HANSER

Leseprobe

Dirk W. Hoffmann

Theoretische Informatik

ISBN (Buch): 978-3-446-44446-1

ISBN (E-Book): 978-3-446-44530-7

Weitere Informationen oder Bestellungen unter

http://www.hanser-fachbuch.de/978-3-446-44446-1

sowie im Buchhandel.

Vorwort

Für die meisten Menschen ist die Informatik fest mit der Entstehungsgeschichte des Computers verbunden; einer Technik, die von außen betrachtet keinen Grenzen zu unterliegen scheint. Wir erleben seit Jahren eine schier ungebremste Entwicklung und sind längst daran gewöhnt, dass der Computer von heute schon morgen überholt ist. Dass sich hinter der Computertechnik eine tiefgründige Wissenschaft verbirgt, die all die großen Erfolge erst möglich macht, bleibt vielen Menschen verborgen. Die Rede ist von der theoretischen Informatik.

In der Grundlagenausbildung hat die theoretische Informatik ihren festen Platz eingenommen. Viele Studierende begegnen ihr mit gemischten Gefühlen und von manchen wird sie gar als bedrohlich empfunden. Mitverantwortlich für diese Misere sind die historischen Wurzeln der theoretischen Informatik. Entstanden aus der Mathematik, wird sie häufig in einer Präzision dargestellt, die in der Informatik ihresgleichen sucht. Manch ein Leser verirrt sich schnell in einem Gewirr aus Definitionen, Sätzen und Beweisen, das die Sicht auf die eigentlichen Konzepte und Methoden unfreiwillig verdeckt. Dass die theoretische Informatik weder schwer noch trocken sein muss, versuche ich mit diesem Buch zu beweisen.

Die folgenden Kapitel werden von zwei Leitmotiven getragen. Zum einen möchte ich die grundlegenden Konzepte, Methoden und Ergebnisse der theoretischen Informatik vermitteln, ohne diese durch einen zu hohen Abstraktionsgrad zu vernebeln. Hierzu werden die Problemstellungen durchweg anhand von Beispielen motiviert und die Grundideen der komplizierteren Beweise an konkreten Probleminstanzen nachvollzogen. Zum anderen habe ich versucht, den Lehrstoff in vielerlei Hinsicht mit Leben zu füllen. An zahlreichen Stellen werden Sie Anmerkungen und Querbezüge vorfinden, die sich mit der historischen Entwicklung dieser einzigartigen Wissenschaftsdisziplin beschäftigen.

Bei allen Versuchen, einen verständlichen Zugang zu der nicht immer einfachen Materie zu schaffen, war es mir ein Anliegen, keinen Verlust an Tiefe zu erleiden. Das Buch ist für den Bachelor-Studiengang konzipiert und deckt die typischen Lehrinhalte ab, die im Grundstudium an den hiesigen Hochschulen und Universitäten unterrichtet werden.

Vorwort zur dritten Auflage

Mittlerweile ist die *Theoretische Informatik* in der dritten Auflage erschienen. Wie schon zur ersten Auflage habe ich auch zur zweiten Auflage von zahlreichen Lesern Zuschriften

erhalten, für die ich mich an dieser Stelle herzlich bedanke. Namentlich erwähnen möchte ich Dr. Klaus Fiedler, Dr. Bernd Grave, Herbert Rößе, Prof. Dr. Stephan Schulz und Prof. Dr. Rudolf Winkel, die mir mit zahlreichen Hinweisen geholfen haben, das Manuskript weiter zu verbessern.

Karlsruhe, im Juli 2015 Dirk W. Hoffmann

Symbolwegweiser

Definition

Satz, Lemma, Korollar

Leichte Übungsaufgabe

Mittelschwere Übungsaufgabe

Schwere Übungsaufgabe

Lösungen zu den Übungsaufgaben

In wenigen Schritten erhalten Sie die Lösungen zu den Übungsaufgaben:

1. Gehen Sie auf die Seite www.dirkwhoffmann.de/TH
2. Geben Sie den neben der Aufgabe abgedruckten Webcode ein
3. Die Musterlösung wird als PDF-Dokument angezeigt

Inhaltsverzeichnis

3 Logik und Deduktion

In diesem Kapitel werden Sie ...

- mit der Aussagenlogik die Grundlagen des logischen Schließens erlernen,
- formale Beweise im Hilbert-Kalkül führen,
- mit dem Resolutions- und dem Tableaukalkül zwei Widerspruchskalküle kennen lernen,
- die Aussagenlogik zur Prädikatenlogik erster Stufe erweitern,
- die Grundprinzipien der logischen Programmierung verstehen,
- einen Einblick in Logiken höherer Stufe erlangen.

In der formalen Logik haben wir es mit zwei Sprachebenen zu tun. Eine davon ist die *Objektebene*. Von hier aus betrachtet ist eine Logikformel nichts weiter als eine Zeichenkette, die nach formalen Bildungsregeln aufgebaut ist. Wie diese Regeln für die Aussagenlogik aussehen, haben wir in Definition 3.1 exakt festgelegt. Unter anderem ist die folgende Zeichenkette eine Formel der Aussagenlogik:

$$((A_1 \wedge A_2) \vee (\neg A_1 \wedge \neg A_2))$$

Im Folgenden werden wir die Logikebene mehrfach verlassen und Formeln von einem metatheoretischen Standpunkt aus betrachten. Beispielsweise könnte uns interessieren, unter welchen Bedingungen eine Formel erfüllbar ist, die sich als Disjunktion anderer Formeln darstellen lässt. Damit ist eine Formel der folgenden Bauart gemeint:

$$F \vee G$$

Anders als im ersten Beispiel ist diese Zeichenkette keine Formel der Aussagenlogik; sie ist eine Formel der *Metaebene*. Zu lesen ist sie als Schablone, die für eine Schar verschiedener Formeln steht. Erst wenn wir die Platzhalter F und G durch konkrete Teilausdrücke ersetzen, erhalten wir eine echte Formel.
Die Vermischung von Objekt- und Metaebene ist eine häufige Ursache von Verständnisschwierigkeiten im Bereich der Logik. Um Verwechslungen vorzubeugen, werden wir für die verschiedenen Formelbestandteile unterschiedlichen Schrifttypen verwenden. Alle Bestandteile der Objektebene werden in einer serifenlosen, steilen Schrift dargestellt ($\mathsf{A}, \mathsf{B}, \ldots$). Platzhalter notieren wir, wie es in der Mathematik üblich ist, weiterhin in einer kursiven Schrift ($F, G, \ldots$). Jetzt ist klar, warum die Symbole A_1 und A_2 in der ersten Formel steil, die Symbole F und G in der zweiten Formel dagegen kursiv dargestellt sind.

3.1 Aussagenlogik

Die Aussagenlogik (PL0) ist die einfachste Spielart des logischen Schließens. Gegenstand der Aussagenlogik sind *atomare Aussagen* („Es regnet“, „Die Straße ist nass“) und die Beziehungen, die zwischen solchen Aussagen bestehen („Wenn es regnet, dann ist die Straße nass“). Sie besticht durch einen einfachen Aufbau und eine klare Struktur, kommt jedoch nicht an die Ausdrucksstärke der anderen hier vorgestellten Logiken heran. Trotzdem ist die Bedeutung der Aussagenlogik beträchtlich. Zum einen ist sie die theoretische Grundlage des Hardware-Entwurfs, da sich das Verhalten einer kombinatorischen Hardware-Schaltung auf der Logikebene eins zu eins auf eine aussagenlogische Formel abbilden lässt. Zum anderen ist sie als Teilmenge in allen anderen Logiken enthalten und damit der kleinste gemeinsame Nenner, über den alle Logiken miteinander verbunden sind.

3.1.1 Syntax und Semantik

Wir nähern uns der Aussagenlogik in zwei Schritten. Zunächst fixieren wir die *Syntax*, d. h., wir definieren, nach welchen Regeln aussagenlogische Ausdrücke (Formeln) aufgebaut sind. Eine Formel ist in diesem Stadium nichts weiter als eine Folge von bedeutungsleeren Symbolen, die in einer festgelegten Art und Weise miteinander kombiniert werden dürfen. Erst die *Semantik* versieht die Ausdrücke mit einer Bedeutung; sie legt fest, wie wir die unterschiedlichen Zeichen und Symbole einer Logikformel zu interpretieren haben.

Definition 3.1 (Syntax der Aussagenlogik)

Die Menge der *aussagenlogischen Formeln* über dem *Variablenvorrat* $V = \{\mathsf{A}_1, \mathsf{A}_2, \ldots\}$ ist rekursiv definiert:

- 0 und 1 sind Formeln.
- Jede Variable $\mathsf{A}_i \in V$ ist eine Formel.
- Sind F und G Formeln, dann sind es auch

$$(\neg F), (F \wedge G), (F \vee G), (F \rightarrow G), (F \leftrightarrow G), (F \nleftrightarrow G).$$

Der Operator $\neg$ ist die *Negation*, $\wedge$ die *Konjunktion* (*UND-Operator*), $\vee$ die *Disjunktion* (*ODER-Operator*) und $\rightarrow$ die *Implikation*. Ferner be-

zeichnen wir $\leftrightarrow$ als *Äquivalenz-* und $\nleftrightarrow$ als *Antivalenzoperator* (*XOR-Operator*). Auch wenn die Namen bereits deutliche Hinweise geben, mit welcher Semantik wir die einzelnen Operatoren später belegen werden, sollten Sie versuchen, eine Formel zunächst als eine Aneinanderreihung von Symbolen zu betrachten, die noch keine konkrete Bedeutung besitzt.

Eine Formel F, die nicht weiter zerlegt werden kann, nennen wir *atomar*. In der Aussagenlogik ist die Menge der atomaren Formeln mit der Menge $V \cup \{0,1\}$ identisch. Eine Formel, die als Teil einer anderen Formel vorkommt, bezeichnen wir als *Teilformel*. Wir verwenden die etwas informelle Notation $F \in G$, um auszudrücken, dass F eine Teilformel von G ist; andernfalls schreiben wir $F \notin G$. Variablen werden wir im Folgenden durchweg mit Großbuchstaben bezeichnen, allerdings von Fall zu Fall den Symbolvorrat anpassen (z. B. $\mathsf{X}, \mathsf{Y}, \mathsf{Z}$ anstelle von $\mathsf{A}_1, \mathsf{A}_2, \mathsf{A}_3$).

Für die eingeführten Operatoren existiert keine einheitliche Notation. Zum einen wurde die Schreibweise in der Vergangenheit mehrfach geändert, zum anderen ist sie auch heute noch regional verschieden. Im oberen Teil von Abbildung 3.1 sind die Symbole zusammengefasst, wie sie zu Beginn des zwanzigsten Jahrhunderts üblich waren und zum Verständnis alter Forschungsbeiträge notwendig sind (vgl. Abbildung 3.2). Der untere Teil enthält die Symbole, wie sie insbesondere im US-amerikanischen Sprachraum verwendet werden. Die Notation $\overline{F}$ für $\neg F$ hat sich auch im deutschen Sprachraum etabliert, da sie für viele Ausdrücke zu einer übersichtlicheren Darstellung führt.

Um die Lesbarkeit zu verbessern, werden wir auf die Angabe mancher Klammerpaare verzichten. Mehrdeutigkeiten werden, wie in Abbildung 3.3 gezeigt, über eine Reihe von Bindungs- und Kettenregeln beseitigt. Bindungsregeln teilen die Operatoren in schwächer bindende und stärker bindende Operatoren ein, Kettenregeln bestimmen den Umgang mit Ausdrücken, in denen der gleiche Operator mehrmals hintereinander vorkommt. Wir legen die übliche Konvention zugrunde, dass die Negation ($\neg$) stärker bindet als die Konjunktion ($\wedge$). Diese bindet wiederum stärker als die Disjunktion ($\vee$). Die Operatoren $\rightarrow$, $\leftrightarrow$ und $\nleftrightarrow$ binden am schwächsten. Kommen sie in einem Ausdruck gemischt vor, erfolgt die Klammerung linksassoziativ. Werden Teilterme mit demselben Operator verknüpft, betrachten wir die entstehende Kette ebenfalls als linksassoziativ geklammert. Die einzige Ausnahme bildet der (einstellige) Negationsoperator, den wir rechtsassoziativ gruppieren.

Die zweistelligen Operatoren $\wedge$ und $\vee$ lassen sich auf natürliche Weise zu mehrstelligen Operatoren verallgemeinern. Hierzu vereinbaren wir

- Frühere Schreibweise

Formel	Schreibweise
$\neg F$	$\sim F$
$F \wedge G$	$F \,.\, G$
$F \vee G$	$F \vee G$
$F \rightarrow G$	$F \supset G$

- Punkt-Strich-Notation

Formel	Schreibweise
$\neg F$	$-F$, $\overline{F}$
$F \wedge G$	$F \cdot G$, FG
$F \vee G$	$F + G$
$F \rightarrow G$	$F \rightarrow G$

Abbildung 3.1: Alternative Schreibweisen aussagenlogischer Formeln

- Drei Formeln der Principia …

$*2{\cdot}03.\quad \vdash : p \supset \sim q \,.\supset.\, q \supset \sim p$
$*2{\cdot}15.\quad \vdash : \sim p \supset q \,.\supset.\, \sim q \supset p$
$*2{\cdot}16.\quad \vdash : p \supset q \,.\supset.\, \sim q \supset \sim p$
$*2{\cdot}17.\quad \vdash : \sim q \supset \sim p \,.\supset.\, p \supset q$

- … und deren moderne Schreibweise

(2.03) $\vdash (\mathsf{p} \rightarrow \neg\mathsf{q}) \rightarrow (\mathsf{q} \rightarrow \neg\mathsf{p})$
(2.15) $\vdash (\neg\mathsf{p} \rightarrow \mathsf{q}) \rightarrow (\neg\mathsf{q} \rightarrow \mathsf{p})$
(2.16) $\vdash (\mathsf{p} \rightarrow \mathsf{q}) \rightarrow (\neg\mathsf{q} \rightarrow \neg\mathsf{p})$
(2.17) $\vdash (\neg\mathsf{q} \rightarrow \neg\mathsf{p}) \rightarrow (\mathsf{p} \rightarrow \mathsf{q})$

Abbildung 3.2: In der Notation der Principia besitzt der Punkt eine Doppelbedeutung. In Abhängigkeit von seiner Position wird er für die konjunktive Verknüpfung oder zum Klammern von Teilausdrücken verwendet. Die Abbildung zeigt vier Formeln aus der Originalausgabe der Principia Mathematica sowie deren Übersetzung in die heute übliche Schreibweise.

- Bindungsregeln

Ausdruck	Bedeutung
$\neg F \wedge G$	$((\neg F) \wedge G)$
$F \vee G \wedge H$	$(F \vee (G \wedge H))$
$F \vee G \rightarrow H$	$((F \vee G) \rightarrow H)$
$F \rightarrow G \nleftrightarrow H$	$((F \rightarrow G) \nleftrightarrow H)$
$F \nleftrightarrow G \rightarrow H$	$((F \nleftrightarrow G) \rightarrow H)$

- Kettenregeln

Ausdruck	Bedeutung
$\neg\neg\neg F$	$\neg(\neg(\neg F))$
$F \wedge G \wedge H$	$(F \wedge G) \wedge H$
$F \vee G \vee H$	$(F \vee G) \vee H$
$F \rightarrow G \rightarrow H$	$(F \rightarrow G) \rightarrow H$
$F \leftrightarrow G \leftrightarrow H$	$(F \leftrightarrow G) \leftrightarrow H$
$F \nleftrightarrow G \nleftrightarrow H$	$(F \nleftrightarrow G) \nleftrightarrow H$

Abbildung 3.3: Zur Vereinfachung der Schreibweise dürfen Klammerpaare weggelassen werden. Zweideutigkeiten werden mithilfe von Bindungs- und Kettenregeln beseitigt. Erstere teilen die Operatoren in schwächer bindende und stärker bindende Operatoren ein, Letztere regeln den Umgang mit Ausdrücken, in denen der gleiche Operator mehrmals hintereinander vorkommt.

für die endlich vielen Formeln $F_1, \ldots, F_n$ die folgende Schreibweise:

$$\left(\bigwedge_{i=1}^{n} F_i\right) := F_1 \wedge \ldots \wedge F_n \qquad \left(\bigvee_{i=1}^{n} F_i\right) := F_1 \vee \ldots \vee F_n$$

Die Semantik einer aussagenlogischen Formel wird über die *Modellrelation* $\models$ festgelegt. Um diese formal definieren zu können, müssen wir vorab klären, was wir unter dem Begriff der *Interpretation* zu verstehen haben:

Definition 3.2 (Interpretation)

Sei F eine aussagenlogische Formel. $\mathsf{A}_1, \ldots, \mathsf{A}_n$ bezeichnen die in F vorkommenden Variablen. Jede Abbildung

$$I : \{\mathsf{A}_1, \ldots, \mathsf{A}_n\} \rightarrow \{0, 1\}$$

heißt eine *Interpretation* von F.

Eine Interpretation ordnet jeder Variablen einer aussagenlogischen Formel F einen der beiden Wahrheitswerte 0 (*falsch*) oder 1 (*wahr*) zu und wird aufgrund dieser Eigenschaft auch als *Belegung* bezeichnet.

Der Begriff der Interpretation ist die Grundlage für die formale Definition der Semantik:

Definition 3.3 (Semantik der Aussagenlogik)

F und G seien aussagenlogische Formeln und I eine Interpretation. Die Semantik der Aussagenlogik ist durch die *Modellrelation* $\models$ gegeben, die induktiv definiert ist:

$$\begin{aligned}
I &\models 1 \\
I &\not\models 0 \\
I \models \mathsf{A}_i &:\Leftrightarrow I(\mathsf{A}_i) = 1 \\
I \models (\neg F) &:\Leftrightarrow I \not\models F \\
I \models (F \wedge G) &:\Leftrightarrow I \models F \text{ und } I \models G \\
I \models (F \vee G) &:\Leftrightarrow I \models F \text{ oder } I \models G \\
I \models (F \rightarrow G) &:\Leftrightarrow I \not\models F \text{ oder } I \models G \\
I \models (F \leftrightarrow G) &:\Leftrightarrow I \models F \text{ genau dann, wenn } I \models G \\
I \models (F \nleftrightarrow G) &:\Leftrightarrow I \not\models (F \leftrightarrow G)
\end{aligned}$$

Eine Interpretation I mit $I \models F$ heißt *Modell* für F.

Wir können jede aussagenlogische Formel F mit n Variablen als *boolesche Funktion* $f^F : \{0,1\}^n \to \{0,1\}$ auffassen, die für eine Belegung I genau dann den Funktionswert 1 annimmt, wenn I ein Modell für F ist. Mit anderen Worten: Weist I den Variablen $\mathsf{A}_1, \ldots, \mathsf{A}_n$ die Wahrheitswerte $b_1, \ldots, b_n$ zu, dann ist der Funktionswert $f^F(b_1, \ldots, b_n)$ durch die folgende Formel bestimmt:

$$f^F(b_1, \ldots, b_n) = \begin{cases} 1 & \text{falls } I \models F \\ 0 & \text{falls } I \not\models F \end{cases}$$

Aufgrund des diskreten Definitionsbereichs lässt sich eine n-stellige boolesche Funktion in Form einer *Wahrheitstabelle* darstellen, indem alle möglichen Kombinationen der Eingangsvariablen $\mathsf{A}_1, \ldots, \mathsf{A}_n$ zusammen mit dem zugeordneten Funktionswert zeilenweise aufgelistet werden. Als Beispiele zeigt Abbildung 3.4 die Wahrheitstafeln der eingeführten aussagenlogischen Operatoren. Die Funktionswerte erschließen sich unmittelbar aus Definition 3.3. Wahrheitstabellen werden in der Literatur auch als *Wahrheitstafeln* oder *Funktions(wert)tabellen* bezeichnet; alle diese Begriffe bezeichnen die gleiche tabellarische Beschreibungsweise einer booleschen Funktion.

Abbildung 3.5 zeigt, wie sich Wahrheitstafeln für zusammengesetzte Ausdrücke erzeugen lassen. Ausgehend von den Basistermen werden zunächst die Teilformeln und anschließend der Gesamtausdruck ausgewertet. Die drei Beispiele wurden bewusst gewählt. Die Formel F_1 ist so beschaffen, dass sie genau zwei Modelle besitzt; sie ist genau dann wahr, wenn die Variablen A, B, C mit dem gleichen Wahrheitswert belegt werden. In der Terminologie der Aussagenlogik wird die Funktion als *erfüllbar* bezeichnet. F_2 ist ebenfalls erfüllbar, besitzt aber im Gegensatz zu F_1 die Eigenschaft, dass ausnahmslos alle Variablenbelegungen ein Modell sind. Solche Formeln heißen *allgemeingültig*. In entsprechender Weise bezeichnen wir F_3 als *unerfüllbare* Formel, da sie kein einziges Modell besitzt. Formal halten wir das Gesagte in der folgenden Definition fest:

Definition 3.4 (Erfüllbarkeit, Allgemeingültigkeit)

Eine aussagenlogische Formel F heißt

- *erfüllbar*, falls F mindestens ein Modell besitzt,
- *unerfüllbar*, falls F kein Modell besitzt,
- *allgemeingültig*, falls $\neg F$ unerfüllbar ist.

Eine allgemeingültige Formel bezeichnen wir auch als *Tautologie*.

- Negation

	A	¬A
0	0	1
1	1	0

- Konjunktion

	A	B	A ∧ B
0	0	0	0
1	0	1	0
2	1	0	0
3	1	1	1

- Disjunktion

	A	B	A ∨ B
0	0	0	0
1	0	1	1
2	1	0	1
3	1	1	1

- Implikation

	A	B	A → B
0	0	0	1
1	0	1	1
2	1	0	0
3	1	1	1

- Äquivalenz

	A	B	A ↔ B
0	0	0	1
1	0	1	0
2	1	0	0
3	1	1	1

- Antivalenz

	A	B	A ↮ B
0	0	0	0
1	0	1	1
2	1	0	1
3	1	1	0

Abbildung 3.4: Wahrheitstabellen der aussagenlogischen Basisoperatoren

- Beispiel 1: $F_1 = \underbrace{\underbrace{(\overline{A} \vee B)}_{G_1} \wedge \underbrace{(\overline{B} \vee C)}_{G_2}}_{G_4} \wedge \underbrace{(\overline{C} \vee A)}_{G_3}$

A	B	C	G_1	G_2	G_3	G_4	F_1
0	0	0	1	1	1	1	1
0	0	1	1	1	0	1	0
0	1	0	1	0	1	0	0
0	1	1	1	1	0	1	0
1	0	0	0	1	1	0	0
1	0	1	0	1	1	0	0
1	1	0	1	0	1	0	0
1	1	1	1	1	1	1	1

- Beispiel 2: $F_2 = \underbrace{\underbrace{((A \to B)}_{G_1} \wedge \underbrace{(B \to C))}_{G_2}}_{G_4} \to \underbrace{(A \to C)}_{G_3}$

A	B	C	G_1	G_2	G_3	G_4	F_2
0	0	0	1	1	1	1	1
0	0	1	1	1	1	1	1
0	1	0	1	0	1	0	1
0	1	1	1	1	1	1	1
1	0	0	0	1	0	0	1
1	0	1	0	1	1	0	1
1	1	0	1	0	0	0	1
1	1	1	1	1	1	1	1

- Beispiel 3: $F_3 = \underbrace{\underbrace{(A \leftrightarrow B)}_{G_1} \wedge \underbrace{(A \leftrightarrow C)}_{G_2}}_{G_4} \wedge \underbrace{(B \leftrightarrow C)}_{G_3}$

A	B	C	G_1	G_2	G_3	G_4	F_3
0	0	0	0	0	0	0	0
0	0	1	0	1	1	0	0
0	1	0	1	0	1	0	0
0	1	1	1	1	0	1	0
1	0	0	1	1	0	1	0
1	0	1	1	0	1	0	0
1	1	0	0	1	1	0	0
1	1	1	0	0	0	0	0

Abbildung 3.5: Wahrheitstabellen zusammengesetzter Funktionen

Abbildung 3.6 demonstriert den Zusammenhang zwischen den eingeführten Begriffen. Alle drei lassen sich in naheliegender Weise auf Mengen von aussagenlogischen Formeln erweitern. Die Formelmenge $M = \{F_1, \ldots, F_n\}$ heißt erfüllbar, wenn eine Interpretation I existiert, die für alle $F_i \in M$ ein Modell ist. Beachten Sie, dass das Modell für alle Formeln das gleiche sein muss; es reicht also nicht aus, dass jede Formel

für sich erfüllbar ist. Die Unerfüllbarkeit und Allgemeingültigkeit von Formelmengen definieren wir analog. M ist unerfüllbar, wenn $F_1, \ldots, F_n$ kein gemeinsames Modell besitzen. Ist dagegen jede Interpretation ein Modell für die Elemente von M, so nennen wir M allgemeingültig.

Mithilfe der Modellrelation können wir den Begriff der *logischen Folgerung* formal definieren:

Definition 3.5 (Logische Folgerung)

$M := \{F_1, \ldots, F_n\}$ sei eine Menge aussagenlogischer Formeln. Wir schreiben

$$M \models G \quad (\text{„}aus\ M\ folgt\ G\text{“}),$$

wenn jedes Modell von M auch ein Modell der aussagenlogischen Formel G ist. Ferner vereinbaren wir die Kurzschreibweise $\models G$ für $\emptyset \models G$ und $F \models G$ für $\{F\} \models G$.

Es gelten die folgenden Zusammenhänge:

- $\models G$ gilt genau dann, wenn G allgemeingültig ist.
- $F \models G$ gilt genau dann, wenn $F \to G$ allgemeingültig ist.
- $\{F_1, F_2, \ldots, F_n\} \models G$ ist äquivalent zu $\{F_2, \ldots, F_n\} \models F_1 \to G$.

In den kommenden Betrachtungen wird der Begriff der *Äquivalenz* immer wieder auftauchen:

Definition 3.6 (Äquivalenz)

Seien F und G zwei aussagenlogische Formeln. Die Relation $\equiv$ ist wie folgt definiert:

$$F \equiv G \;:\Leftrightarrow\; F \models G \text{ und } G \models F$$

Zwei Formeln F und G mit $F \equiv G$ heißen *äquivalent*.

Damit sind zwei Formeln F und G genau dann äquivalent, geschrieben als $F \equiv G$, falls sie exakt dieselben Modelle besitzen. Im mathematischen Sinn ist $\equiv$ eine Äquivalenzrelation auf der Menge der aussagenlogischen Formeln und besitzt die Eigenschaften der Reflexivität, Symmetrie und Transitivität:

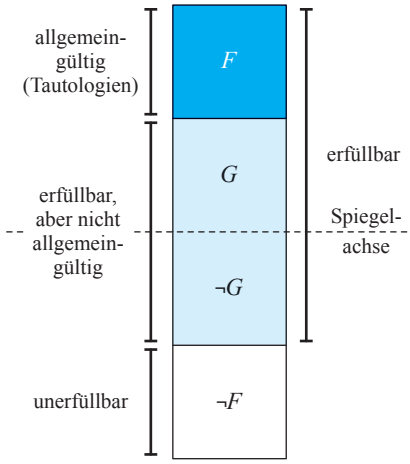

Abbildung 3.6: Das Spiegelungsprinzip visualisiert, wie sich die Eigenschaften der Formeln F und $\neg F$ gegenseitig beeinflussen. Ist F allgemeingültig, so ist $\neg F$ unerfüllbar. Ist F nicht allgemeingültig, aber dennoch erfüllbar, so gilt das Gleiche für $\neg F$. Damit ist die Allgemeingültigkeit eine exklusive Eigenschaft, die nur eine der beiden Formeln F oder $\neg F$ erfüllen kann. Im Gegensatz hierzu können sowohl F als auch $\neg F$ erfüllbar sein.

Kommutativität	Neutralität
$F \wedge G \equiv G \wedge F$ $F \vee G \equiv G \vee F$	$F \wedge 1 \equiv F$ $F \vee 0 \equiv F$
Distributivität	**Inversion**
$F \wedge (G \vee H) \equiv (F \wedge G) \vee (F \wedge H)$ $F \vee (G \wedge H) \equiv (F \vee G) \wedge (F \vee H)$	$F \wedge \neg F \equiv 0$ $F \vee \neg F \equiv 1$
Assoziativität	**Elimination**
$F \wedge (G \wedge H) \equiv (F \wedge G) \wedge H$ $F \vee (G \vee H) \equiv (F \vee G) \vee H$	$F \wedge 0 \equiv 0$ $F \vee 1 \equiv 1$
De Morgan'sche Regeln	**Idempotenz**
$\neg(F \wedge G) \equiv \neg F \vee \neg G$ $\neg(F \vee G) \equiv \neg F \wedge \neg G$	$F \wedge F \equiv F$ $F \vee F \equiv F$
Absorption	**Doppelnegation**
$F \wedge (F \vee G) \equiv F$ $F \vee (F \wedge G) \equiv F$	$\neg\neg F \equiv F$

Tabelle 3.1: Wichtige Äquivalenzen aussagenlogischer Ausdrücke

- Reflexivität: Für alle Formeln F gilt $F \equiv F$.
- Symmetrie: Aus $F \equiv G$ folgt $G \equiv F$.
- Transitivität: Aus $F \equiv G$ und $G \equiv H$ folgt $F \equiv H$.

Es gelten die folgenden Zusammenhänge:

- F ist genau dann allgemeingültig, wenn $F \equiv 1$.
- F ist genau dann unerfüllbar, wenn $F \equiv 0$.

In Tabelle 3.1 sind wichtige Äquivalenzen zusammengefasst, die sich durch das Aufstellen von Wahrheitstafeln leicht verifizieren lassen. Die Bedeutung dieser Äquivalenzen ist zweigeteilt. Zum einen gestatten sie einen Einblick in die elementaren Zusammenhänge zwischen den eingeführten booleschen Operatoren, zum anderen dienen sie als wichtige Umformungsregeln für aussagenlogische Ausdrücke. Grundlage hierfür

ist das *Substitutionstheorem*, das uns gestattet, Teilformeln durch äquivalente Ausdrücke zu ersetzen, ohne die Modelle der Gesamtformel zu beeinflussen.

Satz 3.1 (Substitutionstheorem)

Seien F, G, G' aussagenlogische Formeln mit $G \in F$ und $G \equiv G'$.

$$F[G \leftarrow G']$$

bezeichnet diejenige Formel, die aus F entsteht, indem die Teilformel G durch G' ersetzt wird. Dann gilt:

$$F \equiv F[G \leftarrow G']$$

Mit dem Mittel der strukturellen Induktion aus Abschnitt 2.4.2 lässt sich das Substitutionstheorem induktiv über den Aufbau aussagenlogischer Formeln beweisen.

Vielleicht haben Sie sich gewundert, dass Tabelle 3.1 ausschließlich Rechenregeln für die aussagenlogischen *Elementaroperatoren* $\neg$, $\wedge$ und $\vee$ enthält. Hierbei handelt es sich um keine Einschränkung im eigentlichen Sinne, da sich alle anderen Operatoren auf diese drei zurückführen lassen (vgl. Tabelle 3.2). Die Menge $\{\neg, \wedge, \vee\}$ ist zudem ein *vollständiges Operatorensystem*, d. h., sie besitzt die Eigenschaft, dass sich jede boolesche Funktion durch einen aussagenlogischen Ausdruck beschreiben lässt, in dem ausschließlich Operatoren aus dieser Menge vorkommen. Ein Beweis der Vollständigkeit wird uns im nächsten Abschnitt ohne Zutun in die Hände fallen. Dort werden wir zeigen, wie sich eine entsprechende aussagenlogische Formel systematisch aus der Wahrheitstabelle einer booleschen Funktion erzeugen lässt.

Neben der Menge der Elementaroperatoren existieren weitere vollständige Operatorensysteme, wie z. B. die Menge $\{\neg, \rightarrow\}$. Um die Vollständigkeit zu beweisen, nutzen wir unser Wissen, dass die drei Elementaroperatoren $\wedge$, $\vee$ und $\neg$ zusammen ein vollständiges Operatorensystem bilden. Können wir zeigen, dass $\wedge$ und $\vee$ durch $\neg$ und $\rightarrow$ darstellbar sind, so lässt sich jede boolesche Funktion mit einem aussagenlogischen Ausdruck beschreiben, der ausschließlich die Operatoren $\neg$ und $\rightarrow$ enthält. Kurzum: $\{\neg, \rightarrow\}$ bildet dann ebenfalls ein vollständiges Operatorensystem. Tabelle 3.3 zeigt, wie die notwendigen Reduktionen durchgeführt werden können.

Mithilfe der Aussagenlogik lassen sich viele der kombinatorischen Zusammenhänge beschreiben, die wir im Bereich des mathematischen

Implikation		
$F \rightarrow G$	$\equiv$	$\neg F \vee G$

Äquivalenz		
$F \leftrightarrow G$	$\equiv$	$(\neg F \wedge \neg G) \vee (F \wedge G)$
	$\equiv$	$(\neg F \vee G) \wedge (F \vee \neg G)$

Antivalenz		
$F \nleftrightarrow G$	$\equiv$	$(\neg F \wedge G) \vee (F \wedge \neg G)$
	$\equiv$	$(\neg F \vee \neg G) \wedge (F \vee G)$

Tabelle 3.2: Reduktion der Operatoren $\rightarrow$, $\leftrightarrow$ und $\nleftrightarrow$ auf die Elementaroperatoren $\neg$, $\wedge$ und $\vee$

Reduktion von $\wedge$ auf $\{\rightarrow, \neg\}$		
$F \wedge G$	$\equiv$	$\neg(\neg F \vee \neg G)$
	$\equiv$	$\neg(F \rightarrow \neg G)$

Reduktion von $\vee$ auf $\{\rightarrow, \neg\}$		
$F \vee G$	$\equiv$	$\neg\neg F \vee G$
	$\equiv$	$\neg F \rightarrow G$

Tabelle 3.3: Reduktion der Elementaroperatoren $\wedge$ und $\vee$ auf $\neg$ und $\rightarrow$

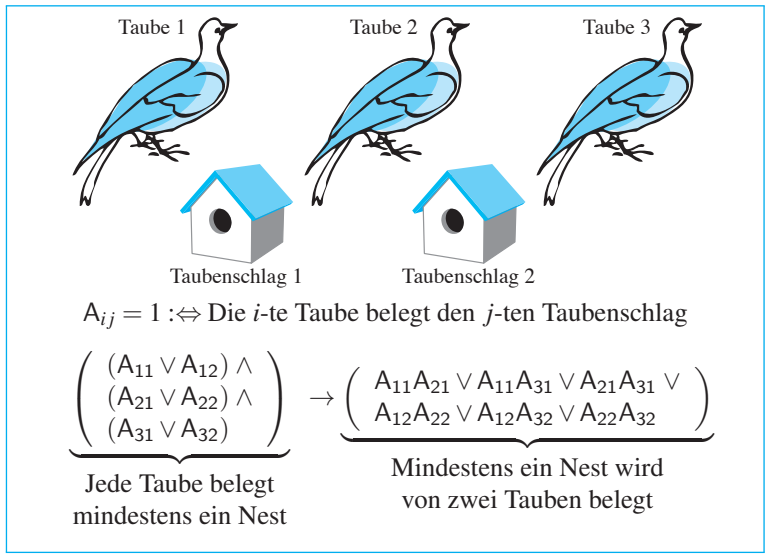

Abbildung 3.7: Das *Dirichlet'sche Schubfachprinzip* ist im angelsächsischen Raum unter dem Namen *pigeonhole principle* (*Taubenschlagprinzip*) bekannt. Verteilen sich, wie in diesem Beispiel, 3 Tauben auf 2 Taubenschläge, so muss mindestens ein Taubenschlag doppelt belegt werden. Mithilfe der Aussagenlogik lässt sich das Prinzip formalisieren und beweisen.

Schließens tagtäglich anwenden. Als Beispiel zeigt Abbildung 3.7 eine Formalisierung des *Dirichlet'schen Schubfachprinzips*. Dieses besagt, dass eine endliche Menge M nicht injektiv auf eine Menge N abgebildet werden kann, wenn N weniger Elemente enthält als M. Das Prinzip ist nach dem deutschen Mathematiker Johann Dirichlet (Abbildung 3.8) benannt und ein häufig angewandtes Beweisargument in der diskreten Mathematik. Jedem von uns ist das Schubfachprinzip aus dem Alltag geläufig. Verteilen wir m Gegenstände auf n Schubfächer und gilt $m > n$, so muss mindestens ein Schubfach mehrere Gegenstände enthalten. Im angelsächsischen Raum wird das Dirichlet'sche Schubfachprinzip als *pigeonhole principle* (*Taubenschlagprinzip*) bezeichnet. Auch hier ist die angestellte Überlegung die gleiche: Verteilen sich m Tauben auf n Nester und gilt $m > n$, so ist mindestens ein Taubenschlag mehrfach besetzt.

Mithilfe der Aussagenlogik können wir das Schubfachprinzip formalisieren und beweisen. Hierzu führen wir für jede mögliche Kombination von Gegenständen und Schubfächern eine aussagenlogische Variable A_{ij} ein, die genau dann den Wert 1 annimmt, wenn sich der i-te Gegenstand im j-ten Schubfach befindet. Um das Schubfachprinzip für n Gegenstände und $n-1$ Schubfächer zu formalisieren, benötigen wir $n \cdot (n-1)$ Variablen. Wir werden nun Schritt für Schritt herausarbeiten, wie sich das Schubfachprinzip mithilfe der booleschen Operatoren $\neg$, $\wedge$, $\vee$ und $\rightarrow$ formal nachbilden lässt:

- „Das i-te Element befindet sich in einem der Schubfächer“

$$\bigvee_{j=1}^{n-1} A_{ij}$$

- „Jedes Element befindet sich in einem der Schubfächer“

$$\bigwedge_{i=1}^{n} \bigvee_{j=1}^{n-1} A_{ij} \tag{3.1}$$

- „In Schubfach j befinden sich das i-te und k-te Element gleichzeitig“

$$A_{ij} \wedge A_{kj}$$

- „In Schubfach j befinden sich mindestens zwei Elemente“

$$\bigvee_{i=1}^{n-1} \bigvee_{k=i+1}^{n} (A_{ij} \wedge A_{kj})$$

- „In einem Schubfach befinden sich mindestens zwei Elemente“

$$\bigvee_{j=1}^{n-1} \bigvee_{i=1}^{n-1} \bigvee_{k=i+1}^{n} (A_{ij} \wedge A_{kj}) \tag{3.2}$$

Verbinden wir die Formeln (3.1) und (3.2) mit dem Implikationsoperator, so erhalten wir die Aussage des Schubfachprinzips: Befinden sich n Elemente in $n-1$ Schubfächern, so enthält eines der Schubfächer mindestens zwei Elemente:

$$\bigwedge_{i=1}^{n} \bigvee_{j=1}^{n-1} A_{ij} \rightarrow \bigvee_{j=1}^{n-1} \bigvee_{i=1}^{n-1} \bigvee_{k=i+1}^{n} (A_{ij} \wedge A_{kj}) \tag{3.3}$$

Die in Abbildung 3.7 dargestellte Formel erhalten wir aus Formel (3.3) für den Spezialfall $n = 3$.

3.1.2 Normalformen

Weiter oben haben wir herausgearbeitet, dass wir eine aussagenlogische Formel F als boolesche Funktion interpretieren können, indem wir jede Interpretation I als Eingabebelegung auffassen und der Funktion genau dann den Wert 1 zuweisen, wenn I ein Modell für F ist. Die enge Beziehung, die zwischen aussagenlogischen Formeln auf der einen Seite

Johann Peter Gustav Lejeune Dirichlet (1805 – 1859)

Abbildung 3.8: Das Schubfachprinzip erhielt seinen Namen durch Johann Peter Gustav Lejeune Dirichlet. Der deutsche Mathematiker wurde im nordrhein-westfälischen Düren geboren, damals Teil des napoleonischen Kaiserreiches. 1822 nahm er das Studium der Mathematik in Paris auf. Bereits drei Jahre später demonstrierte er der Wissenschaftsgemeinde das erste Mal sein Talent, als er die Gültigkeit der Fermat'schen Vermutung für den Fall $n = 5$ bewies. Im Jahr 1826 kehrte er nach Deutschland zurück und graduierte 1827 an der Universität Bonn. Seine akademische Karriere führte ihn über Breslau und Berlin nach Göttingen, wo er im Jahr 1855 den Lehrstuhl von Carl-Friedrich Gauß übernahm. Dort sollte ihm das Schicksal nur eine kurze Schaffenszeit gewähren. Am 5. Mai 1859, nur fünf Monate nach seiner Frau Rebecca Mendelssohn Bartholdy, verstarb er im Alter von 54 Jahren. Dirichlet zählt zu den großen Mathematikern des neunzehnten Jahrhunderts und machte sich vor allem in den Gebieten der partiellen Differentialgleichungen und der algebraischen Zahlentheorie verdient. Zu seinen bedeutendsten Hinterlassenschaften gehört der Dirichlet'sche Einheitensatz, die Dirichlet-Funktion und die Dirichlet'sche Eta-Funktion.

■ Beispiel 1

$$F := (\mathsf{A} \wedge \mathsf{B}) \vee \mathsf{A}$$
$$G := \mathsf{A}$$

	A	B	F	G
0	0	0	0	0
1	0	1	0	0
2	1	0	1	1
3	1	1	1	1

■ Beispiel 2

$$F := (\mathsf{A} \nleftrightarrow \mathsf{B}) \vee (\mathsf{A} \leftrightarrow \mathsf{B})$$
$$G := 1$$

	A	B	F	G
0	0	0	1	1
1	0	1	1	1
2	1	0	1	1
3	1	1	1	1

Abbildung 3.9: Zwei aussagenlogische Formeln können selbst dann äquivalent sein, wenn sie unterschiedliche Variablen enthalten. Wie die Wahrheitstafeln belegen, besitzen F und G jeweils die gleichen Modelle.

und booleschen Funktionen auf der anderen besteht, ist nicht eindeutig. So repräsentieren die Formeln

$$F_1 := \mathsf{A}$$
$$F_2 := \mathsf{A} \wedge \mathsf{A}$$
$$F_3 := \mathsf{A} \wedge \mathsf{A} \wedge \mathsf{A}$$
$$\ldots$$
$$F_n := \mathsf{A} \wedge \mathsf{A} \wedge \ldots \wedge \mathsf{A}$$

allesamt die gleiche boolesche Funktion. Die äußere Form lässt also keinerlei Rückschluss zu, ob zwei boolesche Formeln äquivalent zueinander sind oder nicht. Wie die Beispiele in Abbildung 3.9 zeigen, können zwei Formeln sogar dann äquivalent sein, wenn sie unterschiedliche Variablen enthalten.

Um diesem Problem zu begegnen, wurden in der Vergangenheit mehrere *Normalformen* entwickelt, die eine Eins-zu-eins-Beziehung zwischen aussagenlogischen Formeln und booleschen Funktionen herstellen. Wichtige Normalformen sind die *kanonische konjunktive* und die *kanonische disjunktive Normalform*. Um diese formal zu definieren, benötigen wir die folgenden Hilfsbegriffe:

Definition 3.7 (Literal, Minterm, Maxterm)

Sei F eine aussagenlogische Formel mit den Variablen $\mathsf{A}_1, \ldots, \mathsf{A}_n$.

- Jedes Vorkommen einer Variablen A_i oder ihrer Negation $\neg\mathsf{A}_i$ bezeichnen wir als *Literal*, geschrieben als $(\neg)\mathsf{A}_i$ oder L_i.
- Jeder Ausdruck der Form $(\neg)\mathsf{A}_1 \wedge \ldots \wedge (\neg)\mathsf{A}_n$ heißt *Minterm*.
- Jeder Ausdruck der Form $(\neg)\mathsf{A}_1 \vee \ldots \vee (\neg)\mathsf{A}_n$ heißt *Maxterm*.

Hat ein Literal L die Form $\neg A$, so sprechen wir von einem *negativen*, andernfalls von einem *positiven Literal*. Minterme und Maxterme besitzen die Eigenschaft, dass sämtliche Variablen einer Funktion konjunktiv bzw. disjunktiv miteinander verknüpft werden. Hieraus ergeben sich die folgenden beiden Eigenschaften:

- Ein Minterm ist für genau eine Variablenbelegung wahr.
- Ein Maxterm ist für genau eine Variablenbelegung falsch.

Mithilfe der eingeführten Begriffe definieren wir die kanonische disjunktive und die kanonische konjunktive Normalform wie folgt:

Definition 3.8 (KKNF und KDNF)

Sei F eine aussagenlogische Formel mit den Variablen

$$\mathsf{A}_1, \ldots, \mathsf{A}_n$$

- F liegt in *kanonischer konjunktiver Normalform* (KKNF) vor, wenn sie die Form

$$F = \bigwedge_{i=1}^{m} ((\neg)\mathsf{A}_1 \vee \ldots \vee (\neg)\mathsf{A}_n)$$

besitzt und alle Maxterme paarweise verschieden sind.

- F liegt in *kanonischer disjunktiver Normalform* (KDNF) vor, wenn sie die Form

$$F = \bigvee_{i=1}^{m} ((\neg)\mathsf{A}_1 \wedge \ldots \wedge (\neg)\mathsf{A}_n)$$

besitzt und alle Minterme paarweise verschieden sind.

Die KKNF entspricht einer Kette UND-verknüpfter Maxterme und die KDNF einer Kette ODER-verknüpfter Minterme. Aus der Wahrheitstabelle einer aussagenlogischen Formel F lassen sich die kanonischen Normalformen auf einfache Weise ableiten. Für die KKNF wird für jede Belegung I mit $I \not\models F$ ein Maxterm erzeugt. Anschließend werden diese konjunktiv miteinander verknüpft (vgl. Tabelle 3.4 links). Die KDNF entsteht analog, indem für jede Belegung I mit $I \models F$ ein Minterm erzeugt und diese anschließend disjunktiv miteinander verknüpft werden (vgl. Tabelle 3.4 rechts). Für das gezeigte Beispiel erhalten wir die in Abbildung 3.10 dargestellten Ergebnisse.

Das Konstruktionsschema besitzt zwei wesentliche Eigenschaften. Zum einen können wir es auf beliebige Wahrheitstabellen anwenden und somit zu jeder booleschen Funktion f eine äquivalente aussagenlogische Formel F in kanonischer konjunktiver oder kanonischer disjunktiver Normalform konstruieren. Da in F ausschließlich die drei Elementaroperatoren $\neg$, $\wedge$ und $\vee$ vorkommen, haben wir nebenbei bewiesen, dass die Menge $\{\neg, \wedge, \vee\}$ ein vollständiges Operatorensystem bildet. Zum anderen ist das Konstruktionsschema deterministisch, d. h., wir hatten zu keiner Zeit eine Möglichkeit, die Konstruktion der Min- oder Maxterme in irgendeiner Weise zu beeinflussen. Da jede boolesche Funktion eine eindeutige Wahrheitstafeldarstellung besitzt, ist auch die erzeugte Formeldarstellung eindeutig.

Im strengen Sinn sind die kanonische konjunktive und die kanonische disjunktive Normalform keine echten kanonischen, d. h. syntaktisch eindeutigen, Darstellungen. Verantwortlich hierfür ist eine fehlende Ordnung zwischen den Literalen eines Minterms bzw. Maxterms und den Mintermen bzw. Maxtermen selbst. Ohne diese Ordnung können wir aus einer kanonischen konjunktiven oder kanonischen disjunktiven Normalform eine weitere erzeugen, indem wir die Literale innerhalb eines Minterms umordnen. Ebenso können wir die Position der Minterme oder der Maxterme vertauschen, ohne die Normalformeigenschaft zu verletzen. Mathematisch ausgedrückt handelt es sich bei der KKNF und der KDNF um eine kanonische Darstellung modulo Kommutativität und Assoziativität.
Eine im mathematischen Sinne echte kanonische Darstellung ließe sich erzeugen, indem wir die Literale und Minterme beispielsweise alphabetisch anordnen. Für die meisten Anwendungen ist der Normalformbegriff in der hier eingeführten Form aber völlig ausreichend.

	A	B	C	D	F	
$A \vee B \vee C \vee D$	0	0	0	0	0	
	0	0	0	1	1	$\overline{A} \wedge \overline{B} \wedge \overline{C} \wedge D$
	0	0	1	0	1	$\overline{A} \wedge \overline{B} \wedge C \wedge \overline{D}$
	0	0	1	1	1	$\overline{A} \wedge \overline{B} \wedge C \wedge D$
$A \vee \overline{B} \vee C \vee D$	0	1	0	0	0	
	0	1	0	1	1	$\overline{A} \wedge B \wedge \overline{C} \wedge D$
$A \vee \overline{B} \vee \overline{C} \vee D$	0	1	1	0	0	
$A \vee \overline{B} \vee \overline{C} \vee \overline{D}$	0	1	1	1	0	
$\overline{A} \vee B \vee C \vee D$	1	0	0	0	0	
$\overline{A} \vee B \vee C \vee \overline{D}$	1	0	0	1	0	
	1	0	1	0	1	$A \wedge \overline{B} \wedge C \wedge \overline{D}$
$\overline{A} \vee B \vee \overline{C} \vee \overline{D}$	1	0	1	1	0	
	1	1	0	0	1	$A \wedge B \wedge \overline{C} \wedge \overline{D}$
	1	1	0	1	1	$A \wedge B \wedge \overline{C} \wedge D$
	1	1	1	0	1	$A \wedge B \wedge C \wedge \overline{D}$
$\overline{A} \vee \overline{B} \vee \overline{C} \vee \overline{D}$	1	1	1	1	0	

Tabelle 3.4: Konstruktionsschema der kanonischen Normalformen. Die KKNF wird erzeugt, indem zunächst für jede Belegung I mit $I \not\models F$ ein Maxterm erzeugt wird. Eine Variable wird unverändert in den Maxterm aufgenommen, wenn sie von I mit 0 belegt wird, andernfalls wird sie negiert aufgenommen. Anschließend werden alle Maxterme miteinander konjunktiv verknüpft. Die Konstruktion der KDNF verläuft analog, indem zunächst für jede Belegung I mit $I \models F$ ein Minterm erzeugt wird. Jetzt wird eine Variable unverändert in den Minterm aufgenommen, wenn sie von I mit 1 belegt wird, andernfalls wird sie negiert aufgenommen. Anschließend werden alle Minterme miteinander disjunktiv verknüpft.

- KKNF: $(A \vee B \vee C \vee D) \wedge (A \vee \neg B \vee C \vee D) \wedge (A \vee \neg B \vee \neg C \vee D) \wedge (A \vee \neg B \vee \neg C \vee \neg D) \wedge (\neg A \vee B \vee C \vee D) \wedge (\neg A \vee B \vee C \vee \neg D) \wedge (\neg A \vee B \vee \neg C \vee \neg D) \wedge (\neg A \vee \neg B \vee \neg C \vee \neg D)$

- KDNF: $(\neg A \wedge \neg B \wedge \neg C \wedge D) \vee (\neg A \wedge \neg B \wedge C \wedge \neg D) \vee (\neg A \wedge \neg B \wedge C \wedge D) \vee (\neg A \wedge B \wedge \neg C \wedge D) \vee (A \wedge \neg B \wedge C \wedge \neg D) \vee (A \wedge B \wedge \neg C \wedge \neg D) \vee (A \wedge B \wedge \neg C \wedge D) \vee (A \wedge B \wedge C \wedge \neg D)$

Abbildung 3.10: Kanonische Normalformen für die Funktion aus Tabelle 3.4

In den folgenden Betrachtungen ist die Eigenschaft der Eindeutigkeit nicht von Bedeutung. Aus diesem Grund werden wir auf eine kompaktere Darstellung zurückgreifen, die die zweistufige Grundstruktur der Normalform nicht zerstört. Die neue Darstellung basiert auf der Beobachtung, dass wir zwei Min- bzw. Maxterme immer dann zu einem gemeinsamen Term verschmelzen können, wenn sich diese im Vorzeichen eines einzigen Literals unterscheiden. Für unsere Beispielfunktion sind unter anderem die folgenden Vereinfachungen möglich:

$$\begin{aligned} & (A \wedge B \wedge \neg C \wedge \neg D) \vee (A \wedge B \wedge \neg C \wedge D) \\ \equiv\ & (A \wedge B \wedge \neg C) \wedge (\neg D \vee D) \\ \equiv\ & (A \wedge B \wedge \neg C) \end{aligned}$$

Wenden wir das Vereinfachungsschema durchgängig an, so können wir die 8 Teilausdrücke der KKNF und der KDNF auf jeweils 4 Teilausdrücke reduzieren. Beachten Sie, dass wir die Eigenschaft der Eindeutigkeit durch die Reduktion verlieren. Wie in Abbildung 3.11 gezeigt, existieren zwei verschiedene Möglichkeiten, die ursprüngliche Funktion kompakt darzustellen. Folgerichtig sprechen wir nicht mehr länger von einer *kanonischen Normalform*, sondern nur noch von einer *Normalform* (KNF oder DNF).

Eine konjunktive bzw. disjunktive *Minimalform* liegt vor, wenn die Funktion mit der kleinstmöglichen Anzahl von Literalen dargestellt wird, d. h., wenn keine andere konjunktive bzw. disjunktive Form existiert, die mit weniger Literalen auskommt.

Definition 3.9 (Konjunktive und disjunktive Minimalform)

Sei F eine aussagenlogische Formel.

- F ist in *konjunktiver Normalform* (KNF), wenn sie eine Konjunktion von Disjunktionen von Literalen ist.
- Eine konjunktive Form heißt *minimal*, wenn es keine äquivalente KNF gibt, die mit weniger Literalen auskommt.
- F ist in *disjunktiver Normalform* (DNF), wenn sie eine Disjunktion von Konjunktionen von Literalen ist.
- Eine disjunktive Form heißt *minimal*, wenn es keine äquivalente DNF gibt, die mit weniger Literalen auskommt.

Die Erzeugung einer konjunktiven oder einer disjunktiven Minimalform ist ein gut untersuchtes Teilgebiet der technischen Informatik. Unter dem Begriff der *zweistelligen Logikminimierung* wurde eine Vielzahl von Verfahren entwickelt, mit deren Hilfe sich die Minimalform effizient erzeugen bzw. annähern lässt [49]. Unter anderem kann mit diesen Verfahren gezeigt werden, dass es sich bei den Formeln aus Abbildung 3.11 tatsächlich um Minimalformen handelt.

Für die konjunktive Normalform einer aussagenlogischen Formel F existiert mit der *Klauseldarstellung* eine eigene Notation, von der wir in Abschnitt 3.1.3.2 im Zusammenhang mit dem Resolutionskalkül umfassend Gebrauch machen werden.

Definition 3.10 (Klauseldarstellung)

Eine *Klausel* ist eine Menge von Literalen. Die Klauselmenge

$$\{\{(\neg)A_1,\ldots,(\neg)A_i\},\ldots,\{(\neg)B_1,\ldots,(\neg)B_j\}\}$$

steht stellvertretend für die Formel

$$((\neg)A_1 \vee \ldots \vee (\neg)A_i) \wedge \ldots \wedge ((\neg)B_1 \vee \ldots \vee (\neg)B_j).$$

Die *leere Klausel* $\square$ repräsentiert den Wahrheitswert 0.

- Reduzierte KNF

$$\begin{aligned} F = (&\ B \vee\ C \vee\ D) \wedge \\ (&\ A \vee \neg B \vee\ D) \wedge \\ (\neg B \vee \neg C \vee \neg D) &\wedge \\ (\neg A \vee\ B \vee \neg D)& \end{aligned}$$

$$\begin{aligned} F = (&\ A \vee\ C \vee\ D) \wedge \\ (&\ A \vee \neg B \vee \neg C) \wedge \\ (\neg A \vee \neg C \vee \neg D) &\wedge \\ (\neg A \vee\ B \vee\ C)& \end{aligned}$$

- Reduzierte DNF

$$\begin{aligned} F = (\neg B \wedge\ C \wedge \neg D) &\vee \\ (\ A \wedge\ B \wedge \neg D) &\vee \\ (\ B \wedge \neg C \wedge\ D) &\vee \\ (\neg A \wedge \neg B \wedge\ D)& \end{aligned}$$

$$\begin{aligned} F = (\neg A \wedge \neg C \wedge\ D) &\vee \\ (\neg A \wedge \neg B \wedge\ C) &\vee \\ (\ A \wedge\ C \wedge \neg D) &\vee \\ (\ A \wedge\ B \wedge \neg C)& \end{aligned}$$

Abbildung 3.11: Die betrachtete Beispielfunktion ist so strukturiert, dass sich je zwei Min- bzw. Maxterme zu einem gemeinsamen Term verschmelzen lassen. Die reduzierte Darstellung ist nicht mehr eindeutig, so dass wir die entstehenden Formeln nur noch als Normalformen und nicht mehr als kanonische Normalformen bezeichnen.

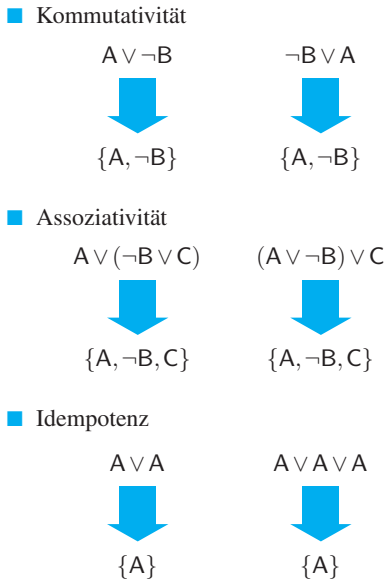

Abbildung 3.12: Zwischen Klauseln und Formeln besteht keine Eins-zu-eins-Beziehung. Jede Formel lässt sich eindeutig einer Klausel zuordnen, aber nicht umgekehrt. In der Klauseldarstellung sind die Eigenschaften der Kommutativität, Assoziativität und Idempotenz implizit vorhanden.

Auch wenn wir Klauseln fast immer wie Formeln behandeln werden, sollten Sie stets daran denken, dass zwischen beiden Darstellungsformen keine Eins-zu-eins-Beziehung besteht. Jede aussagenlogische Formel lässt sich eindeutig in eine Klausel verwandeln, aber nicht umgekehrt (vgl. Abbildung 3.12). Schuld daran ist die Mengendarstellung, in der zum einen die Information über die Reihenfolge der Elemente verloren geht und zum anderen jeder Term nur einmal aufgenommen werden kann. In einigen Anwendungsfällen ist dies gewollt. So ist die Klauseldarstellung der Formeldarstellung immer dann überlegen, wenn die Kommutativität ($F \vee G \equiv G \vee F$), die Assoziativität ($F \vee (G \vee H) \equiv (F \vee G) \vee H$) und die Idempotenz ($F \vee F \equiv F$) keine Rolle spielen.

3.1.3 Beweistheorie

In Abschnitt 3.1.1 haben wir die Semantik der Aussagenlogik über die Modellrelation $\models$ festgelegt und darauf aufbauend den Begriff der *allgemeingültigen Formel* definiert.

In diesem Abschnitt werden wir eine Reihe von Beweissystemen einführen, mit deren Hilfe sich die Allgemeingültigkeit einer Formel formal beweisen lässt. Auch wenn die vorgestellten Systeme äußerlich betrachtet sehr unterschiedlich wirken, folgen sie alle dem gleichen Ansatz: Die Allgemeingültigkeit wird mit einem Regelsystem bewiesen, das auf der symbolischen Manipulation von Zeichenketten beruht. Da ein Beweis vollständig auf der syntaktischen Ebene durchgeführt wird, benötigen wir keinerlei Wissen über Interpretationen, Modelle oder andere Begriffe, die sich mit den semantischen Eigenschaften von Formeln beschäftigen. Ein solches Regelsystem bezeichnen wir fortan als *Kalkül*.

Jeder Kalkül definiert eine *Ableitungsrelation* $\vdash$, die über die folgenden Beziehungen mit der Modellrelation $\models$ verbunden ist:

Definition 3.11 (Korrektheit, Vollständigkeit)

Sei K ein aussagenlogischer Kalkül. Ist eine Formel F innerhalb des Kalküls ableitbar, so schreiben wir $\vdash_K F$.

- K heißt *korrekt*, wenn aus $\vdash_K F$ stets $\models F$ folgt.
- K heißt *vollständig*, wenn aus $\models F$ stets $\vdash_K F$ folgt.