64-040 Modul IP7: Rechnerstrukturen

http://tams.informatik.uni-hamburg.de/lectures/2011ws/vorlesung/rs

Kapitel 8

Andreas Mäder

Universität Hamburg
Fakultät für Mathematik, Informatik und Naturwissenschaften
Fachbereich Informatik
Technische Aspekte Multimodaler Systeme

Wintersemester 2011/2012

Kapitel 8

Wiederholung: Grundbegriffe der Algebra

- Mengen
- Relationen, Verknüpfungen
- Gruppe, Abel'sche Gruppe
- Körper, Ring
- Vektorraum
- usw.

Nutzen einer (abstrakten) Algebra?!

Analyse und Beschreibung von

- gemeinsamen, wichtigen Eigenschaften
- mathematischer Operationen
- mit vielfältigen Anwendungen

Spezifiziert durch

- die Art der Elemente (z.B. ganze Zahlen, Aussagen, usw.)
- die Verknüpfungen (z.B. Addition, Multiplikation)
- zentrale Elemente (z.B. Null-, Eins-, inverse Elemente)

Anwendungen: z.B. fehlerkorrigierende Codes auf CD/DVD

Boole'sche Algebra

- George Boole, 1850: Untersuchung von logischen Aussagen mit den Werten *true* (wahr) und *false* (falsch)
- Definition einer Algebra mit diesen Werten
- Vier grundlegende Funktionen:
  - NEGATION (NOT) Schreibweisen: $\neg a$, $\overline{a}$, ~a
  - UND –"– $a \wedge b$, $a \,\&\, b$
  - ODER –"– $a \vee b$, $a \mid b$
  - XOR –"– $a \oplus b$, $a \,\hat{}\, b$
- Claude Shannon, 1937: Realisierung der Boole'schen Algebra mit Schaltfunktionen (binäre digitale Logik)

Grundverknüpfungen

- zwei Werte: *wahr* (*true*, 1) und *falsch* (*false*, 0)
- vier grundlegende Verknüpfungen:

NOT(x)

x	
0	1
1	0

AND(x, y)

y \ x	0	1
0	0	0
1	0	1

OR(x, y)

y \ x	0	1
0	0	1
1	1	1

XOR(x,y)

y \ x	0	1
0	0	1
1	1	0

- alle logischen Operationen lassen sich mit diesen Funktionen darstellen (*vollständige Basismenge*)

Grundverknüpfungen

- zwei Werte, $\{0, 1\}$
- insgesamt 4 Funktionen mit einer Variable
  $f_0(x) = 0,\ f_1(x) = 1,\ f_2(x) = x,\ f_3(x) = \neg x$
- insgesamt 16 Funktionen zweier Variablen
- allgemein 2^{2^n} Funktionen von n Variablen
- später noch viele Beispiele

Alle Funktionen von zwei Variablen

x = 0 1 0 1 y = 0 0 1 1	Bezeichnung	Notation	Alternativnotation	Java/C-Notation
0 0 0 0	Nullfunktion	0		`0`
0 0 0 1	AND	$x \cap y$		`x&&y`
0 0 1 0	Inhibition	$y > x$		`y>x`
0 0 1 1	Identität y	y		`y`
0 1 0 0	Inhibition	$x > y$		`x>y`
0 1 0 1	Identität x	x		`x`
0 1 1 0	XOR	$x \oplus y$	$x \neq y$	`x!=y`
0 1 1 1	OR	$x \cup y$		`x||y`
1 0 0 0	NOR	$\neg(x \cup y)$		`!(x||y)`
1 0 0 1	Äquivalenz	$\neg(x \oplus y)$	$x = y$	`x==y`
1 0 1 0	NICHT x	$\neg x$	x'	`!x`
1 0 1 1	Implikation	$x \leq y$	$x \to y$	`y>=x`
1 1 0 0	NICHT y	$\neg y$	y'	`!y`
1 1 0 1	Implikation	$x \geq y$	$x \leftarrow y$	`x>=y`
1 1 1 0	NAND	$\neg(x \cap y)$		`!(x&&y)`
1 1 1 1	Einsfunktion	1		`1`

Boole'sche Algebra

- 6-Tupel $< \{0,1\}, \vee, \wedge, \neg, \ 0, 1 >$ bildet eine Algebra

- $\{0,1\}$ Menge mit zwei Elementen
- $\vee$ ist die „Addition"
- $\wedge$ ist die „Multiplikation"
- $\neg$ ist das „Komplement" (nicht das Inverse!)
- 0 (false) ist das Nullelement der Addition
- 1 (true) ist das Einselement der Multiplikation

Rechenregeln: Ring / Algebra

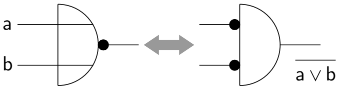
Eigenschaft	Ring der ganzen Zahlen	Boole'sche Algebra
Kommutativgesetz	$a + b = b + a$ $a \times b = b \times a$	$a \vee b = b \vee a$ $a \wedge b = b \wedge a$
Assoziativgesetz	$(a + b) + c = a + (b + c)$ $(a \times b) \times c = a \times (b \times c)$	$(a \vee b) \vee c = a \vee (b \vee c)$ $(a \wedge b) \wedge c = a \wedge (b \wedge c)$
Distributivgesetz	$a \times (b + c) = (a \times b) + (a \times c)$	$a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$
Identitäten	$a + 0 = a$ $a \times 1 = a$	$a \vee 0 = a$ $a \wedge 1 = a$
Vernichtung	$a \times 0 = 0$	$a \wedge 0 = 0$
Auslöschung	$-(-a) = a$	$\neg(\neg a) = a$
Inverses	$a + (-a) = 0$	—

Rechenregeln: Ring / Algebra (cont.)

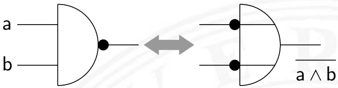
Eigenschaft	Ring der ganzen Zahlen	Boole'sche Algebra
Distributivgesetz	—	$a \lor (b \land c) = (a \lor b) \land (a \lor c)$
Komplement	—	$a \lor \neg a = 1$
	—	$a \land \neg a = 0$
Idempotenz	—	$a \lor a = a$
	—	$a \land a = a$
Absorption	—	$a \lor (a \land b) = a$
	—	$a \land (a \lor b) = a$
De-Morgan Regeln	—	$\neg(a \lor b) = \neg a \land \neg b$
	—	$\neg(a \land b) = \neg a \lor \neg b$

De-Morgan Regeln

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



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



1. Ersetzen von *UND* durch *ODER* und umgekehrt
   $\Rightarrow$ Austausch der Funktion
2. Invertieren aller Ein- und Ausgänge

Verwendung

- bei der Minimierung logischer Ausdrücke
- beim Entwurf von Schaltungen
- siehe Abschnitte: „Schaltfunktionen“ und „Schaltnetze“

XOR: Exklusiv-Oder / Antivalenz

$\Rightarrow$ entweder a oder b (ausschließlich)
a ungleich b ($\Rightarrow$ Antivalenz)

- $a \oplus b = (\neg a \wedge b) \vee (a \wedge \neg b)$
  genau einer von den Termen a und b ist wahr

- $a \oplus b = (a \vee b) \wedge \neg(a \wedge b)$
  entweder a ist wahr, oder b ist wahr, aber nicht beide gleichzeitig

- $a \oplus a = 0$