

Die Aussagenlogik (AL) ist die einfachste Form der Logik. Sie geht schon auf George Boole (1815–1864) zurück und beschreibt einfachste Verknüpfungen zwischen als atomar („unteilbar“) angesehenen Elementaraussagen. Dass dieses Modell aus logischer wie sprachlicher Sicht extrem vereinfachend ist liegt auf der Hand. Wie lassen sich solche Atome weiter strukturieren? Darauf versuchen sowohl die Logik, wie wir noch bei den anderen Formalismen sehen werden, als auch die Sprachwissenschaft Antworten zu geben.

Die praktische Bedeutung der AL in der Informatik kann gar nicht überschätzt werden. In jeder Programmiersprache kommen Boolesche Ausdrücke vor und auch beim Schaltkreisentwurf sind sie unentbehrlich. In der AL lassen sich künstliche, abstrahierte Situationen mit mathematischer Präzision analysieren, so etwa die folgende.

Beispiel

Meiers geben ein Abendessen. Eingeladen ist unter anderen die Familie Müller, bestehend aus Vater, Mutter und den drei Kindern Alfons, Berta und Claus. Wegen der komplizierten Familienverhältnisse bei Müllers gibt es verschiedene Einschränkungen:

- Wenn der Vater kommt, dann auch seine Frau.
- Mindestens einer der beiden Söhne kommt.
- Es kommt genau eine der beiden Frauen.
- Berta und Claus kommen gemeinsam oder gar nicht.
- Wenn Alfons kommt, dann auch Claus und der Vater.
- Welche Mitglieder der Familie Müller kommen zum Abendessen?

Tab. 2.1 Formale Symbole der Aussagenlogik

\wedge	Konjunktion	für „und“
\vee	Disjunktion	für „oder“
\neg	Negation	für „nicht“
\rightarrow	Implikation	für „wenn ... dann“
\leftrightarrow	Äquivalenz	für „genau dann ... wenn“

Übungsaufgabe

Versuchen Sie, dieses Problem zunächst freihändig „mit gesundem Menschenverstand“ zu lösen.

2.1 Syntax

Die Syntax der AL hat folgende Bestandteile:

- Eine Menge von Symbolen, $ASym = \{a, b, p_1, p_2, p_3, \dots\}$. Dies ist eine abzählbar unendliche Menge von Symbolen für Aussagen.
- Eine Menge von fünf formalen Symbolen, den *Junktoren*, wie schon in der Einleitung erwähnt und in Tab. 2.1 ersichtlich.
- Die Konstanten *true* und *false*.

► **Definition (Syntax der AL)** Die Menge *AForm* der *korrekt geformten aussagenlogischen Formeln* wird induktiv definiert:

1. Für jedes Aussagensymbol

$$p \in ASym \text{ gilt :} \quad (2.3)$$

$$p \in AForm. \quad (2.4)$$

2. Für die Konstanten *true* und *false* gilt:

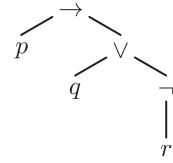
$$true, false \in AForm. \quad (2.5)$$

3. Sind $F, F_1, F_2 \in AForm$, dann sind auch

$$(F_1 \wedge F_2) \in AForm, \quad (2.6)$$

$$(F_1 \vee F_2) \in AForm, \quad (2.7)$$

Abb. 2.1 Ein reduzierter Ableitungsbaum



$$(F_1 \rightarrow F_2) \in AForm, \quad (2.8)$$

$$(F_1 \leftrightarrow F_2) \in AForm, \quad (2.9)$$

$$\neg(F) \in AForm. \quad (2.10)$$

4. $AForm$ enthält nur Zeichenketten, die durch endlich viele Anwendungen der Regeln 1, 2 und 3 entstanden sind.

Die Aussagesymbole stehen für elementare Aussagen der alltäglichen Sprache. Der Aufbau einer AL-Formel kann durch einen (reduzierten) Ableitungsbaum dargestellt werden. Dieser spiegelt den induktiven Aufbau der Formel wieder. Die Blätter sind die Aussagesymbole. An den inneren Knoten stehen die Junktoren. Die Klammern sind entbehrlich.

Beispiel

Der Satz „Wenn jemand grob foul spielt, dann wird er vom Platz gestellt oder der Schiedsrichter sieht das Foul nicht.“ soll formalisiert werden. Es stehen

p	für „Jemand spielt foul.“
q	für „Er wird vom Platz gestellt.“
r	für „Der Schiedsrichter sieht das Foul.“

Dann wird der gesamte Satz wiedergegeben durch

$$p \rightarrow (q \vee \neg r). \quad (2.11)$$

Der Ableitungsbaum zu der Formel des Beispiels ist in Abb. 2.1 zu sehen.

Zur Vereinfachung der Formeln vereinbaren wir auch hier die üblichen Regeln für Klammern:

- Äußere Klammern um eine Formel können entfallen.
- Das \neg bindet am stärksten.
- Die Junktoren \wedge und \vee binden stärker als \rightarrow und \leftrightarrow .
- Der Junktor \wedge bindet stärker als \vee .
- Der Junktor \rightarrow bindet stärker als \leftrightarrow .
- Die entsprechenden Klammern können entfallen. So reduziert sich

$$((\neg F) \wedge G \rightarrow H) \text{ zu } \neg F \wedge G \rightarrow H. \quad (2.12)$$

Übungsaufgabe

Wieso gilt $((\neg(F) \wedge G) \rightarrow H) \in AForm$?

Sie müssen hier eine Induktion über den Aufbau der Formel machen.

Übungsaufgabe

Machen Sie sich klar, dass einige der folgenden Ausdrücke keine korrekten Elemente von $AForm$ sind. Welche?

- $pq \rightarrow r$
- $\neg\neg p$
- $p \wedge \vee q$
- $p \wedge \neg q$

2.2 Semantik der AL

Die Bedeutungszuordnung erfolgt induktiv: Jedes Aussagensymbol $p \in ASym$ kann zwei Werte annehmen. Deshalb wird auch von *zweiwertiger Logik* gesprochen. In einer konkreten Situation wird also jedem Aussagesymbol ein solcher Wert zugeordnet. Eine solche Zuordnung heißt *Belegung der Aussagesymbole*. Diese Vorüberlegungen führen zu einer konkreten

► **Definition** Die Werte, die die Aussagesymbole annehmen können, werden üblicherweise mit 0 , 1 oder f , w (für falsch, wahr) bezeichnet. Sie heißen *Wahrheitswerte* oder *boolesche Werte*. Die Menge der booleschen Werte wird mit B abgekürzt.

- Eine *Belegung der Aussagesymbole* ist eine Abbildung

$$\sigma : ASym \rightarrow \mathbf{B}. \quad (2.13)$$

- Es wird also jedem Aussagesymbol ein Wahrheitswert zugeordnet.
- Sei künftig die Menge aller Belegungen mit Σ bezeichnet.

Wir werden im Folgenden demnach eine strikte Unterscheidung treffen:

- Auf syntaktischer Ebene werden die Konstanten *true* und *false* benutzt,
- sobald es um den semantischen Bereich geht, die booleschen Werte 0 oder 1.
- Zur Semantikdefinition wird jede Belegung jetzt auf eindeutige Weise von den Aussagesymbolen auf die Formeln hochgehoben werden. Damit entsteht eine Abbildung

$$I : AForm \times \Sigma \rightarrow \mathbf{B}. \quad (2.14)$$

Diese Abbildung heißt *Interpretation* oder (*formale*) *Semantik*. Ihre Definition erfolgt für Aussagesymbole durch Einsetzen und für die Junktoren durch Ausrechnen einer Funktionstabelle.

Definition (Semantik der AL) Die Semantik I ist eine Abbildung $I : AForm \times \Sigma \rightarrow B$, die folgendermaßen induktiv definiert ist.

Für alle $\sigma \in \Sigma$ gilt:

1. $I(p, \sigma) = \sigma(p)$ für alle $p \in ASym$,
2. $I(true, \sigma) = 1$ und $I(false, \sigma) = 0$,
3. $I(F_1 \wedge F_2, \sigma) = I(F_1, \sigma) \wedge I(F_2, \sigma)$,

analog für die anderen Junktoren.

In der letzten Formelzeile stehen die beiden \wedge -Zeichen für zwei verschiedene Dinge:

- Auf der linken Seite ist es ein Junktor, ein formales rein syntaktisches Symbol.
- Auf der rechten Seite ist es ein Funktionssymbol, das für eine Funktion $\mathbf{B} \times \mathbf{B} \rightarrow \mathbf{B}$ steht.

Dabei haben die Funktionen, die durch diese Funktionssymbole bezeichnet werden und die die Bedeutung der Junktoren festlegen, die jetzt angegebene Form.

- Für $\wedge : B \times B \rightarrow B$ gilt die folgende Funktionaltabelle:

\wedge	0	1
0	0	0
1	0	1

- Für $\vee : B \times B \rightarrow B$ gilt:

\vee	1	0
0	0	1
1	1	1

- Für $\rightarrow : B \times B \rightarrow B$ gilt:

\rightarrow	0	1
0	1	1
1	0	1

- Für $\leftrightarrow : B \times B \rightarrow B$ gilt:

\leftrightarrow	0	1
0	1	0
1	0	1

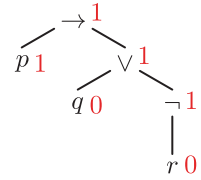
- Für $\neg : B \rightarrow B$ gilt:

\neg	
0	1
1	0

- **Bemerkung** Wird eine Formel $F \in AForm$ festgehalten, so beschreibt die entstehende Funktion

$$I(F, \cdot) : \Sigma \rightarrow B \quad (2.15)$$

die möglichen Auswertungen von F bei den verschiedenen Belegungen. Diese Funktion wird oft durch die wohlbekannten Wahrheitstafeln berechnet.

Abb. 2.2 Ein attributierter Ableitungsbaum

So berechnet man etwa $I(p_1 \wedge p_2, \cdot)$, indem man alle möglichen Belegungen $\sigma \in \Sigma$ durchspielt, hier also vier Stück, und schließlich den entsprechenden Wahrheitswert induktiv ausrechnet. Soll der Wert von $I(F, \cdot)$ nur für eine feste Belegung σ berechnet werden, so kann das leicht mit Hilfe des Ableitungsbaumes von F durchgeführt werden. Dazu werden an den Knoten des Baumes die Wahrheitswerte als Attribute angefügt, zunächst an den Blättern die durch σ gegebenen Werte der Aussagevariablen. Die Werte für die inneren Knoten werden mittels Induktion über den Aufbau von unten nach oben berechnet. In Abb. 2.2 ist der Vorgang des Attributierens eines Ableitungsbaumes für den Baum aus Abb. 2.1 und eine Belegung σ mit $\sigma(p) = 1, \sigma(q) = \sigma(r) = 0$ dargestellt.

Umgekehrt definiert eine festgehaltene Belegung σ eine Funktion

$$I(\cdot, \sigma) : AForm \rightarrow B. \quad (2.16)$$

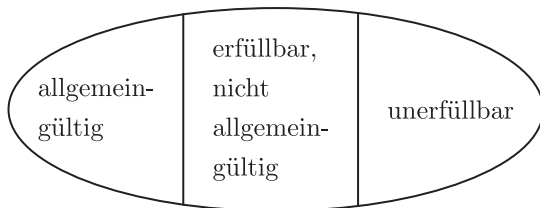
Auch diese Abbildungen, die angeben, ob eine Formel bei einer bestimmten Belegung wahr ist oder nicht, werden wir Interpretation nennen, die aus der Belegung σ *hochgehobene Interpretation*.

Beispiel

Wir berechnen die Werte für die Formel $((p_1 \wedge p_2) \rightarrow p_3) \leftrightarrow (p_1 \rightarrow (p_2 \rightarrow p_3))$ in einer Wahrheitstafel:

p_1	p_2	p_3	\neg	$((p_1 \wedge p_2) \rightarrow p_3)$	\leftrightarrow	$(p_1 \rightarrow (p_2 \rightarrow p_3))$
–	–	–	–	1	2	1
–	–	–	–	1	3	1
–	–	–	–	1	4	1
–	–	–	–	1	–	–
0	0	0	–	0	0	0
1	0	0	–	1	0	1
0	1	0	–	0	0	1
1	1	0	–	1	1	0
0	0	1	–	0	0	1
1	0	1	–	1	0	1
0	1	1	–	0	1	1
1	1	1	–	1	1	1

Abb. 2.3 Formeln je nach Erfüllbarkeit



Dabei sind in den drei links stehenden Spalten alle möglichen Kombinationen von Wahrheitswerten vorgegeben. Dann werden zuerst die mit 1 bezeichneten Spalten von dort kopiert, dann die mit 2 bezeichneten Spalten errechnet, dann die mit 3 behandelt und schließlich liefert die mit 4 bezeichnete Spalte das Ergebnis.

Übungsaufgabe

Lösen Sie das Problem aus der Einleitung zu diesem Kapitel („Müllers zu Gast“) mit Hilfe einer Wahrheitstafel.

Wir brauchen noch verschiedene Sprechweisen:

Definition (Gültigkeit) Seien $F \in AForm$ und σ eine Belegung.

$$1. \quad I, \sigma \models F \Leftrightarrow I(F, \sigma) = 1 \quad (2.17)$$

Man sagt, Formel F sei in der aus σ hochgehobenen Interpretation $I(., \sigma)$ gültig. Spielt die Belegung keine Rolle, oder ist sie aus dem Kontext klar, so schreiben wir einfach $I \models F$ statt $I, \sigma \models F$.

$$2. \quad \models F \Leftrightarrow \quad (2.18)$$

Für alle hochgehobenen Interpretationen I gilt $I \models F$.

Für alle $\sigma \in \Sigma$ gilt in diesem Fall also $I(F, \sigma) = 1$. Dann wird F als *allgemeingültig* oder *Tautologie* bezeichnet.

3. Gilt $I \models F$ für mindestens eine Interpretation, so wird F als *erfüllbar* bezeichnet. Sonst wird F als *nicht erfüllbar* bezeichnet.

4. Sei $F = \{F_1, F_2, \dots\}$ eine unter Umständen unendliche Menge von Formeln. Dann gelte $I \models F$, genau dann wenn

$$I \models F_i \quad (2.19)$$

für alle i gilt.

Ein Beispiel für eine Tautologie liefert die Formel $p \vee \neg p$, wie sich durch eine Wahrheitstafel leicht verifizieren lässt. Tautologien sind offensichtlich immer erfüllbar. Für Formeln gibt es also drei Klassen, die erfüllbaren, die allgemeingültigen und die, die weder eins noch das andere sind, wie in Abb. 2.3 zu sehen ist.

Erfüllbarkeit lässt sich auch gut charakterisieren durch die folgende

► **Definition** Die *Erfüllungsmenge* einer Formel F ist die Menge

$$\delta(F) = \{\sigma \mid I(F, \sigma) = 1\}, \quad (2.20)$$

also die Menge aller Belegungen, die die Formel wahr machen.

Ohne Beweis zitiere ich hier den einfachen

► **Satz** Für zwei aussagenlogische Formeln F und G gilt $F \leftrightarrow G$ genau dann, wenn sie die gleichen Erfüllungsmengen haben, wenn also gilt

$$\delta(F) = \delta(G), \quad (2.21)$$

Übungsaufgabe

Beweisen Sie den Satz.

2.3 Der logische Folgerungsbegriff

Ein zentraler Begriff der Logik ist der der logischen Folgerung, dem deshalb hier ein eigenes Unterkapitel gewidmet wird. Auf den ersten Blick scheint es so, als ob der folgende Ansatz angemessen wäre:

$$F_1, \dots, F_n \text{ haben } G \text{ zur Folge, wenn gilt } F_1 \wedge \dots \wedge F_n \rightarrow G$$

Dieser Ansatz ist jedoch unmöglich, falls von einer unendlichen Formelmengende eine andere Formel zur Folge hat, die Rede ist. Eine Lösung auch für unendliche Formelmengen bietet die

► **Definition (logische Konsequenz)** Sei F eine nicht notwendigerweise endliche Menge von Formeln. Dann heißt $G \in AForm$ eine (logische) Konsequenz von F , in Zeichen $F \models G$, wenn gilt:

Wenn immer $I \models F$ gilt, dann auch $I \models G$.

Man schreibt auch $\models G$ statt $\emptyset \models G$.

In den folgenden Aufgaben seien immer $F, F_i, G, H \in AForm$.

Übungsaufgaben

1. Beweisen Sie: Sei $F = \{F_1, \dots, F_n\}$, also endlich. Dann gilt $F \models G$ genau dann, wenn

$$F_1 \wedge \dots \wedge F_n \rightarrow G$$

eine Tautologie ist.

2. Beweisen Sie oder widerlegen Sie:

- $\{F \rightarrow G, F\} \models G$,
- $\{F \rightarrow G, G\} \models F$,
- $\{F \rightarrow G, \neg G\} \models \neg F$,
- $\{F \rightarrow G, H \vee \neg G, \neg(F \wedge H)\} \models \neg F$,
- $\{F \vee H, G \vee \neg H\} \models F \vee G$.

3. Beweisen Sie die *reductio ad absurdum*:

$F \models G \wedge \neg G$ gilt genau dann, wenn gilt $\models \neg F$.

4. Beweisen Sie oder widerlegen Sie:

- $\models F \rightarrow G$ ist äquivalent zu $\{F\} \models G$.
- $\models F \rightarrow G$ und $\models G \rightarrow F$ ist äquivalent zu $\models F \leftrightarrow G$.
- Aus $F \models G$ und $G \models H$ folgt $F \models H$ (Bei dieser Eigenschaft spricht man aus nahe-
liegenden Gründen von der „*Transitivität des Folgerungsbegriffes*“).
- Aus $F \models G$ oder $F \models H$ folgt $F \models G \vee H$.
- Aus $F \models G$ oder $F \models H$ folgt $F \models G \wedge H$.
- Aus $F \models G$ und $F \models H$ folgt $F \models G \vee H$.
- Aus $F \models G$ und $F \models H$ folgt $F \models G \wedge H$.
- Aus $F \models \neg F$ folgt, dass $\neg F$ eine Tautologie ist.

5. Beweisen Sie:

- Ist F nicht erfüllbar, so gilt $F \models G$ für alle Formeln G .
- Ist G eine Tautologie, so gilt $F \models G$ für alle Formeln F .

Der in diesem Unterkapitel eingeführte Folgerungsbegriff wird später sinngemäß auch auf andere Logiken erweitert werden.

2.4 Metasprachen und Kalküle

Eine der üblicherweise betrachteten Eigenschaften jeder Sprache ist ihre Fähigkeit, über Sprache zu rasonieren. Dann wird von der Sprache als *Metasprache* geredet. Wir müssen also unterscheiden zwischen der Argumentation *in* einer Logik (also in der Sprache der Logik) und der Argumentation *über* Logik (vermutlich in einer ganz anderen Sprache, etwa Deutsch).

► **Bemerkung** Um diesen Unterschied deutlich zu machen, werde ich beispielsweise unterscheiden zwischen den Zeichen

- \rightarrow , das nur syntaktisch zur Konstruktion und Formeln dient, also *in* der Logik benutzt wird, und
- \Rightarrow , bei dem allgemein über mathematische oder logische Aussagen, darunter natürlich auch Formeln, geredet wird. Hier wird dann auf der semantischen Ebene *über* Logik argumentiert. Entsprechendes gilt für \leftrightarrow und \Leftrightarrow .

Soll etwa der Nachweis geführt werden, dass zwei Formeln F und G äquivalent sind, also von $F \Leftrightarrow G$, so kann dies sehr leicht durch die Erstellung und den Vergleich von zwei Wahrheitstafeln geschehen. Nach den bisherigen Ausführungen ist klar, dass es sich dabei um eine semantische Methode handelt. Das Verfahren ist narrensicher, aber umständlich. Man sieht leicht ein, dass der Aufwand exponentiell mit der Größe der Formeln steigt. Ob es dennoch schnellere Verfahren gibt ist unbekannt.

Bei einem alternativen Ansatz könnte man so vorgehen: Man forme F so lange **syntaktisch** durch vorher festgelegte Schlüsse um, bis man G erhalten hat. Bei den Umformungen wird nur auf die Form der Formeln geachtet. Es wird also eine endliche Menge von solchen Umformungsregeln, quasi ein „Werkzeugkasten“, benutzt. Zum Nachweis, dass der Werkzeugkasten nur korrekte Regeln enthält, muss natürlich auf semantischer Grundlage argumentiert werden. **Bei der Anwendung der Regeln spielt die Semantik dann aber keine Rolle mehr.**

Ein solcher „Werkzeugkasten“ wird *Kalkül* genannt.

Definition

1. Ein *Kalkül* ist eine endliche Menge von Regeln.
2. Eine *Regel* ist ein Tripel $R = (\text{Präm}, \text{Conc}, \text{Bed})$ mit:
3. *Präm* (Prämissen) ist eine endliche Menge von Formeln. Die Prämissen kennzeichnen eine Voraussetzung für die Anwendung der Regel.
4. *Conc* (Conclusio) ist eine Formel, die eine Schlussfolgerung aus den Prämissen beschreibt.
5. *Bed* ($= \text{Bed}(\text{Präm}, \text{Conc})$) ist eine Bedingung für die Formeln in *Präm* und *Conc*, deren Wahrheit entschieden werden kann und die eine Nebenbedingung bezeichnet. Es muss entschieden werden können, ob *Bed* (*Präm*, *Conc*) zutrifft oder nicht.
6. Die Prämissenmenge einer Regel kann auch leer sein. In diesem Falle wird die Regel $(\emptyset, \text{Conc}, \text{Bed})$ als *Axiomenschema* bezeichnet. *Conc* beschreibt dann eine Menge von Formeln, auf die *Bed* zutrifft. Ein Element dieser Menge heißt *Axiom*.
7. Formal kann ein Kalkül K auch als Relation auf Formelmengen betrachtet werden:

$$K = \{(\text{Präm}, \text{Conc}) \mid \text{Bed}(\text{Präm}, \text{Conc})\}$$

Kalküle dieser Form werden nach dem deutschen Mathematiker David Hilbert (1862–1943) auch *Hilberttyp-Kalküle* genannt.

Der Kalkülbegriff ist unabhängig von der verwendeten Logik. Sie ist zunächst nur ein formales Spiel mit der Syntax. Soll der Kalkül in den Zusammenhang mit einer Logik gebracht werden, ihm also eine Semantik zugeordnet werden, spielt die jeweilige Logik über den noch zu definierenden Begriff der *Folgerung* eine entscheidende Rolle.

Üblicherweise werden Regeln in der Form

$$\frac{\text{Präm}}{\text{Conc}} \text{ mit Bed} \quad (2.22)$$

geschrieben. Kalküle werden in der Informatik immer wieder benutzt. Sie sind Ausdruck des in der Einleitung erwähnten Machbarkeitsgedankens. Dabei geht man davon aus, dass Kalküle leichter zu implementieren sind als semantische Grundlagen. Ich gebe hier einige

Beispiele

1. Der aus der Programmiersprache Prolog wohlbekannte *SLD-Kalkül*, der nur aus einer einzigen Regel (der *SLD-Regel*) besteht, wie auch in Kap. 4 zu sehen.
2. Für die Aussagenlogik gibt es eine Reihe von Kalkülen. Einige werden weiter unten betrachtet.
3. Es gibt auch Kalküle für andere Logiken, zum Beispiel für die Prädikatenlogik und für die in der Programmverifikation benutzte Hoaresche Logik, mit der wir uns in Kap. 5 befassen werden.
4. Es sollte darauf hingewiesen werden, dass nicht alle Kalküle auf syntaktischer Grundlage arbeiten. Ein Beispiel für einen ausdrucksstarken aber langsamen Kalkül auf semantischer Grundlage liefert die schon erwähnte induktive Berechnung in einer Wahrheitstafel.

Oben ist die Bezeichnung $\neq F$ dafür eingeführt worden, dass F allgemeingültig ist. Einen ähnlichen Begriff liefert die folgende

Definition

1. Eine Formel F heißt *ableitbar* (in einem Kalkül K) *aus einer Formelmenge M in einem Schritt*, wenn K eine Regel (M', F, B) enthält, wobei M' eine endliche Teilmenge von M ist und B eine erfüllte Nebenbedingung.
2. Die Formel F heißt (ganz allgemein) *ableitbar aus einer Formelmenge M* (im Kalkül K), in Zeichen $M \vdash_K F$, wenn es eine Folge von Formeln F_1, \dots, F_n gibt mit $F = F_n$, wobei sich jede Formel F_{i+1} in einem Schritt ableiten lässt aus $M \cup \{F_1, \dots, F_i\}$.
3. Die Folge F_1, \dots, F_n heißt *Beweis* von F , n die *Länge des Beweises*.
4. Eine Formel F heißt *ableitbar in einem Kalkül K* , in Zeichen $\vdash_K F$, wenn F in K aus der leeren Menge ableitbar ist.
5. Zunächst soll auf eine wichtige Begriffsunterscheidung hingewiesen werden:

► Bemerkung

- Im Allgemeinen kann für eine gegebene Formel F und einen gegebenen Kalkül K nicht entschieden werden, ob F in K ableitbar ist.
- Es kann im Allgemeinen aber für eine gegebene Folge von Formeln F_1, \dots, F_n sehr wohl entschieden werden, ob sie einen Beweis einer Formel F im Kalkül K darstellt.

Im Allgemeinen werden einige Anforderungen an einen Kalkül gestellt. Es soll noch einmal darauf hingewiesen werden, dass die folgende Definition auch in anderen Logiken als der AL anwendbar ist; natürlich muss dann der dort gültige Folgerungsbegriff benutzt werden.

► **Definition**

1. Eine Regel $R = (\text{Präm}, \text{Conc}, \text{Bed})$ heißt *korrekt*, wenn bei Vorliegen von *Bed* die Konklusion *Conc* eine logische Folgerung der Prämissenmenge *Präm* ist, wenn also gilt

$$\text{Präm} \models \text{Conc}. \quad (2.23)$$

2. Ein Kalkül K heißt *korrekt*, wenn für alle Formeln F gilt

$$\vdash_K F \Rightarrow \models F. \quad (2.24)$$

3. Ein Kalkül K heißt *vollständig*, wenn für alle Formeln F gilt

$$\models F \Rightarrow \vdash_K F. \quad (2.25)$$

Korrektheit bedeutet, dass alles, was im Kalkül abgeleitet werden kann, wahr ist. Vollständigkeit bedeutet, dass jede wahre Formel im Kalkül abgeleitet werden kann. Das allgemeine Ziel eines Kalküls ist es daher, vollständig und korrekt zu sein, also

$$\models F \Leftrightarrow \vdash_K F. \quad (2.26)$$

Während ein nicht korrekter Kalkül wertlos ist, gibt es in der Praxis viele Kalküle, die nicht vollständig sind. Beispielsweise gibt es nachweisbar gar keinen vollständigen Kalkül für die Hoaresche Logik. Für den abgeschwächten Begriff der „relativen Vollständigkeit“ sei auf das entsprechende Kapitel in diesem Buch hingewiesen. Dagegen ist der SLD-Kalkül vollständig. Auch diese Aussage wird später noch präzisiert. Für die Aussagenlogik gibt es viele vollständige und korrekte Kalküle. Auch der in Unterkapitel 3.6 eingeführte Prädikatenkalkül ist vollständig.

Als grundlegend für die Axiomatisierung der AL gilt die Arbeit Whitehead und Russel (1910) von Alfred N. Whitehead (1862–1943) und Bertrand A. W. Russell (1872–1970). In diesem Werk, den „Principia Mathematica“, einem der einflussreichsten Bücher der Geschichte der Mathematik und Logik, versuchen die Autoren, alle wahren mathematischen Aussagen und Beweise auf eine axiomatisierte symbolische Logik zurückzuführen.

Übungsaufgabe

Beweisen Sie, dass ein Kalkül genau dann korrekt ist, wenn alle seine Regeln korrekt sind.

2.5 Kalküle für die Aussagenlogik

Ein erster vollständiger und korrekter Kalkül für die Aussagenlogik soll jetzt beispielhaft vorgestellt werden. Es handelt sich nicht um einen der klassischen Kalküle, wie sie am Ende des Unterkapitels vorgestellt werden. Insbesondere ist er deutlich umfänglicher als jene. Dafür ist dieser Kalkül insbesondere für nicht mit der Materie Vertraute leichter anzuwenden; es ist klarer zu sehen, wie die einzelnen Beweisschritte ablaufen können. Die einzelnen Regeln werden als Äquivalenzen dargestellt. Dadurch werden sie leichter lesbar. Man kann sich leicht überlegen, dass eine logische Äquivalenz $F \Leftrightarrow G$ genauso gut durch die Regeln

$$\frac{F}{G}, \frac{G}{F} \quad (2.27)$$

entsprechend unserer formalen Definition dargestellt werden kann.

- **Satz** Für alle $F, G, H \in AForm$ gelten folgende einfach nachzuweisende Äquivalenzen:

- Elimination von \rightarrow und \leftrightarrow :

$$F \rightarrow G \Leftrightarrow \neg F \vee G \quad (2.28)$$

$$F \leftrightarrow G \Leftrightarrow (F \rightarrow G) \wedge (G \rightarrow F) \quad (2.29)$$

- Doppelnegation:

$$\neg\neg F \Leftrightarrow F \quad (2.30)$$

- Kontraposition:

$$F \rightarrow G \Leftrightarrow \neg G \rightarrow \neg F \quad (2.31)$$

- De Morgan-Gesetze:

$$\neg(F \wedge G) \Leftrightarrow \neg F \vee \neg G \quad (2.32)$$

$$\neg(F \vee G) \Leftrightarrow \neg F \wedge \neg G \quad (2.33)$$

- Idempotenz:

$$F \wedge F \Leftrightarrow F \quad (2.34)$$

$$F \vee F \Leftrightarrow F \quad (2.35)$$

- Kommutativität:

$$F \wedge G \Leftrightarrow G \wedge F \quad (2.36)$$

$$F \vee G \Leftrightarrow G \vee F \quad (2.37)$$

- Assoziativität:

$$(F \wedge G) \wedge H \Leftrightarrow F \wedge (G \wedge H) \quad (2.38)$$

$$(F \vee G) \vee H \Leftrightarrow F \vee (G \vee H) \quad (2.39)$$

- Distributivität:

$$F \wedge (G \vee H) \Leftrightarrow (F \wedge G) \vee (F \wedge H) \quad (2.40)$$

$$F \vee (G \wedge H) \Leftrightarrow (F \vee G) \wedge (F \vee H) \quad (2.41)$$

- Absorption:

$$F \wedge (F \vee G) \Leftrightarrow F \quad (2.42)$$

$$F \vee (F \wedge G) \Leftrightarrow F \quad (2.43)$$

- Gesetze über *true* und *false*:

$$F \wedge \neg F \Leftrightarrow \text{false} \quad (2.44)$$

$$F \vee \neg F \Leftrightarrow \text{true} \quad (2.45)$$

$$F \wedge \text{false} \Leftrightarrow \text{false} \quad (2.46)$$

$$F \wedge \text{true} \Leftrightarrow F \quad (2.47)$$

$$F \vee \text{false} \Leftrightarrow F \quad (2.48)$$

$$F \vee \text{true} \Leftrightarrow \text{true} \quad (2.49)$$

$$\neg \text{true} \Leftrightarrow \text{false} \quad (2.50)$$

$$\neg \text{false} \Leftrightarrow \text{true} \quad (2.51)$$

Zusätzlich werden in einem kalkülmäßigen Ansatz noch zwei **Ersetzungsprinzipien** genutzt:

- Leibniz'sches Ersetzungsprinzip

Wird in einer Formel H eine Teilformel F durch eine logisch äquivalente Teilformel F_1 ersetzt, so ist die dadurch entstehende Formel H_1 logisch äquivalent zu H .

- Instanziierung von Tautologien

Die Formel F enthalte die Aussagevariablen x_1, \dots, x_n , schreibe sich also in der Form $F(x_1, \dots, x_n)$. Seien ferner $G_1, \dots, G_n \in AForm$ Formeln. Wenn F eine Tautologie ist, dann ist auch $F(G_1, \dots, G_n)$ eine Tautologie.

Beispiele

- Um die Äquivalenz der Formeln $\neg F \rightarrow G$ und $F \vee G$ einzusehen, startet man mit
 - $\neg F \rightarrow G$, eliminiert die Implikation, erhält
 - $\neg \neg F \vee G$ und gelangt durch Beseitigung der doppelten Negation zu
 - $F \vee G$.
- Sei $F(x_i) = x_i \vee \neg x_i$. Bei der Instanziierung von Tautologien kann für x_i jede beliebige Aussage eingesetzt werden, so etwa, eine immer falsche Aussage. Trotzdem ist das Ergebnis immer wahr. Sei $G_1 = H \wedge \neg H$. Dann hat $F(G_1)$ die Form

$$F(G_1) = (H \wedge \neg H) \vee \neg(H \wedge \neg H). \quad (2.52)$$

Man überzeugt sich leicht, dass das eine Tautologie ist.

Zunächst ohne Beweis sei der folgende Satz zitiert. Für die Richtigkeit des Satzes wird im nächsten Unterkapitel mit Hilfe der Normalformen argumentiert, insbesondere für die Vollständigkeit des Kalküls.

- **Satz** Die Äquivalenzen dieses Unterkapitels bilden zusammen mit dem Axiom *true* (also der Forderung, dass *true* eine ableitbare Formel ist), dem Leibniz'schen Ersetzungsprinzip und der Instanziierung von Tautologien einen vollständigen und korrekten Kalkül für die Aussagenlogik.

Dieser Kalkül soll künftig als „Aussagenkalkül“ (AK) bezeichnet werden.

Übungsaufgaben

1. Beweisen Sie, dass der Aussagenkalkül korrekt und vollständig ist.
2. Lösen Sie das Problem aus der Einleitung zu diesem Kapitel („Müllers zu Gast“) mit Hilfe des AK.

3. Beweisen Sie den folgenden häufig benutzten Hilfssatz:

$$\text{Aus } \vdash_{AK} F_1 \text{ und } \vdash_{AK} F_2 \text{ folgt } \vdash_{AK} F_1 \wedge F_2.$$

Zunächst ist aber noch eine Bemerkung angebracht, die für einige andere Logiken wichtig wird:

Das Leibnizsche Ersetzungsprinzip ist eng mit dem schon aus Unterkapitel 1.3 bekannten Extensionalitätsprinzip verknüpft; es ist oft alles andere als trivial. Seine Richtigkeit hängt vom jeweiligen Kontext ab. Interessierte Leser können sich in der Literatur unter den Stichwörtern „*extensionaler Kontext*“ und „*intensionaler Kontext*“ kundig machen. Im Wesentlichen kann im Rahmen eines extensionalen Kontextes Gleiches durch Gleiches ersetzt werden, im Gegensatz zu intensionalen Kontexten.

Dieser Sachverhalt soll jetzt illustriert werden. Natürlich stammt das Beispiel nicht aus der Aussagenlogik, wo das Extensionalitätsprinzip ja gilt, sondern aus der in Kap. 8 behandelten epistemischen Logik. Es zeigt, dass etwas zu wissen keinen extensionalen Kontext darstellt.

Beispiel

Gegeben sei der Satz:

a. „Lenin hat die Oktoberrevolution gemacht.“

Bekanntlich hat Lenin den bürgerlichen Namen „Wladimir Iljitsch Uljanow“ geführt. Man sagt dann, die Begriffe „Lenin“ und „Wladimir Iljitsch Uljanow“ seien extensional gleich. Durch extensional gleiche Ersetzung entsteht der Satz:

b. „Wladimir Iljitsch Uljanow hat die Oktoberrevolution gemacht.“

Diese beiden Sätze sind dann äquivalent. Wird jetzt in dem Satz

c. „Moritz weiß: Lenin hat die Oktoberrevolution gemacht.“

der Teilsatz (a.) durch den Teilsatz (b.) ersetzt, entsteht der Satz

d. „Moritz weiß: Wladimir Iljitsch Uljanow hat die Oktoberrevolution gemacht.“

Obwohl (a.) und (b.) äquivalent sind, müssen es (c.) und (d.) nicht sein, nämlich wenn dem Moritz die extensionale Gleichheit von Lenin und Wladimir Iljitsch Uljanow unbekannt ist.

Der hier angegebene Kalkül für die AL ist sehr eingängig und die Beweise sind einfach zu führen. Jedoch hat der AK auch einen bedeutenden Nachteil: Er ist zu umfangreich.

Ganz allgemein gilt für beliebige Kalküle: Ist der Kalkül zu umfangreich, hat das den Nachteil, dass es zu einer gegebenen konkreten Struktur umständlich wird, nachzuweisen, dass diese Struktur die Regeln des Kalküls erfüllt. Ganz allgemein wird die Arbeit mit dem Kalkül zu aufwendig. Für die Aussagenlogik gibt es einige deutlich kürzere klassische Kalküle.

Definition (Hilbertscher Kalkül) Der *Hilbertsche Kalkül (HK)* besteht aus den folgenden Axiomenschemata und Regeln:

- Für alle Aussagevariablen p, q, r gilt

$$A_1 : (p \vee p) \rightarrow p \quad (2.53)$$

$$A_2 : q \rightarrow (p \vee q) \quad (2.54)$$

$$A_3 : (p \vee q) \rightarrow (q \vee p) \quad (2.55)$$

$$A_4 : (p \rightarrow r) \rightarrow ((p \vee q) \rightarrow (r \vee q)) \quad (2.56)$$

- *Regel 1 (Formelsubstitution):*

Wird in einem Theorem F des HK eine Aussagevariable p an allen Stellen durch eine wohlgeformte Formel G ersetzt, so ist das Ergebnis ebenfalls ein Theorem des HK.

$$\text{Aus} \vdash_{HK} F(p) \text{ folgt } \vdash_{HK} F(G).$$

- *Regel 2 (Modus Ponens):*

Sind F und $F \rightarrow G$ Theoreme des HK, so ist auch G ein Theorem des HK.

$$\text{Aus} \vdash_{HK} F \text{ und } \vdash_{HK} F \rightarrow G \text{ folgt } \vdash_{HK} G$$

Ferner kann $\neg F \vee G$ für $F \rightarrow G$ geschrieben werden.

Definition (Kalkül von Łukasiewicz): Der Kalkül von Łukasiewicz (LK) besteht aus den folgenden Axiomenschemata und Regeln:

- Für alle Aussagevariablen p, q, r gilt

$$A_1 : p \rightarrow (q \rightarrow p) \quad (2.57)$$

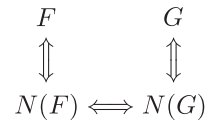
$$A_2 : (p \rightarrow (q \rightarrow r)) \rightarrow ((p \rightarrow q) \rightarrow (p \rightarrow r)) \quad (2.58)$$

$$A_3 : (p \vee q) \rightarrow (q \vee p) \quad (2.59)$$

$$A_4 : (p \rightarrow q) \rightarrow (\neg q \rightarrow \neg p) \quad (2.60)$$

Dazu kommen wieder die beiden Umformungsregeln aus dem HK.

Abb. 2.4 Äquivalenzbeweis
mit Normalformen



Eine dem Modus Ponens verwandte Regel, die zwar in den genannten Kalkülen nicht vorkommt, die aber auch sehr praxisrelevant ist, ist der „*Modus Tollens*“.

Definition (Modus Tollens)

Der *Modus Tollens* ist eine Schlussregel, die besagt:

Sind $\neg G$ und $F \rightarrow G$ Theoreme eines Kalküls, so ist auch $\neg F$ ein Theorem des Kalküls.

Der Modus Tollens ist in der Mathematik letztlich der Kern von Widerspruchsbeweisen und wird manchmal als „*Widerlegungsregel*“ bezeichnet. Dann hat er oft die Form

Sind F und $\neg G \rightarrow \neg F$ Theoreme eines Kalküls, so ist auch G ein Theorem des Kalküls.

Genau zur Widerlegung von Aussagen dient die Regel etwa bei der Anwendung des SLD-Kalküls, siehe Kapitel 4.

2.6 Normalformen in der AL

Informell ist eine *Normalform* $N(F)$ einer Formel F eine semantisch zu F äquivalente Formel, die aber in irgendeiner Form syntaktisch herausgehoben ist. Der Sinn von Normalformen ist es, zu gegebenen Formeln einfacher überblickbare aber äquivalente Darstellungen zu finden. Eine typische Anwendung von Normalformen ist der Nachweis der Äquivalenz von Formeln. Das auch in Abb. 2.4 dargestellte Vorgehen ist dabei das folgende:

1. Gegeben seien die Formeln F und G .
2. Überführe F in eine Normalform $N(F)$.
3. Überführe G in eine Normalform $N(G)$.
4. Überprüfe die Äquivalenz von $N(F)$ und $N(G)$.

Idealerweise sind die Normalformen so beschaffen, dass sich der letzte Schritt leicht durchführen lässt, jedenfalls leichter als der direkte Nachweis der Äquivalenz von F und G . Natürlich soll auch die Überführung in eine Normalform einfach zu bewerkstelligen sein.

Mit Hilfe dieser Überlegungen lässt sich dann die Vollständigkeit des Kalküls aus dem letzten Unterkapitel leicht einsehen. Es ist nachzuweisen, dass sich die Verfahrensschritte 1. – 4. mit Hilfe der angegebenen Regeln durchführen lassen. Das ist einfach.

Einige Normalformen AL sollen jetzt vorgestellt werden.

Definition

1. Ein *Literal* ist eine Aussagevariable (auch als *positives Literal* bezeichnet) oder die Negation einer Aussagevariablen (dann als *negatives Literal* bezeichnet).

2. Ein *Konjunktionsterm* ist eine Konjunktion von Literalen, wobei keine Variable doppelt vorkommt.
3. Ein *Minterm* zu einer Formel F ist ein Konjunktionsterm, bei dem jede Variable von F genau einmal vorkommt.
4. Eine *Disjunktive Normalform (DNF)* ist eine Disjunktion von Konjunktionstermen.
5. Eine *Kanonische Disjunktive Normalform (KDNF)* zu einer Formel F ist eine Disjunktion von Mintermen zu F , wobei keine zwei Minterme zueinander äquivalent sind.

► **Bemerkung** Die DNF ist im Allgemeinen nicht eindeutig. Das zeigt die Äquivalenz

$$\neg p \vee (q \wedge p) \Leftrightarrow \neg p \vee q \quad (2.61)$$

Beide Seiten sind in DNF.

Die KDNF ist eindeutig im folgenden Sinne:

Für jede Formel F existiert bis auf Kommutativität von \wedge und \vee genau eine Formel F' mit

1. $F \Leftrightarrow F'$,
2. F' ist in KDNF.

Diese Formel F' wird dann als *die* KDNF von F bezeichnet.

Beispiel

Die KDNF von $p \rightarrow q$ bestimmt sich durch die Äquivalenz

$$p \rightarrow q \Leftrightarrow (\neg p \wedge q) \vee (\neg p \wedge \neg q) \vee (p \wedge q). \quad (2.62)$$

Diese KDNF ist eindeutig bis auf Vertauschung der einzelnen Literalen innerhalb eines Konjunktionstermes und die Vertauschung der einzelnen Konjunktionsterme.

Übungsaufgaben

1. Bestimmen Sie die KDNF der Formeln
 - $((p \wedge (q \vee r)) \vee (q \wedge (p \vee r)))$,
 - $((\neg p \vee (p \wedge \neg q)) \wedge r \vee (\neg p \wedge q))$
 - $(p \rightarrow (q \leftrightarrow r))$.
2. Beweisen Sie, dass die KDNF jeder Formel bis auf Kommutativität von \wedge und \vee eindeutig ist.

Mit Hilfe der KDNF lässt sich jetzt ein Teil des Programmes zur syntaktischen Bestimmung der Äquivalenz von zwei Formeln vom Beginn dieses Unterkapitels durchführen;

mehr noch: Die Überführung einer Formel in eine äquivalente Formel in KDNF lässt sich auch effektiv auf einem Rechner, und damit beispielsweise in einem logikbasierten System, durchführen. Da es sich bei den Umformungen des Aussagenkalküls in Kap. 2.5 um Äquivalenzumformungen handelt, lässt sich auch der umgekehrte Weg von der KDNF zu einer vorgegebenen Formel gehen.

- **Satz** Jede Formel lässt sich durch den Aussagenkalkül in die zugehörige KDNF umformen.

Statt eines formalen Beweises sollen die nötigen Schritte an einem Beispiel illustriert werden:

Beispiel

Wir beginnen mit einer Formel

$$(\neg p \rightarrow q) \leftrightarrow (q \rightarrow p \wedge q) \quad (2.63)$$

Zunächst wird die Äquivalenz eliminiert, wobei man auch das Ersetzungsprinzip von Leibniz benötigt, (also: $x \leftrightarrow y \Leftrightarrow (x \wedge y) \vee (\neg x \wedge \neg y)$ + *Leibniz*) mit dem Ergebnis

$$((\neg p \rightarrow q) \wedge (q \rightarrow p \wedge q)) \vee (\neg(\neg p \rightarrow q) \wedge \neg(q \rightarrow p \wedge q)) \quad (2.64)$$

Danach wird die Implikation beseitigt, wobei wieder das Ersetzungsprinzip von Leibniz benutzt wird (also: $x \rightarrow y \Leftrightarrow \neg x \vee y$ + *Leibniz*). Man erhält

$$((\neg\neg p \vee q) \wedge (\neg q \vee (p \wedge q))) \vee (\neg(\neg\neg p \vee q) \wedge \neg(\neg q \vee p \wedge q)) \quad (2.65)$$

Danach werden die Negationen nach innen durchgeschoben, bis sie direkt vor den Aussagevariablen stehen. Doppelte Negationen werden beseitigt. Neben der dafür zuständigen Regel werden auch die Gesetze von De Morgan herangezogen. (also: $\neg\neg x \Leftrightarrow x$ + *De Morgan*)

Das Ergebnis im vorliegenden Fall lautet dann:

$$((p \vee q) \wedge (\neg q \vee (p \wedge q))) \vee (\neg p \wedge \neg q \wedge q \wedge (\neg p \vee \neg q)) \quad (2.66)$$

Dieser Ausdruck wird sukzessiv in eine Disjunktion von Konjunktionstermen überführt. Die wichtigsten Hilfsmittel sind dabei die Distributivgesetze. Wir erhalten zunächst

$$((p \vee q) \wedge \neg q) \vee ((p \vee q) \wedge (p \wedge q)) \vee ((\neg p \wedge \neg q) \wedge ((q \wedge \neg p) \vee (q \wedge \neg q))), \quad (2.67)$$

dann

$$(p \wedge \neg q) \vee (q \wedge \neg q) \vee ((p \wedge p \wedge q) \vee (q \wedge p \wedge q)) \vee ((\neg p \wedge \neg q) \wedge ((q \wedge \neg p) \vee false)). \quad (2.68)$$

ferner (unter Weglassung einiger Schritte) mit Hilfe der Absorptionsgesetze und der Gesetze über *true* und *false*

$$(p \wedge \neg q) \vee false \vee (p \wedge q) \vee (p \wedge q) \vee false \vee false, \quad (2.69)$$

und schließlich

$$(p \wedge \neg q) \vee (p \wedge q). \quad (2.70)$$

Das ist die gesuchte KDNF zu der ursprünglichen Formel. Eine weitere Vereinfachung, etwa zu dem Konjunktionsterm p , ist nicht statthaft, da p kein Minterm ist und damit keine KDNF vorläge.

Übungsaufgabe

Beweisen Sie den Satz formal.

Zusammenfassend erhalten wir das

Korollar Der Aussagenkalkül ist vollständig.

Beweisidee Seien F, G gegeben mit $F \Leftrightarrow G$.

Seien $KDNF(F)$ und $KDNF(G)$ zugehörige KDNF. Wegen

$$KDNF(F) \Leftrightarrow F \Leftrightarrow G \Leftrightarrow KDNF(G) \quad (2.71)$$

sind $KDNF(F)$ und $KDNF(G)$ auch KDNF zu G und F . Sie lassen sich also im Kalkül ineinander überführen, da es zu jeder Formel bis auf Kommutativität von \wedge und \vee nur genau eine äquivalente KDNF gibt. Mit dem Kalkül lassen sich ferner F und $KDNF(F)$ ineinander überführen, ebenso $KDNF(G)$ und G .

Das beweist die Behauptung.

Statt mit KNF hätten wir auch einen anderen Ansatz benutzen können. Dieser wird oft „dualer“ Ansatz genannt, ist dem eben vorgestellten sehr ähnlich und wird deshalb hier nur gestreift.

► Definition

1. Ein *Disjunktionsterm* ist eine Disjunktion von Literalen, wobei keine Variable doppelt vorkommt.
2. Ein *Maxterm* zu einer Formel F ist ein Disjunktionsterm, bei dem jede Variable von F genau einmal vorkommt.
3. Eine *Konjunktive Normalform (KNF)* ist eine Konjunktion von Disjunktionstermen.
4. Eine *Kanonische Konjunktive Normalform (KKNF)* zu einer Formel F ist eine Konjunktion von Maxtermen zu F , wobei keine zwei Maxterme zueinander äquivalent sind.

Auch die *KNF* ist im Allgemeinen nicht eindeutig, im Gegensatz zur *KKNF*, die zu jeder Formel existiert, zu dieser äquivalent und bis auf Kommutativität von \wedge und \vee eindeutig ist. Mit Hilfe des Aussagenkalküls lässt sich auch jede Formel in die *KKNF* überführen. Auch auf diese Weise könnte die Vollständigkeit des Aussagenkalküls bewiesen werden.

Übungsaufgabe

Bestimmen Sie die KKNF der Formeln

- $(p \wedge (q \vee r)) \vee (q \wedge (p \vee r))$,
- $((\neg p \vee (p \wedge \neg q)) \wedge (r \vee (\neg p \wedge q)))$,
- $(p \rightarrow (q \leftrightarrow r))$.

Konjunktive Normalformen werden wir noch in Kap. 4 im Zusammenhang mit der Logik-Programmierung etwas näher kennen lernen.

Literatur

Whitehead, A. N., Russell, B. A. W.: Principia Mathematica. Cambridge University Press, Cambridge (1910)

Logikkalküle in der Informatik

Wie wird Logik vom Rechner genutzt?

Schenke, M.

2013, XII, 232 S. 74 Abb., Softcover

ISBN: 978-3-8348-1887-4