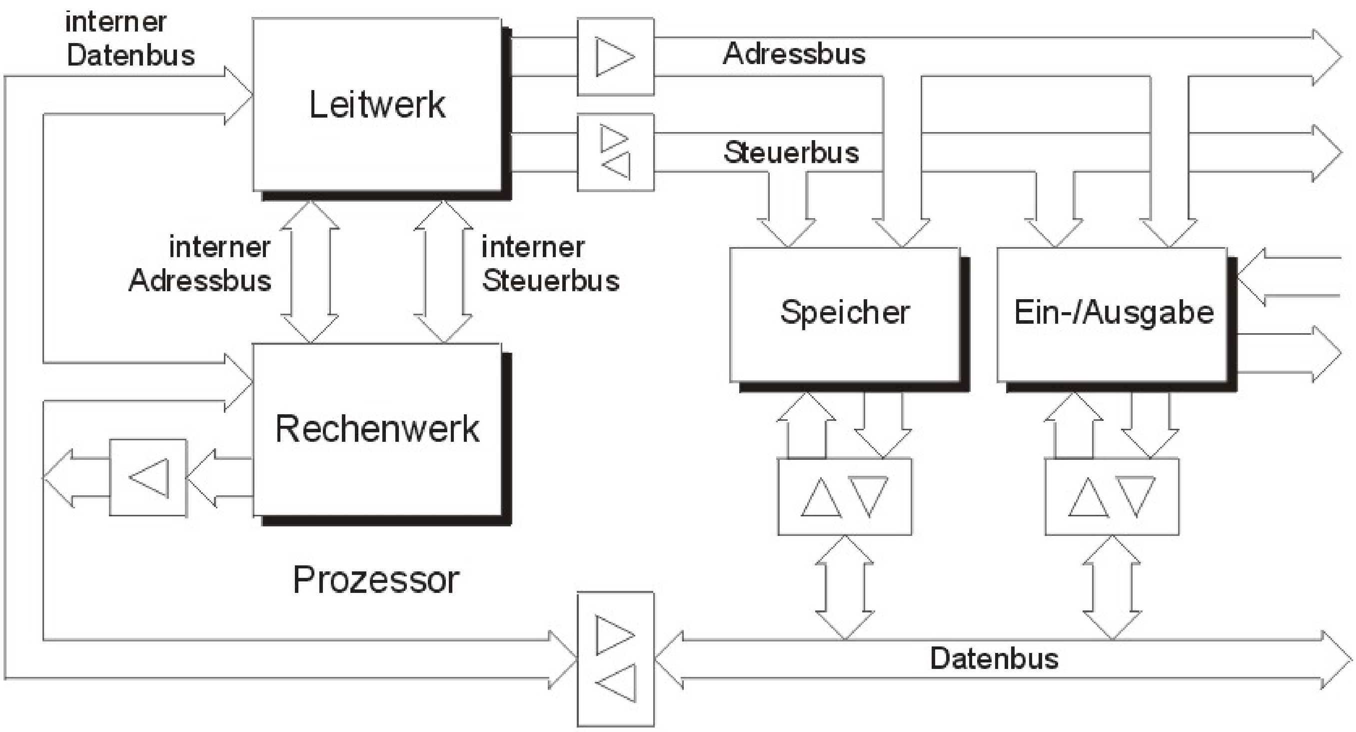
FernUniversität in Hagen

Kurs 1608: Computersysteme I

Kurseinheit 1: Schaltfunktionen und Boole'sche Ausdrücke

Autor: Jörg Keller



© 2008 FernUniversität in Hagen
Fachbereich Informatik
Alle Rechte vorbehalten

01608-1-01-S 1

000 000 000 (10/08)

Das Werk ist urheberrechtlich geschützt. Die dadurch begründeten Rechte, insbesondere das Recht der Vervielfältigung und Verbreitung sowie der Übersetzung und des Nachdrucks, bleiben, auch bei nur auszugsweiser Verwertung, vorbehalten. Kein Teil des Werkes darf in irgendeiner Form (Druck, Fotokopie, Mikrofilm oder ein anderes Verfahren) ohne schriftliche Genehmigung der FernUniversität reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

Inhaltsverzeichnis

Vorwort

Allgemeines

Wir begrüßen Sie herzlich zum Kurs *01608 Computersysteme I*. Dieser Kurs führt Sie in die Grundlagen der Computer-Hardware ein: Schaltfunktionen, Schaltnetze, Speicher, Schaltwerke und Mikroprozessoren. Im Folgekurs *01609 Computersysteme II* werden Sie die Architektur von Hochleistungsprozessoren und –Speichersystemen kennenlernen. Die beiden Kurse können entweder beide in einem Semester oder in zwei aufeinanderfolgenden Semestern belegt werden, da der Kurs 01609 zeitversetzt beginnt. Beide Kurse werden in jedem Semester angeboten.

Die beiden Kurse 01608 und 01609 bilden im Bachelor-Studiengang Informatik das Modul *Computersysteme*, das mit einer Prüfungsklausur endet, die am Ende jedes Semesters stattfindet[1]. Das Modul Computersysteme ist mit 10 Leistungspunkten bewertet, was einem *durchschnittlichen* Arbeitsaufwand eines Studierenden von 300 Stunden entspricht. Zur Bearbeitung des Kurses 01608 müssen Sie also im Mittel mit 150 Stunden Aufwand rechnen. Wenn Sie mit dem Kurs gut zurecht kommen, ist der Aufwand vermutlich geringer. Wenn Sie Schwierigkeiten mit den Kursinhalten haben ist der Aufwand vermutlich höher.

Die Inhalte des Kurses sind unseres Erachtens mit dem Studium des Kurstextes zu erschließen. Allerdings hilft zum vertieften Verständnis, gerade bei schwierigen Kursteilen, die Konsultation von Sekundärliteratur, um einen etwas anderen Blickwinkel auf die gleiche Materie zu erhalten. Hierzu existiert eine Fülle von Lehrbüchern, eine kleine Auswahl zeigt das Literaturverzeichnis. Die meisten dieser Bücher sind in der Universitätsbibliothek der FernUniversität und anderer Universitäten verfügbar. Zur Ausleihe von Büchern konsultieren Sie bitte die Webseiten der Bibliothek.

Kursbeleger fragen oft, warum Studierende der Informatik Kenntnisse über Hardware erwerben sollen, auch wenn sie voraussichtlich nie Prozessoren oder Rechner entwerfen werden. Hierfür gibt es unseres Erachtens mehrere Gründe.

1. So selten sind hardware-nahe Tätigkeiten gar nicht! Eine der größten Branchen in Deutschland ist der Maschinen– und Automobilbau, und heutige Maschinen und Autos beziehen einen großen Teil ihrer Funktionalität aus sogenannten *eingebetteten Systemen*, d.h. in darin integrierten

[1] Zur Verwendung in anderen Studiengängen konsultieren Sie bitte Ihre studiengangsspezifischen Studien– und Prüfungsordnungen.

Hard– und Softwaresystemen. Bei der Entwicklung eingebetteter Systeme findet auch heute noch[2] die Software-Entwicklung sehr hardware-nah statt, so dass Kenntnisse der zugrunde liegenden Hardware und Maschinensprache notwendig sind.

2. Auch bei der klassischen Software-Entwicklung sind Kenntnisse der Hardware, auf der die Software ausgeführt werden soll, wichtig um die Leistung der Software zu steigern. Dies reicht vom Einstellen der korrekten Compiler-Switches bis zur Optimierung von Datenstrukturen in Bezug auf die Nutzung der Prozessor-Caches. Außerdem ist ein grundlegendes Verständnis der Bereiche, die an das eigene Arbeitsfeld angrenzen, stets nützlich, um über den „Tellerrand" schauen zu können. Schließlich verschmelzen bei den zunehmend verwendeten rekonfigurierbaren Schaltungsbausteinen die Bereiche der Software- und Hardware-Entwicklung, so dass auch hier Kenntnisse beider Bereiche notwendig sind.

3. Viele Konzepte, die innerhalb ihres Anwendungsbereichs bei Computersystemen vorgestellt werden, sind in der gesamten Informatik wichtig. Als Beispiel sollen *endliche Automaten* genannt werden, die nicht nur als Zustandsmaschinen komplexer Schaltwerke sondern auch als Konzept zur Erkennung regulärer Sprachen in der theoretischen Informatik und als Steuerungsalgorithmen in der praktischen Informatik verwendet werden.

Der Kurs 01608 *Computersysteme I* besteht aus vier Kurseinheiten.

Kurseinheit 1 beschreibt Schaltfunktionen und ihre verschiedenen Darstellungen, speziell die Beschreibung durch Boole'sche Ausdrücke.

Kurseinheit 2 beschreibt Schaltnetze, die Schaltfunktionen in Hardware realisieren, sowie dafür geeignete Zahlendarstellungen.

Kurseinheit 3 beschreibt Speicher sowie Schaltwerke, d.h. Kombinationen von Speichern und Schaltnetzen.

Kurseinheit 4 beschreibt komplexe Schaltwerke, d.h. Schaltwerke in denen zwischen Daten– und Kontrollsignalen unterschieden wird, und erweitert diese zum Grundkonzept eines einfachen Prozessors.

Zu jeder Kurseinheit gibt es Einsendeaufgaben, die im 2-Wochenrhythmus zu bearbeiten sind. Die genauen Termine entnehmen Sie bitte den Seiten des Kurses in der LVU.

Die ersten beiden Kurseinheiten sind gemessen an ihrem Inhalt recht formal gehalten. Dies geschieht weder um Studierende zu verwirren noch ist es Selbstzweck. Rein textuelle Beschreibungen erklären zwar die häufigen Betriebsfälle einer Schaltung anschaulicher als Formeln, gleichzeitig bleiben oft Unklarheiten über das Verhalten der Schaltung in Ausnahmefällen. Solche Unklarheiten werden bei einer formalen Beschreibung vermieden, da dort in der Regel auffällt, wenn ein Fall fehlt, oder wenn verschiedene Fälle nicht disjunkt sind und damit die Beschreibung nicht widerspruchsfrei ist. Auch das Erlernen dieser Betrachtungsweise ist über den Bereich der Computersysteme hinaus wichtig, da auch Software-Projekte oft unter unentdeckten unvollständigen oder widersprüchlichen Anforderungen leiden. Schließlich erlaubt die formale Beschreibung auch

[2]Dies geschieht aus Ressourcen– und Effizienzgründen.

das Führen von Beweisen, und führt so über die bloße, „kochbuchartige“ Vermittlung von Fakten (die Tiefe des Carry-Chain-Addierers ist zum Beispiel linear zur Bitbreite der Eingaben) zu einem Verständnis warum dies so ist (der Übertrag läuft beim Carry-Chain-Addierer eventuell über alle Stellen). Schließlich übt das formale Vorgehen auch die aus den Kursen der Mathematik kommenden Methoden im Informatik-Umfeld.

Die Inhalte des Moduls Computersysteme werden in Kursen der Kataloge B2 und M2 vertieft. Prozessoren für eingebettete Rechensysteme werden im Kurs 01706 *Anwendungsorientierte Mikroprozessoren* behandelt. Der Entwurf von integrierten Schaltungen wird im Kurs 01721 *VLSI-Entwurfsalgorithmen* vermittelt. Arithmetische Schaltnetze und Schaltwerke werden in Kurs 01726 *Rechnerarithmetik* ausführlich behandelt. Praktische Aspekte von PC-Systemen werden im Kurs 01744 *PC–Technologie* vertieft.

Literatur

1. B. Becker, R. Drechsler, P. Molitor. Technische Informatik. Pearson Studium 2005.
2. K. Gotthardt. Aufgaben zur Informationstechnik Teil I. Logos 2003.
3. H. P. Gumm, M. Sommer. Einführung in die Informatik. 6. Auflage. Oldenbourg 2004.
4. G. Hotz. Einführung in die Informatik. Teubner 1990.
5. J. Keller, W. J. Paul. Hardware Design, 3. Auflage. Teubner 2005.
6. W. Oberschelp, G. Vossen. Rechneraufbau und Rechnerstrukturen. Oldenbourg 2000.
7. W. Schiffmann, R. Schmitz. Technische Informatik I+II. 5. Auflage, Springer 2005.
8. A. Tanenbaum. Computerarchitektur. 5. Auflage. Pearson 2005.
9. H.-D. Wuttke, K. Henke. Schaltsysteme. Pearson 2003.

Kurseinheit 1

Schaltfunktionen und Boole'sche Ausdrücke

Lernziele

Die Lernziele dieser Kurseinheit sind:

- Kenntnis der verschiedenen Darstellungen von Schaltfunktionen,
- Sicherheit im Umgang mit und Kenntnis der Eigenschaften von Boole'schen Ausdrücken,
- Fähigkeit zur Bestimmung der Kosten von Boole'schen Ausdrücken, speziell von Minimalpolynomen.

1.1 Vorbemerkungen

1.1.1 Mengen und Funktionen

In diesem Abschnitt befassen wir uns mit Funktionen auf endlichen Mengen. Diese Funktionen werden sich durch Schaltnetze berechnen lassen. Wir setzen ein allgemeines Verständnis von Mengen und Funktionen voraus, und werden hier einige speziellere Sachverhalte wiederholen.

Wir bezeichnen mit

$$\mathbf{N} = \{1, 2, 3, \ldots\} \text{ bzw. } \mathbf{N}_0 = \mathbf{N} \cup \{0\}$$

natürliche Zahlen
reelle Zahlen
leere Menge

die *Menge der natürlichen Zahlen* (ohne bzw. mit Null) und mit $\mathbf{R}$ die *Menge der reellen Zahlen*. Mit Z_n bezeichnen wir die Menge der Zahlen von 0 bis $n-1$. Weitere Mengen bezeichnen wir mit Großbuchstaben, z.B. $M = \{1, 2, 5\}$. Die leere Menge bezeichnen wir mit $\emptyset$.

abzählbar unendlich
überabzählbar unendlich

Wir bezeichnen die Mächtigkeit einer Menge M mit $\#M$. Bei einer endlichen Menge ist $\#M \in \mathbf{N}_0$. Zum Beispiel ist $\#\{1, 2, 5\} = 3$. Zwei Mengen M und N sind gleichmächtig, wenn es eine Bijektion $f : M \to N$ gibt, d.h. wenn jedem Element von M eineindeutig ein Element aus N zugeordnet werden kann. Während diese Festlegung für endliche Mengen (bei denen man ja lediglich die Anzahlen vergleichen müsste) aufwändig erscheint, ist sie bei unendlichen Mengen angebracht. Eine unendliche Menge, die gleichmächtig wie $\mathbf{N}$ ist, heißt *abzählbar unendlich*, ansonsten *überabzählbar unendlich*. Zum Beispiel ist die Menge der reellen Zahlen überabzählbar unendlich, die Menge M der geraden natürlichen Zahlen ist hingegen abzählbar unendlich, da die Funktion $f : \mathbf{N} \to M$, $f(x) = 2x$ eine Bijektion zwischen diesen Mengen darstellt.

kartesisches Produkt

Das *kartesische Produkt* $M \times N$ zweier Mengen M und N ist definiert als

$$M \times N = \{(a, b) \ : \ a \in M, b \in N\} .$$

Hierbei gilt für endliche Mengen M und N: $\#(M \times N) = \#M \cdot \#N$. Beispielsweise ist

$$Z_2 \times Z_3 = \{0, 1\} \times \{0, 1, 2\} = \{(0, 0), (0, 1), (0, 2), (1, 0), (1, 1), (1, 2)\} .$$

und

$$\#(Z_2 \times Z_3) = 6 = 2 \cdot 3 = \#Z_2 \cdot \#Z_3 .$$

In gleicher Weise kann man das kartesische Produkt aus $n \in \mathbf{N}$ Mengen $M_0, \ldots, M_{n-1}$ definieren. Sind alle Mengen $M_i = M$ identisch, dann schreibt man statt $M \times \cdots \times M$ auch M^n, und es gilt $\#M^n = (\#M)^n$. Beispielsweise ist

$$\begin{aligned} \{0,1\}^3 &= \{(0,0,0), (0,0,1), (0,1,0), (0,1,1), \\ &\quad\ (1,0,0), (1,0,1), (1,1,0), (1,1,1)\} , \\ \#\{0,1\}^3 &= (\#\{0,1\})^3 = 2^3 = 8 . \end{aligned}$$

Korrekterweise müsste man das n-fache kartesische Produkt induktiv definieren und hätte dann sehr viele Klammern zu setzen. Wir setzen allerdings nur die äußeren Klammern, d.h. statt

$$a = (\cdots(a_0, a_1), a_2) \cdots), a_{n-1})$$

schreiben wir

$$a = (a_0, a_1, a_2, \ldots, a_{n-1}) \ .$$

Wir nennen a eine *Folge der Länge* n, in Zeichen $l(a) = n$. Weiterhin vereinbaren wir, wenn $\cup$ die Vereinigung von Mengen bezeichnet, die Notation

Folge

$$A^+ = \bigcup_{i \in \mathbf{N}} A^i \ .$$

Eine endliche Menge A von Symbolen oder Zeichen nennen wir ein *Alphabet*. Die Folgen über A, d.h. die Elemente aus A^+ nennen wir *Zeichenreihen*.

Alphabet

Zeichenreihe

Die eindeutige Zeichenreihe mit Länge 0 wird *das leere Wort* genannt und häufig mit ϵ abgekürzt. Für ein beliebiges Alphabet A bezeichnet man mit A^0 die Menge, die nur das leere Wort enthält, also

leeres Wort

$$A^0 = \{\epsilon\} \ .$$

Man bezeichnet mit

$$A^* = A^+ \cup \{\epsilon\} = \bigcup_{i \in \mathbf{N}_0} A^i$$

die Menge aller Zeichenreihen mit Zeichen aus A einschließlich des leeren Worts.

Damit ist $\{0,1\}^* = \{\epsilon, 0, 1, (0,0), (0,1), (1,0), (1,1), \ldots\}$.

Für die Anzahl von Zeichenreihen der Länge höchstens n gilt, da die vereinigten Mengen disjunkt sind

$$\# \left(\bigcup_{i=0}^{n} A^i \right) = \#A^0 + \#A^1 + \cdots + \#A^n = \sum_{i=0}^{n} \#A^i = \sum_{i=0}^{n} (\#A)^i \ . \tag{1.1}$$

Um die Summe der rechten Seite zu vereinfachen, benötigen wir das folgende Lemma 1.1.

Lemma 1.1 *Seien x und n zwei natürliche Zahlen mit $x \neq 1$. Dann gilt*

$$\sum_{i=0}^{n} x^i = \frac{x^{n+1} - 1}{x - 1} \ .$$

Beweis: Sei $s = \sum_{i=0}^{n} x^i$. Dann gilt

$$x \cdot s = \sum_{i=0}^{n} x^i \cdot x = \sum_{i=1}^{n+1} x^i \ .$$

Weiterhin ist $(x-1) \cdot s = x \cdot s - s$. Wir setzen für s und für $x \cdot s$ jeweils die obigen Summenformeln ein und erhalten

$$(x-1) \cdot s = \sum_{i=1}^{n+1} x^i - \sum_{i=0}^{n} x^i = x^{n+1} - 1 \ .$$

Dividiert man die linke und rechte Seite der Gleichungskette durch $x - 1$, so erhält man die Behauptung. ∎

Den Ausdruck $\sum_{i=0}^{n} x^i = x^0 + x^1 + \cdots x^n$ nennt man *geometrische Reihe.*

geometrische Reihe

Damit ist die Anzahl der Zeichenketten der Länge höchstens 2 über dem Alphabet $\{0, 1\}$, also gerade der oben ausformulierte Teil der Menge $\{0, 1\}^*$, $(2^3 - 1)/(2 - 1) = 7$.

Eine Folgerung aus Lemma 1.1 ist

$$\sum_{i=m}^{n-1} x^i = \frac{x^n - x^m}{x - 1} . \tag{1.2}$$

wobei m eine natürliche Zahl kleiner als n sein soll. Zum Beispiel ist

$$\sum_{i=0}^{n-1} 2^i = 2^n - 1 \text{ und } \sum_{i=m}^{n-1} 2^i = 2^n - 2^m .$$

Selbsttestaufgabe 1.1 *Beweisen Sie Gleichung (1.2).*

Eine *Funktion* kann nun ebenfalls über ein kartesisches Produkt definiert werden.

Funktion

Funktionswert

Definitionsbereich

Wertebereich

Definition 1.1 *Es seien X und Y Mengen. Eine* Funktion *f von X nach Y ist eine Teilmenge f von $X \times Y$, für die gilt: für jedes $x \in X$ gibt es genau ein $y \in Y$ mit $(x, y) \in f$. Dieses y heißt der* Funktionswert *von f an der Stelle x. Die Menge X heißt* Definitionsbereich *von f, die Menge Y heißt* Wertebereich *von f.*

Statt $(x, y) \in f$ schreibt man gewöhnlich $f(x) = y$. Statt „$f \subseteq X \times Y$ ist eine Funktion" schreibt man gewöhnlich $f : X \to Y$.

1.1.2 Gerichtete Graphen

Zusammenhänge zwischen Elementen einer Menge werden in der Informatik häufig durch Graphen dargestellt. Wir werden die wichtigsten Begriffe hier einführen.

gerichteter Graph

Definition 1.2 *Ein* endlicher gerichteter Graph *wird spezifiziert durch ein Paar $G = (V, E)$. Hierbei gilt*

- *V ist eine endliche Menge. Die Elemente von V heißen die* Knoten *des Graphen.*
- $$E \subseteq V \times V = \{(u, v) \mid u, v \in V\} .$$

  Ein Element $(u, v) \in E$ heißt eine gerichtete Kante *von u nach v. Ist $(u, v) \in E$, so heißt u* direkter Vorgänger *von v und v heißt* direkter Nachfolger *von u.*

Kante, Vorgänger, Nachfolger

Da V endlich ist, ist notwendig auch E endlich. Wir betrachten bis auf weiteres weder unendliche Graphen noch ungerichtete Graphen. Wir schreiben deshalb statt „endlicher gerichteter Graph" meistens einfach „gerichteter Graph" oder „Graph". Man zeichnet gerichtete Graphen, indem man die Knoten $v \in V$ als Kreise oder Punkte mit Beschriftung v und gerichtete Kanten (u, v) als Pfeile von u nach v malt.

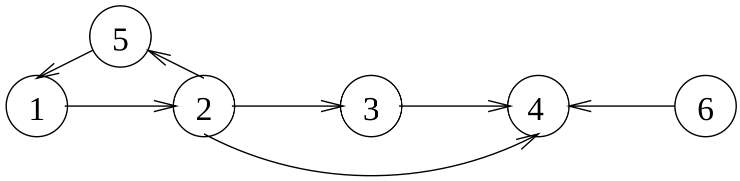


Abbildung 1.1: Beispiel eines Graphen

Definition 1.3 *Sei $G = (V, E)$ ein gerichteter Graph. Eine Folge von Kanten*

$$e_i = (v_i, w_i) \in E, i = 1, \ldots, l \; ,$$

heißt Pfad, *falls $v_{i+1} = w_i$ für $i = 1, \ldots, l-1$. Wir nennen l die* Länge *des Pfades. Gilt $w_l = v_1$, so heißt der Pfad* Zyklus. *Gerichtete Graphen, in denen es keine Zyklen gibt, heißen* zykelfrei.

Pfad
Zyklus

Definition 1.4 *Sei $G = (V, E)$ ein gerichteter Graph. Für einen Knoten $v \in V$ ist der* Ingrad *$indeg(v)$ und der* Outgrad *$outdeg(v)$ definiert als*

Ingrad
Outgrad

$$\begin{aligned} indeg(v) &= \#\{u \mid (u,v) \in E\} \\ outdeg(v) &= \#\{u \mid (v,u) \in E\} \; . \end{aligned}$$

Knoten v mit $indeg(v) = 0$ heißen Quellen *des Graphen. Knoten v mit $outdeg(v) = 0$ heißen* Senken *des Graphen. Für den gesamten Graphen definiert man*

Quelle
Senke

$$\begin{aligned} indeg(G) &= \max\{indeg(v) \mid v \in V\} \\ outdeg(G) &= \max\{outdeg(v) \mid v \in V\} \end{aligned}$$

Beispiel 1.1 *Abbildung 1.1 zeigt einen gerichteten Graphen mit sechs Knoten und sieben Kanten. Es gibt einen Pfad der Länge 3 und einen Pfad der Länge 2 von Knoten 1 nach Knoten 4, es gibt einen Zyklus der Länge 3 aus den Kanten $(1, 2)$, $(2, 5)$, $(5, 1)$. Der Ingrad von Knoten 1 ist 1, der Outgrad von Knoten 2 ist 3. Knoten 4 ist eine Senke, Knoten 6 eine Quelle.*

Definition 1.5 *Sei $G = (V, E)$ ein gerichteter Graph und $v \in V$. Die* Tiefe *$\mathrm{T}(v)$ von v wird definiert als die Länge eines längsten Pfades von einer Quelle zu v, falls ein solcher längster Pfad existiert. Andernfalls ist die Tiefe von v nicht definiert.*

Tiefe

Wir fassen einzelne Knoten auf als (unechte) Pfade der Länge 0. Damit können wir die obige Definition noch auf die Quellen gerichteter Graphen ausdehnen und ihre Tiefe als 0 definieren. Die Tiefe von Knoten, die auf einem Zyklus liegen, ist nicht definiert. Man kann zeigen, dass in einem zyklenfreien Graphen jeder Knoten eine Tiefe hat. Für zykelfreie gerichtete Graphen $G = (V, E)$ können wir nun die *Tiefe* $\mathrm{T}(G)$ des Graphen G definieren als

$$\mathrm{T}(G) = \max\{\mathrm{T}(v) \mid v \in V\} \; .$$

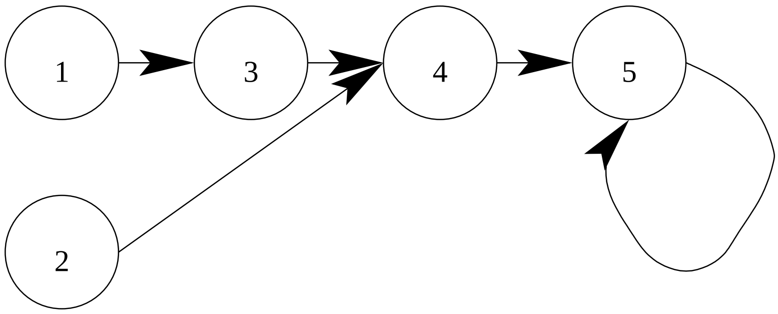


Abbildung 1.2: Zu analysierender Graph der Selbsttestaufg. 1.2

Beispiel 1.2 *In Abbildung 1.1 hat Knoten 6 als Quelle die Tiefe 0, und für Knoten 4 gilt* $T(4) = 1$*. Die anderen Knoten sind nicht von einer Quelle aus zu erreichen, ihre Tiefe ist nicht definiert.*

Selbsttestaufgabe 1.2 *Bestimmen Sie im Graphen aus Abbildung 1.2 die Quellen und die Senken. Bestimmen Sie für alle Knoten den Ingrad, den Outgrad und die Tiefe.*

Baum

Definition 1.6 *Ein* Baum *ist ein gerichteter zykelfreier Graph G für den gilt:*

1. *outdeg*$(G) = 1$ *und*
2. *G hat genau eine Senke.*

Blatt eines Baumes
Wurzel eines Baumes
binärer Baum
balancierter Baum

Die Quellen eines Baums nennt man Blätter*, die Senke des Baums nennt man die* Wurzel *des Baums. Alle Knoten die keine Blätter sind heißen* innere Knoten*. Ein* binärer Baum *ist ein Baum, in dem alle Knoten außer den Quellen Ingrad 2 haben. Ein Baum ist* balanciert *wenn die Pfade von allen Blättern zur Wurzel gleichlang sind.*

Wir weisen darauf hin, dass man einen Baum ebenfalls definieren kann, wenn man die Rollen von Quellen und Senken sowie von Ingrad und Outgrad vertauscht.

Beispiel 1.3 *Der Graph aus Abbildung 1.3(a) ist ein binärer Baum mit Blättern* 1 *und* 4 *bis* 6 *und Wurzel* 0*. Die Graphen aus Abbildung 1.3(b) und (c) sind keine Bäume. Bei (b) gibt es zwei Senken, bei (c) hat ein Knoten Outgrad 2.*

Man kann zeigen, dass ein balancierter binärer Baum, bei dem die Pfade von Blättern zur Wurzel alle Länge n haben, aus 2^n Blättern und $2^n - 1$ inneren Knoten besteht (siehe Kurseinheit 2).

1.2 Boole'sche Ausdrücke

Sei $n \in \mathbf{N}$. Wir interessieren uns für Schaltungen mit n Eingängen $X_1, \ldots, X_n$ und einem Ausgang Y. An jedem Eingang sowie am Ausgang sollen nur zwei

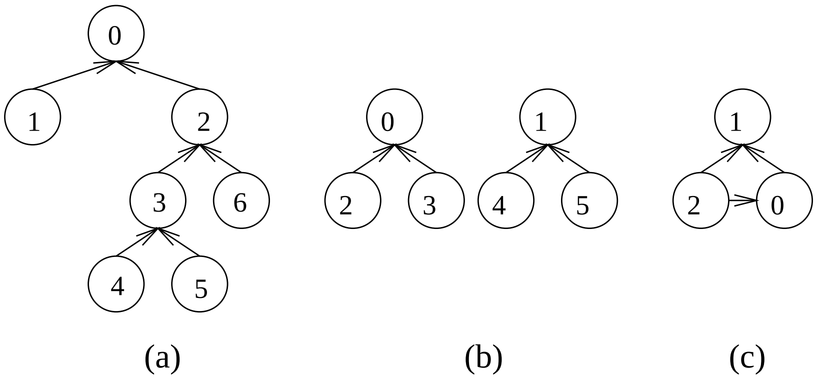


Abbildung 1.3: Beispiel und Gegenbeispiele für Bäume

Signale vorkommen können, die wir der Einfachheit halber mit 0 und 1 bezeichnen. Das Signal am Ausgang soll durch die Signale an den Eingängen eindeutig festgelegt sein. Das Ein–/Ausgabeverhalten einer solchen Schaltung lässt sich dann offensichtlich beschreiben durch eine Funktion $f : \{0,1\}^n \to \{0,1\}$, die jeder Kombination von Eingangssignalen das hierdurch festgelegte Ausgangssignal zuordnet. Dies motiviert die folgende Definition 1.7.

Definition 1.7 *Sei $n \in \mathbf{N}$. Eine n-stellige* Schaltfunktion *ist eine Abbildung $f : \{0,1\}^n \to \{0,1\}$.* Schaltfunktion

Lemma 1.2 *Es gibt 2^{2^n} n-stellige Schaltfunktionen.*

Beweis: Die Mächtigkeit des Definitionsbereichs $Z_2^n = \{0,1\}^n$ ist $m = 2^n$. An jeder Stelle des Definitionsbereichs kann einer von zwei möglichen Funktionswerten genommen werden. Damit gibt es insgesamt, $2^m = 2^{2^n}$ Möglichkeiten, die Funktion zu definieren. ∎

Zum Beispiel beträgt die Anzahl der 5-stelligen Schaltfunktionen $2^{2^5} = 2^{32} \approx 4$ Milliarden. Zum Schätzen der Größenordnung von Zweierpotenzen benutzt man den Zusammenhang $2^{10} = 1024 \approx 10^3 = 1000$, siehe auch Kurseinheit 2.

Um eine Schaltfunktion jemandem anderen mitteilen zu können, muss man sie auf irgendeine Art und Weise darstellen. Hierzu sind die folgenden Methoden gebräuchlich, wobei die Liste keinen Anspruch auf Vollständigkeit erhebt:

- Wertetabellen,
- Karnaugh-Diagramme,
- Boole'sche Ausdrücke,
- Schaltnetz-Zeichnungen,
- geordnete binäre Entscheidungsbäume (OBDDs).

Eine sehr allgemeine Art der Darstellung ist die *Wertetabelle*. Man schreibt in der linken Spalte alle Elemente des Definitionsbereichs untereinander, und in Wertetabelle

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

Tabelle 1.1: Wertetabelle einer 2-stelligen Schaltfunktion

	$X_1 = 1$	$X_1 = 1$	$X_1 = 0$	$X_1 = 0$	
$X_2 = 1$	1	1	1	0	$X_4 = 0$
$X_2 = 1$	1	1	1	1	$X_4 = 1$
$X_2 = 0$	1	1	1	0	$X_4 = 1$
$X_2 = 0$	0	1	0	0	$X_4 = 0$
	$X_3 = 0$	$X_3 = 1$	$X_3 = 1$	$X_3 = 0$	

$$f(X_1, X_2, X_3, X_4) = \begin{cases} 1 & \text{falls mindestens zwei der } X_i = 1, \\ 0 & \text{sonst} \end{cases}$$

Abbildung 1.4: Karnaugh-Diagramm für $n = 4$

der rechten Spalte schreibt man neben jedes Element des Definitionsbereichs den zugehörigen Funktionswert. Während Wertetabellen für beliebige Funktionen anwendbar sind, sind die weiteren Darstellungen auf Schaltfunktionen spezialisiert. Tabelle 1.1 zeigt die Wertetabelle einer 2-stelligen Schaltfunktion, die gerade dann den Wert 1 annimmt, wenn beide Variablen den Wert 1 haben. Diese Schaltfunktion wird uns in Kürze unter dem Namen Konjunktion oder UND-Verknüpfung wiederbegegnen.

Selbsttestaufgabe 1.3 *Bestimmen Sie die Anzahl der 2-stelligen Schaltfunktionen, und tragen Sie alle diese Funktionen in einer Wertetabelle zusammen. Welche dieser Schaltfunktionen können auch als 1-stellige Schaltfunktion des ersten Arguments betrachtet werden, da ihre Funktionswerte vom zweiten Argument unabhängig sind?*

Karnaugh-Diagramm

Ein *Karnaugh-Diagramm*, das auch Karnaugh-Veitch-Diagramm oder KV-Diagramm genannt wird, ist anwendbar für $n \leq 4$ Variablen, es ist ein Rechteck mit 2^n Feldern, und einer Markierung der Seiten mit Variablenwerten, so dass jedes Feld eineindeutig einem Wert des Definitionsbereichs zugeordnet werden kann. In jedes Feld schreibt man den zugehörigen Funktionswert. Abbildung 1.4 zeigt ein Beispiel eines Karnaugh-Diagramms für eine 4-stellige Schaltfunktion. Abbildung 1.5 zeigt eine übliche verkürzende Schreibweise für Karnaugh-Diagramme, bei der nur die Spalten bzw. Zeilen mit einer Variablen markiert werden, bei denen die Variable den Wert 1 hat. Welche Variable an welcher Seite steht, ist eigentlich egal. Allerdings muss garantiert sein, dass sich von jeder Zeile zur nächsten und von jeder Spalte zur nächsten jeweils genau ein Variablenwert ändert, da ansonsten die Eindeutigkeit nicht gelten kann.

X_1

X_2

1	1	1	0
1	1	1	1
1	1	1	0
0	1	0	0

X_4

X_3

Abbildung 1.5: Vereinfachtes Karnaugh-Diagramm für $n = 4$

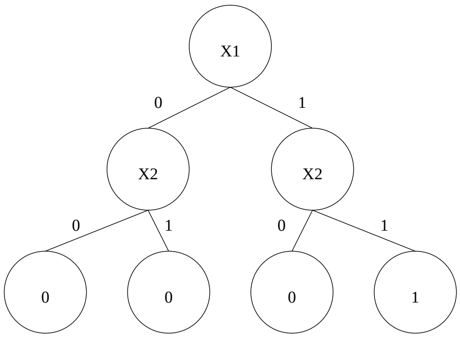


Abbildung 1.6: Geordneter binärer Entscheidungsbaum für eine 2-stellige Schaltfunktion

Selbsttestaufgabe 1.4 *Erstellen Sie ein Karnaugh-Diagramm für die Schaltfunktion aus Tabelle 1.1.*

Mit Boole'schen Ausdrücken wollen wir uns im Folgenden beschäftigen, so dass wir hier auf eine Erklärung verzichten. Gleiches gilt für Schaltnetz-Zeichnungen, die wir in Kurseinheit 2 besprechen werden.

Die letzte Darstellung die wir kurz ansprechen wollen ist eine Darstellung als spezieller Graph, nämlich als *geordneter binärer Entscheidungsbaum* (ordered binary decision diagram, OBDD). Hierbei handelt es sich um einen balancierten binären Baum, d.h. um einen Baum, der von der Wurzel zu den Blättern hin gerichtet ist, bei dem jeder innere Knoten 2 Kinder hat, und bei denen alle Blätter die gleiche Tiefe haben. Alle inneren Knoten der gleichen Tiefe sind mit der gleichen Variable, und die je zwei Kanten die einen Knoten verlassen mit 0 und 1 markiert. Die Blätter sind mit den Funktionswerten markiert. Möchte man den Funktionswert an der Stelle $(a_1, \ldots, a_n)$ herausfinden, so startet man an der Wurzel, und bei jedem inneren Knoten folgt man der linken, mit 0 markierten Kante, falls die Variable X_i, mit der Knoten markiert ist, den Wert $a_i = 0$ hat, sonst folgt man der rechten Kante.

Beispiel 1.4 *Abbildung 1.6 zeigt ein OBDD für die Schaltfunktion aus Tabelle 1.1. Um den Funktionswert an der Stelle $X_1X_2 = 01$ festzustellen, folgt man in der Wurzel der linken Kante, da die Wurzel mit X_1 markiert ist und $X_1 = 0$. In dem linken Knoten der Tiefe 1 folgt man der rechten Kante, da er mit X_2 markiert ist und $X_2 = 1$. Der Funktionswert beträgt 0.*

x_1	x_2	$x_1 \wedge x_2$	$x_1 \vee x_2$
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	1

x_1	$\sim x_1$
0	1
1	0

Tabelle 1.2: Wertetabellen von Konjunktion, Disjunktion und Negation

partiell definiert

Man kann eine Schaltfunktion auch nur auf einer Teilmenge von $\{0,1\}^n$ definieren, man nennt sie dann *partiell definiert*. In diesem Fall lässt man die betreffenden Zeilen in der Wertetabelle weg; im Karnaugh-Diagramm markiert man die Felder, die nicht zum Definitionsbereich gehören mit dem Symbol X.

Konjunktion

Disjunktion

Negation

Spezielle Schaltfunktionen sind die *Konjunktion* $\wedge : \{0,1\}^2 \to \{0,1\}$, die *Disjunktion* $\vee : \{0,1\}^2 \to \{0,1\}$ und die *Negation* $\sim : \{0,1\} \to \{0,1\}$. Ihre Wertetabellen sind in Tabelle 1.2 angegeben.

Gatter

Offensichtlich gilt $\wedge(X_1, X_2) = X_1 \wedge X_2 = 1 \Leftrightarrow X_1 = 1$**und** $X_2 = 1$. Eine Schaltung mit diesem Ein-Ausgabeverhalten heißt deshalb AND-*Gatter*. Weiter gilt $\vee(X_1, X_2) = X_1 \vee X_2 = 1 \Leftrightarrow X_1 = 1$**oder** $X_2 = 1$. Eine Schaltung mit diesem Ein-Ausgabeverhalten heißt deshalb OR-*Gatter*. Schließlich gilt $\sim X_1 = 1 \Leftrightarrow X_1 \neq 1$. Eine Schaltung mit diesem Ein-Ausgabeverhalten heißt deshalb NOT-*Gatter* oder auch *Inverter*. Diese Gatter kann man in Halbleiterschaltungen leicht realisieren. Man benutzt sie als Bausteine zur Konstruktion von komplizierteren Schaltungen (s. Kurseinheit 2).

Inverter

Sei nun $V = \{X_1, \ldots, X_n\}$ eine im Folgenden feste Menge von Variablen. Gelegentlich werden wir diese Variablen zu einem Vektor $X = (X_1, \ldots, X_n)$ zusammenfassen. Wir werden im Rest dieses Abschnitts folgendes tun:

Operatorensystem

1. Wir definieren mit Hilfe von $\wedge, \vee$ und $\sim$ die Menge der Boole'schen Ausdrücke mit Variablen in V. Die Menge $\{\wedge, \vee, \sim\}$ heißt auch *Operatorensystem*.

2. Wir ordnen jedem solchen Ausdruck e eine Schaltfunktion $\|e\| : \{0,1\}^n \to \{0,1\}$ zu. Die Funktion $\|e\|$ heißt die durch Ausdruck e *berechnete Funktion*.

3. Wir geben Regeln für das Rechnen und Lösen von Gleichungen mit Boole'schen Ausdrücken an.

4. Wir zeigen, daß es zu jeder n-stelligen Schaltfunktion f einen Boole'schen Ausdruck e gibt mit $f = \|e\|$. Das gewählte Operatorensystem ist also *vollständig*.

Damit haben wir zweierlei erreicht. Zum einen besitzen wir Schaltfunktionen als *Spezifikation* des Ein–/Ausgabeverhaltens von Schaltnetzen. Zum anderen besitzen wir Boole'sche Ausdrücke, die als Beschreibungen oder Realisierungen einer Schaltfunktion dienen können, da sie alle Schaltfunktionen beschreiben können. Da es zu jeder Schaltfunktion mehrere Boole'sche Ausdrücke gibt, wird man je nach Einsatzzweck den einen oder anderen nehmen.

Dies entspricht der Vorgehensweise in der Software-Entwicklung, wo man zunächst die Aufruf-Semantik einer Prozedur festlegt, und erst später entscheidet, wie die Prozedur intern realisiert wird, d.h. ob ein langsamer aber speicherplatzsparender Algorithmus verwendet wird oder ein schnellerer Algorithmus, der aber mehr Speicherplatz verbraucht.

1.2.1 Vollständig geklammerte Ausdrücke

Sei

$$A = \{0, 1, \wedge, \vee, \sim, X_1, \ldots, X_n, (,)\} \ .$$

Die Menge B der *vollständig geklammerten Boole'schen Ausdrücke* ist eine Menge von Zeichenreihen in A^*. Sie wird auf folgende Weise induktiv definiert: Boole'scher Ausdruck

Definition 1.8 *Es ist $B_1 = \{0, 1, X_1, \ldots, X_n\}$. Sei $i \in \mathbf{N}$, und seien $a, b \in B_i$. Dann liegen folgende Zeichenreichen in B_{i+1}:*

1. a
2. $(\sim a)$
3. $(a \vee b)$
4. $(a \wedge b)$

Eine Zeichenreihe $z \in A^+$ liegt genau dann in B, wenn z in einer der Mengen B_i liegt, d.h. $B = \cup_{i \in \mathbf{N}} B_i$.

Beispiel 1.5 *Der Ausdruck $((0 \vee (\sim 1)) \wedge (X_2 \vee (\sim(\sim X_{52}))))$ ist ein vollständig geklammerter Boole'scher Ausdruck, denn es gilt*

$$\begin{array}{rcl} 0, 1, X_2, X_{52} & \in & B_1, \\ (\sim 1), (\sim X_{52}) & \in & B_2, \\ (0 \vee (\sim 1)), (\sim(\sim X_{52})) & \in & B_3, \\ (X_2 \vee (\sim(\sim X_{52}))) & \in & B_4 \quad \textit{und} \\ ((0 \vee (\sim 1)) \wedge (X_2 \vee (\sim(\sim X_{52})))) & \in & B_5. \end{array}$$

Die obige Definition hat für das praktische Rechnen einen entscheidenden Schönheitsfehler: wir rechnen nicht allein mit Boole'schen Ausdrücken, die nur mit den drei Funktionen $\wedge$, $\vee$ und $\sim$ gebildet sind. Vielmehr definieren wir in vielfältiger Weise neue Funktionen f und bilden dann mit Hilfe dieser Funktionen Ausdrücke wie zum Beispiel

$$X_1 \wedge f(X_2, 0, (X_1 \vee X_3)) \ .$$

Deshalb definieren wir nun die Menge EB der *erweiterten Boole'schen Ausdrücke*. Hierfür sei $F = \{f_1, f_2, \ldots\}$ eine abzählbare Menge von Funktionsnamen für Schaltfunktionen. Mit Hilfe der Funktion $s : F \to \mathbf{N}_0$ ordnen wir jeder Funktion $f \in F$ eine *Stelligkeit* zu, die einfach die Anzahl der Argumente von Funktion f angibt. Das unendliche Alphabet A wird definiert durch erweiterter Boole'scher Ausdruck Stelligkeit

$$A = V \cup F \cup \{0, 1, \wedge, \vee, \sim, (,), , \} \ .$$

Es enthält insbesondere das Komma. Die Menge EB der vollständig geklammerten *erweiterten Boole'schen Ausdrücke* wird nun induktiv definiert.

Definition 1.9 *Es ist* $\mathrm{EB}_1 = \{0,1\} \cup V$. *Sei* $i \in \mathbf{N}$, *und seien* $e_1, e_2, \ldots \in \mathrm{EB}_i$. *Dann liegen folgende Zeichenreihen in* EB_{i+1}.

1. e_1
2. $(\sim e_1)$
3. $(e_1 \wedge e_2)$
4. $(e_1 \vee e_2)$
5. $f(e_1, \ldots, e_{s(f)})$ *für alle* $f \in F$.

Man beachte, dass ein Ausdruck, der sich in EB_i befindet, sich auch wegen der ersten Regel in $\mathrm{EB}_{i+1}, \mathrm{EB}_{i+2}, \ldots$ befindet. Daraus folgt, dass die verschiedenen EB_i nicht disjunkt sind. In der Praxis bedeutet es, dass man, um zu zeigen, dass zum Beispiel $(\sim e_1) \in \mathrm{EB}_{i+1}$ ist, nur zeigen muss, dass $e_1 \in \mathrm{EB}_j$ für irgendein $j \leq i$.

Beispiel 1.6 *Sei* $s(f) = 3$, *d.h.* f *bezeichnet eine 3-stellige Schaltfunktion. Dann ist*

- $X_1, X_2, X_3, 0 \in \mathrm{EB}_1$
- $(X_1 \vee X_3) \in \mathrm{EB}_2$
- $f(X_2, 0, (X_1 \vee X_3)) \in \mathrm{EB}_3$
- $(X_1 \wedge f(X_2, 0, (X_1 \vee X_3))) \in \mathrm{EB}_4$.

Für den späteren Gebrauch verabreden wir noch die Abkürzung $f(X)$ für $f(X_1, \ldots, X_{s(f)})$.

Selbsttestaufgabe 1.5 *Zeigen Sie, dass* $(((X_1 \wedge X_2) \wedge X_3) \vee f_1(X_1, X_2))$ *in* EB *liegt, wenn* $s(f_1) = 2$. *Wie müsste man den Ausdruck* $f_1(X_1, X_2) \vee (X_1 \wedge X_2 \vee X_3))$ *ergänzen, damit er in* EB *liegt?*

1.2.2 Einsetzungen

Jeder Boole'sche Ausdruck $e \in B$ kann als Vorschrift zur Berechnung einer Funktion $\|e\| : \{0,1\}^n \to \{0,1\}$ aufgefaßt werden: für $a = (a_1, \ldots, a_n) \in \{0,1\}^n$ berechnet man den Wert der Funktion $\|e\|$ an der Stelle a, indem man für alle i die Konstante a_i für die Variable X_i einsetzt und dann auf die übliche Art auswertet. Die folgenden Definitionen formalisieren diese Vorgehensweise. Das mag manchem Leser überflüssig erscheinen, aber ohne präzise Definitionen kann man eben nichts beweisen.

Einsetzung

Definition 1.10 *Eine* Einsetzung *ist eine Abbildung* $\phi : V \to \{0,1\}$.

Für alle i ist $\phi(X_i) \in \{0,1\}$ gerade die Konstante, die für die Variable X_i eingesetzt werden soll. Durch eine Einsetzung ϕ ist bereits für *jeden* Boole'schen Ausdruck e der *Wert von e an der Stelle* $(\phi(X_1),\dots,\phi(X_n))$ festgelegt. Wir nennen diesen Wert $\phi(e)$ und definieren ihn formal, indem wir induktiv die Funktion ϕ von $V \subseteq \mathrm{EB}$ auf die ganze Menge EB ausdehnen. Wir definieren $\phi(0)=0$ und $\phi(1)=1$. Damit ist ϕ auf EB_1 erklärt.

Definition 1.11 *Seien $e_1, e_2, \ldots \in \mathrm{EB}$. Wir definieren*

$$\begin{aligned}
\phi(\sim e_1) &= \sim\phi(e_1),\\
\phi(e_1 \wedge e_2) &= \phi(e_1) \wedge \phi(e_2),\\
\phi(e_1 \vee e_2) &= \phi(e_1) \vee \phi(e_2),\\
\phi(f_i(e_1,\ldots,e_{s(f_i)})) &= f_i(\phi(e_1),\ldots,\phi(e_{s(f_i)}))\ \textit{für alle } i \in \mathbf{N}\,.
\end{aligned}$$

Man beachte, daß wir hier die Bezeichner $\wedge$, $\vee$ und $\sim$ sowie f_i in zweifacher Weise verwendet haben. Auf der linken Seite stellen sie ein Zeichen in einem Boole'schen Ausdruck dar, auf der rechten Seite sind sie eine Aufforderung zum Auswerten von Funktionen.

Für die Funktionen '$\wedge$','$\vee$' und '$\sim$' enthält Tabelle 1.2 die Regeln zum Auswerten. Für irgendwelche weiteren Funktionen f_i muß man zuerst Auswertungsvorschriften festlegen, bevor man die obige Definition konkret anwenden kann.

Beispiel 1.7 *Sei ϕ eine Einsetzung mit $\phi(X_1)=1$, $\phi(X_2)=0$ und $\phi(X_3)=1$. Für den Ausdruck $((X_1 \wedge X_2) \vee X_3)$ gilt dann $\phi((X_1\wedge X_2)\vee X_3) = \phi(X_1\wedge X_2)\vee \phi(X_3)$. Für den ersten Teilausdruck ergibt sich $\phi(X_1 \wedge X_2) = \phi(X_1)\wedge\phi(X_2) = 1\wedge 0 = 0$. Damit folgt $\phi((X_1\wedge X_2)\vee X_3) = 0 \vee 1 = 1$.*

Beispiel 1.8 *Die 3-stellige Schaltfunktion f sei durch die Funktionstabelle 1.3 definiert. Wie im vorigen Beispiel sei ϕ eine Einsetzung mit $\phi(X_1)=1$, $\phi(X_2)=0$ und $\phi(X_3)=1$. Für den Ausdruck $X_1 \wedge f(X_2, 0, (X_1 \vee X_3))$ gilt dann*

$$\phi(X_1 \wedge f(X_2,0,(X_1\vee X_3))) = \phi(X_1) \wedge \phi(f(X_2,0,(X_1\vee X_3)))\,.$$

Für den zweiten Teilausdruck gilt

$$\phi(f(X_2,0,(X_1\vee X_3))) = f(\phi(X_2),\phi(0),\phi(X_1\vee X_3))\,.$$

Da $\phi(X_1\vee X_3) = \phi(X_1)\vee\phi(X_3) = 1\vee 1 = 1$, gilt für den zweiten Teilausdruck $\phi(f(X_2,0,(X_1\vee X_3))) = f(0,0,1) = 0$. Damit gilt für den gesamten Ausdruck

$$\phi(X_1 \wedge f(X_2,0,(X_1\vee X_3))) = 1 \wedge 0 = 0\,.$$

Ein subtiler Punkt ist an dieser Stelle die Tatsache, daß durch Definition 1.11 jedem Ausdruck e ein und *nur* ein Wert $\phi(e)$ zugewiesen wird. Hierfür muß man zeigen, daß es zu jedem vollständig geklammerten Ausdruck eine und *nur* eine Zerlegung in Teilausdrücke gibt, auf die man Definition 1.11 anwenden kann. Das ist im Wesentlichen der Inhalt des Zerlegungssatzes, den wir hier nicht ausführen wollen.

a_1	a_2	a_3	$f(a_1, a_2, a_3)$
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

Tabelle 1.3: Wertetabelle der Funktion f aus Beispiel 1.8

Selbsttestaufgabe 1.6 *Bestimmen Sie den Wert des Ausdrucks* $(((X_1 \wedge X_2) \wedge X_3) \vee f_1(X_1, X_2))$ *an den Stellen* $a = (1, 1, 1)$ *und* $b = (1, 0, 1)$*, d.h. für die Einsetzungen* ϕ_a *und* ϕ_b *mit* $\phi_a(X_1) = \phi_a(X_2) = \phi_a(X_3) = 1$ *und* $\phi_b(X_1) = \phi_B(X_3) = 1$*,* $\phi_b(X_2) = 0$*. Hierbei soll* $f_1(10) = 1$ *und* $f_1(00) = f_1(01) = f_1(11) = 0$ *gelten.*

1.2.3 Identitäten und Ungleichungen

Ausdrücke kann man nicht nur auswerten, man kann auch mit ihnen rechnen. Dabei verfolgt man meistens eine der zwei folgenden Aktivitäten:

1. man formt Ausdrücke äquivalent um oder

2. man löst Gleichungen.

Definition 1.12 *Es seien* $e_1, e_2 \in \mathrm{EB}$ *erweiterte Boole'sche Ausdrücke. Es gilt* $e_1 \equiv e_2$ *genau dann, wenn* $\phi(e_1) = \phi(e_2)$ *für alle Einsetzungen* ϕ *gilt.*

Identität

Für $e_1 \equiv e_2$ sagt man auch „e_1 und e_2 sind *äquivalent*“, und man nennt die Zeichenreihe „$e_1 \equiv e_2$“ eine *Identität.*

Beim konkreten Rechnen schreibt man häufig statt '$e_1 \equiv e_2$' einfach '$e_1 = e_2$' und sagt '$e_1 = e_2$ gilt identisch' oder noch einfacher 'e_1 gleich e_2'. Dabei mißhandelt man strikt gesprochen das Gleichheitszeichen, dann man setzt Ausdrücke einander gleich, die als Zeichenreihen betrachtet in der Regel *nicht* gleich sind. Wir werden jedoch gelegentlich die Schreibweise '$e_1 \equiv e_2$' verwenden.

Satz 1.3 *Sei* $e \in \mathrm{EB}$ *ein erweiterter Boole'scher Ausdruck. Dann gibt es genau eine* n*-stellige Schaltfunktion* f *so daß* $f(X) \equiv e$ *gilt.*

Die Funktion f mit $f(X) \equiv e$ heißt die durch Ausdruck e *berechnete Funktion.*

Beweis: Wir definieren zuerst die Funktion f. Für $a = (a_1, \ldots, a_n) \in \{0, 1\}^n$ sei $\phi_a : V \to \{0, 1\}$ die Einsetzung mit $\phi_a(X_i) = a_i$ für alle i. Damit $f(X) \equiv e$

gilt, muß $\phi(f(X)) = \phi(e)$ für alle Einsetzungen ϕ gelten, also insbesondere für $\phi = \phi_a$. Es folgt

$$\phi_a(e) = \phi_a(f(X)) = f(\phi_a(X_1), \ldots, \phi_a(X_n)) = f(a) \; .$$

Also ist f eindeutig bestimmt, und um den Wert der Funktion f an der Stelle a zu berechnen muß man einfach:

- für jede Variable X_i die Konstante a_i einsetzen (ϕ_a bilden) und dann
- auswerten ($\phi_a(e)$ bilden).

Sei nun $\phi : V \to \{0,1\}$ eine beliebige Einsetzung. Dann ist für $a = (\phi(X_1), \ldots, \phi(X_n))$ auch $\phi = \phi_a$. Es folgt

$$\phi(e) = \phi_a(e) = f(a) = \phi_a(f(X)) = \phi(f(X)) \; ,$$

also gilt $e \equiv f$. ■

Beispiele für Identitäten liefert der folgende

Satz 1.4

(B1)	$(X_1 \wedge X_2) \equiv (X_2 \wedge X_1)$	*Kommutativität*
	$(X_1 \vee X_2) \equiv (X_2 \vee X_1)$	
(B2)	$((X_1 \vee X_2) \vee X_3) \equiv (X_1 \vee (X_2 \vee X_3))$	*Assoziativität*
	$((X_1 \wedge X_2) \wedge X_3) \equiv (X_1 \wedge (X_2 \wedge X_3))$	
(B3)	$(X_1 \wedge (X_2 \vee X_3)) \equiv ((X_1 \wedge X_2) \vee (X_1 \wedge X_3))$	*Distributivität*
	$(X_1 \vee (X_2 \wedge X_3)) \equiv ((X_1 \vee X_2) \wedge (X_1 \vee X_3))$	
(B4)	$(X_1 \vee (X_1 \wedge X_2)) \equiv X_1$	
	$(X_1 \wedge (X_1 \vee X_2)) \equiv X_1$	
(B5)	$(X_1 \vee (X_2 \wedge (\sim X_2))) \equiv X_1$	
	$(X_1 \wedge (X_2 \vee (\sim X_2))) \equiv X_1$	
(B6)	$(X_1 \vee (\sim X_1)) \equiv 1$	
	$(X_1 \wedge (\sim X_1)) \equiv 0$	
(B7)	$(X_1 \vee 1) \equiv 1$	
	$(X_1 \vee 0) \equiv X_1$	
	$(X_1 \wedge 1) \equiv X_1$	
	$(X_1 \wedge 0) \equiv 0$	
(B8)	$(\sim(X_1 \vee X_2)) \equiv ((\sim X_1) \wedge (\sim X_2))$	*Morgan-Formeln*
	$(\sim(X_1 \wedge X_2)) \equiv ((\sim X_1) \vee (\sim X_2))$	
(B9)	$(\sim(\sim X_1)) \equiv X_1$	
(B10)	$(X_1 \vee X_1) \equiv X_1$	
	$(X_1 \wedge X_1) \equiv X_1$	

Man kann Satz 1.4 beweisen, indem man für jede der Identitäten ganz stur die höchstens acht verschiedenen Belegungen der vorkommenden Variablen aufzählt und für jede der Belegungen den Wert beider Seiten der Identitäten auswertet. Das kann man ganz schematisch in Tabellenform tun. Für die erste der Identitäten (B6) haben wir das in Tabelle 1.4 ausgeführt. Mit Hilfe der Identitäten aus Satz 1.4 kann man bis auf die vielen Klammern schon fast in gewohnter Weise rechnen. Mit Hilfe von (B3) kann man beispielsweise rechnen:

$\phi(e_1)$	$\phi(\sim e_1)$	$\phi((e_1 \vee (\sim e_1)))$
0	1	1
1	0	1

Tabelle 1.4: Beweis von Identität (B6)

$$(X_7 \vee (X_1 \wedge (X_4 \wedge (1 \vee X_2)))) = (X_7 \vee (X_1 \wedge ((X_4 \wedge 1) \vee (X_4 \wedge X_2)))).$$

Hierbei haben wir zwei Dinge getan, nämlich:

1. Wir haben in (B3) die Variablen umbenannt und teilweise durch Konstanten ersetzt und

2. wir haben in einem Ausdruck einen Teilausdruck durch einen äquivalenten Ausdruck ersetzt.

In der Schule wurde beim Rechnen mit arithmetischen Ausdrücken die Regel 'Punktrechnung geht vor Strichrechnung' vereinbart. Der einzige Sinn dieser Regel ist das Sparen von Schreibarbeit, da man Ausdrücke nun nicht mehr vollständig klammern muß. Für Boole'sche Ausdrücke verabreden wir die Regeln

- $\sim$ bindet stärker als $\wedge$ und

- $\wedge$ bindet stärker als $\vee$.

Insbesondere behandeln wir also '$\vee$' wie '+' (Strichrechnung) und '$\wedge$' wie '$\cdot$'. (Punktrechnung). Nun können wir in gewohnter Weise Klammern weglassen.

Beispiel 1.9 $X_1 \vee \sim X_2 \wedge X_3 \vee X_4$ *ist Abkürzung für* $((X_1 \vee ((\sim X_2) \wedge X_3)) \vee X_4)$.

unvollständig geklammerter Ausdruck

Die *unvollständig geklammerten Ausdrücke*, die durch das Weglassen von Klammern entstehen, sind nichts weiter als Abkürzungen für die ursprünglichen — hoffentlich eindeutig rekonstruierbaren — vollständig geklammerten Ausdrücke. Eine strenge Beschreibung und Rechtfertigung dieses Vorgehens ist mit erheblichem Aufwand verbunden. Der interessierte Leser findet die entsprechenden Konstruktionen und Sätze zum Beispiel in Kapitel 1 von Keller/Paul: Hardware Design.

Wir vereinfachen die Schreibweise noch weiter. Ist e ein erweiterter Boole'scher Ausdruck, so schreibt man statt $\sim e$ oft $\bar{e}$.

Beispiel 1.10 *Statt* $\sim(X_1 \wedge X_2)$ *schreibt man oft* $\overline{X_1 \wedge X_2}$.

In arithmetischen Ausdrücken läßt man oft das Multiplikationszeichen '$\cdot$' weg. Ebenso läßt man in Boole'schen Ausdrücken oft das '$\wedge$' weg.

Beispiel 1.11 *Statt* $X_1 \wedge X_2 \wedge \overline{X_3}$ *schreibt man oft* $X_1 X_2 \overline{X_3}$.

Selbsttestaufgabe 1.7 *Nutzen Sie Regel (B3), um den Ausdruck $X_1(X_2 \vee X_3)$ in einen unvollständig geklammerten Ausdruck zu transformieren, in dem bei Beachtung der Punkt-vor-Strich-Regel überhaupt keine Klammern mehr notwendig sind. Nutzen Sie die Regeln (B6) und (B7), um den Ausdruck X_1X_3 so zu transformieren, dass auch die Variable X_2 vorkommt, und keine Klammern notwendig sind.*

Aus den Morgan-Formeln von Satz 1.4 kann man durch Induktion direkt die *allgemeinen Morgan-Formeln*

allgemeine Morgan–Formeln

$$\begin{aligned} \overline{X_1 \vee \cdots \vee X_n} &\equiv \overline{X_1} \wedge \cdots \wedge \overline{X_n} \\ \overline{X_1 \wedge \cdots \wedge X_n} &\equiv \overline{X_1} \vee \cdots \vee \overline{X_n} \end{aligned} \tag{1.3}$$

herleiten. Außerdem folgen aus Regeln (B3) und (B6) von Satz 1.4 direkt die sogenannten *Resolutionsregeln*

$$\begin{aligned} X_1X_3 \vee X_2\overline{X_3} &\equiv X_1X_3 \vee X_2\overline{X_3} \vee X_1X_2 \\ (X_1 \vee X_3)(X_2 \vee \overline{X_3}) &\equiv (X_1 \vee X_3)(X_2 \vee \overline{X_3})(X_1 \vee X_2) \end{aligned} \tag{1.4}$$

In Analogie zur Summennotation von arithmetischen Ausdrücken verabreden wir für erweiterte Boole'sche Ausdrücke $e_1, \ldots, e_m$ die Schreibweisen

$$\begin{aligned} \bigwedge_{i=1}^{m} e_i &= e_1 \wedge \ldots \wedge e_m, \\ \bigvee_{i=1}^{m} e_i &= e_1 \vee \ldots \vee e_m \, . \end{aligned}$$

Für den Sonderfall, daß man das UND bzw. ODER von einer leeren Menge von Ausdrücken bildet, verabreden wir

$$\bigwedge_{i\in\emptyset} e_i = 1 \quad \text{und} \quad \bigvee_{i\in\emptyset} e_i = 0 \, . \tag{1.5}$$

Definition 1.13 *Es seien e_1 und e_2 erweiterte Boole'sche Ausdrücke. Es gilt $e_1 \leq e_2$ genau dann, wenn $\phi(e_1) \leq \phi(e_2)$ für alle Einsetzungen ϕ gilt.*

Aus den Definitionen schließt man unmittelbar für Ausdrücke $a, a', b, b' \in$ EB:

1. aus $a \leq b$ und $a' \leq b$ folgt $a \vee a' \leq b$ und
2. aus $a \leq b$ und $a' \leq b'$ folgt $a \vee a' \leq b \vee b'$.

Beispiel 1.12 *Es ist $\wedge_{i=1}^{3} X_i = X_1X_2X_3$. Durch Betrachten der Funktionstabelle 1.2 erkennt man, dass $X_1 \wedge X_2 \leq X_1 \vee X_2$ gilt.*

1.2.4 Lösen von Gleichungen

Das Gleichheitszeichen zwischen verschiedenen Ausdrücken e_1 und e_2 kommt außer beim äquivalenten Umformen noch in einem ganz anderen Zusammenhang vor, nämlich beim Lösen von Gleichungen.

Gleichung

Definition 1.14 *Eine* Gleichung *ist eine Zeichenreihe der Form '$e_1 = e_2$', wobei e_1 und e_2 beliebige Ausdrücke sein dürfen. Man* löst *eine Gleichung, indem man alle Einsetzungen $\phi : V \to \{0,1\}$ bestimmt, so daß $\phi(e_1) = \phi(e_2)$ gilt.*

Beispiel 1.13 *Die Gleichung $X_1\overline{X_2} \vee \overline{X_1}X_2 = 1$ hat zwei Lösungen, nämlich*

1. $\phi(X_1) = 1, \phi(X_2) = 0$ *und*
2. $\phi(X_1) = 0, \phi(X_2) = 1$.

Wir leiten einige Regeln zum Lösen von Gleichungen her. Es seien $e_1, \ldots, e_n$ vollständig geklammerte Boole'sche Ausdrücke und es sei ϕ eine Einsetzung. Aus Definition 1.11 und Tabelle 1.2 folgt direkt:

$$\begin{aligned} \phi((e_1 \wedge e_2)) = 1 &\Leftrightarrow \phi(e_1) \wedge \phi(e_2) = 1 \\ &\Leftrightarrow \phi(e_1) = 1 \text{ und } \phi(e_2) = 1 \;. \end{aligned}$$

Durch Induktion über n folgt:

$$\phi((e_1 \wedge \ldots \wedge e_n)) = 1 \Leftrightarrow \phi(e_i) = 1 \text{ für alle } i \in \{1, \ldots, n\}$$

Dem Leser wird auffallen, daß man eine Menge Schreibarbeit sparen kann, wenn man beim Gleichungslösen die ϕ's einfach wegfallen läßt. Aus dem Zusammenhang des Gleichungslösens geht dann hervor, daß man statt den Ausdrücken e in Wirklichkeit die Werte $\phi(e)$ meint. Das ist in der Tat gängige Praxis, der wir auch folgen werden. Nur bei ganz seltenen Anlässen muß man sich daran erinnern, daß man diese Vereinfachung vorgenommen hat. Insbesondere hätte man oben ohne Bezugnahme auf ϕ *nicht* folgern können:

$$(e_1 \wedge e_2) = 1 \Leftrightarrow e_1 = 1 \text{ und } e_2 = 1 \;.$$

Nach dem gleichen Muster beweist man das folgende Lemma. Es ist in der vereinfachten Form formuliert, aber für den Induktionsanfang der Beweise muß die Vereinfachung rückgängig gemacht werden.

Lemma 1.5 *Seien $e_1, \ldots, e_n$ vollständig geklammerte Boole'sche Ausdrücke. Dann gilt:*

1. $e_1 \wedge \ldots \wedge e_n = 1 \Leftrightarrow e_i = 1$ *für alle* $i \in \{1, \ldots, n\}$
2. $e_1 \wedge \ldots \wedge e_n = 0 \Leftrightarrow e_i = 0$ *für (mindestens) ein* $i \in \{1, \ldots, n\}$
3. $e_1 \vee \ldots \vee e_n = 1 \Leftrightarrow e_i = 1$ *für ein* $i \in \{1, \ldots, n\}$
4. $e_1 \vee \ldots \vee e_n = 0 \Leftrightarrow e_i = 0$ *für alle* $i \in \{1, \ldots, n\}$
5. $\overline{e_1} = 1 \Leftrightarrow e_1 = 0$

Beispiel 1.14 *Für den Ausdruck $X_1(X_2 \vee X_3)$ finden wir alle Einsetzungen, bei denen der Ausdruck den Wert 1 hat, d.h. wir lösen die Gleichung $X_1(X_2 \vee X_3) = 1$. Zumindest muss für jede solche Einsetzung $\phi(X_1) = 1$ gelten, denn ansonsten gilt wegen der zweiten Regel von Lemma 1.5, dass der Ausdruck den Wert 0 hat. Auch der Ausdruck in der Klammer muss den Wert 1 haben, was wegen der dritten Regel des Lemmas dann der Fall ist, wenn mindestens eine der Variablen X_2 und X_3 den Wert 1 hat. Also gibt es drei verschiedene solcher Einsetzungen:*
$\phi_1(X_1) = 1,\ \phi_1(X_2) = 1,\ \phi_1(X_3) = 0,$
$\phi_2(X_1) = 1,\ \phi_2(X_2) = 0,\ \phi_2(X_3) = 1,$
$\phi_3(X_1) = 1,\ \phi_3(X_2) = 1,\ \phi_3(X_3) = 1.$

Selbsttestaufgabe 1.8 *Zeigen Sie, dass der Ausdruck $(\bar{X}_1 \vee \bar{X}_2) \wedge X_1 X_2$ unter keiner Einsetzung den Wert 1 annehmen kann.*

1.2.5 Der Darstellungssatz

Der zentrale Satz dieses Abschnitts läßt sich nun sehr leicht herleiten. Für Variablen $X_i \in V$ und $\epsilon \in \{0,1\}$ verabreden wir die Schreibweise

$$X_i^\epsilon = \begin{cases} \overline{X_i} \text{ falls } \epsilon = 0 \\ X_i \text{ falls } \epsilon = 1\,. \end{cases}$$

Offensichtlich gilt

$$X_i^\epsilon = 1 \Leftrightarrow X_i = \epsilon\,.$$

Boole'sche Ausdrücke der Form X_i^ϵ nennt man *Literale*. Literal

Definition 1.15 *Für $a = (a_1, \ldots, a_n) \in \{0,1\}^n$ definieren wir die Boole'schen Ausdrücke $m(a)$ und $c(a)$ durch*

$$\begin{aligned} m(a) &= \bigwedge_{i=1}^{n} X_i^{a_i}\,, \\ c(a) &= \bigvee_{i=1}^{n} X_i^{\bar{a}_i}\,. \end{aligned}$$

Der Ausdruck $m(a)$ heißt der zu a gehörige Minterm *und $c(a)$ der zu a gehörige* Maxterm*.* Minterm Maxterm

Beispiel 1.15 *Es ist $m(0,1,0) = \overline{X_1} X_2 \overline{X_3}$ und $c(0,1,0) = X_1 \vee \overline{X_2} \vee X_3$.*

Aus Lemma 1.5 folgt

$$\begin{aligned} m(a) = \bigwedge_{i=1}^{n} X_i^{a_i} = 1 \quad &\Leftrightarrow \quad X_i^{a_i} = 1 \text{ für alle } i \in \{1, \ldots, n\} \\ &\Leftrightarrow \quad X = (X_1, \ldots, X_n) = a & (1.6) \\ c(a) = 0 \quad &\Leftrightarrow \quad X = a & (1.7) \end{aligned}$$

Für n-stellige Schaltfunktionen f heißt die Menge

$$\mathrm{Tr}(f) = \{a \in \{0,1\}^n \mid f(a) = 1\}$$

der *Träger* von f. Offenbar ist

$$\mathrm{Tr}(f) = f^{-1}(1) \text{ und } \{0,1\}^n \setminus \mathrm{Tr}(f) = f^{-1}(0) .$$

Träger

Hierbei ist f^{-1} die Umkehrabbildung zu f, d.h. $f^{-1}(y)$ ist die Menge aller x für die $f(x) = y$ gilt. Es gilt

Satz 1.6 (Darstellungssatz) *Sei $f : \{0,1\}^n \to \{0,1\}$ eine Schaltfunktion. Dann gilt*

$$\begin{aligned} f(X) &\equiv \bigvee_{a \in \mathrm{Tr}(f)} m(a) \\ f(X) &\equiv \bigwedge_{a \notin \mathrm{Tr}(f)} c(a) \end{aligned}$$

kanonische disjunktive Normalform
kanonische konjunktive Normalform

Die erste Darstellung heißt die *kanonische disjunktive Normalform* von f, die zweite Darstellung die *kanonische konjunktive Normalform*. Der Ausdruck „kanonisch" rührt daher, dass diese Formen jeweils bis auf die Reihenfolge Min– bzw. Maxterme eindeutig sind.

Beispiel 1.16 *Sei f die in Tabelle 1.3 definierte Funktion. Dann gilt*

$$\begin{aligned} f(X) &\equiv \overline{X_1} X_2 \overline{X_3} \vee X_1 X_2 \overline{X_3} \\ &\equiv (X_1 \vee X_2 \vee X_3) \wedge (X_1 \vee X_2 \vee \overline{X_3}) \wedge (X_1 \vee \overline{X_2} \vee \overline{X_3}) \\ &\quad \wedge (\overline{X_1} \vee X_2 \vee X_3) \wedge (\overline{X_1} \vee X_2 \vee \overline{X_3}) \wedge (\overline{X_1} \vee \overline{X_2} \vee \overline{X_3}) . \end{aligned}$$

Beweis des Darstellungssatzes: Es gilt

$$\begin{aligned} \bigvee_{a \in \mathrm{Tr}(f)} m(a) = 1 &\Leftrightarrow m(b) = 1 \text{ für ein } b \in \mathrm{Tr}(f) \\ &\Leftrightarrow X = b \text{ für ein } b \in \mathrm{Tr}(f) . \end{aligned}$$

Behauptung 1 folgt nun direkt aus Lemma 1.5. Behauptung 2 beweist man ebenso. ■

Selbsttestaufgabe 1.9 *Bestimmen Sie den Träger der Funktion $\vee$. Bestimmen Sie die zu den Elementen des Trägers gehörigen Minterme und die kanonische disjunktive Normalform von $\vee$.*

1.2.6 Kosten von Ausdrücken

Wir suchen im Folgenden sehr oft zu einer vorgegebenen Schaltfunktion f möglichst *einfache* Ausdrücke e, die f berechnen. Hierbei messen wir die Kompliziertheit eines Ausdrucks einfach durch die folgende Kostenfunktion.

Definition 1.16 *Sei $e \in B$ ein Boole'scher Ausdruck. Die* Kosten *$L(e)$ von e sind definiert als die Anzahl von Vorkommen der Zeichen $\wedge$, $\vee$ und $\sim$ in e.*

Beispiel 1.17 $L(X_1 \wedge \sim X_2 \wedge X_3) = 3$.

Die obige Definition scheint wörtlich genommen nur sinnvoll zu sein für Ausdrücke e, bei denen wir gewisse vereinfachte Schreibweisen nicht verwenden. Wir erinnern jedoch daran, daß für uns vereinfacht aufgeschriebene Ausdrücke ebenso wie unvollständig geklammerte Ausdrücke bloß Abkürzungen für vollständig geklammerte Ausdrücke aus B sind. Es ist deshalb

$$L(X_1\overline{X_2}X_3) = L(X_1 \wedge {\sim}X_2 \wedge X_3) = L((X_1 \wedge (({\sim}X_2) \wedge X_3))) = 3 \ .$$

Offenbar ist $L(X_i^\epsilon) \in \{0,1\}$, d.h. Literale haben stets Kosten 0 oder 1. Sei nun f eine n-stellige Schaltfunktion. Jeder Minterm m der vollständigen disjunktiven Normalform von f besteht aus genau n Literalen und $n-1$ $\wedge$–Zeichen. Es folgt $L(m) \leq 2n-1$. Sei nun p die vollständige disjunktive Normalform von f. Dann besteht p aus genau $\#\mathrm{Tr}(f)$ Mintermen. Für die Anzahl v der $\vee$–Zeichen in p gilt

$$v = \begin{cases} \#\mathrm{Tr}(f) - 1 & \text{falls } \#\mathrm{Tr}(f) \geq 2 \\ 0 & \text{falls } \#\mathrm{Tr}(f) \leq 1 \end{cases}$$

Wegen $\#\mathrm{Tr}(f) \leq \#\{0,1\}^n = 2^n$ folgt

$$L(p) \leq n2^{n+1} \ .$$

Für jede n-stellige Schaltfunktion f gibt es also einen Boole'schen Ausdruck e mit Kosten höchstens $n2^{n+1}$, der f berechnet. Wir wären natürlich gern in der Lage, zu jeder vorgegebenen Schaltfunktion einen *billigsten* Ausdruck mit dieser Eigenschaft sowie seine Kosten zu bestimmen.

Definition 1.17 *Für Schaltfunktionen f heißt die Zahl*

$$L(f) = \min\{L(e) \mid e \in B, e \equiv f(X)\}$$

die Formelgröße *(engl. formula size) von f.* Formelgröße

Aus dem oben Gesagten folgt sofort

Satz 1.7 *Für jede n-stellige Schaltfunktion f gilt $L(f) \leq n2^{n+1}$.*

Genau genommen folgt aus der obigen Konstruktion $L(f) \leq n2^{n+1} - 1$. Allerdings ist die Aussage aus Satz 1.7 natürlich auch richtig. Sie ist außerdem besser zu merken und enthält die wichtige Information, dass man eine obere Schranke angeben kann, die exponentiell in der Anzahl n der Variablen ist und einen linearen Vorfaktor hat. Oft möchte man, wenn man das Wachstum von Funktionen beschreibt, auch von konstanten Faktoren abstrahieren, da man an der „Größenordnung" des Wachstums interessiert ist. Um den Begriff „größenordnungsmäßig" formal zu fassen, führen wir Notationen für asymptotisches Wachstum ein.

Definition 1.18 *Seien $f, g : \mathbf{N} \to \mathbf{N}$ Funktionen. Wir sagen f* ist asymptotisch durch g beschränkt*, in Zeichen $f \leq_a g$, falls es ein $n_0 \in \mathbf{N}$ gibt mit $f(n) \leq g(n)$ für alle $n \geq n_0$. Wir definieren*

$$\begin{aligned} O(g) &= \{f \mid \exists k \in \mathbf{N} : f \leq_a k \cdot g\}, \\ \Omega(g) &= \{f \mid \exists k \in \mathbf{N} : g \leq_a k \cdot f\}, \\ \Theta(g) &= O(g) \cap \Omega(g) \quad . \end{aligned}$$

Normalerweise schreibt man $f = O(g)$ statt $f \in O(g)$.

Beispiel 1.18 *Es ist* $3n^2 - 4n + 5 \in O(n^3)$, $e^n = \Omega(n^{10})$ *und* $4n^5 = \Theta(n^5)$. *Allerdings ist* $2n \notin O(\log_2 n)$.

Damit läßt sich Satz 1.7 umformulieren zu:

Für jede n*-stellige Schaltfunktion* f *gilt* $L(f) = O(n2^n)$.

Selbsttestaufgabe 1.10 *Bestimmen Sie eine möglichst gute obere Schranke für die Formelgröße von* $f(X_1, X_2, X_3) = X_1 \wedge (X_2 \vee X_3)$. *Geben Sie auch Schranke aus Satz 1.7 an.*

1.3 Minimalpolynome

1.3.1 Polynome und Primimplikanten

Wir untersuchen im Folgenden besonders einfache Mengen von Boole'schen Ausdrücken, nämlich die sogenannten *Boole'schen Polynome* und die *konjunktiven Normalformen.*

Definition 1.19

Literal

- *Ein* Literal *ist ein Ausdruck der Form* X_i^ϵ *mit* $X_i \in V$ *und* $\epsilon \in \{0, 1\}$.

Monom
Konjunktionsterm

- *Ein* Monom *oder* Konjunktionsterm *ist ein Ausdruck der Form* $\bigwedge_{i \in I} L_i$, *wobei die* L_i *Literale sind für alle* i *in einer endlichen Indexmenge* I.

Boole'sches Polynom
disjunktive Normalform

- *Ein (Boole'sches)* Polynom *oder* disjunktive Normalform (DNF) *ist ein Ausdruck der Form* $\bigvee_{i \in I} M_i$, *wobei die* M_i *Monome sind für alle* i *in einer endlichen Indexmenge* I.

Klausel
Disjunktionsterm

- *Eine* Klausel *oder* Disjunktionsterm *ist ein Ausdruck der Form* $\bigvee_{i \in I} L_i$, *wobei die* L_i *Literale sind für alle* i *in einer endlichen Indexmenge* I.

konjunktive Normalform

- *Eine* konjunktive Normalform *(KNF) ist ein Ausdruck der Form* $\bigwedge_{i \in I} C_i$, *wobei die* C_i *Klauseln sind für alle* i *in einer endlichen Indexmenge* I.

Alle Minterme sind Monome, und alle Maxterme sind Klauseln. Jede kanonische disjunktive Normalform ist ein Polynom, und jede kanonische konjunktive Normalform ist eine konjunktive Normalform.

In den obigen Definitionen sind auch leere Indexmengen I erlaubt. Es folgt, daß 0 sowohl ein Polynom und als auch eine Klausel ist, und daß 1 sowohl ein Monom als auch eine konjunktive Normalform ist.

Selbsttestaufgabe 1.11 *Ist der Ausdruck* $X_1(X_2 \vee X_3)$ *eine disjunktive oder konjunktive Normalform? Falls nicht, formen Sie ihn um.*

Naturgemäß interessiert man sich zu einer vorgegebenen Schaltfunktion f für *billigste* Polynome p, die f berechnen.

Definition 1.20 *Sei f eine Schaltfunktion und p ein Boole'sches Polynom. Dann heißt p ein* Minimalpolynom *oder* kürzeste disjunktive Normalform *von f, falls die folgenden beiden Bedingungen gelten:*

1. *$p \equiv f(X)$, d.h. p berechnet f.*
2. *$L(p) = \min\{L(q) \mid q$ ist Boole'sches Polynom und $q \equiv f(X)\}$, d.h. p ist ein billigstes Polynom mit dieser Eigenschaft.*

Wir werden im Folgenden zu vorgegebener Funktion f die Monome, die in Minimalpolynomen von f auftreten können, charakterisieren, und wir werden angeben, wie man diese Monome finden kann. Im unmittelbaren Anschluß daran stoßen wir schon auf das mit Abstand berühmteste offene Problem der Informatik.

Definition 1.21 *Seien m und m' Monome. Dann heißt m'* Teilmonom *von m, falls die folgenden beiden Bedingungen gelten:* Teilmonom

1. *jedes Literal in m' kommt auch in m vor, oder $m' = 1$;*
2. *in m kommt mindestens ein Literal vor, das nicht in m' vorkommt.*

Beispiel 1.19 *Die Monome X_1X_4, 1 und $X_2\overline{X_3}$ sind Teilmonome von $X_1X_2\overline{X_3}X_4$, die Monome $X_1X_2X_3X_4$ und $X_1X_2\overline{X_3}X_4$ hingegen nicht.*

Lemma 1.8 *Es sei m' Teilmonom von m. Dann gilt $m \leq m'$.*

Beweis: Falls $m' = 1$ dann ist $m \leq m'$ offensichtlich wegen $m \leq 1$. Es sei also $m' = \bigwedge_{i \in J} L_i$, wobei die L_i Literale sind und J eine nicht-leere Indexmenge. Dann ist $m = \bigwedge_{i \in I} L_i$, wobei $I \supset J$, da jedes Literal aus m' nach Definition auch in m enthalten ist. Auch können wir $I = J$ ausschließen, da es nach Definition mindestens ein Literal in m geben muss, das nicht in m' enthalten ist. Wir betrachten im folgenden nur den Fall dass m bei einer Variablenbelegung den Wert 1 annimmt, denn wenn es den Wert 0 annimmt ist offensichtlich $m \leq m'$ wegen $m' \geq 0$. Aus Lemma 1.5 folgt dann unter Berücksichtigung von $J \subset I$:

$$\begin{aligned} m = 1 \quad &\Leftrightarrow \quad L_i = 1 \text{ für alle } i \in I \\ &\Rightarrow \quad L_i = 1 \text{ für alle } i \in J \\ &\Leftrightarrow \quad m' = 1\,. \end{aligned}$$

Also gilt auch in dem Fall, dass m bei einer Variablenbelegung den Wert 1 annimmt, $m \leq m'$. ∎

Beispiel 1.20 *Die Monome $m'_1 = X_1X_4$, $m'_2 = 1$ und $m'_3 = X_2\overline{X_3}$ sind Teilmonome von $m = X_1X_2\overline{X_3}X_4$. Das Monom $m = X_1X_2\overline{X_3}X_4$ nimmt den Wert 1 nur an der Stelle $X_1X_2X_3X_4 = 1101$ an. Sonst nimmt es den Wert Null an. An der Stelle 1101 haben auch die Teilmonome den Wert 1. Damit gilt für jedes Teilmomom m'_i: $m \leq m'_i$. An der Stelle $X_1X_2X_3X_4 = 1111$ haben die ersten beiden Teilmonome den Wert 1 aber m den Wert 0, an der Stelle* 0100 *hat das letzte Teilmonom den Wert 1, m aber nicht.*

Implikant

Primimplikant

Definition 1.22 *Sei f eine Schaltfunktion und m ein Monom. Dann heißt m ein* Implikant *von f falls $m \leq f(X)$ gilt, d.h. falls aus $m = 1$ auch $f(X) = 1$ folgt. Ein Implikant von f heißt ein* Primimplikant *oder* Primterm *von f falls kein Teilmonom von m Implikant von f ist.*

Die konstante Funktion f mit $f(a) = 0$ für alle a hat nur ein Minimalpolynom, nämlich 0. Für alle anderen Schaltfunktionen f werden die Implikanten, die in Minimalpolynomen von f vorkommen können, charakterisiert durch

Satz 1.9 *Es sei f eine Schaltfunktion, und f sei nicht identisch gleich* 0*. Es sei p ein Minimalpolynom von f. Dann besteht p nur aus Primimplikanten von f.*

Beweis: Ist f identisch gleich 1, so hat f nur ein Minimalpolynom, nämlich 1, und der Satz gilt offensichtlich. Andernfalls gilt

$$f(X) \equiv p = m_1 \vee \ldots \vee m_s$$

für ein $s \in \mathbf{N}$ und Monome m_i, $i \in \{1, \ldots, s\}$. Jedes der Monome m_i ist ein Implikant von f, da es sonst eine Einsetzung ϕ gäbe mit $\phi(f) = 0$, aber $1 = \phi(m_i) = \phi(p)$.

Wir nehmen nun an, daß mindestens eins der Monome m_i kein Primimplikant von f ist. Ohne Beschränkung der Allgemeinheit können wir $i = 1$ annehmen (sonst numerieren wir die Monome um.) Sei nun m_1' Teilmonom von m_1 und Implikant von f. Wir bilden das Polynom p', indem wir in p das Monom m_1 durch das billigere Monom m_1' ersetzen:

$$p' = m_1' \vee m_2 \vee \ldots \vee m_s \; .$$

Offensichtlich ist dann $L(p') < L(p)$. Wegen Lemma 1.8 ist $m_1 \leq m_1'$ und deshalb $p \leq p'$. Andererseits gilt $m_1' \leq f$ und $m_i \leq f$ für alle i, denn sowohl m_1' als auch alle m_i sind Implikanten von f. Es folgt $p' \leq f \equiv p$. Es folgt $p' \equiv p \equiv f$. Also war p kein Minimalpolynom von f. ∎

Selbsttestaufgabe 1.12 *Sind $X_1\bar{X}_2$, $\bar{X}_1 X_2$ und $X_1 X_2$ Implikanten der Schaltfunktion $f : \{0,1\}^2 \to \{0,1\}$, die den Wert 1 genau dann annimmt, wenn genau eines ihrer Argumente den Wert 1 hat? Falls ja, sind es Primimplikanten?*

1.3.2 Bestimmung von Minimalpolynomen

Um ein Minimalpolynom zu bestimmen, gibt es eine Reihe von Verfahren. Bei den meisten bildet man ausgehend von der Wertetabelle oder der kanonischen DNF zunächst alle Primimplikanten. Hierunter ist das bekannteste das Verfahren von Quine und McCluskey. Wir werden hier lediglich exemplarisch zeigen, wie man die Primimplikanten mittels des Karnaugh-Diagramms bestimmt.

Hierzu erinnern wir daran, dass jedes Feld des Karnaugh-Diagramms eineindeutig einem Element des Definitionsbereichs entspricht. Damit entspricht jedes mit 1 markierte Feld einem Minterm. Zum Beispiel entspricht in Abbildung 1.4 auf Seite 14 das Feld in der linken oberen Ecke dem Minterm

$X_1X_2\bar{X}_3\bar{X}_4$ und das Feld rechts daneben entspricht dem Minterm $X_1X_2X_3\bar{X}_4$. Da sich beim Wechsel der Zeile oder der Spalte der Wert genau einer Variable ändert, haben die Minterme zweier nebeneinanderliegender mit 1 markierter Felder die Form m_1X_i und $m_1\bar{X}_i$, wobei m_1 ein Monom ist, das die Variable X_i nicht enthält. Damit entspricht das 2×1-Rechteck, das aus diesen beiden Feldern gebildet wird, dem Monom $m_1X_i \vee m_1\bar{X}_i = m_1(X_i \vee \bar{X}_i = m_1$. Dieses Monom ist offensichtlich Teilmonom der beiden Monome (in diesem Fall Minterme) aus denen es entstanden ist. Die beiden Minterme sind Implikanten, also ist auch das resultierende Monom ein Implikant.

Auf die gleiche Weise kann man natürlich auch zwei 2×1-Rechtecke die mit Einsen markiert sind weiter zusammenfassen, und erhält wiederum ein Teilmonom das ein Implikant ist. Zum Beispiel bilden die vier Einsen der ersten und zweiten Zeile und Spalte in Abbildung 1.4 ein 2×2-Rechteck, das dem Monom X_1X_2 entspricht.

Insgesamt können wir festhalten, dass jedes Rechteck in einem Karnaugh-Diagramm, dessen Seitenlängen Zweierpotenzen sind, einem Monom entspricht. Folglich ist jedes dieser Rechtecke, das in einem Karnaugh-Diagramm nur Einsen überdeckt, ein Implikant. Hierbei ist zu beachten, dass man das Karnaugh-Diagramm so interpretieren muss, als sei es „rundgeklebt“, d.h. wenn man am linken Rand herausfällt, macht man am rechten Rand weiter, ebenso mit oberem und unterem Rand. Lässt sich kein größeres Monom-Rechteck finden, das das gegenwärtige Rechteck enthält, und nur Einsen überdeckt, dann ist das gegenwärtige Rechteck schon ein Primimplikant.

Beispiel 1.21 *In dem Karnaugh-Diagramm aus Abbildung 1.4 (Seite 14) lassen sich sechs Primimplikanten finden: die zweite Zeile bildet ein Rechteck mit Seitenlängen 4 und 1. Sie entspricht dem Monom X_2X_4. Die zweite Spalte bildet ein Rechteck mit Seitenlängen 1 und 4, sie entspricht dem Monom X_1X_3. In dem linken oberen 3×3-Block aus Einsen finden sich vier Quadrate mit Seitenlänge 2. Sie entsprechen den Monomen X_1X_2, X_2X_3, X_1X_4, X_3X_4.*

Beispiel 1.22 *In dem Karnaugh-Diagramm aus Abbildung 1.7 finden sich drei ungewöhnliche Primimplikanten. Die vier Ecken bilden wegen des Rundklebens ein Quadrat mit Seitenlänge 2, das dem Monom $\bar{X}_3\bar{X}_4$ entspricht. Die erste und letzte Zeile der ersten und zweiten Spalte bilden ein Quadrat mit Seitenlänge 2, das dem Monom $X_1\bar{X}_4$ entspricht. Die ersten beiden Zeilen der ersten und letzten Spalte bilden wiederum ein Quadrat mit Seitenlänge 2, das dem Monom $X_2\bar{X}_3$ entspricht.*

Ist die Schaltfunktion nur partiell definiert, so kann das Symbol X im Karnaugh-Diagramm als 1 oder 0 interpretiert werden, je nachdem wie es besser passt.

Selbsttestaufgabe 1.13 *Stellen Sie eine Wertetabelle auf für die Schaltfunktion $f : \{0,1\}^4 \to \{0,1\}$, die genau dann den Wert 1 annimmt, wenn höchstens zwei ihrer vier Argumente den Wert 1 annehmen, und die undefiniert ist, wenn genau 3 ihrer Argumente den Wert 1 annehmen. Bestimmen Sie den Träger und erstellen Sie die kanonische disjunktive Normal-*

$$
\begin{array}{c|cccc|c}
 & \multicolumn{2}{c}{\overline{\quad X_1 \quad}} & & & \\
X_2 & 1 & 1 & 0 & 1 & \\
X_2 & 1 & 0 & 0 & 1 & X_4 \\
 & 0 & 0 & 0 & 0 & X_4 \\
 & 1 & 1 & 0 & 1 & \\
 & & \multicolumn{2}{c}{\underline{\quad X_3 \quad}} & &
\end{array}
$$

Abbildung 1.7: Vereinfachtes Karnaugh-Diagramm für $n = 4$

form. Bestimmen Sie die Kosten der KDNF. Übertragen Sie die Wertetabelle in ein Karnaugh-Diagramm. Bestimmen Sie die Primimplikanten aus dem Karnaugh-Diagramm.

Es bleibt das auf den ersten Blick einfache Restproblem, aus den Primimplikanten einer Schaltfunktion ein Minimalpolynom zusammenzubauen. Ein solches Minimalpolynom wird i.A. nicht aus allen Primimplikanten bestehen. Man muß deshalb eventuell unter den Primimplikanten eine Auswahl treffen. Hierfür beschreiben wir im weiteren Regeln.

Überdeckung

Definition 1.23 *Sei e ein Boole'scher Ausdruck und $a \in \{0,1\}^n$. Wir sagen e* überdeckt *a genau dann, wenn $\phi_a(e) = 1$ gilt. Ist M eine Menge von Monomen und $F \subseteq \{0,1\}^n$, so heißt die Abbildung $I : M \times F \to \{0,1\}$,*

$$I(m,a) = \begin{cases} 1 \text{ falls } m \text{ überdeckt } a \\ 0 \text{ falls sonst} \end{cases}$$

Implikantentafel

die Implikantentafel *von M und F.*

Der Name 'Implikantentafel' kommt daher, daß man I als Matrix aufschreiben kann, deren Zeilen mit den Elementen $m \in M$ und deren Spalten mit den Elementen $a \in F$ indiziert sind. Ist M die Menge der Primimplikanten einer Schaltfunktion f und ist $F = \{a \mid f(a) = 1\}$ der Träger der Schaltfunktion, so heißt I die *Primimplikantentafel* oder *Primtermtabelle* von f. Die Primimplikantentafeln der beiden Schaltfunktionen $f_1 : \{0,1\}^4 \to \{0,1\}$ und $f_2 : \{0,1\}^3 \to \{0,1\}$, deren Funktionstabellen in Tabelle 1.5 zu sehen sind, findet man in Tabelle 1.6.

Jeder Menge S von Monomen ordnen wir das Polynom

$$p(S) = \bigvee_{m \in S} m$$

zu. Eine Implikantentafel I definiert ein zugehöriges *Überdeckungsproblem*: finde eine Teilmenge $S \subseteq M$ von Monomen, so daß gilt: $p(S)$ überdeckt f. Eine solche Teilmenge heißt eine *Lösung* des Überdeckungsproblems. Ist I die Primimplikantentafel von f, so gilt offensichtlich

$$p(S) \equiv f(X)$$

a_1	a_2	a_3	a_4	$f_1(a_1,a_2,a_3,a_4)$	$f_2(a_1,a_2,a_3)$
0	0	0	0	1	0
0	0	0	1	1	
0	0	1	0	1	1
0	0	1	1	1	
0	1	0	0	0	1
0	1	0	1	0	
0	1	1	0	0	1
0	1	1	1	1	
1	0	0	0	0	1
1	0	0	1	0	
1	0	1	0	0	1
1	0	1	1	0	
1	1	0	0	0	1
1	1	0	1	0	
1	1	1	0	0	0
1	1	1	1	1	

Tabelle 1.5: Funktionstabellen der Schaltfunktionen f_1 und f_2

(a)

f_1	0000	0001	0010	0011	0111	1111
$\overline{X_1}X_3X_4$	0	0	0	1	1	0
$X_2X_3X_4$	0	0	0	0	1	1
$\overline{X_1}\,\overline{X_2}$	1	1	1	1	0	0

(b)

f_2	001	010	011	100	101	110
$\overline{X_1}X_3$	1	0	1	0	0	0
$\overline{X_2}X_3$	1	0	0	0	1	0
$\overline{X_1}X_2$	0	1	1	0	0	0
$X_2\overline{X_3}$	0	1	0	0	0	1
$X_1\overline{X_3}$	0	0	0	1	0	1
$X_1\overline{X_2}$	0	0	0	1	1	0

Tabelle 1.6: Primimplikantentafeln für f_1 und f_2

	0011	0101	0110	0111	1001	1010	1011	1100	1101	1110	1111
X_1X_2								1	1	1	1
X_1X_3						1	1			1	1
X_1X_4					1		1		1		1
X_2X_3			1	1						1	1
X_2X_4		1		1					1		1
X_3X_4	1			1			1				1

Tabelle 1.7: Primimplikantentafel zur Schaltfunktion aus Abb. 1.4

für alle Lösungen S von I, da sie gerade alle Implikanten der Funktion f enthält. Die Lösungen S, für die $p(S)$ minimale Kosten hat, sind offensichtlich gerade die Minimalpolynome von f.

Beispiel 1.23 *Wir erstellen die Primimplikantentafel zu der Schaltfunktion aus Abbildung 1.4 (Seite 14). Hierzu bestimmen wir zunächst den Träger der Funktion, d.h. alle Belegungen $a \in \{0,1\}^4$ mit $f(a) = 1$. Das sind nach Definition der Schaltfunktion gerade alle a die mindestens zwei Einsen enthalten. Die Primimplikanten haben wir in Beispiel 1.21 bereits bestimmt. Die Primimplikantentafel ist in Tabelle 1.7 dargestellt. Zur Verdeutlichung haben wir nur die Einträge mit 1 dargestellt. Die Einträge mit 0 sind leer. Man sieht, dass jeder Primimplikant gerade vier Monome überdeckt, nämlich die vier, mittels derer man ihn im Karnaugh-Diagramm identifiziert hat.*

Um eine Lösung eines Überdeckungsproblems zu finden, bestimmen wir zunächst die Monome, die in der Lösung unbedingt enthalten sein müssen.

wesentliches Monom

Definition 1.24 *Sei $I : M \times F \to \{0,1\}$ ein Überdeckungsproblem und $m \in M$. Dann heißt m* wesentlich, *falls es ein $a \in F$ gibt, so daß a nur von m und keinem anderen Monom in M überdeckt wird.*

Kernimplikant

Die wesentlichen Primimplikanten der Primimplikantentafel nennt man auch *Kernimplikant.* Einen Kernimplikanten entdeckt man in der Primimplikantentafel dadurch, dass eine der Einsen in seiner Zeile die einzige Eins in der betreffenden Spalte ist.

Beispiel 1.24 *Wie man aus Tabelle 1.6 erkennt sind für die Funktion f_1 aus Tabelle 1.5 die Monome $\overline{X_1}\,\overline{X_2}$ wegen der ersten drei Einsen und $X_2X_3X_4$ wegen der letzten Eins wesentlich. Monom $\overline{X_1}X_3X_4$ ist nicht wesentlich. Bei Funktion f_2 ist kein Monom wesentlich. In der Tabelle 1.7 sind alle Primimplikanten wesentlich.*

Das Überdeckungsproblem $I' = I(m) : M' \times F' \to \{0,1\}$ enstehe aus I durch Entfernen von m aus M und durch Entfernen aller von m überdeckten Tupel a aus F. Anschaulich gesprochen entfernen wir alle Spalten, in denen in der Zeile von m eine 1 war, und dann die Zeile von m.

Eine Teilmenge S' von M' ist genau dann eine Lösung $S(m)$ von $I(m)$ wenn $S' \cup \{m\}$ Lösung von I ist. Eine billigste Lösung von Problem I kann also stets aus einer billigsten Lösung des kleineren Problems I' gewonnen werden.

f_1	0111	1111
$\overline{X_1}X_3X_4$	1	0
$X_2X_3X_4$	1	1

Tabelle 1.8: Vereinfachte Primimplikantentafel für f_1

Wir lösen also das Überdeckungsproblem, indem wir alle wesentlichen Monome nacheinander entfernen und eine billigste Lösung für das Restproblem suchen.

Beispiel 1.25 *Sei I das Überdeckungsproblem aus Tabelle 1.6(a) und $m = \overline{X_1}\,\overline{X_2}$. Dann ist I' das Problem aus Tabelle 1.8.*

Wir können natürlich das gleiche Kriterium nochmals anwenden, um in diesem Fall die einzige Lösung minimaler Kosten zu bestimmen.

Tabelle 1.8 illustriert auch ein Kriterium mit dem man Hinweise gewinnt, wie man eine billigste Lösung des Restproblems nach Entfernen der wesentlichen Monome findet:

Sei $I : M \times F \to \{0,1\}$ ein Überdeckungsproblem, und es seien $m, m' \in M$. Wir sagen, daß m das Monom m' *dominiert*, falls $L(m) \leq L(m')$ und falls jedes Tupel a, das von m' überdeckt wird auch von m überdeckt wird.

Beispiel 1.26 *In Tabelle 1.8 dominiert $X_2X_3X_4$ das Monom $\overline{X_1}X_3X_4$.*

Wird m' von m dominiert, so kann man m' in jeder Lösung von I durch m ersetzen. Man erhält wieder eine Lösung, und diese ist nicht teurer als die alte Lösung. Deshalb kann man in diesem Fall m' einfach aus M entfernen.

Man kommt leider häufig in Situationen, in denen das Kriterium nicht anwendbar ist, beispielsweise in Tabelle 1.6(b). Im Allgemeinen kommt man an dieser Stelle nur noch mit roher Gewalt weiter: Man sucht für *alle* $m \in M$ eine billigste Lösung $S(m)$ des Problems $I(m)$ und sucht dann unter den Lösungen $m \vee \bigvee_{r \in S(m)} r$ eine billigste aus. Da man bei den entstehenden kleineren Problemen immer wieder in die gleiche Situation geraten kann, wird das sehr schnell extrem aufwendig:

Für $k \in \mathbf{N}$ sei $i(k)$ die größte Zahl von Überdeckungsproblemen, die insgesamt generiert werden, wenn man mit einem Überdeckungsproblem mit k Monomen startet. Dann ist

$$i(1) = 1,$$

und für $k > 1$ können wir nach dem oben Gesagten $i(k)$ nur abschätzen durch

$$i(k) \leq k \cdot i(k-1) \,.$$

Durch Induktion über k folgt

$$i(k) \leq k! = 1 \cdot 2 \cdot \ldots \cdot k = \Theta(k^k) \,.$$

Ob man solche Probleme sehr viel schneller lösen kann, d.h. ob es eine Lösung gibt mit einem Aufwand der polynomiell in k bleibt, ist eine offene Frage, deren Bedeutung weit über die Grenzen der Optimierung von Boole'schen

	m_1	m_2	m_3	m_4	m_5	m_6
pi_1	1	1				
pi_2		1	1	1		
pi_3			1	1	1	
pi_4					1	1
pi_5				1		1

Abbildung 1.8: Primimplikantentafel zu Selbsttestaufg. 1.14

Ausdrücken hinausreicht. In der theoretischen Informatik werden Ihnen solche Fragestellungen als NP-vollständige Probleme wieder begegnen.

In der Praxis ist das Restproblem oft klein, so dass man durch Ausprobieren direkt eine Lösung findet. Enthält das Restproblem zum Beispiel noch drei Primimplikanten, so kann man zunächst alle Möglichkeiten suchen, zwei dieser Primimplikanten zu kombinieren, so dass sie die restlichen Monome vollständig überdecken. Unter diesen sucht man dann die billigste Variante.

Beispiel 1.27 *Die vereinfachte Primimplikantentafel für Schaltfunktion f_2 aus Tabelle 1.5 entspricht ihrer Primimplikantentafel aus Tabelle 1.6(b). Hier gibt es sechs Primimplikanten, von denen keiner den anderen dominiert. Alle haben gleiche Kosten 3, so dass diese bei der Auswahl keine Rolle spielen. Da jeder Primimplikant zwei Monome überdeckt, und sechs Monome zu überdecken sind, braucht eine Lösung des Restüberdeckungsproblems mindestens drei Primimplikanten. Durch genaues Hinsehen findet man drei solche Primimplikanten auch schnell, zum Beispiel $\bar{X}_1 X_3$, $X_2 \bar{X}_3$ und$X_1 \bar{X}_2$. Ein Minimalpolynom der Schaltfunktion f_2 lautet also*

$$p(X_1, X_2, X_3) = \bar{X}_1 X_3 \vee X_2 \bar{X}_3 \vee X_1 \bar{X}_2 \ .$$

Selbsttestaufgabe 1.14 *Bestimmen Sie aus der Primimplikantentafel der Abbildung 1.8 die Kernimplikanten und stellen Sie die vereinfachte Primimplikantentafel auf. Die Primimplikanten sind dort mit pi_i gekennzeichnet, die Monome des Trägers mit m_j. Vereinfachen Sie diese Tafel mit den Kriterien der Wesentlichkeit und der Dominanz. Bilden Sie eine Lösung des Restproblems. Geben Sie ein Minimalpolynom an. Gehen Sie davon aus, dass die Primimplikanten alle gleiche Kosten haben.*

Index

Lösungen der Selbsttestaufgaben

Zu Aufgabe 1.1

Es gilt

$$\sum_{i=m}^{n-1} x^i = \sum_{i=0}^{n-1} x^i - \sum_{i=0}^{m-1} x^i \,.$$

Die beiden Summen auf der rechten Seite lassen sich mittels Lemma 1.1 ausdrücken als

$$\frac{x^n - 1}{x - 1} \text{ und } \frac{x^m - 1}{x - 1} \,.$$

Subtrahiert man diese beiden Brüche, erhält man die rechte Seite von Gleichung (1.2).

Zu Aufgabe 1.2

Die Knoten 1 und 2 bilden die Quellen. Es gibt keine Senke im Graphen, da alle Knoten einen Outgrad größer als Null haben. Es gilt $T(1) = T(2) = 0$, da die beiden Knoten Quellen sind, und $T(3) = 1$, da dieser Knoten mit einem Pfad der Länge 1 von Quelle 1 aus erreichbar ist. Knoten 4 ist von Quelle 1 aus mit einem Pfad der Länge 2 erreichbar, und von Quelle 2 aus mit einem Pfad der Länge 1 aus erreichbar. Die Tiefe von Knoten 4 beträgt also 2, da der längste Pfad zählt. Die Tiefe von Knoten 5 ist nicht definiert, da dieser einen Zyklus der Länge 1 mit sich selbst bildet.

Zu Aufgabe 1.3

Die Anzahl der 2-stelligen Schaltfunktionen beträgt $16 = 2^{2^2} = 2^4$. Sie lauten:

X_1 X_2	f_0	f_0	f_0	f_0	f_0	f_0	f_0	f_0	f_0	f_0	f_0	f_0	f_0	f_0	f_0	f_0
00	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
01	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
10	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
11	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Als 1-stellige Schaltfunktion des ersten Arguments können die Funktionen $f_0 = 0$, $f_3 = X_1$, $f_{12} = 1 - X_1$, $f_{15} = 1$ interpretiert werden. Man erhält gerade die Anzahl möglicher 1-stelliger Schaltfunktionen: $4 = 2^{2^1} = 2^2$. Man erkennt die Unabhängigkeit vom zweiten Argument daran, dass die Funktionswerte bei 00 und 01 sowie die Funktionswerte bei 10 und 11 gleich sind.

Zu Aufgabe 1.4

Das Karnaugh-Diagramm hat 4 Felder und bildet also ein 2×2-Quadrat. Um Eindeutigkeit zu erzielen, muss jeweils die Hälfte der Spalten und die Hälfte der Zeilen mit einer Variable markiert werden. Die genaue Zuordnung ob X_0 die Spalten oder die Zeilen markiert, und ob die linke bzw. rechte Spalte markiert wird, spielt in diesem Fall keine Rolle. Ein mögliches KV-Diagramm zeigt die folgende Abbildung.

$$\begin{array}{|c|c|l}
\multicolumn{1}{c}{} & \multicolumn{1}{c}{X_0} & \\
\hline
0 & 0 & \\
\hline
0 & 1 & X_1 \\
\hline
\end{array}$$

Zu Aufgabe 1.5

Es gilt $X_1, X_2, X_3 \in \mathrm{EB}_1$ und damit über die erste Regel von Definition 1.9 auch in allen weiteren EB_i. Dann sind $(X_1 \wedge X_2)$ und $f_1(X_1, X_2)$ in EB_2, $((X_1 \wedge X_2) \wedge X_3) \in \mathrm{EB}_3$ und $(((X_1 \wedge X_2) \wedge X_3) \vee f_1(X_1, X_2)) \in \mathrm{EB}_4$. In dem zweiten Ausdruck fehlt die äußere öffnende Klammer zu Anfang des Ausdrucks, sowie eine innere Klammer zur Strukturierung von $(X_1 \wedge X_2 \vee X_3)$. Die letzte Ergänzung ist allerdings nicht eindeutig. Die zwei möglichen Ausdrücke aus EB sind

$$(f_1(X_1, X_2) \vee ((X_1 \wedge X_2) \vee X_3)) \text{ und } (f_1(X_1, X_2) \vee (X_1 \wedge (X_2 \vee X_3))) \,.$$

Zu Aufgabe 1.6

An der Stelle $a = (1, 1, 1)$ gilt $\phi_a(X_1) = \phi_a(X_2) = 1$, also $\phi_a(X_1 \wedge X_2) = 1 \wedge 1 = 1$. Weiterhin ist $\phi_a(f_1(X_1, X_2)) = f_1(\phi_a(X_1), \phi(X_2)) = f_1(1, 1) = 0$. Hieraus folgt $\phi_a((X_1 \wedge X_2) \wedge X_3) = \phi_a(X_1 \wedge X_2) \wedge \phi_a(X_3) = 1 \wedge 1 = 1$ und $\phi_a(((X_1 \wedge X_2) \wedge X_3) \vee f_1(X_1, X_2)) = \phi_a(((X_1 \wedge X_2) \wedge X_3)) \vee \phi_a(f_1(X_1, X_2)) = 1 \vee 0 = 1$

An der Stelle $b = (1, 0, 1)$ gilt $\phi_b(X_1 \wedge X_2) = 1 \wedge 0 = 0$ und damit $\phi_b((X_1 \wedge X_2) \wedge X_3) = 0 \wedge 1 = 0$. Gleichzeitig gilt $\phi_b(f_1(X_1, X_2)) = f_1(1, 0) = 1$. Damit folgt $\phi_b(((X_1 \wedge X_2) \wedge X_3) \vee f_1(X_1, X_2)) = \phi_b(((X_1 \wedge X_2) \wedge X_3)) \vee \phi_b(f_1(X_1, X_2)) = 0 \vee 1 = 1$.

Zu Aufgabe 1.7

Wir wenden die erste Regel unter (B3) an und erhalten $X_1(X_2 \vee X_3) \equiv X_1X_2 \vee X_1X_3$. Beim zweiten Ausdruck wenden wir zunächst die dritte Regel unter (B7) an und erhalten $X_1X_3 \equiv X_1 \wedge X_3 \wedge 1$. Nun ersetzen wir die 1 mittels der zweiten Regel unter (B6) und erhalten $X_1 \wedge X_3 \wedge 1 \equiv X_1 \wedge X_3 \wedge (X_2 \vee \bar{X}_2)$. Schließlich nutzen wir wieder die erste Regel unter (B3) und erhalten $X_1 \wedge X_3 \wedge (X_2 \vee \bar{X}_2) \equiv X_1X_2X_3 \vee X_1\bar{X}_2X_3$.

Zu Aufgabe 1.8

Wir formen den Ausdruck zunächst mittels der Morgan-Formel um und erhalten $\overline{X_1 X_2} \wedge (X_1 X_2)$. Wir ersetzen nun zur Verdeutlichung $X_1 X_2$ durch e und erhalten $\bar{e} \wedge e$, was aber nach Regel (B6) identisch zu Null ist. Damit kann es keine Einsetzung geben, unter der der Ausdruck den Wert 1 hat.

Zu Aufgabe 1.9

Nach Tabelle 1.2 gilt $\mathrm{Tr}(\vee) = \{(0,1), (1,0), (1,1)\}$. Damit gilt $m(0,1) = \bar{X}_1 X_2$, $m(1,0) = X_1 \bar{X}_2$, und $m(1,1) = X_1 X_2$. Schließlich ist die kanonische disjunktive Normalform von $\vee$:

$$\bar{X}_1 X_2 \vee X_1 \bar{X}_2 \vee X_1 X_2 \; .$$

Zu Aufgabe 1.10

Die gegebene Beschreibung mittels des Ausdrucks $X_1(X_2 \vee X_3)$ liefert bereits $L(f) \leq 2$. Hier sieht man auch, dass die Schranke aus Satz 1.7 mit $3 \cdot 2^4 = 48$ oft sehr unscharf ist.

Zu Aufgabe 1.11

Der Ausdruck $X_1(X_2 \vee X_3)$ ist eine konjunktive Normalform, da X_1 und $X_2 \vee X_3$ Klauseln sind. Er ist keine disjunktive Normalform, da er kein Monom darstellt. Durch Anwendung des Distributionsgesetzes (B3) kann man ihn aber in die disjunktive Normalform $X_1 X_2 \vee X_1 X_3$ umformen.

Zu Aufgabe 1.12

Die Funktion f mit $f(00) = f(11) = 0$, $f(01) = f(10) = 1$ heißt auch exklusives Oder. Wir prüfen für jedes der Monome, welchen Wert es an den Stellen annimmt, an denen die Funktion den Wert 0 annimmt, denn nur dort kann eine Verletzung der Implikanteneigenschaft '$\leq$' vorkommen. Das Monom $\bar{X}_1 X_2$ hat an den Stellen 00 und 11 den Wert 0, das Monom $X_1 \bar{X}_2$ ebenfalls. Diese beiden Monome sind also Implikanten. Das Monom $X_1 X_2$ hat an der Stelle 11 den Wert 1, also ist $X_1 X_2 \not\leq f$, und dieses Monom ist kein Implikant. Die beiden ersten Monome stellen auch Primimplikanten dar, denn ihre Teilmonome sind X_1, $\bar{X}_1$, X_2, und $\bar{X}_2$, und alle diese Ausdrücke sind keine Implikanten, da sie entweder an der Stelle 00 oder an der Stelle 11 den Wert 1 annehmen, die Funktion f hingegen nicht.

Zu Aufgabe 1.13

Die Wertetabelle ist in Tabelle 1.9 dargestellt. Der Träger der Funktion ist die Menge

$$\mathrm{Tr}(f) = \{0000, 0001, 0010, 0011, 0100, 0101, 0110, 1000, 1001, 1010, 1100\} \; .$$

$X_1X_2X_3X_4$	$f(X)$
0000	1
0001	1
0010	1
0011	1
0100	1
0101	1
0110	1
0111	X
1000	1
1001	1
1010	1
1011	X
1100	1
1101	X
1110	X
1111	0

Tabelle 1.9: Wertetabelle einer zu analysierenden Schaltfunktion f

Die kanonische disjunktive Normalform lautet

$$\begin{aligned} f(X) &= \bar{X}_1\bar{X}_2\bar{X}_3\bar{X}_4 \vee \bar{X}_1\bar{X}_2\bar{X}_3X_4 \vee \bar{X}_1\bar{X}_2X_3\bar{X}_4 \vee \bar{X}_1\bar{X}_2X_3X_4 \\ &\vee \bar{X}_1X_2\bar{X}_3\bar{X}_4 \vee \bar{X}_1X_2\bar{X}_3X_4 \vee \bar{X}_1X_2X_3\bar{X}_4 \vee X_1\bar{X}_2\bar{X}_3\bar{X}_4 \\ &\vee X_1\bar{X}_2\bar{X}_3X_4 \vee X_1\bar{X}_2X_3\bar{X}_4 \vee X_1X_2\bar{X}_3\bar{X}_4 \ . \end{aligned}$$

Die KDNF enthält 10 $\vee$-Operatoren, $33 = 11 \cdot 3$ $\wedge$-Operatoren, und 28 Inverter, ihre Kosten betragen also 71.

Das Karnaugh-Diagramm ist in Abbildung 1.9 abgebildet. Hier ist es sinnvoll, die mit X markierten Felder als 1 zu interpretieren, da sich hierdurch größere Rechtecke bilden lassen. Es ergeben sich vier Primimplikanten. Die erste und die letzte Zeile bilden ein 4×2-Rechteck, das dem Monom $\bar{X}_4$ entspricht. Die dritte und vierte Zeile bilden ein 4×2-Rechteck, das dem Monom $\bar{X}_2$ entspricht. Die erste und die letzte Spalte bilden ein 2×4-Rechteck, das dem Monom $\bar{X}_3$ entspricht. Die dritte und die vierte Spalte bilden ein 2×4-Rechteck, das dem Monom $\bar{X}_1$ entspricht. Jeder dieser Primimplikanten ist ein Kernimplikant. Das Minimalpolynom lautet

$$f(X) = \bigvee_{i=1}^{4} \bar{X}_i \ .$$

Zu Aufgabe 1.14

Der einzige Kernimplikant ist pi_1, da die Spalte m_1 als einzige nur eine 1 enthält. Die vereinfachte Primimplikantentafel entsteht durch Streichung der Spalten m_1 und m_2 sowie der Zeile pi_1 und ist in Abbildung 1.10 zu sehen.

X_1

X_2	1	X	1	1	
	X	0	X	1	X_4
	1	X	1	1	
	1	1	1	1	

X_3

Abbildung 1.9: Karnaugh-Diagramm

	m_3	m_4	m_5	m_6
pi_2	1	1		
pi_3	1	1	1	
pi_4			1	1
pi_5		1		1

Abbildung 1.10: Vereinfachte Primimplikantentafel zu Selbsttestaufg. 1.14

In dieser Tafel wird pi_2 durch pi_3 dominiert, und die betreffende Zeile kann weggelassen werden. Dann ist pi_3 aber wieder wesentlich, und wir können die Spalten m_3, m_4 und m_5 sowie die Zeile mit pi_3 entfernen. Übrig bleibt die Tabelle des Restproblems in Abbildung 1.11. Die beiden Primimplikanten dominieren sich gegenseitig. Da sie gleiche Kosten haben, wählen wir einen aus, zum Beispiel pi_4. Das resultierende Minimalpolynom ist

$$pi_1 \vee pi_3 \vee pi_4 \ .$$

	m_6
pi_4	1
pi_5	1

Abbildung 1.11: Primimplikantentafel des Restproblems zu Selbsttestaufg. 1.14