Universität Bielefeld
Technische Fakultät

R|V|S

Rechnernetze und Verteilte Systeme

Technische Informatik I

Vorlesung 3: Bool'sche Algebra

Mirco Hilbert
mail@Mirco-Hilbert.de

Übersicht

- Bool'sche Algebra
  - Operatoren und Eigenschaften
  - Schaltzeichen
  - Wertetabellen
  - Bearbeitung durch algebraische Gleichungen
- Schaltfunktionen
- einfache Schaltnetze
- Normalformen
  - disjunktive Normalform
  - konjunktive Normalform

Informationsrepräsentation

- Informationen wie Buchstaben und Zahlen können als Summe von Faktoren mit fester Basis dargestellt werden
- Diese Informations-Repräsentation wird physikalisch realisiert. Also wird folgendes erwünscht:
  - Die Verarbeitung der physikalischen Repräsentation sei identisch mit der Umformung der logischen Repräsentation
  - Zu jedem Zeitpunkt existiert eine eineindeutige Abbildung zwischen logischer und physikalischer Repräsentation

Mögliche physikalische Repräsentationen

Verschiedene physikalische Repräsentationen sind denkbar:

- Stärke eines hydraulischen oder elektrischen Stroms
- Farbe, Intensität oder Phasenlage von Licht
- stufenlos veränderbare Spannungspegel
- drei diskrete Spannungspegel (3-wertige Logik): negative / keine / positive Spannung
- zwei diskrete Spannungspegel (2-wertige **binäre** Logik): Spannung / keine Spannung

Heutiger Stand der Technik

- binäre Basis, d.h. Darstellung durch die Menge der **bool'schen** Werte

  2 = {0, 1}
- **Info-Repräsentation** durch Binärwörter.
  - Ein Binärwort der Länge n ist ein Element von
    - exp(**2**, n) = **2** x **2** x .... x **2**
      = {0,1} x {0,1} x ... x {0,1}
  - **Sequenz/Tupel von 0 und 1, Länge n**

Heutige Stand der Technik

- Elektrische **Info-Verarbeitung** durch Funktionen auf Binärwörtern (**Bool'sche Funktionen, Schaltfunktionen**)
- Schaltnetze als Realisierung (physikalische Repräsentation) logischer Schaltfunktionen

Bool'sche Algebra

Definition (Binäre Bool'sche Algebra)

Ein algebraisches System (**2**, $\wedge$, $\vee$, $\neg$)

- $\wedge$ (logisches **UND**) binäre Funktion
- $\vee$ (logisches **ODER**) binäre Funktion
- $\neg$ (logisches Komplement, **NICHT**) unäre Fkt

- Wertebereich {0,1}
- Funktionen definierbar durch Tabellen, wie folgt

Bool'sche Algebra

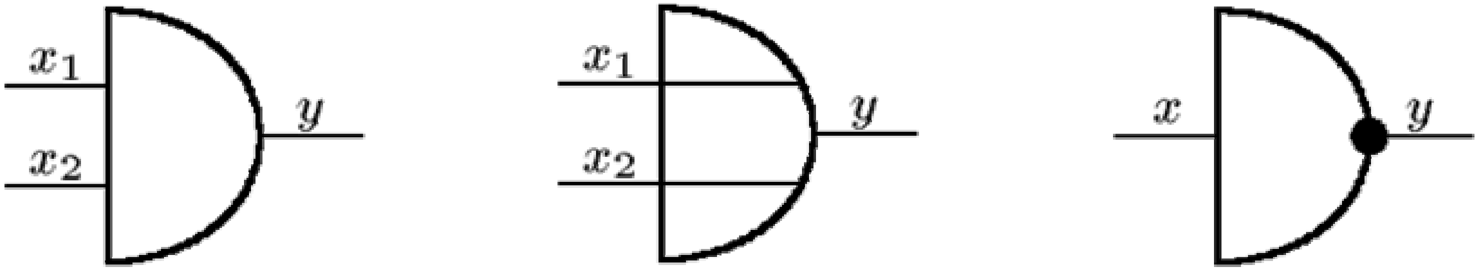
- Wertebereich {0, 1}
- Unäre Funktion definiert durch
  - f(0) = ??
  - f(1) = ??
- Binäre Funkion definiert durch
  - g(0,0) = ??
  - g(0,1) = ??
  - g(1,0) = ??
  - g(1,1) = ??

Bool'sche Algebra

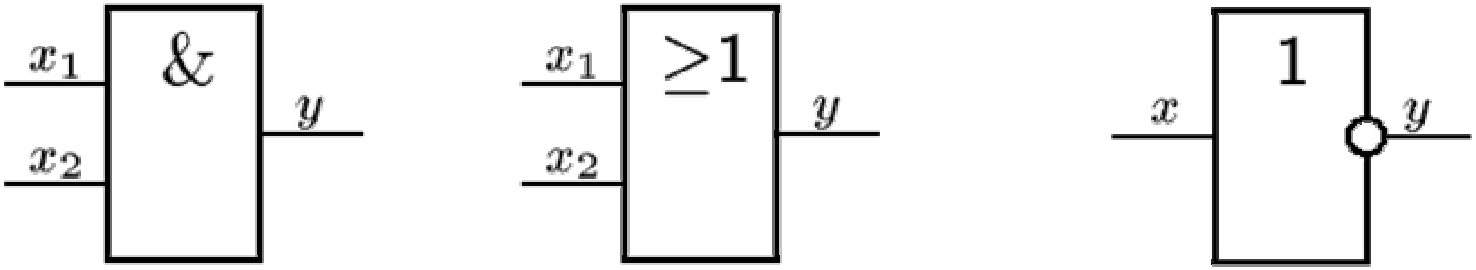
- Wir schreiben $\neg$ in Präfix-Notation: $\neg a$ oder $\overline{a}$
- Wir schreiben $\wedge$ und $\vee$ in Infix-Notation:
  - Statt $\wedge(a,b)$ schreiben wir $(a \wedge b)$
  - Das gleiche für $\vee$
- Funktionen also definiert, wie folgt:
  - $\neg 0 = 1 \qquad \neg 1 = 0$
  - $0 \wedge 0 = 0 \qquad 1 \wedge 1 = 1$
    $0 \wedge 1 = 1 \wedge 0 = 0$
  - $0 \vee 0 = 0 \qquad 1 \vee 1 = 1$
    $1 \vee 0 = 0 \vee 1 = 1$

Schaltzeichen

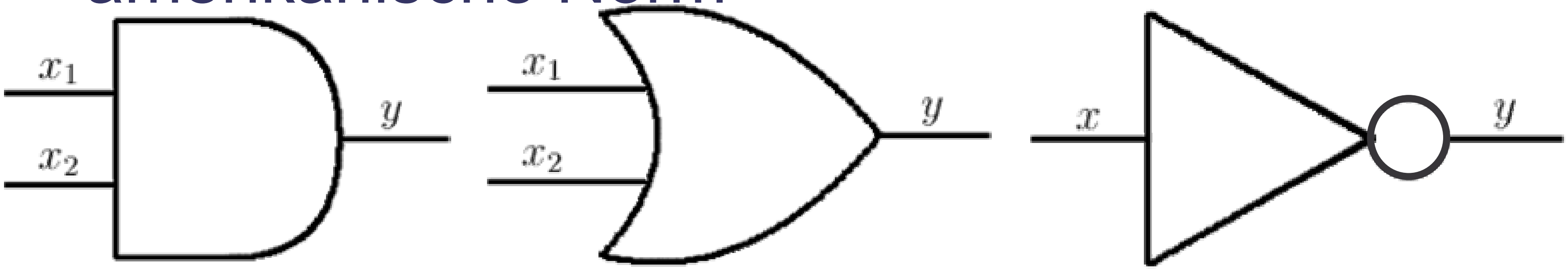
- alte Norm



- neue Norm



- amerikanische Norm



UND (AND) $\wedge$ **ODER (OR) $\vee$** **NICHT (NOT) $\neg$**

Bool'sche Funktionen

- bool'sche Funktionen

$y = f(x_1, x_2) = x_1 \wedge x_2 \quad y = f(x_1, x_2) = x_1 \vee x_2 \quad y = f(x) = \neg x$

- Wertetabellen

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

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

x	y
0	1
1	0

Bool‘sche Basis-Operatoren I

Theorem: Alle binäre bool'schen Funktionen sind eine Komposition der beiden Basis-Operatoren $\wedge$ und $\neg$; bzw. $\vee$ und $\neg$.

Beweis: Es gibt 16 binäre Bfnen (siehe unten).

Man schreibt die Kompositionen, wie gewünscht.

Beispiel: Nach dem Gesetz von DE MORGAN läßt sich $\vee$ ausdrücken durch:

$$\boldsymbol{a} \vee b = \neg(\neg a \wedge \neg b)$$

Ebenso läßt sich $\wedge$ ausdrücken durch:

$$\boldsymbol{a} \wedge b = \neg(\neg a \vee \neg b)$$

Bool'sche Gleichungen

- De Morgan'sche Gesetze
  - $\neg (a \wedge b) = (\neg a) \vee (\neg b)$
    - a=0, b=0:
      $\neg (0 \wedge 0) = \neg 0 = 1 = 1 \vee 1 = (\neg 0) \vee (\neg 0)$
    - a=0, b=1:
      $\neg (0 \wedge 1) = \neg 0 = 1 = 1 \vee 0 = (\neg 0) \vee (\neg 1)$
    - a=1, b=0:
      $\neg (1 \wedge 0) = \neg 0 = 1 = 0 \vee 1 = (\neg 1) \vee (\neg 0)$
    - a=1, b=1:
      $\neg (1 \wedge 1) = \neg 1 = 0 = 0 \vee 0 = (\neg 1) \vee (\neg 1)$
  - $\neg (a \vee b) = (\neg a) \wedge (\neg b)$
    - Ähnlicher Beweis

Ein Basis-Theorem

Theorem: Alle bool'schen Funktionen (unäre, binäre, n-äre für jede positive ganze Zahl n) sind eine Komposition der beiden Basis-Operatoren $\wedge$ und $\neg$; bzw. $\vee$ und $\neg$.

Beweis: Ausgelassen.

Bool'scher Basis-Operator II

Theorem: Alle binären bool'schen Funktionen sind eine Komposition des Operators **NAND**:

$$a \text{ NAND } b := \neg(a \wedge b)$$

Beweis: Die Negation $\neg$ läßt sich ausdrücken durch:

$$\neg a = \neg(a \wedge a) = a \text{ NAND } a$$

Die Konjunktion $\wedge$ und Disjunktion $\vee$ durch:

$$a \wedge b = \neg\neg(a \wedge b) = \neg(a \text{ NAND } b)$$
$$= (a \text{ NAND } b) \text{ NAND } (a \text{ NAND } b)$$

$$a \vee b = \neg\neg a \vee \neg\neg b = \neg(\neg a \wedge \neg b) = \neg a \text{ NAND } \neg b$$
$$= (a \text{ NAND } a) \text{ NAND } (b \text{ NAND } b)$$

Bool'scher Basis-Operator III

Theorem: Das gleiche gilt für den Scheffer'schen Strich | (**NOR**):

$$\boldsymbol{a} \mid \boldsymbol{b} := \neg a \wedge \neg b$$

Beweis:

Für die Negation:

$$\neg a = \neg a \wedge \neg a = a \mid a$$

Die Disjunktion:

$$\boldsymbol{a} \vee b = \neg(\neg a \wedge \neg b) = \neg(a \mid b) = (a \mid b) \mid (a \mid b)$$

Die Konjunktion: **Übung !**

Mögliche Kombinationen zweier boolscher Variablen

x_1	x_2	0	$x_1 \wedge x_2$	$x_1 \wedge \overline{x_2}$	x_1	$\overline{x_1} \wedge x_2$	x_2	$(x_1 \wedge \overline{x_2}) \vee (\overline{x_1} \wedge x_2)$	$x_1 \vee x_2$
0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1
		Null	Konjunktion	Inhibition	Transfer	Inhibition	Transfer	Antivalenz	Disjunktion
			AND					XOR	OR

x_1	x_2	1	$\overline{x_1 \wedge x_2}$	$\overline{x_1} \vee x_2$	$\neg x_1$	$x_1 \vee \overline{x_2}$	$\neg x_2$	$(\overline{x_1} \vee x_2) \wedge (x_1 \vee \overline{x_2})$	$\overline{x_1 \vee x_2}$
0	0	1	1	1	1	1	1	1	1
0	1	1	1	1	1	0	0	0	0
1	0	1	1	0	0	1	1	0	0
1	1	1	0	1	0	1	0	1	0
		Eins		Implikation	Negation	Implikation	Negation	Äqivalenz	
			NAND		NOT		NOT		NOR

Anzahl von Bool'sche Funktionen

- Unäre Schaltfunktionen:
  - Domäne-Wertebereich: **{0,1}** also zwei Elemente
  - **Ziel-Wertebereich: {0,1}** also zwei Elemente
  - **Zwei Elemente in der Domäne, jeweils ein Wert aus dem Zielwertebereich und zwei möglich Werte**
  - **Auswahlsmöglichkeiten also exp(2,2) = 4**
- **f(0) = 0, f(1) = 0 : False**
- **f(0) = 0, f(1) = 1: Identität**
- **f(0) = 1, f(1) = 0: Negation $\neg$**
- **f(0) = 1, f(1) = 1: True**

Anzahl von Bool'sche Funktionen

- Binäre Schaltfunktionen:
  - Domäne-Wertebereich: **{0,1} x {0,1}** also exp(2.2) = 4 Elemente
  - **Ziel-Wertebereich: {0,1}** also zwei Elemente
  - **Vier Elemente in der Domäne, jeweils ein Wert aus dem Zielwertebereich und zwei möglich Werte**
  - **Auswahlsmöglichkeiten also exp(2,4) = 16 binäre Schaltfunktionen**

Anzahl möglicher Schaltfunktionen I

- Wir haben gesehen: Es gibt 16 verschiedene mögliche Schaltfunktionen bei gerade mal 2 Eingangsvariablen.
- Wie viele Funktionen sind aber bei n Eingangsvariablen möglich?
  - Sei f: $\mathbf{2}^n \rightarrow \mathbf{2}$ eine Schaltfunktion mit n frei belegbaren Variablen.
  - Dann gibt es 2^n mögliche Kombinationen von Wertebelegungen für die n Variablen
  - *Eine bestimmte Funktion* f ist für jede dieser $w = 2^n$ Eingangskombinationen definiert, sie produziert also w Ergebnisse (jeweils 0 oder 1).

Anzahl möglicher Schaltfunktionen II

- Durch die Wahl unterschiedlicher Schaltfunktionen f: $\mathbf{2}^n \to \mathbf{2}$ lassen sich für diese $\boldsymbol{w}$ Ergebnisse 2^w Ergebniskombinationen vorschreiben.
- Es existieren also genau $2^w = 2^{2^n}$ bool‘sche Funktionen, die alle möglichen Wertkombinationen der Eingangsvariablen auf alle Kombinationen von Resultaten abbilden.

- Bei $n = 2$ Eingangsvariablen existieren also $2^{2^2} = 2^4 = 16$ mögliche Schaltfunktionen.
- Bei $n = 3$ schon $2^{2^3} = 2^8 = 256$.

Auswahl von Schaltfunktionen

- Wir kürzen, wie folgt: XYZW bedeutet
  - **f(0,0) = X, f(0,1) = Y, f(1,0) = Z, f(1,1) = W**
- **Wichtigste Funktionen sind**
  - **0000: False; 1111: True**
  - **0011: 1er Projektion; 0101: 2er Projektion**
  - **0001: and; 0111: or**
  - **1110; nand; 1000: nor, oder Scheffer'sche Strich**
  - **0110: xor**
  - **1001: äquivalenz; 1101: implikation (material conditional); 1011: reverse-implikation;**

Schaltfunktion

- Eine n-äre Schaltfunktion bzw. Bool'sche Funktion ist eine Abbildung
  - f: exp(**2**,n) $\rightarrow$ **2**
    {0,1} x {0,1} x ... x {0,1} $\rightarrow$ {0,1}
  - **z.B. f(x,y,z): exp(2,3) $\rightarrow$ 2**
    {0,1} x {0,1} x {0,1} $\rightarrow$ {0,1}
    f(0,0,0) = 0
    f(0,0,1) = 1
    f(0,1,0) = 1
    f(0,1,1) = 0 ....

Beispiel 1

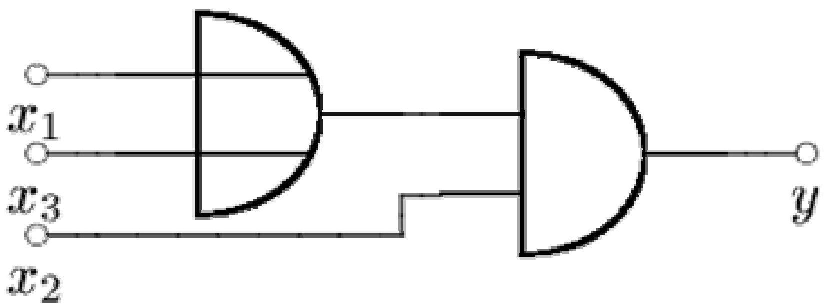
- Schaltfunktion:

  $f : 2^3 \to 2$ mit $f(x_1, x_2, x_3) = x_2 \wedge (x_1 \vee x_3)$
- Wertetabelle:

x_1	x_2	x_3	$f(x_1, x_2, x_3)$
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Beispiel 1 - Schaltnetz

- Schaltfunktion:
  $f : 2^3 \to 2$ mit $f(x_1, x_2, x_3) = x_2 \wedge (x_1 \vee x_3)$
- Realisierung als Schaltnetz



- Funktion ist eine Komposition von $\wedge$ und $\vee$

Beispiel 2: Antivalenz (XOR)

- Schaltfunktion:

$$f : 2^2 \to 2 \text{ mit } f(x_1, x_2) = (x_1 \wedge \overline{x_2}) \vee (\overline{x_1} \wedge x_2)$$

- Wertetabelle:

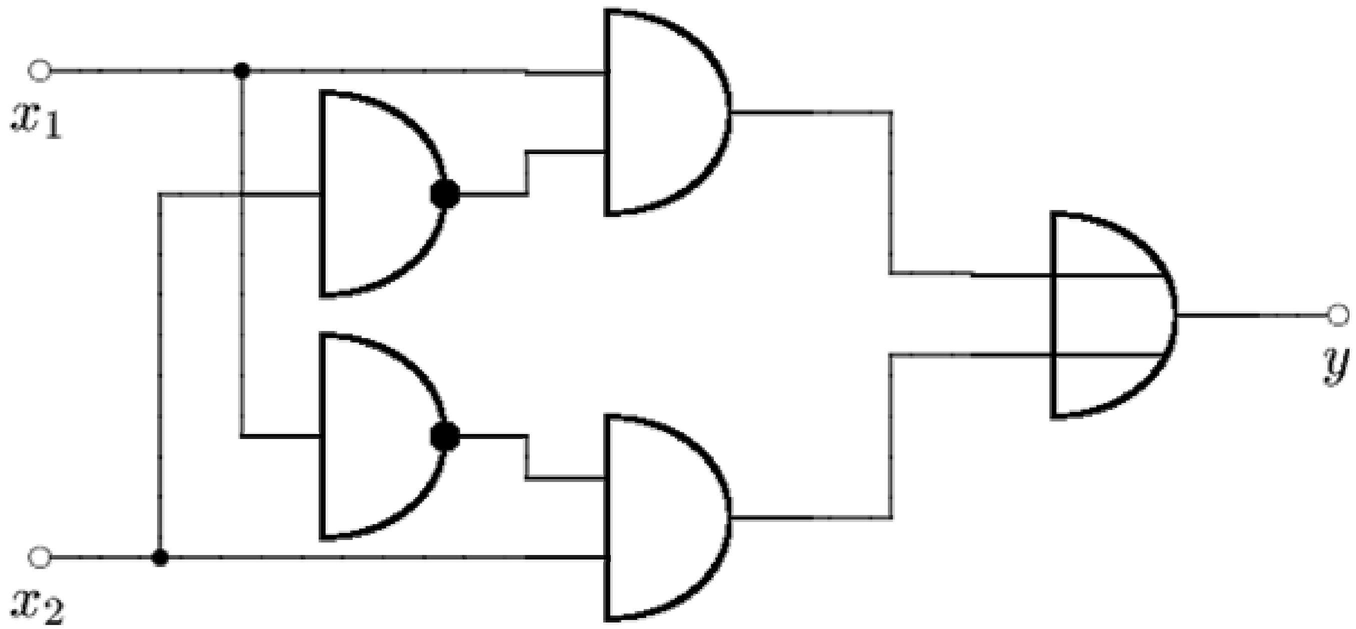
x_1	x_2	$f(x_1, x_2)$
0	0	0
0	1	1
1	0	1
1	1	0

Beispiel 2 - Schaltnetz

- Schaltfunktion:

$$f : 2^2 \to 2 \text{ mit } f(x_1, x_2) = (x_1 \wedge \overline{x_2}) \vee (\overline{x_1} \wedge x_2)$$

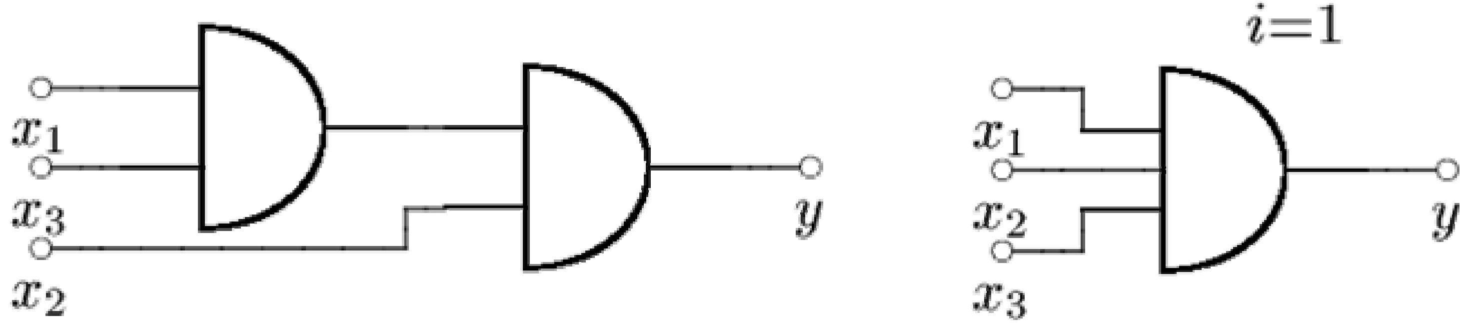
- Realisierung als Schaltnetz:



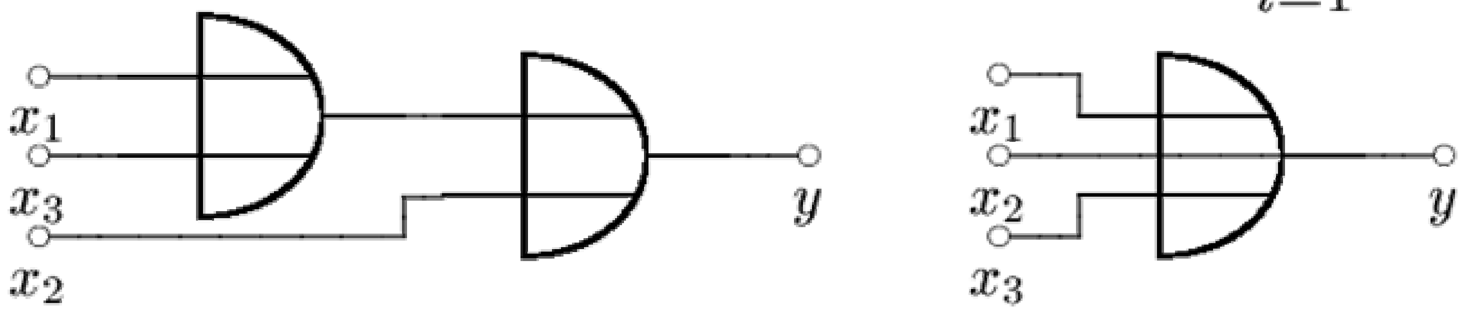
Abkürzungsnotation I

Da das Assoziativgesetz gilt, läßt sich

$(x_1 \wedge x_2) \wedge x_3$ auch notieren als $\bigwedge_{i=1}^{3} x_i$



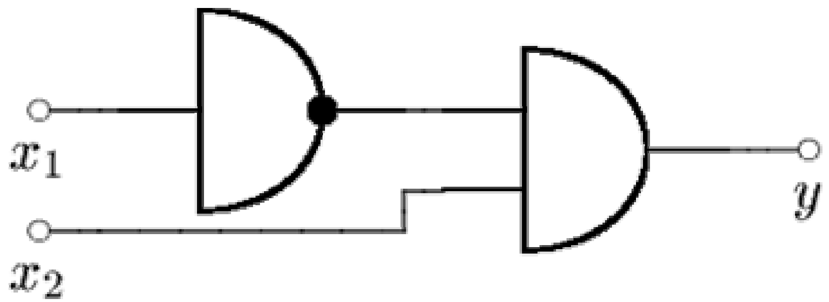
$(x_1 \vee x_2) \vee x_3$ auch notieren als $\bigvee_{i=1}^{3} x_i$



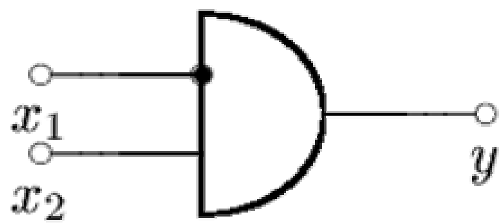
analog für beliebig großes i.

Abkürzungsnotation II

Der Übersichtlichkeit halber werden Inverter (NICHT-Schaltzeichen) nicht explizit eingezeichnet sondern durch invertierte Eingänge der nachfolgenden Schaltzeichen ausgedrückt.
Somit läßt sich

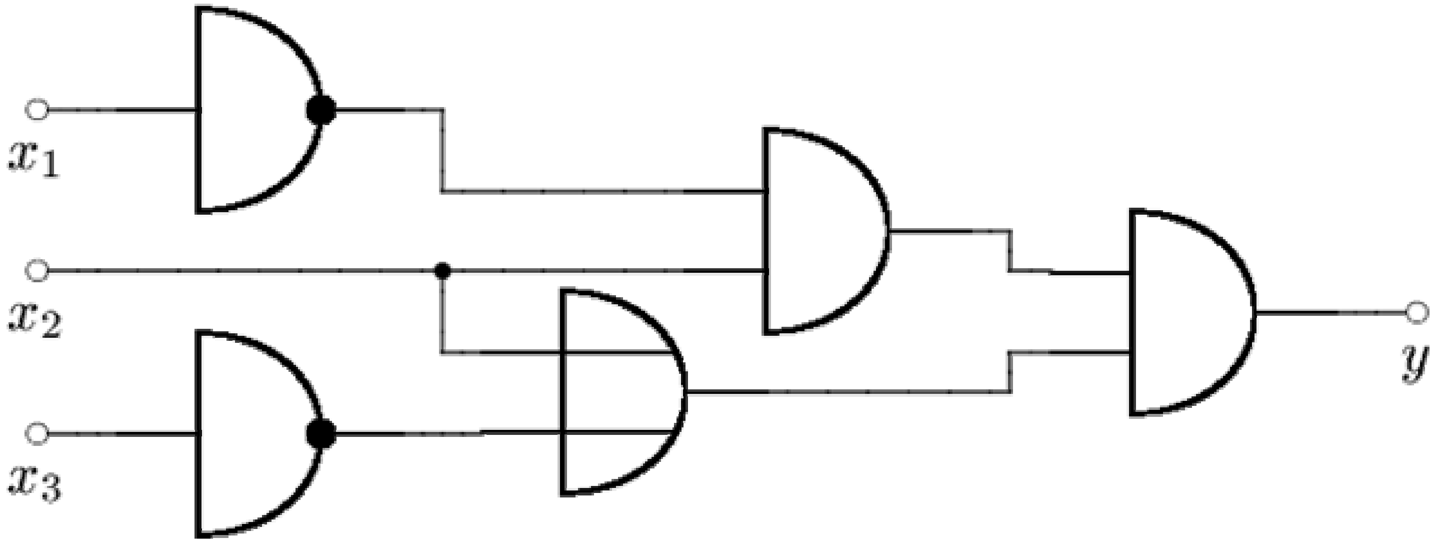


auch notieren als

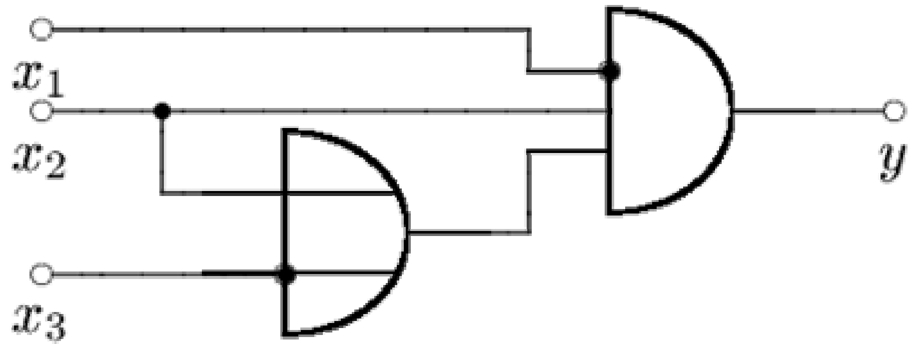


Beispiel 3

$$f(x_1, x_2, x_3) = \overline{x_1} \wedge (x_2 \vee \overline{x_3}) \wedge x_2$$



Abkürzende Notation:



Beispiel 3 - vereinfachbar?

Kann man die Schaltfunktion

$$f(x_1, x_2, x_3) = \overline{x_1} \wedge (x_2 \vee \overline{x_3}) \wedge x_2$$

aus Beispiel 3 noch vereinfachen?

Schauen wir uns folgende Wertetabelle an:

x_1	x_2	x_3	$\overline{x_1} \wedge (x_2 \vee \overline{x_3}) \wedge x_2$	$\overline{x_1} \wedge x_2$
0	0	0	0	0
0	0	1	0	0
0	1	0	1	1
0	1	1	1	1
1	0	0	0	0
1	0	1	0	0
1	1	0	0	0
1	1	1	0	0

Beispiel 3 - vereinfachbar?

Die Wahrheitswert-Belegungen der beiden bool'schen Funktionen

$\overline{x_1} \wedge (x_2 \vee \overline{x_3}) \wedge x_2$ und $\overline{x_1} \wedge x_2$

sind gleich.

Somit sind sie **äquivalent**.

Wie aber läßt sich das formal beweisen?

Theoreme der Bool'schen Algebra I

- Kommutativgesetz

  $a \wedge b = b \wedge a$ $\qquad$ $a \vee b = b \vee a$

- Assoziativgesetz

  $(a \wedge b) \wedge c = a \wedge (b \wedge c)$ $\qquad$ $(a \vee b) \vee c = a \vee (b \vee c)$

- Absorptionsgesetz

  $a \wedge (a \vee b) = a$ $\qquad$ $a \vee (a \wedge b) = a$

- Idempotenzgesetz

  $a \wedge a = a$ $\qquad$ $a \vee a = a$

Theoreme der Bool'schen Algebra II

- Distributivgesetz

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

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

- Gesetz von De Morgan

  $\neg(a \wedge b) = \neg a \vee \neg b$ $\quad$ $\neg(a \vee b) = \neg a \wedge \neg b$

- ***Beweise? Gleich wie bei De Morgan***
- ***Theorem: Alle wahre Gleichungen (Gesetze) in*** $\neg, \wedge, \vee$ sind Konsequenzen dieser Gesetze

  Beweis: Ausgelassen

Beispiel 3 - Vereinfachung

$$\overline{x_1} \wedge (x_2 \vee \overline{x_3}) \wedge x_2$$

läßt sich nach dem Kommutativgesetz umformen zu: $\overline{x_1} \wedge x_2 \wedge (x_2 \vee \overline{x_3})$

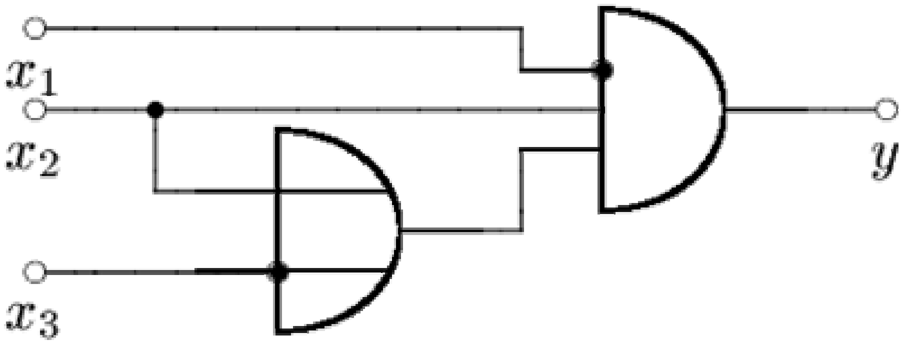
Das läßt sich nach dem Absorptionsgesetz umformen zu:

$$\overline{x_1} \wedge x_2$$

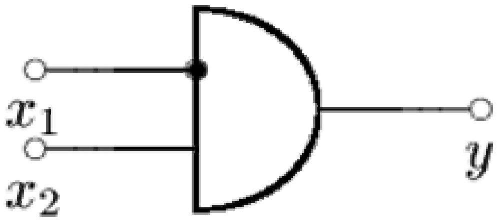
Somit ist die Äquivalenz der beiden Schaltfunktionen bewiesen.

Beispiel 3 - Vereinfachung

Somit läßt sich die Schaltung



vereinfachen zu



Literatur und Links

- Structured Computer Organization
  Andrew S. Tanenbaum, Prentice Hall, 1999
- Lectures on Boolean Algebra
  Paul Halmos, Springer-Verlag
- Rechneraufbau und Rechnerstrukturen
  W. Oberschelp und G. Vossen, 6. Aufl.,
  R. Oldenbourg-Verlag, 1994
- Kurz-Zusammenstellung
  „Formale Methoden der Linguistik I“
  Mirco Hilbert, WS 2000/01
- elearn.rvs.uni-bielefeld.de

Universität Bielefeld
Technische Fakultät

RVS
Rechnernetze und
Verteilte Systeme

Technische Informatik I

Nächste Woche:
Vorlesung 4: Vereinfachung von Schaltfunktionen

Mirco Hilbert
mail@Mirco-Hilbert.de