiten und ebenso Unterschiede zwischen der Aussagenlogik und der Mengenalgebra. In der folgenden Tabelle werden jeweils die entsprechenden Objekte aus den beiden Theorien gegenübergestellt. Das bietet eine Grundlage, bezüglich der die beiden Seiten verglichen werden können.

Aussagenlogik	**Mengenlehre**
Sprachelemente und syntaktisches Vokabular	Elemente
Aussagen	Mengen
Aussagenvariablen	Mengenvariablen
Zuordnung Variable $\mapsto$ Wahrheitswert	Zuordnung Menge $\mapsto$ Mächtigkeit
Junktoren $\neg, \wedge, \vee, \ldots$	Operationen $\cup, \cap, \ldots$
Gesetze: Kommutativitat etc.	Duale Gesetze: Kommutativität etc.

Offenbar gibt es zwischen den Begriffen links und denen rechts eine Entsprechung. Diejenigen Leser, die schon etwas Schaltalgebra kennen, wissen, dass man genauso diese der Aussagenlogik oder der Mengenalgebra gegenüberstellen kann. In diesem Skript beschäftigen wir uns mit der Aussagenlogik und im Speziellen mit der einstelligen und zweistelligen Boolschen Algebra.

3 Theoretische Grundlagen

3.1 Boolesche Algebra

Die sogenannte Boolesche Algebra geht zurück auf den irischen Mathematiker George Boole (1815-1864), der sie 1847 zuerst formulierte. E.Huntington war es, der 1904 ein Axiomensystem für die Boolesche Algebra entwickelte. Erst etwa ein Jahrhundert später, im Jahr 1937, wurde ihre Anwendbarkeit als sogenannte Schaltalgebra in den Natur- und Ingenieurwissenschaften, zuerst durch A.Nakshima bei der Optimierung von Telefonvermittlungen und ein Jahr später durch C.E.Shannon bei der Optimierung von Relaisnetzwerken erkannt.

George Boole (1815-1864)

- Anwendung von algebraischen Methoden auf die Aussagenlogik
- '0' und '1' als Logikwerte für „falsch“ und „wahr“
- Variablen, die den Wahrheitswert von Aussagen modellieren

Ernst Schröder (1841-1902), **Edward V. Huntington** (1874-1952)

- Axiomatisierung der Booleschen Algebra
- Anwendung auf den Entwurf von Telefonvermittlungsanlagen

Claude Elwood Shannon (1916-2001)

- Anwendung der Booleschen Algebra auf die Analyse und den Entwurf von digitalen (elektromechanischen) Schaltungen

3.2 Quantoren

In mathematischer Sprechweise werden die Quantoren folgendermaßen definiert:

- Allquantor oder Generalisator: $\forall\ x\ \in\ \Omega : A(x)$
  „Für alle x aus der Menge/Klasse Ω ist die Aussage A(x) wahr.“
- Existenzquantor oder Partikularisator: $\exists\ x\ \in\ \Omega : A(x)$
  „Es gibt (mindestens) ein x in Ω, so dass A(x) wahr ist.“
- Existenz und Eindeutigkeit: $\exists_1\ x\ \in\ \Omega : A(x)$
  „Es gibt genau ein x in Ω , so dass A(x) wahr ist.“

Man beachte: „$\forall\ x\ \in\ \Omega : A(x)$“ oder „$\exists\ x\ \in\ \Omega : A(x)$“ sind Aussagen. Die Variablen x und y werden durch die Quantoren gebunden. Insbesondere sind die Aussagen $\forall\ x\ \in\ \Omega : A(x)$ und $\forall\ z\ \in\ \Omega : A(z)$ gleichbedeutend!

Verneinungsregel: $\neg\,[\forall\ x\ \in \Omega : A(x)]\ \Leftrightarrow\ \exists\ x\ \in\ \Omega : [\neg A(x)]$

Erweitert man den Aussagenkalkül durch die Sprachelemente „Quantoren“, „Subjekte“ und „Prädikatsvariable“ (Eigenschaften), so erhält man den Prädikatenkalkül der ersten Stufe. Auch hier sind genau die korrekten Formeln aus dem Axiomensystem ableitbar. Das ist der Inhalt des Vollständigkeitssatzes von Kurt Gödel (1930).

Eine weitere Erweiterung der Sprachelemente, bei der auch Prädikate von Prädikaten und Quantoren über Prädikatsvariblen zugelassen sind, führt zu einem Prädikatenkalkül höherer Stufe. Dieser ist allerdings stets unvollständig (Unvollständigkeitssatz von Gödel aus dem Jahr 1931). Mehr noch, es gibt keinen (endlichen) Algorithmus, mit dem man entscheiden kann, ob eine vorgegebene Formel allgemeingültig ist (Unentscheidbarkeitssatz von Alan Turing und Alonzo Church aus dem Jahr 1936).

Beispiel

- $\forall\, x\ \in \mathbb{R}:\ x = 3$ ist falsch
- $\exists\, x\ \in \mathbb{R}:\ x = 3$ ist wahr
- $\exists_1\, x\ \in \mathbb{R}:\ x = 3$ ist falsch
- $\forall\, z\ \in \mathbb{R}:\ z + 1\ \in\ \mathbb{R}$ ist wahr
- $\forall\, y\ \in \mathbb{R}:\ y \geq |y|\ \in\ \mathbb{R}$ ist falsch

3.3 Aussagen und Operationen

Definition (1.2)

Aussagen sind Sätze, die entweder wahr oder falsch sind.

„Tertium non datur“ (Es gibt keine dritte Möglichkeit)!

Ist x eine Aussage, so bezeichnet $w(x)$ ihren Wahrheitswert, w(x) = 1 falls x eine wahre Aussage ist und $w(x) = 0$ andernfalls.

Definition (1.3)

Aussagenformen sind sprachliche Gebilde mit Leerstellen (Aussagevariablen). Dort können Subjekte (Dinge), Prädikate (Eigenschaften) oder Aussagen eingesetzt werden. Wenn man einsetzt, erhält man eine Aussage. Im Allgemeinen muss wohldefiniert werden, was eingesetzt werden darf!

Beispiel

- $P(x)$ sei „x ist die Hauptstadt von Deutschland“; $P(x)$ ist eine Aussageform (keine Aussage!); für die Variable x dürfen wir Städtenamen einsetzen. P(München) (ist eine Aussage) ist falsch, P(Berlin) ist wahr.
- $Q(z) :\leftrightarrow z > 3$. Wir setzen reelle Zahlen ein, $z\ \in\ \mathbb{R}$. $Q(11)$ ist wahr, $Q(-2)$ ist falsch.

Verknüpfung von Aussagen durch Junktoren - Seien x und y Aussagen. Dann werden die folgenden Aussagen definiert:

- Negation $\neg x$ („nicht x“)
- Konjunktion $x \wedge y$ („x und y“, $\wedge$ vom lateinischen *aut*)
- Disjunktion $x \vee y$ („x oder y“, $\vee$ vom lateinischen *vel*))

- Implikation $x \rightarrow y$ („wenn x, dann y“)
- Äquivalenz $x \leftrightarrow y$ („x ist äquivalent zu y“)

Die Booleschen Formeln setzen sich aus Variablen und logischen Operatoren zusammen.

Für die Aussage „falsch“ und „wahr“ setzen wir nun die binäre Darstellung „0_b“ und „1_b“ (der Index b macht deutlich, dass es sich um eine Binärzahl handelt). Zur Vereinfachung schreiben wir im Kontext dieses Skriptes „$0_b = 0$“ und „$1_b = 1$“. Variablen werden mit Kleinbuchstaben $(a, b, ..., x, y, z)$ beschrieben. Aussagen denen die Aussage „falsch“ oder „wahr“ zugeordnet werden können, sind mit Großbuchstaben bezeichnet.

Beispiel

- $x \Leftrightarrow$ „Es regnet.“, $y :\leftrightarrow$ „Es ist Montag.“,
- $\neg y \Leftrightarrow$ „Es ist nicht Montag.“
- $(x \wedge y) \Leftrightarrow$ „Es regnet und es ist Montag.“
- $(x \vee y) \Leftrightarrow$ „Es regnet oder es ist Montag.“
- $(x \rightarrow y) \Leftrightarrow$ „Immer wenn es regnet, ist es Montag.“
- $(y \rightarrow x) \Leftrightarrow$ „Wenn Montag ist, dann regnet es.“
- $(y \rightarrow (1 + 1 = 2))$ ist eine wahre Aussage, da $1 + 1 = 2$ wahr ist (die Aussage ist also an jedem Wochentag wahr).

3.4 Boolesche Operation

Satz (1.0): Alle 16 Junktoren lassen sich mit den elementaren Junktoren Negation $\neg$, Konjunktion $\wedge$ und Disjunktion $\vee$ darstellen. D.h. es gibt Aussageformen $w_i(x, y)$, welche nur diese Verknüpfungen verwenden, so dass

$$x \; o_i \; y \Leftrightarrow \; w_i(x, y), \; i = 1, \ldots, 16$$

gilt.

Definition (1.4)

- NAND Verknüpfung: $x|y \Leftrightarrow \neg(x \wedge y)$
- NOR Verknüpfung: $x \nabla y \Leftrightarrow \neg(x \vee y)$

Die **NAND**-Verknüpfung heißt auch **Sheffer-Operator**, die **NOR**-Verknüpfung auch **Pierce-Operator**. Diese Operatoren lassen sich in der digitalen Schaltungstechnik sehr vorteilhaft realisieren.

Die Booleschen Formeln setzen sich aus Variablen und logischen Operatoren zusammen. Für die Aussage ‚falsch‘ und ‚wahr‘ setzen wir nun die binäre Darstellung ‚0_b‘ und ‚1_b‘ (der Index b macht deutlich, dass es sich um eine Binärzahl handelt). Zur Vereinfachung schreiben wir im Kontext dieses Skriptes ‚$0_b = 0$‘ und

‚$1_b = 1$‘. Variablen werden mit Kleinbuchstaben $(a, b, ..., x, y, z)$ beschrieben. Aussagen denen die Aussage ‚falsch‘ oder ‚wahr‘ zugeordnet werden können, sind mit Großbuchstaben bezeichnet.

x	0	0	1	1	verbale Form	Boolesche	Algebraische	Bezeichnung
y	0	1	0	1		Darstellung	Darstellung	
$x\ o_1\ y$	0	0	0	0	konstant 0	$\neg x \wedge x$	$\overline{x} \cdot x$	Nullfunktion
$x\ o_2\ y$	0	0	0	1	x und y	$x \wedge y$	$x \cdot y$	Konjunktion
$x\ o_3\ y$	0	0	1	0	nicht y, aber x	$x \wedge \neg y$	$x \cdot \overline{y}$	Inhibition
$x\ o_4\ y$	0	0	1	1	identisch x	x	x	Indentität
$x\ o_5\ y$	0	1	0	0	nicht x, aber y	$\neg x \wedge y$	$\overline{x} \cdot y$	Inhibition
$x\ o_6\ y$	0	1	0	1	identisch y	y	y	Indentität
$x\ o_7\ y$	0	1	1	0	x ungleich y	$x \not\leftrightarrow y$	$x \oplus y$	Antivalenz
$x\ o_8\ y$	0	1	1	1	x oder y	$x \vee y$	$x + y$	Disjunktion
$x\ o_9\ y$	1	0	0	0	nicht (x oder y)	$\neg(x \vee y)$	$\overline{(x + y)}$	Neg. Disjunk.
$x\ o_{10}\ y$	1	0	0	1	x gleich y	$x \leftrightarrow y$	$\overline{x \oplus y}$	Äquivalenz
$x\ o_{11}\ y$	1	0	1	0	nicht y	$\neg y$	$\overline{y}$	Negation
$x\ o_{12}\ y$	1	1	0	1	wenn x dann y	$x \rightarrow y$	$x \leq y$	Implikation
$x\ o_{13}\ y$	1	1	0	0	nicht x	$\neg x$	$\overline{x}$	Negation
$x\ o_{14}\ y$	1	0	1	1	wenn y dann x	$y \rightarrow x$	$y \leq x$	Implikation
$x\ o_{15}\ y$	1	1	1	0	nicht(x und y)	$\neg(x \wedge y)$	$\overline{(x \cdot y)}$	Neg. Konjuk.
$x\ o_{16}\ y$	1	1	1	1	konstant 1	1	1	Tautologie

- Warum einstellige Binärfunktion? ⇒ Nur von einer Variablen abhängig!
- Warum zweistellige Binärfunktion? ⇒ Von mindestens zwei Variablen abhängig!

Satz (1.1): Alle 16 Junktoren lassen sich mit der NAND-Verknüpfung darstellen. Analoges gilt für die NOR-Verknüpfung. Insbesondere bedeutet dies, dass sich jede Verknüpfung zweier Aussagen allein als Komposition von NAND-Verknüpfungen (oder NOR-Verknüpfungen) darstellen lässt.

Definition (1.5)

Aussageformen, in denen die Leerstellen Aussagevariablen sind, heißen **Formeln der Aussagenlogik**. Formeln heißen **erfüllbar**, falls die Variablen so durch Aussagen ersetzt werden können, dass eine wahre Aussage entsteht.

Definition (1.6)

Eine Formel heißt **allgemeingültig** oder auch eine **Tautologie**, falls bei jeder Ersetzung der Leerstellen durch Aussagen eine wahre Aussage entsteht.

3.5 Axiome und Theoreme

Eine abstrakte Algebra ist ein Verband einer nicht-leeren Menge B und einer Menge F von Operationen, die auf B oder einem kartesischen Produkt von B definiert sind. Sofern die Menge B nur zwei Elemente hat $(B = 0, 1)$, spricht man von einer logischen Algebra. Die alternativen Werte von B können mit

- falsch/wahr
- 0/1
- low/high

assoziiert werden. Im Allgemeinen ist für F ein vollständiger Satz von Operationen erforderlich. Da es viele solcher Sätze von Operationen gibt, können viele Algebren definiert werden. Nur wenige sind bisher ausgearbeitet worden. Wir beschränken uns auf die Boolesche Algebra.

$$\begin{aligned} \text{Algebra:} \quad & A = (\{B\}, \{F\}) \\ \text{Logische Algebra:} \quad & A = (\{0, 1\}, \{F\}) \\ \text{Boolesche Algebra:} \quad & A = (\{0, 1\}, \{\text{UND , ODER , NICHT }\}) \end{aligned}$$

Definition (1.0)

Eine Variable x heißt binäre Variable, wenn Sie nur die Werte 0 und 1 annehmen kann, $x \in \{0, 1\}$.

Für die Operation der Konjunktion, Disjunktion und der Negation existieren in der Literatur verschiedene Symbole:

- Konjunktion (**UND**): Boolesche Notation $x \wedge y$, Algebraische Notation: $x \cdot y$
- Disjunktion (**ODER**): Boolesche Notation $x \vee y$, Algebraische Notation: $x + y$
- Negation (**NICHT**): Boolesche Notation $\neg x$, Algebraische Notation: $\bar{x}$

Wir verwenden das Symbol ‘$\wedge$, für die Konjunktion und das Symbol ‘$\vee$, für die Disjunktion, da sie im deutschsprachigen Schrifttum weit verbreitet sind. Die Algebraische Schreibweise ist auch durchaus üblich. Jedoch gilt zu beachten, dass ein Mischen der beiden Schreibweisen zu vermeiden ist. In der diversen Fachliteratur kann es vorkommen, dass auch andere Schreibweisen und Darstellungen Verwendung finden. Es wird festgelegt, dass für dieses Skript und der dazugehörigen Vorlesung keine Mischformen erlaubt sind.

$$\begin{aligned} f = \neg x \vee \neg y \wedge z \quad & \rightarrow \quad \text{richtige Schreibweise} \\ f = \overline{x} + \overline{y} \cdot z \quad & \rightarrow \quad \text{richtige Schreibweise} \\ f = \overline{x} \vee \overline{y} \cdot z \quad & \rightarrow \quad \text{nicht zulässige Schreibweise!} \end{aligned}$$

Bezüglich der Notation ist es in manchen Fachbüchern üblich, das Konjunktionszeichen ($\wedge$) wegzulassen. Das ist durchaus zulässig auf Grund der Vorrangregel „Konjunktion vor Disjunktion, Äquivalenz, Antivalenz“, wobei für die letzten drei Operationen keine Vorrangregel üblich ist. Beispielsweise gilt:

$$f = x \wedge y \vee y \wedge z = x\ y \vee y\ z$$

Es ist empfohlen, die ausführliche Schreibweise anzuwenden.

Den Überbau dieser Algebra bilden sechs Axiome:

Abgeschlossenheit:

$$(1a) \quad \forall x, y \in B : x \vee y \in B \qquad \text{Geschlossenheit}$$

$$(1b) \quad \forall x, y \in B : x \wedge y \in B$$

Neutrales Element:

$$(2a) \quad \exists 0 \in B : x \vee 0 = x \qquad \text{Existenz des Null-Elements}$$

$$(2b) \quad \exists 1 \in B : x \wedge 1 = x \qquad \text{Existenz des Eins-Elements}$$

Kommutativ-Gesetz:

$$(3a) \quad \forall x, y \in B : x \vee y = y \vee x$$

$$(3b) \quad \forall x, y \in B : x \wedge y = y \wedge x$$

Distributiv-Gesetze:

$$(4a) \quad \forall\, x, y, z \in B : x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$$

$$(4b) \quad \forall\, x, y, z \in B : x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$$

Identitätselemente:

$$(5a) \quad \forall x \in B : x \vee \neg x = 1$$

$$(5b) \quad \forall x \in B : x \wedge \neg x = 0$$

Assoziativgesetz:

$$(6a) \quad \forall x, y, z \in B : (x \vee y) \vee z = x \vee (y \vee z)$$

$$(6b) \quad \forall x, y, z \in B : (x \wedge y) \wedge z = x \wedge (y \wedge z)$$

Aus den Axiomen folgen einige wichtige Theoreme, welche mit Hilfe der Axiome nachweisbar sind:

Absorptionsgesetz:

$$(7a) \quad \forall x, y \in B : x \vee (x \wedge y) = x$$

$$(7b) \quad \forall x, y \in B : x \wedge (x \vee y) = x$$

DeMorgan'sche Gesetze:

$$(8a) \quad \forall x, y \in B : \neg(x \vee y) = \neg x \wedge \neg y$$

$$(8b) \quad \forall x \in B : \neg(x \wedge y) = \neg x \vee \neg y$$

Idempotenz

$$(9a) \quad \forall x \in B : \neg(x \vee x) = x$$

$$(9b) \quad \forall x \in B : \neg(x \wedge x) = x$$

$$(9c) \quad \forall x \in B : \neg(\neg x) = x$$

Ferner gilt:

$$(x \wedge y) \vee (x \wedge \neg y) = x$$

$$(x \vee y) \wedge (x \vee \neg y) = x$$

$$x \vee (\neg x \wedge y) = x \vee y$$

$$x \wedge (\neg x \vee y) = x \wedge y$$

$$(x \wedge y) \vee (x \wedge \neg y \wedge z) = x \wedge y \vee x \wedge z$$

$$(x \vee y) \wedge (x \vee \neg y \vee z) = (x \vee y) \wedge (x \vee z)$$

3.6 Nachweis von Tautologien

Was ist eine Tautologie?

Eine Tautologie („dasselbe“), auch Verum (lat. verum „wahr“) genannt, ist in der Logik eine allgemein gültige Aussage, das heißt eine Aussage, die aus logischen Gründen immer wahr ist. Beispiel für eine Tautologien ist die Aussagen „Wenn es regnet, dann regnet es“.

Tautologien werden auf folgende Methoden untersucht:

- **Semantisches Verfahren** - Man stellt die Wahrheitswertetafel auf und belegt die auftretenden Aussagevariablen mit allen möglichen Kombinationen von Wahrheitswerten.
- **Syntaktisches Verfahren** - Man versucht, die Formel, die man als Tautologie nachweisen will, durch Anwendung bekannter Tautologien direkt umzuformen.

Beispiel

Zu beweisen ist die Äquivalenz der beiden Formeln:

$$f = x \wedge (y \vee z) \quad \text{und}$$
$$g = ((\neg\neg x \wedge x) \wedge y) \vee (x \wedge (x \vee z) \wedge z)$$

1. durch syntaktische Umformun
2. und durch die Wahrheitstafeln.

1. Syntaktische Umformung:

$$\begin{aligned} g &= ((\neg\neg x \wedge x) \wedge y) \vee (\underbrace{x \wedge (x \vee z)}_{\text{Absorbtionsgesetz}} \wedge z) \\ &= ((\underbrace{\neg\neg x}_{\text{Doppelnegation}} \wedge x) \wedge y) \vee (x \wedge z) \\ &= (\underbrace{(x \wedge x)}_{Idempotenz} \wedge y) \vee (x \wedge z) \\ &= \underbrace{(x \wedge y) \vee (x \wedge z)}_{\text{Distributivgesetz}} \\ &= x \wedge (y \vee z) \equiv f \end{aligned}$$

2. Um die **Wahrheitstabelle** für den jeweiligen Ausdruck zu erstellen, bietet sich an, die Formel schrittweise zu entwickeln. Bei drei Variablen (x,y,z) ergeben sich 8 Kombination. Zur besseren Übersicht und Lesbarkeit kann man alle 4 Zeilen einen kleinen Abstand einfügen. Es folgt:

$(i)_{10}$	x	y	z	$(y \vee z)$	f
0	0	0	0	0	0
1	0	0	1	1	0
2	0	1	0	1	0
3	0	1	1	1	0
4	1	0	0	0	0
5	1	0	1	1	1
6	1	1	0	1	1
7	1	1	1	1	1

$(i)_{10}$	x	y	z	$(x \wedge y)$	$(x \wedge (x \vee z) \wedge z)$	g
0	0	0	0	0	0	0
1	0	0	1	0	0	0
2	0	1	0	0	0	0
3	0	1	1	0	0	0
4	1	0	0	0	0	0
5	1	0	1	0	1	1
6	1	1	0	1	0	1
7	1	1	1	1	1	1

Somit ist bewiesen, dass $f \equiv g$ ist.

Aussagenkalkül - Betrachtet man nur Aussagen und Aussagenverknüpfungen mit den obigen Junktoren, so erhält man das Aussagenkalkül. Genauer besteht das Aussagenkalkül aus Axiomen, dass sind grundlegende Tautologien und Regeln, wie man durch Umformung aus diesen neue Tautologien gewinnt. Es gilt der Vollständigkeitssatz des Aussagenkalküls. Jede korrekte Formel (Tautologie) ist mit den Regeln des Aussagenkalküls ableitbar. Für die Darstellung der meisten mathematischen Sachverhalte reichen jedoch die bisherigen Sprachelemente nicht aus!

Vollständigkeitssatz des Aussagenkalküls - Jede korrekte Formel (Tautologie) ist mit den Regeln des Aussagenkalküls ableitbar. Für die Darstellung der meisten mathematischen Sachverhalte reichen jedoch die bisherigen Sprachelemente nicht aus!

Wichtige Tautologien:

a)	$x \vee \neg x$	Ausgeschlossenes Drittes
b)	$\neg(x \wedge \neg x)$	Satz vom Widerspruch
c)	$\neg\neg x \Leftrightarrow x$	Doppelte Verneinung
d)	$\neg(x \wedge y) \Leftrightarrow \neg x \vee \neg y$	I. Regel von de Morgan
e)	$\neg(x \vee y) \Leftrightarrow \neg x \wedge \neg y$	II. Regel von de Morgan
f)	$(x \to y) \Leftrightarrow (\neg y \to \neg x)$	Kontraposition
g)	$(x \to y) \wedge x \Leftrightarrow y$	modus ponens
h)	$(x \to y) \wedge \neg y \Leftrightarrow \neg x$	modus tollens
i)	$(x \to y) \Leftrightarrow (y \vee \neg x)$	
j)	$(x \to y) \wedge (y \to z) \Leftrightarrow (x \to z)$	modus barbara
k)	$x \wedge (y \wedge z) \Leftrightarrow (x \wedge y) \wedge z$	I. Assoziativgesetz
l)	$x \vee (y \vee z) \Leftrightarrow (x \vee y) \vee z$	II. Assoziativgesetz
m)	$x \wedge (y \vee z) \Leftrightarrow (x \wedge y) \vee (x \wedge z)$	I. Distributivgesetz
n)	$x \vee (y \wedge z) \Leftrightarrow (x \vee y) \wedge (x \vee z)$	II. Distributivgesetz

Beispiel

- Zeige: $\neg x \rightarrow (x \rightarrow y)$ ist eine Tautologie.

$$\begin{aligned} \neg x \rightarrow (x \rightarrow y) &\Leftrightarrow [(x \rightarrow y) \vee \neg\neg x] \\ &\Leftrightarrow [(y \vee \neg x) \vee x] \\ &\Leftrightarrow [y \vee (\neg x \vee x)] \end{aligned}$$

- $x \wedge (\neg x)$ ist nicht erfüllbar, dagegen ist $x \vee (\neg x)$ eine Tautologie

4 Aufgaben

4.1 Minimierung mittels Rechenregeln

a) Minimieren Sie mit Hilfe der Axiome und Gesetze der Booleschen Algebra die folgenden algebraischen Ausdrücke.

1. $f_1 = (\overline{x_0} \cdot x_1) + (x_0 + \overline{x_1}) \cdot (x_0 \cdot x_1)$
2. $f_2 = (x_0 + x_1) \cdot (x_0 + \overline{x_1} + x_2)$
3. $f_3 = (x_0 \cdot \overline{x_1}) + (x_0 + x_1) \cdot (x_0 \cdot x_1)$
4. $f_4 = (x_0 + x_2) \cdot (x_0 + \overline{x_1} + x_2)$

b) Geben Sie die minimierten Gleichungen in Boolescher Schreibweise an.

1. $f_1 = \ldots$
2. $f_2 = \ldots$
3. $f_3 = \ldots$
4. $f_4 = \ldots$

c) Geben Sie die minimierten Booleschen Gleichungen in LogiSim ein und verifizieren Sie die funktionale Richtigkeit.

4.2 Erfüllbarkeit und Tautologie

Welche der folgenden Formeln sind (un-)erfüllbar? Ist eine der Formeln eine Tautologie (gültig)?

a) $p \vee p \wedge q$

b) $\neg(p \wedge (p \rightarrow q) \rightarrow q)$

c) $\neg(p \wedge (p \rightarrow q)) \rightarrow q$

d) $(p \wedge (p \rightarrow q) \rightarrow q)$

Hinweis: Verwenden Sie die Funktionstabelle zur Beweisführung!

4.3 Äquivalenz Boolescher Funktionen (1)

Zeigen Sie, dass die Booleschen Funktionen

$$(x \vee y) \wedge (\neg x \vee \neg y) \quad \text{und} \quad (x \wedge \neg y) \vee (y \wedge \neg x)$$

äquivalent sind.

a) per Wahrheitstabelle

x	y	$(x \vee y)$	$(\neg x \vee \neg y)$	$(x \wedge \neg y)$	$(y \wedge \neg x)$	$(x \vee y) \wedge (\neg x \vee \neg y)$	$(x \wedge \neg y) \vee (y \wedge \neg x)$	All
0	0							
0	1							
1	0							
1	1							

b) mithilfe der grundlegenden „Äquivalenzen“!

4.4 Beweis der Konsensusregel

Der Beweis der Konsensusregel verwendet das Shannon'sche Expansionstheorem (Fallunterscheidung).

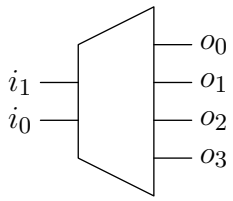
Beweisen Sie die Konsensusregel

$$(x \wedge y) \vee (\neg x \wedge z) \vee (y \wedge z) \equiv (x \wedge y) \vee (\neg x \wedge z)$$

ausschließlich mithilfe der grundlegenden Äquivalenzen!

4.5 Äquivalenz-Operation mit 2-zu-4-Decoder

Ein 2-zu-4-Decoder wandelt eine 2-Bit-Binärzahl in eine 4-Bit-Zahl in „One Hot"-Darstellung (siehe Funktionstabelle):



i_0	i_1	o_0	o_1	o_2	o_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

a) Zeigen Sie, dass man die Äquivalenz-Operation mithilfe mehrerer 2-zu-4-Decodern implementieren kann!

b) Beweisen Sie mithilfe grundlegender logischer Äquivalenzen, dass Ihre Lösung korrekt ist!

4.6 NOR als Basis

Zeigen Sie, dass die NOR-Operation $\neg(a \vee b)$ eine Basis ist!

4.7 Die Exklusiv-Oder-Operation ist keine Basis

Zeigen Sie, dass die Exklusiv-Oder-Operation **k**eine Basis ist! Die Definition von $\not\Leftrightarrow$ ist wie folgt:

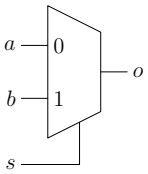
$$a \not\Leftrightarrow b \equiv (a \wedge \neg b) \vee (\neg a \wedge b)$$

4.8 Multiplexer als Basis

Untenstehend finden Sie die Wahrheitstabelle der Booleschen $\leq$-Funktion.

a	b	$a \leq b$
0	0	1
0	1	1
1	0	0
1	1	1

a) Implementieren Sie diese Funktion mithilfe von nur einem Multiplexer und Konstanten.



Der Ausgabewert o eines *Multiplexers* (siehe Abbildung links) entspricht dem Eingabewert b, wenn s gesetzt ist, und dem Wert von a, wenn s gleich Null ist.

b Zeigen Sie (in Wort oder Bild), dass der Multiplexer eine Basis darstellt. Begründen Sie Ihre Antwort!

4.9 Shannon-Expansion

Zeigen Sie mithilfe des Shannon'schen Expansionstheorems, dass

$$(x \vee \neg y) \wedge (\neg x \vee y) \equiv (x \wedge y) \vee (\neg x \wedge \neg y) \ .$$

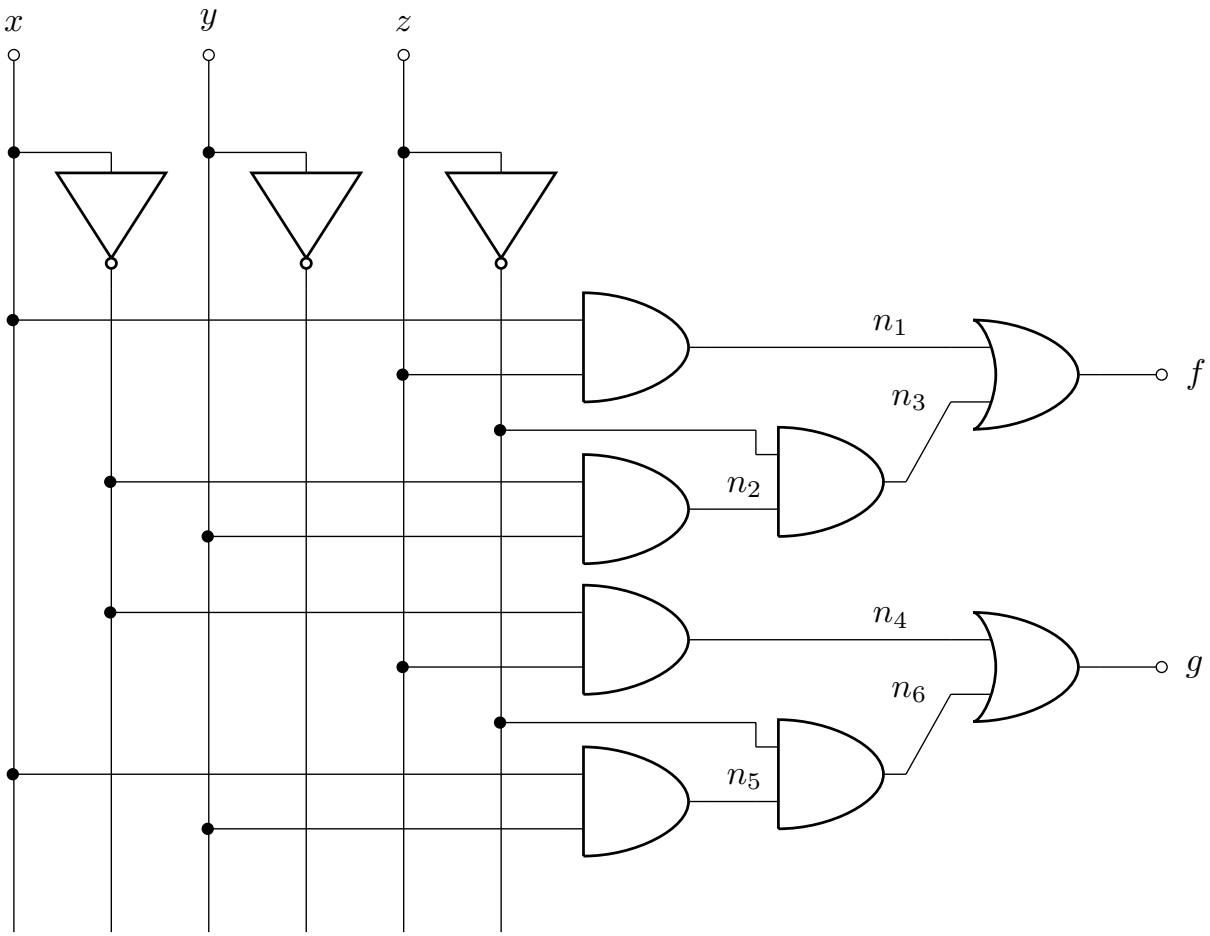
4.10 Umformung und Vereinfachung Boolescher Formeln

Vereinfachen Sie folgenden Booleschen Ausdruck und geben Sie bei jedem Schritt die verwendeten Regeln an:

$$g \quad = \quad (a \wedge b) \wedge \{\neg(b \vee c) \not\leftrightarrow c\}$$

4.11 Schaltungsanalyse (2)

Gegeben ist folgendes Schaltbild.



1. Lesen Sie die Funktionsgleichung aus dem Bild aus. Geben Sie die Funktionsgleichungen in boolescher als auch algebraischer Schreibweise an.

$f_{Boole}(x, y, z) =$ $\qquad$ $g_{Boole}(x, y, z) =$

$f_{Alg}(x, y, z) =$ $\qquad$ $g_{Alg}(x, y, z) =$

2. Füllen Sie die Wahrheitstabelle aus. Formulieren Sie $f(x, y, z)$ und $g(x, y, z)$ mit den Mintermen.

$(i)_{10}$	x	y	z	$f(x, y, z)$	Klausel	$g(x, y, z)$	Klausel
0	0	0	0				
1	0	0	1				
2	0	1	0				
3	0	1	1				
4	1	0	0				
5	1	0	1				
6	1	1	0				
7	1	1	1				

5 Literaturstudium

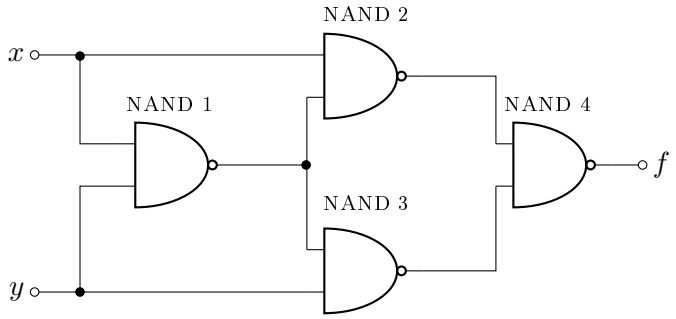
Zur Vor- und Nachbereitung des Laborversuchs sei folgendes Fachbuch empfohlen:

(a) Titelseite (b) Kapitel 2

Abbildung 1: Literaturhinweis `https://link.springer.com/`

6 Eine Musterlösung - XOR auf NAND-Basis

Aus der Tabelle geht hervor, dass Sie die XOR-Verknüpfung als NAND-Schaltung realisieren sollen. Eine Recherche im Internet und/oder ein Literaturstudium haben zu folgenden Schaltbild als Realisierung der XOR-Funktion mit NAND-Gattern geführt:



Die XOR-Verknüpfung wird auf NAND-Basis wie folgt formuliert:

$$
\begin{aligned}
f(x,y) &= x \not\leftrightarrow y \\
&= \neg\Big\{\neg\big[\underbrace{\neg\,(x \wedge y)}_{\text{NAND 1}} \wedge x\big]\Big\} \wedge \neg\big[\underbrace{\neg\,(x \wedge y)}_{\text{NAND 1}} \wedge y\big] \\
&\qquad \underbrace{\qquad\qquad\qquad\qquad\qquad\qquad}_{\text{NAND 4}}
\end{aligned}
$$

- Verifikation der NAND-Realisierung per Funktionstabelle

$(i)_{10}$	x	y	$x \not\leftrightarrow y$	$\neg\,(x \wedge y)$	$\underbrace{\neg\big[\neg\,(x \wedge y) \wedge x\big]}_{T_1}$	$\underbrace{\neg\big[\neg\,(x \wedge y) \wedge y\big]}_{T_2}$	$f(x,y) = \neg\big\{T_1 \wedge T_2\big\}$
0	0	0	0	1	1	1	0
1	0	1	1	1	1	0	1
2	1	0	1	1	0	1	1
3	1	1	0	0	1	1	0

Es folgt somit:

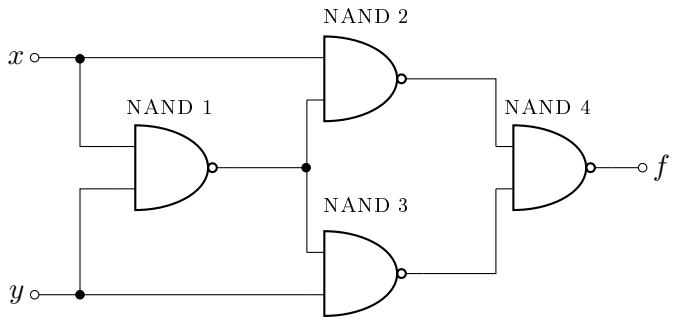
$$x \not\leftrightarrow y \equiv \neg\Big\{\neg\big[\neg\,(x \wedge y) \wedge x\big] \wedge \neg\big[\neg\,(x \wedge y) \wedge y\big]\Big\}$$

- analytischer Beweis der NAND-Realisierung

Aus der obigen Booleschen Funktion für die NAND-basierte XOR-Realisierung kann unter Anwendung der DeMorgan'schen Gesetze eine bekannte Form der Darstellung abgeleitet werden. Dies ist zugleich die analytische Beweisführung der Äquivalenz der beiden funktionalen Darstellungen.

$$
\begin{aligned}
f(x,y) &= \neg\Big\{\neg\big[\neg\,(x \wedge y) \wedge x)\big] \wedge \neg\big[\neg\,(x \wedge y) \wedge y\big]\Big\} \\
&= \neg\Big\{\neg\big[(\neg x \vee \neg y) \wedge x\big] \wedge \neg\big[(\neg x \vee \neg y) \wedge y\big]\Big\} \\
&= \neg\Big\{\neg\big[\underbrace{\neg x \wedge x}_{=0} \vee \neg y \wedge x\big] \wedge \neg\big[\neg x \wedge y \vee \underbrace{\neg y \wedge y}_{=0}\big]\Big\} \\
&= \neg\Big\{\neg\big[\neg y \wedge x\big] \wedge \neg\big[\neg x \wedge y\big]\Big\} \\
&= \neg\Big\{[y \vee \neg x] \wedge [x \vee \neg y]\Big\} \\
&= \neg\,[y \vee \neg x] \vee \neg\,[x \vee \neg y] \\
&= [\neg y \wedge x] \vee [\neg x \wedge y] \\
&= x \not\leftrightarrow y
\end{aligned}
$$

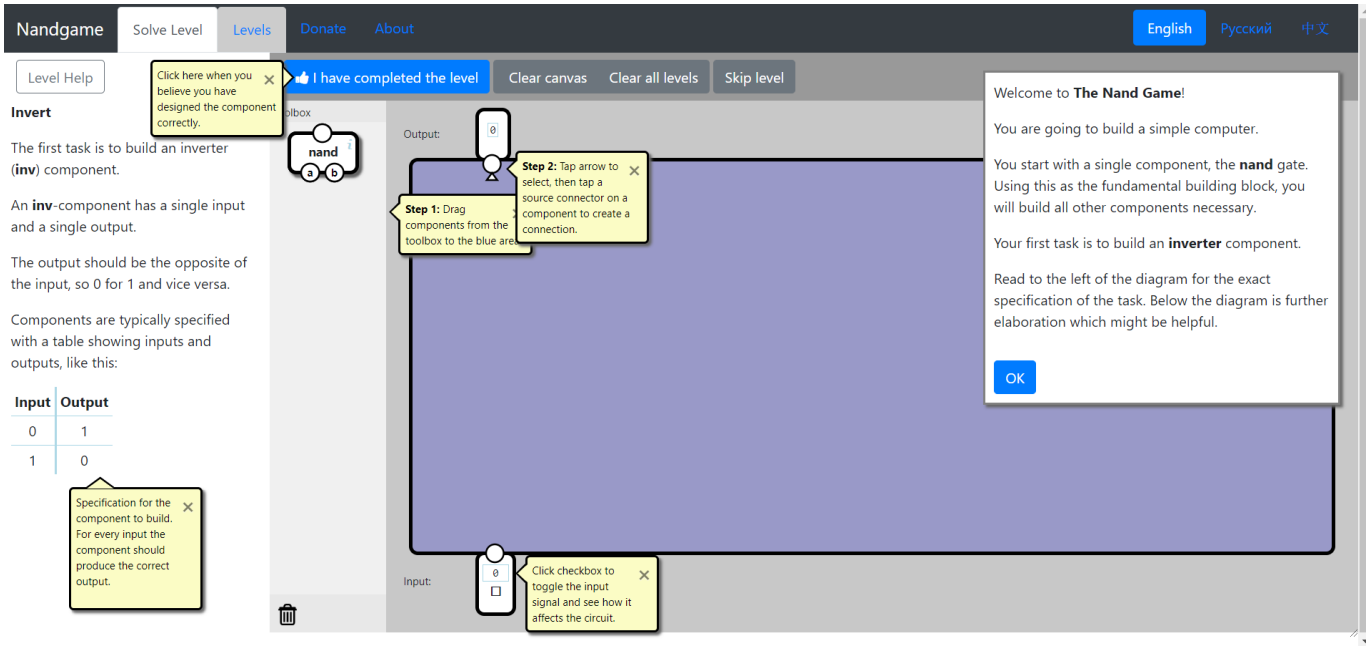
- Herleitung der NAND-Realisierung einer XOR-Verknüpfung



$$
\begin{aligned}
f(x,y) &= x \not\leftrightarrow y \\
&= [\neg y \wedge x] \vee [\neg x \wedge y] && \text{Basis-Operatoren} \\
&= \neg\,[y \vee \neg x] \vee \neg\,[x \vee \neg y] && \text{Auskl. der Neg. pro Term} \\
&= \neg\Big\{[y \vee \neg x] \wedge [x \vee \neg y]\Big\} && \text{Auskl. der Neg. über den ges. Ausdruck, NAND 4} \\
&= \neg\Big\{\underbrace{\neg\big[\neg y \wedge x\big]}_{\text{Term 1}} \wedge \underbrace{\neg\big[\neg x \wedge y\big]}_{\text{Term 2}}\Big\} && \text{Auskl. der Neg., Umwandlung Term 1/2} \\
&= \neg\Big\{\neg\big[\neg x \wedge x \vee \neg y \wedge x\big] \wedge \neg\big[\neg x \wedge y \vee \neg y \wedge y\big]\Big\} && \text{Erweiterung kompl. Element} \\
&= \neg\Big\{\neg\big[(\neg x \vee \neg y) \wedge x\big] \wedge \neg\big[(\neg x \vee \neg y) \wedge y\big]\Big\} && \text{Auskl. der x/y, NAND 2/3} \\
&= \neg\Big\{\neg\big[\neg\,(x \wedge y) \wedge x)\big] \wedge \neg\big[\neg\,(x \wedge y) \wedge y\big]\Big\} && \text{Auskl. der Neg., NAND 1}
\end{aligned}
$$

7 Sonstiges

Etwas zum spielen:



`http://nandgame.com/`

Notizen: