2. Deduktion in der Aussagenlogik

Problemreduktion

Im vorangegangenen Kapitel haben wir von der Bedeutung der Prädikatenlogik für unser Thema Inferenz gesprochen. Leider erweist sich unser Hauptproblem hierbei, nämlich logische Beziehungen der Prädikatenlogik auf ihre Gültigkeit hin zu prüfen, als äußerst komplex. Es ist daher für das Verständnis des Lesers unabdingbar, den Gesamtfragenkomplex in einzelne Teile zu zergliedern.

Anders als in vielen anderen Fragestellungen ist hier jedoch schon diese Aufgabe der Untergliederung höchst problematisch, weil sich das genannte Hauptproblem nicht wirklich in Teilprobleme zerlegen läßt, deren Lösung zusammengefügt dann eine Lösung des Gesamtproblems ergeben würde. Vielmehr ist das Gesamtproblem letztlich nur in einem Stück lösbar.

Aussagenlogik

Um aus diesem Dilemma herauszukommen, bleibt nur der Ausweg der Reduktion. Wir abstrahieren von Teilaspekten und befassen uns zunächst nur mit einer reduzierten Form des Gesamtproblems. Die Aussagenlogik *ist eine solch reduzierte Form der Prädikatenlogik. Wir werden daher in diesem Kapitel die Behandlung des Hauptproblems erst einmal in dieser "reduzierten Prädikatenlogik" durchführen und uns dann im nächsten Kapitel überlegen, welche zusätzlichen Maßnahmen erforderlich sind, um eine Lösung des Hauptproblems in seiner nicht-reduzierten Form anzugreifen. Zunächst beginnen wir damit, die genannte Reduktion genauer zu erläutern.*

2.1 Problemreduktion

Im Abschnitt 1.3 haben wir das Beispiel eines prädikatenlogischen Schlusses kennengelernt, nämlich

$$\frac{\begin{array}{c}\forall x\ [A(x)\ \rightarrow\ S(x)]\\ \forall x\ [S(x)\ \rightarrow\ H(x)]\end{array}}{\forall x\ [A(x)\ \rightarrow\ H(x)]}$$

Dort wurde erläutert, wie diese Form durch Abstraktion (und damit Reduktion) aus der natürlichen Form menschlichen Schließens entsteht. Eine solche Problemreduktion ist immer dann erforderlich, wenn die Problemstellung sich andernfalls als zu kompliziert erweist, was in unserem Fall zweifelsohne in extremer Weise gegeben ist. Es ist allerdings wichtig, nach der Problemreduktion das ursprüngliche Problem nicht ganz zu vergessen. Vielmehr sollte man sich nach Lösung des reduzierten Problems dann wieder an die des ursprünglichen Problems heranmachen. In dem vorliegenden Fall ist selbst das reduzierte Problem von einer befriedigenden Lösung leider noch allzuweit entfernt, so daß wir uns mit der Lösung des Problems des menschlichen Schließens in seiner natürlichen Form wohl noch einige Zeit gedulden müssen. In diesem Abschnitt führen wir nun noch weitere Reduktionsschritte durch, um uns an eine Lösung in dieser Weise heranzuarbeiten.

Sequentielle Darstellung

Zunächst einmal ist die oben illustrierte tabellarische Darstellung eines Schlusses unnötig platzaufwending, so daß wir sie lieber durch eine sequentielle Darstellung ersetzen, die bei dem gleichen Beispiel zu folgender Formel führt.

$$\forall x\ [A(x)\ \rightarrow\ S(x)]\ \&\ \forall x\ [S(x)\ \rightarrow\ H(x)]\ \vdash\ \forall x\ [A(x)\ \rightarrow\ H(x)]$$

Wie aus dem Vergleich beider Darstellungen zu ersehen, werden die *Prämissen* des Schlusses untereinander durch das Zeichen "&" verknüpft (anstatt sie der Reihe nach untereinander aufzulisten); der Prämissenteil wird mit der *Konklusion* des Schlusses durch das *Ableitbarkeitssymbol* $\vdash$ verknüpft (anstelle des waagerechten Schlußstriches vorher).

Objekt- und Metasprache

Diese beiden Zeichen sind nicht zufällig so gewählt, handelt es sich doch logisch im ersteren Fall um eine Und–Verknüpfung und im zweiten Fall um eine Wenn-Dann–Beziehung. Beide Beziehungen gibt es schon *innerhalb* der Prämissen– und Konklusionsteile eines Schlusses, also in der *Objektsprache*, wie man hierzu auch sagt. Im Gegensatz dazu werden die neuen Zeichen nun in der *Metasprache* eingesetzt und zwar in der Weise, daß mit ihnen "Formeln der Metasprache", nämlich Schlüsse, gebildet werden, wobei die Formeln der Objektsprache (also die Prämissen und die Konklusion) als Grundzeichen der Metasprache aufgefaßt werden.

Wenn es sich nun aber auf beiden Sprachebenen, der Objekt– und der Metaebene, um die gleichen logischen Operationen, nämlich dem "und" bzw. dem "wenn–dann", handelt, warum dann überhaupt die Unterscheidung in verschiedene Sprachebenen? In der Tat vereinfacht

sich unsere Problemstellung durch eine solche Verschmelzung der beiden Sprachebenen zu einer neuen Objektsprache, die in diesem Fall jedoch identisch mit der alten ist. Das heißt, wir verwenden tatsächlich anstelle des "&" die Und–Verknüpfung der Prädikatenlogik, die üblicherweise mit $\wedge$ bezeichnet wird (in unserem Beispiel aber zufällig nicht auftritt), und anstelle des "$\vdash$" das übliche "$\rightarrow$". Unsere Formel lautet damit

$$\forall x\ [A(x) \rightarrow S(x)] \wedge \forall x\ [S(x) \rightarrow H(x)] \rightarrow \forall x\ [A(x) \rightarrow H(x)]$$

Hierbei hat sich die Metasprache nur verschoben und ist nicht völlig verschwunden, denn genaugenommen müßte das Ableitbarkeitssymbol nun vor die gesamte Formel geschrieben werden. Mit dieser Umformung haben wir das *Deduktionstheorem* der Prädikatenlogik illustriert (siehe Theorem 2.3.1).

Damit ist die erste Problemreduktion dieses Abschnitts abgeschlossen. Wie eingangs erwähnt, sollte eine solche Reduktion nicht vollends in Vergessenheit geraten, da mit ihr praktisch immer etwas verloren geht. Im vorliegenden Fall ist, besonders in komplexeren Fällen, nun nicht mehr klar erkennbar, was als Prämisse bzw. als Konklusion fungiert. Zum Beispiel läßt sich in der letzteren Form nicht mehr unterscheiden, ob es sich ursprünglich um einen Schluß mit 0, 1 oder 2 Prämissen handelt. Dieser Informationsverlust kann sich später nachteilig bei der Problemlösung bemerkbar machen, worauf wir hier ausdrücklich aufmerksam machen.

Aussagenlogische Formel

Im Hinblick auf die zweite und entscheidende Reduktion dieses Abschnitts (hin zur Aussagenlogik) überlegen wir uns, wie man sich natürlicherweise von der Gültigkeit dieser Formel überzeugen würde. Das heißt, wir wollen beweisen, daß diese Konklusion aus diesen beiden Prämissen folgt. Die Konklusion stellt eine All–Aussage dar. Wie in der Schulmathematik gelernt, wählen wir uns daher ein *festes*, aber *beliebiges* Element aus dem betreffenden Bereich, das wir im Beispiel mit c bezeichnen wollen. Mit anderen Worten, anstelle der Konklusion $\forall x\ [A(x) \rightarrow H(x)]$ betrachten wir die vereinfachte Konklusion $A(c) \rightarrow H(c)$. Können wir die letztere beweisen, dann gilt sie für alle Elemente, da ja c völlig beliebig gewählt war, also gilt die ursprüngliche Konklusion. Beide Prämissen sind wiederum All–Aussagen. Wenn sie gelten (was wir bei Prämissen annehmen können), dann gelten sie auch für jedes einzelne Element, zB. für unser gewähltes c. Angenommen nun, die Voraussetzung der so vereinfachten Konklusion, $A(c)$, gilt. Dann schließen wir mit der vereinfachten ersten Prämisse hieraus, daß auch $S(c)$ gelten muß, also, wegen der vereinfachten zweiten Prämisse, auch $H(c)$, was insgesamt in der vereinfachten Konklusion behauptet wurde, womit insgesamt der gesuchte Beweis geführt ist. Bei genauer Analyse dieses Beweises sieht man leicht,

daß er über den Weg eines Beweises der Formel

$$[A(c) \rightarrow S(c)] \wedge [S(c) \rightarrow H(c)] \rightarrow [A(c) \rightarrow H(c)]$$

erbracht wurde. Beim Beweis dieser Formel hat die innere Struktur der Formelteile zwischen den logischen Operatoren (wie zB. $A(c)$), keine Rolle mehr gespielt. Wir können unsere Problemstellung also dadurch reduzieren, daß wir diese innere Struktur ganz ignorieren. Dies wird mittels Ersetzung dieser Formelteile durch einfache Zeichen erreicht. Ersetzen wir zB. $A(c)$ durch P, $S(c)$ durch Q und $H(c)$ durch R, so entsteht eine Formel der Aussagenlogik.

$$[P \rightarrow Q] \wedge [Q \rightarrow R] \rightarrow [P \rightarrow R]$$

Wie wir sehen, besteht eine solche Formel der Aussagenlogik aus Zeichen wie P usw., den sogenannten Aussagenvariablen, die durch die bekannten logischen Operationen miteinander verknüpft sind. Die Aussagenvariablen stehen für einfache Aussagen. So steht in unserem Beispiel P ursprünglich für die Aussage "Das Objekt c ist ein Affe". Formeln stehen dann für logisch verknüpfte Aussagen.

Das Ergebnis unserer Reduktion können wir nun so zusammenfassen. Beim Beweis der Gültigkeit von prädikatenlogischen Formeln spielen Beweise von entsprechenden aussagenlogischen Formeln eine mitentscheidende Rolle. Ein Studium der letzteren wird daher auch für den allgemeinen Fall von Bedeutung sein. Andererseits sind aussagenlogische Beweise wesentlich einfacher — wenn auch noch immer kompliziert genug, daß man weitere Reduktionen in Betracht ziehen kann (was im Verlauf dieses Kapitels geschehen wird). Wir werden uns für den Rest dieses Kapitels daher nun ausschließlich mit der Aussagenlogik beschäftigen und auf die Prädikatenlogik erst wieder im nächsten Kapitel zurückkommen.

2.2 Die Sprache der Aussagenlogik

Matrizen

Die syntaktische Struktur von Formeln der Aussagenlogik (wie der letzten Formel des vorangegangenen Abschnitts) dürfen wir als bekannt voraussetzen (siehe auch Abschnitt 3.1). Für unsere Zwecke wird sich jedoch die Darstellung aussagenlogischer Formeln in Form von Matrizen als zweckmäßig erweisen. Wir beginnen daher mit der Definition solcher Matrizen und zeigen anschließend, wie sie als aussagenlogische Formeln zu interpretieren sind.

Definition 2.2.1
$\mathcal{P}^0$ sei ein Alphabet, also eine nichtleere, endliche oder abzählbar unendliche Menge. Seine Elemente nennen wir Aussagenvariablen *(oder* Atome*)*

und bezeichnen sie (in der Regel) mit den Buchstaben P, Q oder R. Solche Bezeichnungsvereinbarungen schließen immer auch die Möglichkeit mit ein, die vereinbarten Zeichen mit Indizes, Strichen oder anderen Verzierungen versehen zu können. Aussagenvariablen lassen sich als null*stellige Prädikatszeichen auffassen, womit sich der Index in $\mathcal{P}^0$ erklärt.*

Literale *sind Atome P oder deren Negation $\neg P$ (entsprechend werden sie auch als* positiv *und* negativ *unterschieden). Zum Unterschied von Literalen mit Variablen in der Prädikatenlogik sprechen wir hier manchmal auch von* Grund*literalen. Sie werden mit K, L oder M bezeichnet.*

Klauseln *sind (endliche, möglicherweise leere) Mengen von Literalen.* Matrizen *sind (endliche, möglicherweise leere) Mengen von Klauseln. Klauseln bezeichnen wir mit c, d und e, Matrizen mit D, E und F. Für ein Literal L in der Klausel c schreiben wir auch L^c.*

Beispiele von Matrizen sind die folgenden. $\{\{P\},\{\neg P\}\}$ enthält zwei Klauseln mit je einem Literal, sogenannte Einerklauseln.

$$\{\{P,\neg Q,R\},\{\neg R,R\},\{\}\}$$

enthält drei Klauseln, von denen die letzte leer ist. Hier tritt das Literal R in der ersten und der zweiten Klausel — nennen wir diese Klauseln c und d — auf. Die eingeführte Indexnotation dient dazu, solche mehrfachen Vorkommen von gleichen Literalen voneinander unterscheiden zu können. Den Index werden wir aber nur nach Bedarf anfügen und würden im Extremfall dann für diese Matrix

$$\{\{P^c,(\neg Q)^c,R^c\},\{(\neg R)^d,R^d\},\{\}\}$$

schreiben. $\{\}$ bezeichnet hier die leere Klausel (in anderem Kontext auch die leere Matrix). Für sie schreiben wir auch das für die leere Menge übliche $\emptyset$.

Vorteile

Einer der Vorteile von Matrizen gegenüber Formeln ist ihre einfache Mengenstruktur. Dies hat zur Folge, daß Vertauschungen von Elementen die Matrix nicht ändern. Allgemein ist es das Ziel einer guten Repräsentation, daß sie zwei Objekte genau dann verschieden repräsentiert, wenn sich diese in mindestens einer *interessanten* Eigenschaft unterscheiden. Welche Eigenschaften interessant sind, ist dabei kontextabhängig. In dieser Weise sind (für unsere Zwecke uninteressante) Eigenschaften der logischen Operationen (nämlich ihre gegenseitige Austauschbarkeit sowie Assoziativität, Kommutativität und Idempotenz) unmittelbar in die Struktur miteingebaut.

Ein weiterer Vorteil ist die Neutralität von Matrizen (als Mengen) gegenüber der Art der intendierten Beweisführung, sei es eine direkte (affirmative) oder indirekte (mittels Widerspruch). Mit anderen Worten, bei Matrizen gibt es diesen Unterschied im Gegensatz zu Formeln nicht mehr. Andererseits wollen wir uns unter Matrizen durchaus auch logische Formeln vorstellen können. Welche Formel eine gegebene Matrix repräsentiert, muß demnach davon abhängen, welche dieser beiden verschiedenen Beweismöglichkeiten man bevorzugt.

Definition 2.2.2

Beziehung zu Formeln

Sei F eine Matrix, also von der Gestalt $\{c_1, \ldots, c_n\}$ mit $c_i = \{L_{i1}, \ldots, L_{im_i}\}$. In bezug auf direkte Beweise repräsentiert

- *ein Literal sich selbst,*
- *eine Klausel die Konjunktion ihrer Literale und*
- *eine Formel die Disjunktion ihrer Klauseln.*

In bezug auf indirekte Beweise repräsentiert

- *ein Literal seine negierte Form (dh. P repräsentiert $\neg P$ und $\neg P$ repräsentiert P),*
- *eine Klausel die Disjunktion ihrer Literale und*
- *eine Formel die Konjunktion ihrer Klauseln.*

Im ersten Fall sprechen wir auch von positiver*, im zweiten von* negativer *Repräsentation.*

Die Vertauschung des "Vorzeichens" bei Literalen im Falle von indirekten Beweisen rührt natürlich davon her, daß man die gegebene Formel negiert (um dann einen Widerspruch herzuleiten). In diesem Zusammenhang sei auch daran erinnert, daß nach den de–Morganschen Gesetzen eine negierte Konjunktion zur Disjunktion — und umgekehrt — wird.

Wegen des, vermöge dieser Definition hergestellten, engen Zusammenhangs zwischen Matrizen und Formeln verwenden wir beide Begriffe nahezu synonym. Die repräsentierten Formeln sind im positiven Fall in disjunktiver, im negativen Fall in konjunktiver Normalform. Bekanntlich läßt sich jede aussagenlogische Formel in jede dieser Normalformen überführen. In diesem Sinne repräsentieren Matrizen auch Formeln, die nicht in Normalform sind, worauf wir im Abschnitt 2.5 noch genauer zu sprechen kommen.

Klauselform

In der Resolutionsliteratur beschränkt man sich ausschließlich auf indirekte Beweise. Hier versteht man unter einer Klausel eine Disjunktion (statt einer Menge) von Literalen. Unter der Darstellung einer Formel in *Klauselform* versteht man dann eine Menge solcher Disjunktionen.

Die Klauselform ist also eine Mischung zwischen der üblichen logischen Form und der in der Definition angegebenen Mengenform. Der

Grund für diese unschöne Mischform liegt darin, daß ein in einer Klausel mehrfach auftretendes Literal in der Klauselform auch mehrfach repräsentiert werden kann, während die Mengenform es nur einfach darstellt. Dieser Grund ist jedoch alles andere als überzeugend, weshalb wir dieser Praxis, mag sie auch noch so verbreitet sein, nicht folgen können. Zum einen könnte man das gleiche Argument auch für die Klauseln selbst geltend machen, da es ja in einer Matrix auch mehr gleiche Klauseln geben kann, so daß man dann konsequenterweise auch von einer Konjunktion (statt einer Menge) von Klauseln ausgehen müßte. Umgekehrt läßt sich das mehrfache Auftreten jedoch auch in der Mengenform leicht realisieren, indem man jedes Literal noch zusätzlich durch sein Auftreten markiert (siehe [Bib87]), was wir hier nur deswegen unterlassen haben, um den Leser mit dem daraus resultierenden und für das folgende relativ unwichtigen technischen Ballast zu verschonen.

Zweidimensionale Darstellung

Im Hinblick auf die im übernächsten Abschnitt eingeführte Charakterisierung gültiger Formeln wird sich eine zweidimensionale Darstellung in der Form tatsächlicher Matrizen (wie aus der linearen Algebra bekannt) als besonders anschaulich erweisen. Zu diesem Zweck stellen wir eine Disjunktion als horizontale Folge und eine Konjunktion als vertikale Folge ihrer Elemente dar. Zur Klammerersparnis bedienen wir uns der üblichen Vereinbarung, daß die logischen Operatoren durch die Folge $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftarrow$, $\leftrightarrow$ nach abnehmender Präzedenz geordnet sind. Die Matrix

$$\{\{\neg P, Q, R\}, \{\neg Q, \neg R\}, \{P, Q\}\}$$

repräsentiert demnach positiv die Formel

$$\neg P \wedge Q \wedge R \vee \neg Q \wedge \neg R \vee P \wedge Q$$

und wird in Matrixform wie folgt dargestellt.

$$\left|\begin{array}{ccc} \neg P & \neg Q & P \\ Q & \neg R & Q \\ R & & \end{array}\right|$$

Sie hat drei Spalten, die die drei Klauseln repräsentieren. Die gleiche Matrix repräsentiert negativ die Formel

$$(P \vee \neg Q \vee \neg R) \wedge (Q \vee R) \wedge (\neg P \vee \neg Q).$$

Sie ist die Negation der vorangegangenen Formel. In Matrixform lautet sie wie folgt.

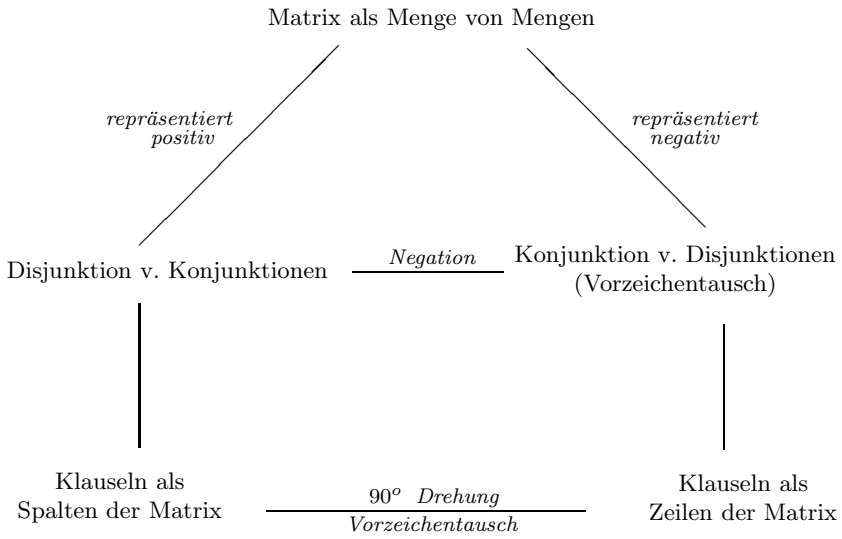


Abbildung 2.1: Darstellungen von Formeln

$$\left|\begin{array}{lll} \neg P & \neg Q & \\ Q & R & \\ P & \neg Q & \neg R \end{array}\right|$$

Es sind hier die drei Zeilen, die die drei Klauseln repräsentieren. Man beachte, daß die erste in die zweite Matrixdarstellung durch eine 90^o Drehung gegen den Uhrzeigersinn sowie durch eine Vertauschung aller Vorzeichen übergeht. Wegen dieses einfachen Zusammenhanges ist es leicht, von der einen in die andere Darstellung überzuwechseln. Im folgenden bevorzugen wir in der Regel die positive Darstellung. Der Leser sei sich dieser Wahl voll bewußt, da die meisten Autoren einer negativen Darstellung den Vorzug geben. Da dies dazu geführt hat, daß man sich der vorhandenen Alternative kaum mehr überhaupt bewußt war, soll unser Vorgehen diese eigentlich natürlichere Alternative wieder mehr in Erinnerung bringen. Was wir mit unserem Beispiel bereits illustriert haben, sei allgemein noch in der Abbildung 2.1 und wie folgt festgehalten.

Lemma 2.2.1
Ist eine Formel durch eine Matrix positiv repräsentiert, dann ist die Negation dieser Formel durch die gleiche Matrix negativ repräsentiert.

Der Sonderfall einer leeren Matrix repräsentiert eine Disjunktion ohne Disjunkte, die wir auch mit $\bot$ bezeichnen, während eine leere Klausel eine Konjunktion ohne Konjunkte repräsentiert, die wir mit $\top$ bezeichnen. Ein besonders wichtiger Spezialfall von Formeln ist durch die folgende Definition gegeben.

Definition 2.2.3
Eine Klausel heißt Horn *Klausel, wenn sie höchstens ein negiertes Atom enthält. Eine Matrix heißt* Horn *Matrix, wenn sie nur aus Hornklauseln besteht.*

Hornklausel

Die obige Matrix ist nicht Horn, da sie die Klausel $\{\neg Q, \neg R\}$ mit zwei negierten Atomen enthält. Die Matrix

$$\{\{P\}, \{\neg P, Q\}, \{\neg Q, R\}, \{\neg R\}\}$$

ist dagegen Horn. In Matrixform lautet sie

$$\left|\begin{array}{cccc} & \neg P & \neg Q & \neg R \\ P & Q & R & \end{array}\right|$$

Wählt man die negative Repräsentation, so lautet die Matrixform

$$\left|\begin{array}{cc} R & \\ Q & \neg R \\ P & \neg Q \\ & \neg P \end{array}\right|$$

Setzt man zur Vermeidung von Negationen in diese letztere Form die Implikation ein, so ergibt sich die in der Programmiersprache PROLOG so (oder ähnlich) verwendete Form.

PROLOG

$$\begin{array}{lcl} R. & & \\ Q & \leftarrow & R. \\ P & \leftarrow & Q. \\ & \leftarrow & P? \end{array}$$

In dieser Form spricht man daher von *Logikprogrammen*. In ihnen werden die einzelnen Klauseln (wie natürlichsprachliche Sätze) durch einen Punkt abgeschlossen. Zur Kennzeichnung der Anfrage verwenden wir analog das Fragezeichen. Die Zeichengebung variiert aber bei den verschiedenen PROLOG–Realisierungen.

Logikprogramm

Auf der linken Seite jeder Klausel enthält ein Logikprogramm genau ein Literal, das nur in der Anfrageklausel fehlen darf. In jüngster

Zeit werden auch *disjunktive* Logikprogramme untersucht, in denen im Unterschied zu den eben besprochenen auch auf der linken Seite jeder Klausel mehr als ein Literal auftreten darf. Auf diese letztere Form läßt sich jedoch jede aussagenlogische Formel in einfacher Weise umformen. Disjunktive Logikprogramme unterscheiden sich daher nicht merklich von der allgemeinen Form aussagenlogischer Problemstellungen.

2.3 Die Semantik der Aussagenlogik

Sätze der natürlichen Sprache beziehen sich auf die uns umgebende Wirklichkeit. Wir sagen dazu, daß sie etwas bedeuten. Was "bedeuten" heißt, wird in der *Semantik* untersucht.

Interpretation

Die Sprache der Aussagenlogik haben wir als eine reduzierte Form der natürlichen Sprache erklärt, also sollten wir uns auch hier die zugehörige Semantik klarmachen. In Abschnitt 2.1 haben wir gesehen, daß Aussagenvariablen für einfache Aussagen stehen. Wenn wir P schreiben, kann damit etwa die Aussage "Das Objekt c ist ein Affe" gemeint sein. Schon der Name "Variable" deutet an, daß eine solche Bedeutungsbindung nicht als fest zu verstehen ist, sondern daß sie variieren kann. Mit anderen Worten, mit P kann *irgendeine* Aussage gemeint sein. Eine *Interpretation* legt fest, welche Bedeutungsbindung wir im konkreten Fall meinen.

Modell

Da unser Interesse besonders auf das Beweisen von logischen Zusammenhängen konzentriert ist, kommt es uns in der Regel auf den Inhalt der Aussage gar nicht im einzelnen an. Vielmehr genügt es meist zu wissen, ob die betreffende Aussage als wahr oder falsch zu betrachten ist. Da mit der Menge der wahren Aussagen die der falschen (als Komplementmenge) mitbestimmt und damit insgesamt das Modell einer möglichen Welt beschrieben ist, ergibt sich also die folgende Definition, die aus der Menge $\mathcal{P}^0$ der Aussagenvariablen (Definition 2.2.1) die Menge der wahren Aussagen als Modell bestimmt.

Definition 2.3.1
Ein Modell *für* $\mathcal{P}^0$ *ist eine endliche Teilmenge von* $\mathcal{P}^0$ *.*

Wahrheitswert

Hat man ein Modell und bildet mit einer logischen Operation aus einfachen Aussagen eine kompliziertere Aussage, dann läßt sich einfach feststellen, ob die kompliziertere Aussage in dem Modell wahr ist oder nicht. Denn für die einfachen Aussagen wissen wir den Wahrheitswert aus dem Modell und der Wahrheitswert der komplizierteren Aussage berechnet sich daraus etwa mittels der bekannten Wahrheitstabellen (siehe unten), die die natürliche Bedeutung der logischen Operationen widerspiegeln.

Da unsere im letzten Abschnitt eingeführten Matrizen solche komplizierteren Aussagen repräsentieren, gilt das gleiche für sie, was in der folgenden Definition zu Ausdruck kommt.

Definition 2.3.2
Unter einem Modell $\mathcal{M}$ hat bei positiver Repräsentation

- *ein Atom P den Wert $\top$ (wahr), wenn $P \in \mathcal{M}$, andernfalls den Wert $\bot$ (falsch),*
- *ein negiertes Atom $\neg P$ den Wert $\bot$, wenn $P \in \mathcal{M}$, andernfalls den Wert $\top$,*
- *eine Klausel den Wert $\top$, wenn alle Literale in ihr den Wert $\top$ haben, andernfalls den Wert $\bot$, und*
- *eine Matrix den Wert $\top$, wenn mindestens eine Klausel in ihr den Wert $\top$ hat, andernfalls den Wert $\bot$.*

Der Fall der negativen Repräsentation entsteht hieraus durch Vertauschung von $\top$ mit $\bot$. Hat eine Matrix F den Wert $\top$, dann schreiben wir $\mathcal{M} \models F$ und sagen, daß $\mathcal{M}$ ein Modell für F ist. Gibt es für eine Formel ein solches Modell, dann heißt sie auch erfüllbar.

Gültiger Schluß

Diese Definition erlaubt uns, die Diskussion aus dem Abschnitt 1.3 auf einer formaleren Ebene fortzusetzen. Insbesondere haben wir dort von den gültigen Schlüssen gesprochen, die aus rein logischen Gründen, unabhängig von den Inhalten der Aussagen gelten. Der zugrundeliegende Begriff kann formal wie folgt gefaßt werden.

Definition 2.3.3
Ein Schluß von einer Menge $\mathcal{E}$ von Formeln auf eine Formel F heißt gültig*, wenn für jedes Modell $\mathcal{M}$ aus $\mathcal{M} \models E$, für alle $E \in \mathcal{E}$, $\mathcal{M} \models F$ folgt, dh. ein Modell für alle Formeln in $\mathcal{E}$ ist notwendig auch ein Modell von F. Im Falle von gültigen Schlüssen schreiben wir $\mathcal{E} \models F$ und sagen, F folgt logisch aus $\mathcal{E}$.*

Definition 2.3.4
Eine erfüllbare Formelmenge zusammen mit allen Formeln, die aus ihr logisch folgen, nennt man auch eine Theorie.

Theorie

Die erste in Abschnitt 2.1 durchgeführte Reduktion läßt sich nun mit dem folgenden *Deduktionstheorem* formal rechtfertigen.

Theorem 2.3.1
Für jede endliche Menge $\mathcal{E}$ von Formeln und zwei weiteren Formeln E, F gilt $\mathcal{E} \cup \{E\} \models F$ genau dann, wenn $\mathcal{E} \models E \rightarrow F$.

Deduktionstheorem

Durch mehrfache Anwendung dieses Theorems läßt sich die Frage nach der Gültigkeit eines Schlusses immer auf die Frage nach der logischen Gültigkeit in dem folgenden Sinne zurückführen.

Definition 2.3.5

Gültige Formel

Eine Formel F heißt (logisch) gültig *(oder eine* Tautologie*), wenn $\models F$, dh. wenn, für alle Modelle $\mathcal{M}$, $\mathcal{M} \models F$ gilt.*

Wahrheitstabelle

Unsere zentrale Frage ist damit zu der nach der Gültigkeit logischer Formeln geworden (bzw. zu der nach *Unerfüllbarkeit*, wenn man der negativen Repräsentation den Vorzug gäbe). Sie läßt sich zB. durch das bekannte Verfahren beantworten, wonach der Wahrheitswert einer gegebenen Formel für alle möglichen Wahrheitsbelegungen der Aussagenvariablen einfach der Reihe nach anhand der *Wahrheitstabellen* ausgerechnet wird (siehe Abschnitt 2.7.7). Der folgende Abschnitt wird den Grundstock für ein effizienteres Vorgehen legen.

2.4 Eine Charakterisierung der Gültigkeit

Schon im Abschnitt 1.3 haben wir festgestellt, daß sich die Gültigkeit eines Schlusses (bzw. einer Formel, wie wir jetzt nach dem im letzten Abschnitt Festgestellten sagen) an strukturellen Merkmalen zeigen muß. Eine solch strukturelle Charakterisierung werden wir in diesem Abschnitt angeben.

Die einfachste Form einer gültigen Formel ist $P \rightarrow P$. Sie ist gleichwertig mit $\neg P \vee P$. Hierbei ist zu bemerken, daß wir in diesem Buch ausschließlich die klassische Logik behandeln (in der — in Abschnitt 5.4 kurz erwähnten — intuitionistischen Logik wäre nämlich diese Gleichwertigkeit nicht gegebenen). Was immer der Wahrwert ist, der P in einem Modell zukommt, eines der Literale erhält in jedem Fall den Wert wahr, also wird diese Disjunktion immer wahr sein. Sie bleibt auch wahr, wenn man noch weitere, beliebige Disjunktionsglieder hinzufügt. Umgekehrt hat jede gültige Disjunktion von Literalen eine solche Gestalt. Zur Prüfung ihrer Gültigkeit muß man also nur für jedes Disjunktionsglied feststellen, ob auch dessen Negation als Disjunktionsglied auftritt.

Pfade

Damit bietet sich ein weiteres Verfahren — neben der bereits im letzten Abschnitt erwähnten Wahrheitswertmethode — zum Nachweis der Gültigkeit einer beliebigen Formel an. Nämlich man transformiert in bekannter Weise die gegebene Formel in konjunktive Normalform. Aufgrund der Eigenschaften der Konjunktion ist eine solche Formel gültig genau dann, wenn jedes der Konjunktionsglieder gültig ist. Letztere sind

aber genau von der eben besprochenen Form. Der Nachteil dieses Verfahrens besteht in dem großen Aufwand, der für die Transformation in Normalform zu erbringen ist. Die nun folgende Charakterisierung beruht auf der Idee, daß man diese Konjunktionsglieder bereits aus der gegebenen Formel ablesen kann, ohne die Transformation explizit durchzuführen. Es handelt sich nämlich genau um die wie folgt definierten Pfade.

Definition 2.4.1
Ist $\{c_1, \ldots, c_n\}$ *eine Matrix, so ist jede Menge* $\{L_1^{c_1}, \ldots, L_n^{c_n}\}$ *ein* Pfad durch *diese Matrix. (Die indizierten Literale waren in Definition 2.2.1 eingeführt.)*

So hat die Matrix

$$\left|\begin{array}{cccc} & \neg P & \neg Q & \neg R \\ P & Q & R & \end{array}\right|$$

die vier Pfade

$$\{P, \neg P, \neg Q, \neg R\}, \{P, \neg P, R, \neg R\}, \{P, Q, \neg Q, \neg R\}, \{P, Q, R, \neg R\}\,.$$

Man kann sie sich anschaulich als Pfade vorstellen, die die Matrix von links nach rechts über die Literale durchlaufen.

Definition 2.4.2
Eine Konnektion *in einer Matrix ist eine Teilmenge der Form* $\{P, \neg P\}$ *eines Pfades durch die Matrix. Eine Menge von Konnektionen in einer Matrix heißt eine* Paarung *(englisch mating). Eine Paarung heißt* aufspannend*, wenn jeder Pfad mindestens eine ihrer Konnektionen enthält.*

Konnektion

Die Literale einer Konnektion werden auch *komplementär* zueinander genannt. Ebenso heißt ein Pfad *komplementär*, wenn er eine Konnektion (als Teilmenge) enthält. Eine Matrix heißt *komplementär*, wenn jeder ihrer Pfade komplementär ist, dh. wenn es eine Paarung gibt. Die letzteren Begriffe sind in der Aussagenlogik redundant und gewinnen erst im nächsten Kapitel an Bedeutung.

Komplementarität

Die obige Matrix enthält eine solche aufspannende Paarung, die in Abbildung 2.2 veranschaulicht ist. Nach dem Vorangegangenen gilt das folgende Charakterisierungstheorem der Konnektionsmethode.

Theorem 2.4.1
Eine Formel ist gültig genau dann, wenn sie eine aufspannende Paarung aufweist.

Charakterisierungstheorem

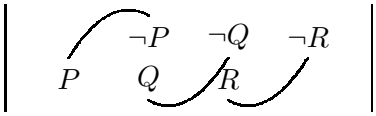


Abbildung 2.2: Eine aufspannende Paarung

Komplementarität

Unter einem *Beweisverfahren* verstehen wir ein Verfahren zum Nachweis der Gültigkeit einer Formel. Unter der *Konnektionsmethode* verstehen wir die Familie der Beweisverfahren, die auf dem vorangegangenen Theorem beruhen. Eines dieser Verfahren haben wir oben bereits beschrieben. Weitere werden folgen, nachdem wir uns im nächsten Abschnitt mehr Klarheit über die Tragweite solcher Verfahren für Formeln verschafft haben, die nicht in Normalform sind.

2.5 Normalformtransformationen

Die im Abschnitt 2.2 eingeführten Matrizen repräsentieren, wie dort erwähnt, Formeln in Normalform. Im allgemeinen sind wir natürlich auch an der Frage nach der Gültigkeit von Formeln interessiert, die nicht in Normalform sind. Diese lassen sich jedoch in Normalform äquivalent überführen, dh. die Ausgangsformel und die resultierende Formel der Transformation haben bei jeder Interpretation den gleichen Wahrheitswert (siehe hierzu Abschnitt 3.4).

Distributionsverfahren

Dazu ersetzt man im wesentlichen jede Teilformel der Gestalt $F \leftrightarrow G$ durch die Formel $F \wedge G \vee \neg F \wedge \neg G$ (oder auch durch die Formel $(F \rightarrow G) \wedge (F \leftarrow G)$, was aber im weiteren Verlauf weniger von Vorteil ist), jede Teilformel der Gestalt $F \rightarrow G$ durch $\neg F \vee G$ und eliminiert alle Negationszeichen mit Ausnahme derjenigen in den Literalen durch Anwendung der de–Morganschen Gesetze. Die entstehende *Und–Oder–Formel* ist dann in sogenannter *Negationsnormalform*. Anschließend ersetzt man unter Anwendung des Distributivgesetzes jede Teilformel der Gestalt $E \wedge (F \vee G)$ durch $E \wedge F \vee E \wedge G$, woraus schließlich eine Formel in disjunktiver Normalform resultiert, die dann als Matrix dargestellt werden kann. Wir wollen dieses Transformationsverfahren als das *Distributionsverfahren* bezeichnen. Man beachte das verdoppelte Auftreten von E im Ergebnis ebenso wie die Verdoppelung des Formelteils bei der Ersetzung der Äquivalenz. Beides führt zu einer erheblichen, tatsächlich exponentiellen Formelaufblähung durch dieses Verfahren.

So entstehen bei der Anwendung des Distributionsverfahrens auf die

Formel $(A_1 \vee B_1) \wedge \ldots \wedge (A_n \vee B_n)$ insgesamt 2^n Klauseln.

Der Rechenaufwand, den ein solch exponentielles Verfahren verursacht, läßt sich wohl auch in Zukunft von keiner Rechenmaschine bewältigen. Man beachte dabei jedoch, daß sich solche Aussagen immer auf den denkbar schlechtesten Fall beziehen, so daß ein exponentielles Verfahren in der Praxis durchaus brauchbar sein kann.

Nicht-normalform

Umgekehrt könnten wir zur Behandlung von Formeln, die nicht in Normalform sind, auch den Begriff der Matrix dadurch verallgemeinern, daß wir in den Klauseln statt Literale selbst wieder Matrizen zuließen. Solch beliebig tief geschachtelte Matrizen (also Mengen von Mengen von ... von Mengen von Literalen) repräsentieren Formeln in Negationsnormalform. Mit ihnen läßt sich alles durchführen, was wir bisher mit den Normalformmatrizen gemacht haben und weiter machen werden. Insbesondere gilt das Charakterisierungstheorem der Konnektionsmethode analog auch für diesen allgemeinen Fall illustrieren (siehe Kapitel II in [Bib87]). Wir werden dies an vereinzelten Beispielen (zB. in Abschnitt 2.7.3 im Zusammenhang mit der Reduktion FACTOR sowie in der Abbildung 5.1).

Definitorische Transformation

Der Nachteil eines solchen Vorgehens ist der unvermeidlich umfangreichere technische Aufwand, der viele Leser abschreckt. Man gewinnt dabei aber andererseits den enormen Vorteil, daß die durch die Normalformtransformation verursachte, oben erwähnte Formelaufblähung völlig vermieden werden kann. Insoweit ergibt sich also die Wahl zwischen einem exponentiellen und damit im Hinblick auf die Erfolgschancen äußerst ungünstigen Verfahren (dem Distributionsverfahren) und einem technisch sehr anspruchsvollen Verfahren, das auf der Verwendung allgemeiner Matrizen beruht. Erfreulicherweise gibt es noch das folgende *definitorische* Transformationsverfahren, bei dem die Aufblähung nur relativ harmlose Ausmaße annimmt. Dieses wollen wir im folgenden kurz erläutern.

Betrachten wir etwa das Beispiel einer Äquivalenz $F \leftrightarrow G$. Wir formulieren diese Formel zur Formel $[E \leftrightarrow (F \leftrightarrow G)] \rightarrow E$ um. Sie behauptet nun E unter der Voraussetzung, daß E als Abkürzung für die ursprüngliche Äquivalenz steht. Tatsächlich genügt in dem die Abkürzung definierenden Teil der Pfeil in die E definierende Richtung, also $[E \leftarrow (F \leftrightarrow G)] \rightarrow E$. Denn erweist sich diese Formel als gültig (was im anschließenden Beweisverfahren dann getestet wird), so folgt die Gültigkeit auch für die ursprüngliche Formel, was man sich in einfacher Weise überlegen kann.

Die eben beschriebene Umformung läßt sich auf jede Teilformel anwenden. Nehmen wir zum Beispiel an, daß in unserer Ausgangsformel G tatsächlich von der Form $G_1 \vee G_2$ ist, die wir nur durch G abgekürzt

haben. Dann ergibt die Umformung

$$[G \leftarrow G_1 \vee G_2] \wedge [E \leftarrow (F \leftrightarrow G)] \rightarrow E$$

Wendet man auf diese Formel das am Beginn dieses Abschnitts beschriebene, übliche Transformationsverfahren an, so ergibt sich die folgende Formel (in disjunktiver Normalform).

$$\neg G \wedge G_1 \vee \neg G \wedge G_2 \vee \neg E \wedge F \wedge G \vee \neg E \wedge \neg F \wedge \neg G \vee E$$

Hätten wir dieses gleiche Transformationsverfahren ohne die eingeführten Abkürzungen auf die ursprüngliche Formel direkt angewandt, so ergäbe sich die folgende Formel als Ergebnis.

$$F \wedge G_1 \vee F \wedge G_2 \vee \neg F \wedge \neg G_1 \vee \neg F \wedge \neg G_2$$

In diesem einfachen Fall ist die erstere Formel mit elf Literalen noch geringfügig länger als die zweite mit acht Literalen. Man beachte jedoch, daß die zweite Formel von den ursprünglichen nichtlogischen Formelzeichen F, G_1, G_2 doppelt so viele Vorkommen wie die erste hat. Wenn diese Zeichen selbst wieder für (möglicherweise lange) Formeln stehen, dann erweist sich umgekehrt die zweite Formel als die eindeutig längere (da ja die neu eingeführten Zeichen immer atomar sind).

Das Verfahren besteht also darin, daß man nach Umformung der gegebenen Formel in die Und-Oder–Form für jede Teilformel, die nicht Literal ist, einen Namen einführt, die Konjunktion der Definitionen dieser Namen in der oben beschriebenen Weise bildet und die Gültigkeit des Namens der Gesamtformel unter der Voraussetzung dieser Konjunktion behauptet.

Vorteil

Der Grund für den Vorteil des hiermit illustrierten definitorischen Transformationsverfahrens im Vergleich mit dem Distributionsverfahren ist auch allgemein leicht einzusehen. Es wendet nämlich die aufblähenden Umformungsgesetze immer nur auf die Definitionsglieder der Gesamtformel an, die wegen der Namensverwendung immer nur von geringer Schachtelungstiefe sind. Im Gegensatz dazu werden im üblichen Verfahren diese Gesetze auf Formeln unbeschränkter Schachtelungstiefe angewandt, was genau zu der exponentiellen Aufblähung Anlaß geben kann. Eine genaue Analyse der Komplexität ergibt in der Tat, daß das definitorische Verfahren linear in der Länge der Ausgangsformel ist [Ede91] (entgegen dem von *kleinen* Beispielen erweckten Anschein). Im Vergleich zu dem exponentiellen Verhalten des üblichen Verfahrens erweist sich dies daher tatsächlich als harmlose Aufblähung, die man, jedenfalls in einem ersten Anlauf, in Kauf nehmen kann.

2.6 Ein einfacher Konnektionskalkül

Wir kommen nun zurück auf die Frage des vorvorherigen Abschnitts nach einem besseren Verfahren zum Test der Existenz aufspannender Paarungen. Zum Einstieg in derartige Überlegungen sollte man sich zunächst das folgende klarmachen.

Anzahl der Pfade

Eine Matrix mit einer Klausel von m Literalen hat m Pfade. Fügt man eine weitere Klausel mit n Literalen hinzu, so fächert sich jeder bisherige Pfad in n Pfade auf, so daß insgesamt $m \cdot n$ Pfade entstehen; usw. Die Anzahl der Pfade ist also das Produkt der Anzahlen der Literale in den Klauseln. Sie ist also eine exponentielle Funktion von der Größe der gegebenen Matrix. Ein Vorgehen, das erst alle Pfade berechnet (wie es bei der Transformation in konjunktive Normalform geschieht) und danach jeden Pfad auf Konnektionen prüft, wird daher meist schon im ersten Teil dieses Vorgehens mangels ausreichender Ressourcen scheitern.

Anzahl der Konnektionen

Andererseits kann jede Paarung aus höchstens quadratisch vielen Elementen bestehen. Denn wenn es n Literale in der Matrix gibt, dann kann eines davon nur mit n weiteren konnektiert werden und dies eben höchstens n mal (offensichtlich sogar eine sehr großzügige Abschätzung). Da es also viel mehr Pfade als Konnektionen gibt, muß eine Konnektion in der Regel gleich eine Reihe von Pfaden als komplementär erweisen. Man muß also tunlichst von den Konnektionen, und nicht von den Pfaden, ausgehen.

2.6.1 Das Extensionsverfahren

Betrachten wir unser Beispiel vom vorvorherigen Abschnitt, angereichert mit zwei weiteren Literalen.

$$\left|\begin{array}{cccc} & \neg P & \neg Q & \neg R \\ P & Q & R & \\ Q & R & & \\ \uparrow & & & \end{array}\right|$$

Extensionsschritt

Wählen wir eine erste Konnektion mit einem Element der durch einen senkrechten Pfeil markierten Klausel, etwa die P–Konnektion. Sie erweist alle Pfade als komplementär, in der sie enthalten ist (in den folgenden Abbildungen durch einen Punkt hinter $\neg P$ gekennzeichnet). In diesem Fall sind dies genau zwei. Zu prüfen ist nun noch die verbleibende Restmenge von Pfaden. Diese unterteilen wir in die Menge von Pfaden, die das P enthalten (gekennzeichnet durch die Box um das P in der Abbildung 2.3), und den Rest. In unserem Fall besteht der Rest genau aus den Pfaden, die das Q in der ersten Klausel enthalten. Diesen Rest

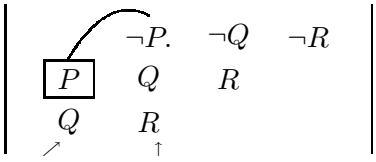


Abbildung 2.3: Erster Extensionsschritt

stellen wir zur Behandlung zurück, was in der Abbildung 2.3 durch einen schrägen Pfeil angedeutet ist, der auf das erste Element einer Folge von offenen Literalen weist. Wir sagen, daß die nachstehende Matrix aus der vorangegangenen durch einen *Extensionsschritt* hervorgegangen ist.

Im zweiten Schritt konzentrieren wir uns nun also auf die Menge der P–Pfade (ohne die bereits behandelten) und wählen daher darin eine Konnektion mit einem Literal der wieder durch einen senkrechten Pfeil markierten nächsten Klausel. Mit anderen Worten, wir wiederholen genau das, was wir im ersten Schritt durchgeführt haben, nur beschränkt auf die inzwischen reduzierte Pfadmenge. Dieser und alle weiteren Schritte sind in der Abbildung 2.4 dargestellt. Betrachten wir zum Beispiel das Ergebnis des dritten Extensionsschrittes. An diesem Punkt sind die beiden durch das Q der zweiten Klausel gehenden Pfade in der derzeit betrachteten Menge getestet (daher der Punkt hinter Q vor Beginn des nächsten Schrittes). Wir können uns daher an dieser Stelle nun auf die vorher zurückgestellte Pfadmenge durch das R der zweiten Klausel konzentrieren und von dort aus mit weiteren Extensionsschritt fortfahren. Um in die Ausgangslage eines Extensionsschrittes zu gelangen, wird in einem Zwischenschritt die Kodierung des bisher Erreichten bereinigt, wie in der Abbildung mit dem Zeichen $\sim$ illustriert. Ein solcher *Bereinigungsschritt* wird nochmals nach dem nächsten Extensionsschritt sowie am Ende des gesamten Verfahren fällig.

Bereinigungsschritt

Wir wollen das so illustrierte Verfahren *Extensionsverfahren* nennen. Allgemein wird in ihm eine Startklausel mit nur Atomen (daher auch eine *positive* Klausel oder *Zielklausel* genannt) und darin ein Literal ausgewählt. Mit diesem Literal wird ein Extensionsschritt durchgeführt, bei dem die anderen Literale der Startklausel unter der Menge der offenen Literale abgespeichert werden. Ein erneuter Extensionsschritt wird dann von der hierdurch konnektierten Klausel aus nach Auswahl eines Literals darin ausgeführt, das von dem vorher konnektierten Literal verschieden ist. Alternativen zu der dabei ausgewählten Konnektion werden

Extensionsverfahren

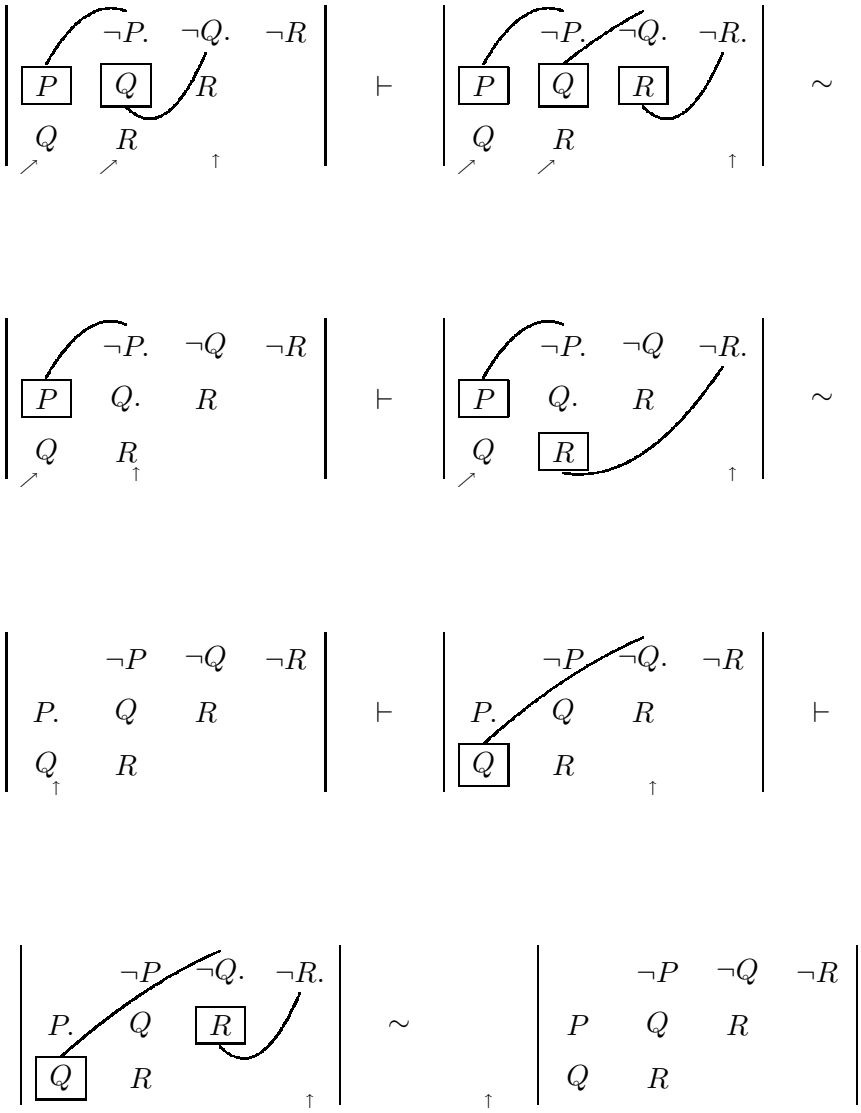


Abbildung 2.4: Weitere Schritte der Ableitung

in einer weiteren Menge (von Alternativen) verwaltet. In dieser Weise werden Extensionsschritte so lange ausgeführt, bis kein geeignetes Literal mehr verfügbar ist. In diesem Fall wird in einem Bereinigungsschritt das erste Element in der Folge der offenen Literale zur Behandlung hervorgeholt, durch das dann wieder eine Serie von Extensionsschritten ausgelöst wird, und so fort. Das Verfahren endet erfolgreich mit dem Nachweis der *Gültig*keit der gegebenen Formel, wenn schließlich die aus Literalen bestehende Menge offener Literale leer geräumt ist. Die Gesamtheit aller Schritte, die zu einer solchen Endkonfiguration geführt haben, nennt man auch eine (Extensions–) *Ableitung.*

Rücksetz-technik

Wenn das Verfahren mangels einer geeigneten Konnektion einen nötigen Extensionsschritt nicht durchführen kann, dann wird das Verfahren mit der bei solchen Suchverfahren üblichen *Rücksetztechnik* (englisch backtracking) mit der obersten Konnektion in der Alternativenmenge in genau der gleichen Situation wie bei der entsprechenden vorangegangenen Alternative neu weitergeführt. Ist die Alternativenmenge in einem solchen Falle jedoch leer, so endet das Verfahren an dieser Stelle mit dem Nachweis, daß die Formel *nicht gültig* ist.

Startklausel

Wenn wir in diesem Verfahren von der Startklausel fordern, daß sie nur Atome enthält, so bedeutet dies keine Einschränkung der Allgemeinheit wegen des folgenden, leicht zu beweisenden Sachverhalts, der auch durch unser Beispiel illustriert ist.

Lemma 2.6.1
Eine gültige (und nichtleere) Matrix enthält mindestens eine Klausel mit nur unnegierten Atomen und mindestens eine Klausel mit nur negierten Atomen.

Weiter ergibt sich die Tatsache, daß das so allgemein beschriebene Extensionsverfahren bei Beendigung die gegebene Formel je nach der beschriebenen Abschlußsituation tatsächlich als gültig bzw. nicht gültig erweist, durch eine Verallgemeinerung der am obigen Beispiel illustrierten Argumentation bezüglich der genannten Pfadmengen. Zur Präzisierung und Vervollständigung dieser Aussage wollen wir die hierzu nötigen Eigenschaften von Beweisverfahren definieren.

Definition 2.6.1

Korrektheit und Vollständigkeit

Wird eine Formel F von einem Beweisverfahren $\mathcal{B}$ als gültig erwiesen, so schreiben wir dafür auch $\vdash_{\mathcal{B}} F$.

Ein Beweisverfahren $\mathcal{B}$ heißt korrekt*, wenn jede von ihm als gültig nachgewiesene Formel F auch tatsächlich gültig ist; dh. wenn $\vdash_{\mathcal{B}} F$ gilt, dann gilt auch $\models F$. $\mathcal{B}$ heißt* vollständig *in bezug auf eine Formelklasse,*

wenn es jede gültige Formel F aus der Klasse auch als gültig nachweisen kann; dh. wenn $\models F$ gilt, dann gilt auch $\vdash_{\mathcal{B}} F$.

Ein Entscheidungsverfahren *für eine Formelklasse ist ein Beweisverfahren, das für jede Formel aus der Klasse nach endlich vielen Schritten entscheiden kann, ob sie gültig ist oder nicht.*

Entscheidungsverfahren

Theorem 2.6.1
Das Extensionsverfahren ist korrekt und (bei Start mit der Zielklausel) vollständig für die Klasse der Hornformeln .

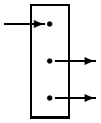
Beim Korrektheitsteil handelt es sich um die soeben informell gegebene Aussage. Wie dieser ist auch der Vollständigkeitsteil in einer Weise einzusehen, wie sie am Beispiel oben illustriert wurde. Warum die Vollständigkeit nur für Hornformeln gegeben ist, wird sich in Abschnitt 2.3.5 klären. Vorweg läßt sich sagen, daß der Extensionsschritt immer von Atomen ausgeht, die je mit einem negierten Atom in einer neugewählten Klausel konnektiert wird. Hätte diese neugewählte Klausel noch ein weiteres negatives Literal, so wüßte daher das Verfahren mit ihm nichts anzufangen.

Das Extensionsverfahren ist der Prototyp einer Reihe von praktisch besonders bedeutsamen Beweisverfahren, von denen wir einige im nächsten Abschnitt kurz besprechen werden. Vorweg wollen wir uns seine wichtigsten Merkmale nochmals vor Augen führen.

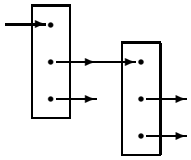
Merkmale

Das wichtigste Merkmal des Extensionsverfahrens ist die *lineare Verkettung*, dh. seine Verkettung von Klauseln mittels linearer Folgen von Konnektionen, wie sie aus Abbildung 2.4 ersichtlich ist. Man betrachte dort etwa die Folge der Konnektionen $(\{P, \neg P\}, \{Q, \neg Q\}, \{R, \neg R\})$, die zusammen mit den vier beteiligten Klauseln eine solche lineare Kette bildet. Jede beteiligte Klausel läßt sich dabei schematisch wie folgt darstellen.

lineare Verkettung



Eine Konnektion führt in die Klausel hinein. Im Falle der Startklausel fehlt dieses Kettenglied. Aus der Klausel führen dann keine, eine oder mehrere Konnektionen heraus, je nachdem wieviele Literale in ihr enthalten sind; hiervon wird jedoch in jeder Kette nur eine als nächstes Kettenglied weiterverfolgt.



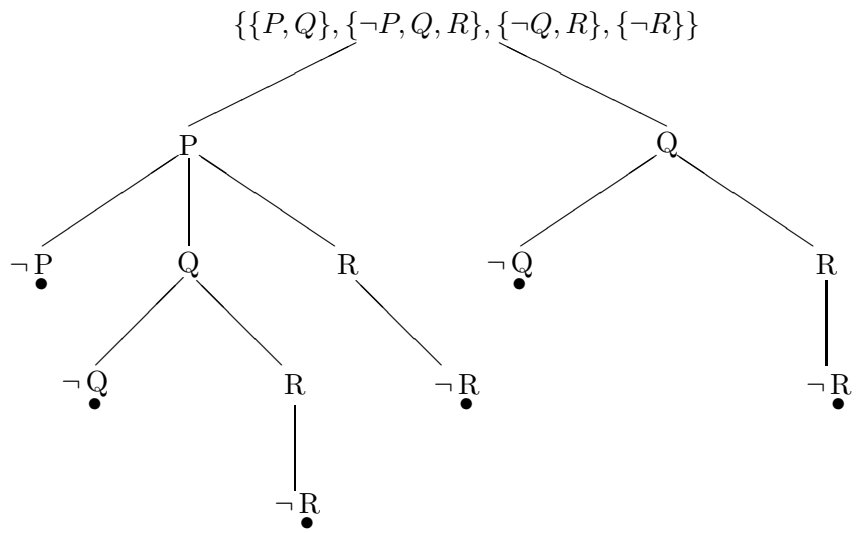


Abbildung 2.5: Eine Tableau-Ableitung

Das Endglied einer solchen Kette ist dann erreicht, wenn es aus der Klausel keine herausführende Konnektion gibt. Jede der gegebenen Klauseln ist an einer solchen Kette höchstens einmal beteiligt, so daß die Kettenlänge durch die Anzahl der gegebenen Klauseln beschränkt ist. Ein vollständiger Beweis läßt sich als ein von den Kettengliedern gebildeter Baum auffassen, dessen Äste aus den linearen Ketten besteht.

PROLOG

Der Verkettungsprozeß ist durch die betreffende Klausel und ihre Konnektionen voll bestimmbar und daher völlig lokaler Natur. Er kann daher höchst effizient implementiert werden. Genau dies wird in der Programmiersprache *PROLOG* ausgenutzt. Ein Programm in PROLOG ist im wesentlichen eine Hornformel (der Prädikatenlogik). Der Interpreter von PROLOG ist im wesentlichen ein Beweisverfahren, das hinsichtlich der aussagenlogischen Struktur auf dem Extensionsverfahren beruht.

2.6.2 Varianten des Extensionsverfahrens

Tableau-Verfahren

In [Bet55] (vgl. auch [Hin55]) wurden die heute so genannten *analytischen Tableaux* eingeführt und in [Smu71] weiter ausgearbeitet. Solche Tableaux sind Bäume, die im Falle unserer Matrizen die in Abbildung 2.5 an unserem Beispiel des vergangenen Unterabschnitts illustrierte Gestalt haben.

Die Wurzel des Tableau ist mit der gegebenen Matrix markiert. Die Nachfolger eines jeden Knotens sind die Literale einer der Klauseln der

Matrix. Ein Ast des Tableau heißt *abgeschlossen*, wenn er zwei komplementäre Literale enthält, was durch einen Punkt gekennzeichnet wird. Im Beispiel sind alle Äste abgeschlossen, was die Gültigkeit der Formel signalisiert. Ein Vergleich mit der Abbildung 2.4 macht sofort offensichtlich, daß die Äste dieses Tableau genau den im Extensionsverfahren betrachteten Pfaden entsprechen. Einen weiterführenden Vergleich werden wir in Abschnitt 4.2 geben.

Vor-Ort-Verfahren

Schon hier läßt sich also feststellen, daß das Tableau–Verfahren der Familie der Konnektionsverfahren angehört. Tatsächlich bietet es eine übersichtlichere Form, Beweise darzustellen. Bei der *Suche* nach Beweisen und in der Entwicklung von Beweisstrategien ist es dem Extensionsverfahren weit unterlegen, da dieses die vorhandene Information nicht wie beim Tableau zerstreut, sondern an ihrem Ort beläßt. Extensionsartige Verfahren sind in diesem Sinne *Vor–Ort*–Verfahren (englisch in-place method), was eine ihrer Stärken ausmacht.

Ebenso wie das Extensionsverfahren erst den Kern eines allgemeineren Verfahrens ausmacht, ist auch das Tableau–Verfahren nicht auf den bisher behandelten Fall der linearen Verkettung beschränkt. Die hier festgestellte enge Verwandtschaft gilt auch noch in der Verallgemeinerung.

Modell Elimination

Des gleichen Geistes Kind ist auch das *Modell–Eliminations–Verfahren* [Lov69]. Betrachten wir wieder unser Beispiel und denken in diesem Fall ausnahmsweise einmal im Sinne der negativen Repräsentation. Dann müßten wir die Unerfüllbarkeit nachweisen, dh. in jedem Modell muß mindestens eine Klausel, also alle ihre Literale, den Wert $\perp$ erhalten. Der Versuch, das Gegenteil zu erreichen, muß also scheitern, wovon wir uns nun überzeugen wollen. Der Leser möge sich dabei an der Abbildung 2.4 orientieren.

Nehmen wir an, P sei wahr, dann ist die erste Klausel wahr; die gesuchte falsche Klausel kann sich also nur unter den restlichen drei Klauseln befinden. Unter der bisherigen Annahme ist das erste Literal der zweiten Klausel falsch. Die Klausel wird aber trotzdem wahr, wenn wir zusätzlich Q als wahr annehmen. Exakt die gleiche Überlegung führt uns bei der dritten Klausel zu der zusätzlichen Annahme, daß auch noch R wahr sei. Nun ist aber die vierte Klausel tatsächlich falsch, so daß dieser Versuch, ein denkbares Modell zu konstruieren, gescheitert ist. Dieses Modell ist damit aus der Betrachtung *eliminiert*. Im zweiten Schritt hätten wir statt Q auch R als wahr annehmen können; dann ergibt sich aber wieder die vierte Klausel als falsch. Bleibt noch die Wahl von Q gleich im ersten Schritt, was aber scheitert, wie wir bereits gesehen haben. Damit sind alle denkbaren Modelle eliminiert, die Formel also unerfüllbar (bzw. gültig in der positiven Repräsentation).

Mag auch die Vorstellung bei diesen Überlegungen eine völlig andere sein als beim Extensionsverfahren, so sehen wir doch, daß sie zu exakt dem gleichen Vorgehen führt. Das so illustrierte Modell–Eliminations–Verfahren ist also de facto identisch mit dem Extensionsverfahren (bzw. dessen Erweiterung auf den allgemeinen Fall) und gehört damit ebenfalls zur Familie der Konnektionsverfahren.

Inverse Methode

Eine weitere Variante bildet die *inverse Methode* von S. J. Maslov [Mas68] (siehe auch [Dav73, Zam90]), die wir abschließend erklären. Am Beginn des Abschnitts 2.6.1 haben wir die Schritte der Ableitung des in Abbildung 2.4 angegebenen Beispiels erläutert. Wir sind von der P–Konnektion ausgegangen, haben damit alle diese Konnektion enthaltenden Pfade als komplementär erkannt und sodann unsere Aufmerksamkeit auf die verbliebene Restmenge konzentriert. Diese wurde unterteilt in solche Pfade, die das Q der ersten Klausel enthalten, und die restlichen Pfade. Die Behandlung der ersteren Menge wurde zurückgestellt.

Genausogut könnten wir natürlich die Reihenfolge der Behandlung vertauschen. Genau dies ist das Vorgehen der inversen Methode. Sie ist damit bis auf das strategische Vorgehen wiederum identisch mit dem Extensionsverfahren. Im einzelnen ist sie wie folgt definiert.

Die inverse Methode geht aus von einer Menge von Konnektionen in der Matrix und wendet die folgende Schlußregel so lange an, bis die leere Klausel entsteht.

$$c_1 \cup \{L_1\}, \ldots, c_n \cup \{L_n\} \vdash c_1 \cup \ldots \cup c_n$$

wobei $\{L_1, \ldots, L_n\}$ eine Klausel der Ausgangsmatrix, der sogenannte Nukleus, ist und die c_i irgendwelche Klauseln sind. Für unser Standardbeispiel starten wir mit der Menge aller Konnektionen, so daß sich die folgende Ableitung ergibt.

$$\begin{aligned}
\{\underline{P^1}, \neg P^2\}\,\{\underline{Q^1}, \neg Q^3\}\,\{Q^2, \neg Q^3\}\,\{R^2, \neg R^4\}\,\{R^3, \neg R^4\} &\vdash \\
\{\underline{\neg P^2}, \neg Q^3\}\,\{\underline{Q^2}, \neg Q^3\}\,\{\underline{R^2}, \neg R^4\}\,\{R^3, \neg R^4\} &\vdash \\
\{\underline{\neg Q^3}, \neg R^4\}\,\{\underline{R^3}, \neg R^4\} &\vdash \\
\{\underline{\neg R^4}\} &\vdash \{\}
\end{aligned}$$

Hierbei haben wir die vier Klauseln durchnumeriert und das Auftreten der Literale in einer Klausel durch Anfügen der Klauselnummer als oberer Index gekennzeichnet. Überdies wurden zur Verdeutlichung die Literale des Nukleus eines jeden Schlusses jeweils durch Unterstreichen gekennzeichnet. Abbildung 2.6 ist eine andere (zweidimensionale) Darstellung der gleichen Ableitung. Da hier eine abgeleitete Formel offensichtlich mehrfach als Prämisse auftreten kann, muß der Schluß genaugenommen so definiert werden, daß die Prämisse zusätzlich auch in der Konklusion

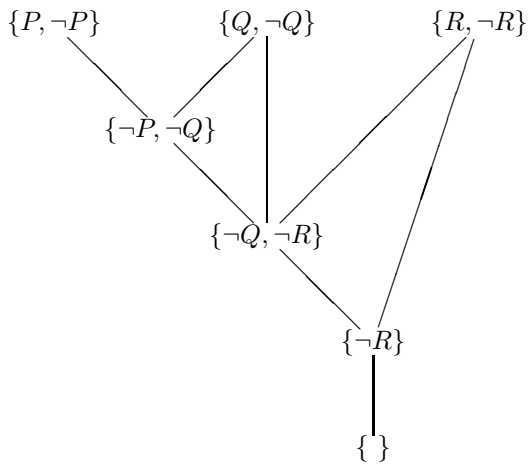


Abbildung 2.6: Maslovableitung in zweidimensionaler Darstellung

unverändert mit auftritt (vgl. auch Abschnitt 2.6.4). Wir überlassen es dem Leser, sich zu verdeutlichen, daß in der Tat diese Ableitung in der einen oder anderen Darstellung nur eine andere Form der Kodierung des oben zur inversen Methode beschriebenen strategischen Vorgehens ist. Im Abschnitt 2.6.5 werden wir auf solche strategischen Varianten noch einmal zu sprechen kommen.

2.6.3 Konsolution

Konsolution ist ein einfacher und doch sehr mächtiger Konnektionskalkül. Insbesondere ist er vollständig für die gesamte Aussagenlogik (bzw. in verallgemeinerter Form für die gesamte Prädikatenlogik). Konsolution

Erinnern wir uns, daß aufgrund des Theorems 2.4.1 alle Pfade durch eine Matrix auf ihre Komplementarität hin überprüft werden müssen, um die Gültigkeit der Matrix nachzuweisen. Die Idee der Konsolution beruht auf der folgenden Kodierung dieser Pfade, zu deren Illustration wir das folgende Beispiel heranziehen.

$$(P \wedge Q) \vee (\neg P \wedge Q) \vee \neg Q\,.$$

Ein Pfad durch eines seiner Literale, sagen wir P, ist eine einelementige Menge, $\{P\}$. Wir betrachten die Matrix daher als eine Menge von Klauseln, die aus solchen Teilpfaden bestehen.

$$\{\{\{P\},\{Q\}\},\{\{\neg P\},\{Q\}\},\{\{\neg Q\}\}\}$$

Wie immer sind die Pfade leichter aus der folgenden Matrixdarstellung dieser Klauselmenge ersichtlich.

$$\left|\begin{array}{lll} \{P\} & \{\neg P\} & \{\neg Q\} \\ \{Q\} & \{Q\} & \end{array}\right|$$

Diese Darstellung hat die folgende Eigenschaft: *Wählt man aus jeder Klausel je ein Element aus, so ist die Vereinigung dieser ausgewählten Teilpfade ein Pfad durch die Matrix, und umgekehrt kann auch jeder Pfad durch die Matrix in dieser Weise gebildet werden.* Diese Eigenschaft bleibt erhalten, wenn man auf zwei beliebig ausgewählte Klauseln die folgende Produktoperation anwendet. Man ersetze die beiden Klauseln durch eine Klausel, die aus allen möglichen Teilpfaden durch die beiden Elternklauseln besteht. Betrachten wir zum Beispiel die ersten beiden Klauseln der obigen Matrix. Die beschriebene Operation führt zur Klausel

$$\{\{P, \neg P\}, \{P, Q\}, \{Q, \neg P\}, \{Q\}\}\,.$$

Es ist offensichtlich, daß die nach dieser Operation verbleibenden beiden Klauseln, nämlich die soeben gezeigte und $\{\{\neg Q\}\}$ immer noch alle Pfade durch die ursprüngliche Matrix in der in der Eigenschaft beschriebenen Weise repräsentieren. Komplementäre Teilpfade können bei dieser Operation aber immer nur wieder zu komplementären Teilpfaden führen, so daß nach Generierung eines komplementären Teilpfades aufgrund der beschriebenen Operation dieser im weiteren Verlauf hinsichtlich der Komplementarität nicht mehr beachtet werden muß.

Genau dies macht sich die Konsolution zunutze. Sie führt die beschriebene Operation durch, eliminiert jedoch dabei entstehende komplementäre Teilpfade wie das $\{P, \neg P\}$ im letzten Beispiel. Die Menge der durch die verbleibenden Klauseln repräsentierten Pfade sind genau diejenigen, deren Komplementarität noch nachgewiesen werden muß. Wenn die leere Klausel entsteht, ist der Beweis erbracht. Der Baum in Abbildung 2.7 stellt daher einen Beweis unseres Beispiels mit der Konsolution dar. Jeder Knoten des Baumes ist mit einer Menge von Teilpfaden in der ursprünglichen Matrix markiert. Man beachte dabei den begrifflichen Unterschied zwischen einem Pfad *durch* und einem Teilpfad *in* einer Matrix. Der verwendete Begriff eines Teilpfades für die Elemente der Markierung der Knoten ist allerdings etwas schlampig. Im allgemeinen handelt es sich nämlich um die Vereinigung von solchen Teilpfaden in der ursprünglichen Matrix und nicht nur um einen einzigen. Präzise müßte man also von "Teilpfadmengen" sprechen. Wir nehmen diese kleine Schlamperei

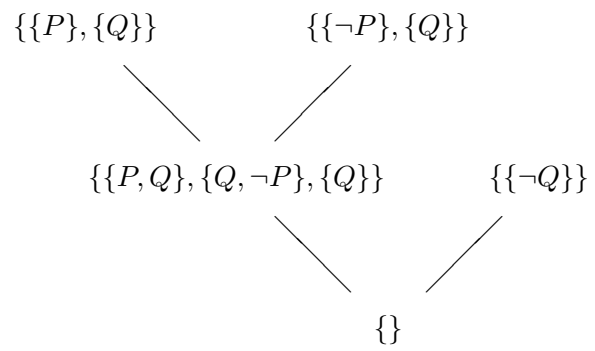


Abbildung 2.7: Eine Konsolutionsableitung

in Kauf, um die etwas umständliche Bezeichnung vermeiden zu können.

Ein Blatt des Beweisbaumes ist markiert mit der Menge $\{\{L\} \mid L \in c\}$ der einelementigen Teilpfade durch die Literale einer Klausel. Damit repräsentieren die Blätter des Baumes die Klauseln der Ausgangsmatrix.

Konsolutionsregel

Die Inferenzregel des Konsolutionskalküls, die wir ebenfalls mit *Konsolution* bezeichnen, bildet aus den Pfadmengen $\mathcal{P}$ und $\mathcal{Q}$ der beiden Elternklauseln das *Produkt*

$$\mathcal{PQ} = \{p \cup q \mid p \in \mathcal{P} \text{ und } q \in \mathcal{Q}\}$$

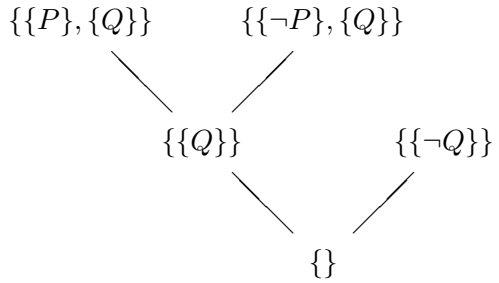
das jedoch vereinfacht wird. In jedem Fall werden bei dieser Vereinfachung komplementäre Pfade eliminiert. Jedoch sind weitere Vereinfachungen erlaubt, wie wir gleich noch sehen werden. Das Ergebnis dieser gesamten Operation nennen wir die *Konsolvente*. Insgesamt besteht dann ein Konsolutionsbeweis aus einer Ableitung der leeren Klausel aus den Ausgangsklauseln, deren einzige Operation aus Anwendungen der Konsolutionsregel besteht.

Vereinfachung

Zur Vervollständigung der Kalkülbeschreibung fehlt nur noch eine Präzisierung der weiteren möglichen Vereinfachungen, die bei der Bildung der Konsolvente erlaubt sind. Jeder Pfad p in $\mathcal{PQ}$ darf durch eine beliebige Teilmenge von p ersetzt werden.

Auch zusammen mit dieser zusätzlichen Vereinfachung ist der Konsolutionskalkül vollständig und konsistent, wenn wir zusätzlich erlauben, daß Klauseln mehrfach als Prämissen eines Schlusses auftreten, insbesondere also nicht durch die Konsolvente *ersetzt* werden, wie dies ohne diese zusätzliche Vereinfachung möglich wäre. Die Vereinfachung besagt

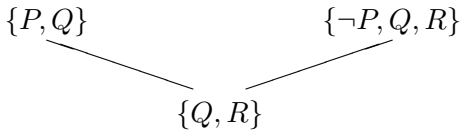
nämlich, daß man auch solche Pfade im weiteren Verlauf noch mitbetrachtet, die eigentlich schon als komplementär nachgewiesen sind. Der Sinn liegt darin, daß ohne eine solche Vereinfachung die Menge der Pfade exponentiell anwächst, was durch die Vereinfachung verhindert werden kann. So kann in unserer Beispielableitung jeder Teilpfad der ersten Konsolvente $\{\{P,Q\},\{Q,\neg P\},\{Q\}\}$ auf diese Weise zu $\{Q\}$ vereinfacht werden, so daß alle drei Pfade zu einem verschmelzen, also die Konsolvente $\{\{Q\}\}$ entsteht. Dies illustriert, daß die Menge der Pfade in der Konsolvente hierdurch radikal reduziert werden kann. Mit dieser Vereinfachung ergibt sich die folgende alternative Konsolutionsableitung für unsere Beispielmatrix.



2.6.4 Resolution

Resolution

Das verbreitetste unter den Beweisverfahren ist das *Resolutionsverfahren* [Bla37, Rob65b]. Es beruht auf der wie folgt illustrierten Resolutionsregel.



Ausgangspunkt der Regel sind zwei Klauseln, die ein komplementäres Literalpaar (hier $P, \neg P$) enthalten. Zur Bildung des Ergebnisses der Regel werden diese Literale gestrichen und der verbleibende Rest in einer neuen Klausel zusammengefaßt. Allgemein ist die Regel wie folgt definiert.

$$c_1 \cup \{L\}, c_2 \cup \{\neg L\} \vdash c_1 \cup c_2$$

Die beiden Prämissen der Regel nennt man *Elternklauseln*, die Konklusion *Resolvente*. Die beiden komplementären Literale in den Elternklauseln, über die *resolviert* wird, bilden eine Konnektion in unserem bisherigen Sinne. Man nennt sie auch den *Nukleus* des Resolutionsschlusses.

Genaugenommen heißt die Resolvente in unserem Fall

Faktorisierung

$$\{Q^{c_1}, Q^{c_2}, R^{c_2}\}\,,$$

wobei c_1, c_2 die beiden Elternklauseln bezeichnen. Offensichtlich lassen sich die beiden Vorkommen von Q jedoch *faktorisieren*, wie man sagt. Unter Faktorisierung werden wir auch, wie in der Mathematik üblich, das Herausziehen eines gemeinsamen Faktors verstehen, wie es in Abschnitt 2.7.3 unter FACTOR erklärt ist. Die Faktorisierung im hier bei der Resolution eingeführten Sinne ist davon zu unterscheiden; sie kann jedoch unter jenen Begriff der Faktorisierung subsumiert werden. Zur Illustration dieser Subsumtion betrachte man die beiden Elternklauseln in dem angegebenen Beispiel. Das gemeinsame Q läßt sich im üblichen Sinne als gemeinsamer Faktor herausziehen. Wendet man dann die Resolution auf die beiden verbleibenden Reste als neue Elternklauseln an und fügt an die Resolvente den Faktor wieder hinzu, dann entsteht genau die Resolvente im obigen Sinne, die mit der Faktorisierung im Sinne der Resolution gebildet wurde. Diese Form der Faktorisierung ist in der Prädikatenlogik jedoch nicht mehr so leicht möglich wie hier und unterbleibt dort daher oft auch. Aus diesem Grunde wird in der Resolutionsliteratur eine Klausel (einer Formel in Klauselform) auch häufig als Disjunktion, statt als Menge von Literalen definiert. Wir ziehen hier jedoch aus den in Abschnitt 2.2 erläuterten Gründen die einheitlichere Mengennotation vor.

Die allgemeinste Form des Resolutionsverfahrens besteht darin, daß man, startend mit der gegebenen Klauselmenge, so lange durch Anwendung der Resolutionsregel neue Resolventen bildet und hinzufügt, bis gegebenenfalls die leere Klausel entsteht. Eine vollständige Resolutionsableitung (mit Faktorisierung) für unser Beispiel ist in der Abbildung2.8 wiedergegeben. Wie ein Vergleich mit der im vorangegangenen Abschnitt eingeführten Konsolution zeigt, kann die Resolution als ein Spezialfall der Konsolution betrachtet werden. Man vergleiche etwa die letzte Ableitung mit Konsolution des vorangegangenen Unterabschnitts mit der Resolutionsableitung in Abbildung 2.8, nachdem man sich aus letzterer alle Vorkommen von R gestrichen denkt — beide Ausgangsformeln und beide Ableitungen sind dann offensichtlich identisch. Insbesondere werden bei diesem Spezialfall Resolution die in der Konsolution zu bildenden Teilpfade immer auf die Länge 1 vereinfacht.

Resolutionsverfahren

Unser Beispiel ist nicht allgemein genug, um demonstrieren zu können, daß die Resolvente die Elternklauseln nicht ersetzen kann, sondern (wie bei der Konsolution im letzten und bei der inversen Methode im vorletzten Abschnitt) zusätzlich betrachtet werden muß. Genaugenommen lautet zB. der erste dieser vier Schlüsse wie folgt.

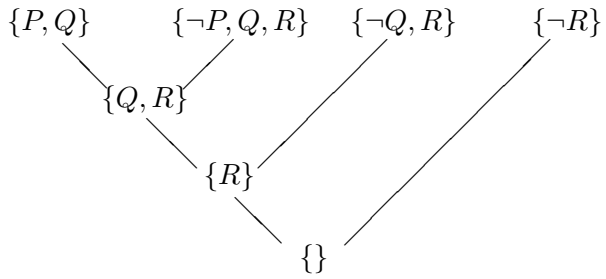


Abbildung 2.8: Eine Resolutionsableitung

$$\{\{P\},\{\neg P,Q,R\},\{\neg Q,R\},\{\neg R\}\} \vdash \{\{P\},\{\neg P,Q,R\},\{\neg Q,R\},\{\neg R\},\{Q,R\}\}$$

Für den Vergleich mit anderen Verfahren ist dieser Sachverhalt von großer Wichtigkeit.

Lineare Eingaberesolution

Unsere Beispielableitung weist zwei spezielle Eigenschaften auf, die im allgemeinen nicht gegeben sein müssen. Erstens ist die jeweils vorangegangene Resolvente gleichzeitig Elternklausel des nächsten Schlusses der Ableitung. Beschränkt man sich auf Ableitungen dieser speziellen Gestalt, so spricht man von *linearer* Resolution. Zweitens ist bei jedem Schluß eine der beiden Elternklauseln eine Eingabeklausel. Bei Ableitungen dieser speziellen Gestalt spricht man von *Eingaberesolution* (englisch *input resolution*). Unsere Ableitung verwendet also die lineare Eingaberesolution.

Es gibt viele *Verfeinerungen* von Resolution von dieser Art (siehe auch den Abschnitt 2.7.5), deren Anliegen es ist, die Fülle der Möglichkeiten zur Bildung von Resolventen einzuschränken, um auf diese Weise schneller die gesuchte Ableitung der leeren Klausel finden zu können. Dabei muß man aber darauf achten, daß man einerseits mit der Einschränkung nicht zu weit geht und das resultierende Verfahren unvollständig wird. Die lineare Eingaberesolution ist für Hornformeln vollständig. Eingaberesolution (linear oder nicht) ist aber nicht mehr vollständig für beliebige Formeln. Andererseits können Einschränkungen auch zu längeren Ableitungen führen.

LUSH Resolution

Die lineare Eingaberesolution ohne Faktorisierung nennt man auch *LUSH* Resolution [Hil74] (**L**inear resolution with **U**nrestricted **S**election function on **H**orn clauses). Während sie an die Wahl des Literals in jedem Schritt keine weiteren Bedingungen stellt, muß bei der ansonsten

identischen *SLD* Resolution (**L**inear resolution with **S**election function restricted to **D**efinite clauses; letztere sind diejenigen Hornklauseln, die genau ein negatives Literal enthalten) immer ein Literal aus der zuletzt hinzugekommenen Literalmenge gewählt werden. *SL* Resolution ist das gleiche Verfahren ohne die Beschränkung auf definite Klauseln. [KK71]

SLD Resolution

Historisch gesehen bildet SLD Resolution die eigentliche Grundlage von PROLOG. Ihr Verhalten ist im Ergebnis der durchgeführten Schritte sowie der in jedem einzelnen Schritt verwendeten Konnektion jedoch völlig identisch mit dem des Extensionsverfahrens. Der Unterschied liegt einzig in der Art und Weise, wie die Abarbeitung der einzelnen Pfade kodiert ist. Dazu betrachte man nochmals den ersten Schritt der Resolutionsableitung. Die entstehende Resolvente enthält ohne Faktorisierung genau die Literale, die im Extensionsverfahren nach dem ersten Schritt noch zur Abarbeitung anstehen. Der Leser möge sich die Resolutionsableitung ohne Faktorisierung aufschreiben, um zu sehen, daß diese Aussage analog bei allen weiteren Schritten gilt. Diese Übereinstimmung gilt jedoch nur in dem Fall der Hornformeln.

Dualität

Interessant ist der Unterschied zwischen der Resolution und der im vorletzten Abschnitt beschriebenen inversen Methode. Ein Vergleich der beiden Regeln zeigt nämlich, daß sie in der Form übereinstimmen, jedoch sind im Falle der Resolution die Ausgangsklauseln die gegebenen Formelklauseln und der Nukleus jeweils eine Konnektion der Matrix, während bei der inversen Methode umgekehrt die Ausgangsklauseln die Konnektionen (der gegebenen Formelklauseln) und der Nukleus jeweils eine Formelklausel sind. Mit anderen Worten, beide Verfahren sind in diesem Sinne dual zueinander und unterscheiden sich dadurch, daß die Rolle der Konnektionen und Klauseln vertauscht werden, worauf in dieser Klarheit G. V. Davydov [Dav73] als erster hingewiesen hat (siehe auch [Mas69, Zam90]).

Semantischer Baum

Diesen Rollentausch kann man auch an den *semantischen Bäumen* ersehen, die für die Resolution das Analogon zu den analytischen Tableaux (bzw. zu der Modellelimination) der Konnektionsmethode darstellen. In Abbildung 2.9 ist ein zu dem Tableau der Abbildung 2.5 analoger semantischer Baum zu sehen. Während dort die Menge der Nachfolgeknoten eine Klausel bildete, ist es hier je eine Konnektion. Ein Ast hieß dort abgeschlossen, wenn er eine Konnektion enthält, während hier dazu gefordert wird, daß er eine Klausel der Matrix enthält (deren Nummer im Bild jeweils angegeben wird). Die einem semantischen Baum zugrundeliegende Idee ist die systematische Berechnung der Wahrheitswerte der Matrix bei allen denkbaren Interpretationen. So werden in der ersten Verzweigung die Fälle P wahr bzw. falsch unterschieden usw. Mit den semantischen Bäumen ist es leicht, sich die der Resolution zugrundelie-

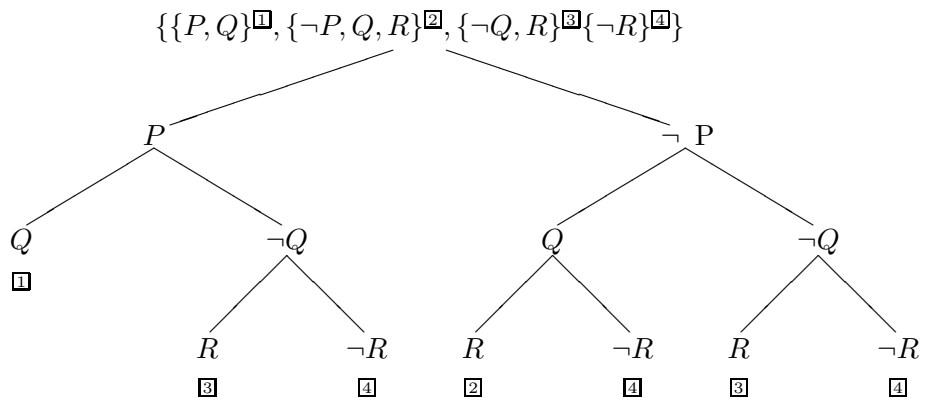


Abbildung 2.9: Ein semantischer Baum

gende Idee klarzumachen.

2.6.5 Strategien

Rückwärts-strategie

Schon in dem einfachen Fall des im Abschnitt 2.6.1 besprochenen Extensionsverfahrens haben wir eine Reihe von jedenfalls zunächst willkürlich erscheinenden strategischen Entscheidungen getroffen. Dies beginnt schon mit der Wahl der Startklausel. Unsere Wahl fiel hier auf die positive Klausel, die in der Regel das zu beweisende Ziel bzw. (im Falle von PROLOG) die Anfrage repräsentiert. Unter der Vorstellung, daß die anderen Klauseln zum Teil Regeln repräsentieren, die von den Fakten hin zu diesem Ziel führen, spricht man in diesem Fall von einer *Rückwärtsstrategie* (englisch backward chaining strategy), die ausgehend vom Ziel den Weg über die rückwärts angewandten Regeln bis zu den Ausgangsfakten zurückverfolgt. Unter der alternativen Vorstellung, daß das Ziel den Gipfel eines Berges darstellt, spricht man hier auch von einer *Abwärtsstrategie* (englisch top-down strategy). In jedem Fall spricht man auch von einem *zielorientierten* Vorgehen.

Eine weitere Wahl haben wir im Extensionsschritt getroffen, indem die verbleibenden Literale in der Ausgangsklausel abgespeichert wurden, während die in der konnektierten Klausel eine vorrangige Behandlung erfuhren. Genausogut hätten wir uns für die umgekehrte Wahl entscheiden können, wie es tatsächlich bei der dort auch besprochenen inversen Methode geschieht. Dies führt aber nur zu einer Vertauschung der Reihenfolge der Schritte. Wenn ein Beweis schließlich gefunden wird, der

diesen Extensionsschritt miteinschließt, dann wirkt sich eine solche Vertauschung auf die Beweislänge und den Suchaufwand nicht aus. Wie wir im vorangegangenen Abschnitt erwähnt haben, gewährt SLD–Resolution diese Wahlmöglichkeit — im Gegensatz zur LUSH–Resolution — nicht.

Vorwärtsstrategie

Im Falle der Wahl der Startklausel macht es durchaus Sinn, die Alternativen mit in Erwägung zu ziehen. So kann man umgekehrt von den Fakten ausgehen und sich entlang der Regeln bis zum Ziel vorarbeiten. Diese *Vorwärts-* oder *Aufwärtsstrategie* (englisch forward bzw. bottom-up strategy) muß aber mit Vorsicht eingesetzt werden, da sonst zuviele Zwischenaussagen abgeleitet werden, die mit dem Ziel absolut nichts zu tun haben (was bei der Abwärtsstrategie automatisch ausgeschlossen ist). Als besonders erfolgreich erweisen sich Mischstrategien, die während einer Vorbereitungsphase in beschränktem Maße zunächst vorwärts und danach rückwärts vorgehen.

Strategie

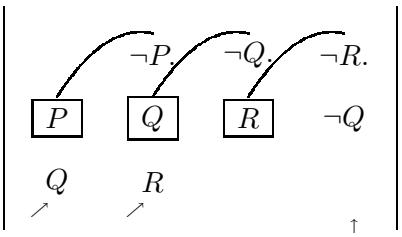
Über diese einfachsten Strategieformen hinaus werden wir im Verlauf des Buches noch eine Reihe weiterer Strategien kennenlernen. Dabei verwenden wir den Begriff der *Strategie* im Zusammenhang mit Deduktionsverfahren immer in dem Sinne einer Steuerung des Vorgehens, die die Korrektheit und Vollständigkeit des zugrundeliegenden Kalküls erhält.

2.7 Allgemeine Tautologiebestimmung

Das im letzten Abschnitt vorgestellte Extensionsverfahren hat sich als vollständig nur für den Bereich der Hornformeln erwiesen. Zum Beweis beliebiger Formeln müssen also allgemeinere Techniken herangezogen werden. Grundlegende solcher Techniken haben wir mit der Konsolution, der Resolution, den semantischen Bäumen, den Tableaux usw. bereits kennengelert. Weitere werden wir in diesem Abschnitt besprechen.

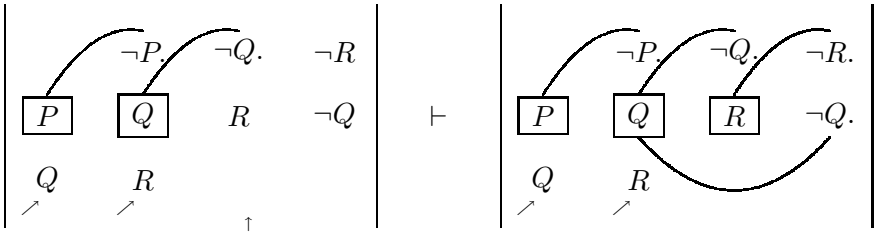
2.7.1 Das allgemeine Extensionsverfahren

Tatsächlich bedarf es zur Behandlung allgemeiner Formeln in Normalform nur eines geringfügigen Zusatzes zu dem im Abschnitt 2.6.1 behandelten Extensionsverfahren. Betrachten wir zur Illustration nochmals die in der Abbildung 2.4 dargestellte Situation nach Vollendung des dritten Extensionsschrittes. Mit einem zusätzlichen Literal $\neg Q$ in der vierten Klausel lautet sie wie folgt.



Allgemeiner Extensionsschritt

Ein Extensionsschritt im bisherigen Sinne ist nicht möglich. Zur Behandlung hinsichtlich der Prüfung auf Komplementarität stehen in dieser Situation alle Pfade durch $P, Q, R, \neg Q$ an, von denen es in diesem Fall genau einen gibt. Dieser ist aber offensichtlich komplementär, da er die Konnektion $\{Q, \neg Q\}$ enthält. Um dies zu erkennen, verallgemeinern wir den Extensionsschritt in folgender Weise. Es wird nämlich in einem Extensionsschritt nicht nur das Literal des letzten "Kettengliedes" mit einem Punkt versehen (wie $\neg R$ in unserem Beispiel), sondern darüber hinaus alle Literale (wie im Beispiel das $\neg Q$) in dieser Klausel, zu denen es ein komplementäres Literal in der Menge der eingerahmten Literale gibt. Diese Menge nennt man auch den *aktiven Pfad.* Der hier betrachtete allgemeine Extensionsschritt lautet also wie folgt.



Schleife

Wie die Abbildung illustriert, bilden die beiden Q–Konnektionen zusammen mit der R–Konnektion eine *Schleife*. Sie besteht aus einer linearen Kette, also einer Folge von Kettengliedern und einer Konnektion zurück zum Kettenanfang. Zu der bisher betrachteten linearen Verkettung treten hier also zusätzlich Schleifen dieser Form hinzu. Insgesamt verfahren wir also im *allgemeinen Extensionsverfahren* genauso wie im einfachen Extensionsverfahren, verwenden jedoch in jedem Schritt die soeben erklärte allgemeine Extension. Insbesondere ist hier auch die Ruecksetztechnik@Rücksetztechnik wie bisher erforderlich.

Separation

Wie im Abschnitt 2.6.5 erläutert, kann sich das Extensionsverfahren bei Verwendung von Aufwärtsstrategien in Beweisketten verlieren, die keine Verbindung mit der zu beweisenden Aussage haben. Zur Integration dieses Falles muß man noch einen sogenannten *Separationsschritt* vorsehen. Er wird verwendet, wenn es überhaupt keine Konnektion mit einem Literal aus dem aktiven Pfad zur Ausführung eines Extensionsschrittes

und auch keine noch nicht betrachteten Alternativen zu bisher durchgeführten Extensionsschritten gibt. In diesem Fall werden alle Klauseln mit Literalen des aktuellen Pfades abgesondert und das Verfahren mit den verbleibenden Klauseln völlig neu gestartet. Wir sprechen zur Unterscheidung vom *allgemeinen Extensionsverfahren mit Separation.*

Varianten

Neben dieser ersten Variante des allgemeinen Extensionsverfahrens gibt es noch viele weitere (siehe Abschnitt 2.7.4). Wir nennen hier nur noch die Möglichkeit, auf die strikte lineare Verkettung im Extensionsschritt zu verzichten und die Extension gegebenenfalls nur auf eine Konnektion mit irgendeinem aktuellen Pfadliteral (also nicht notwendig dem letzten) zu gründen. Diese Variante des allgemeinen Extensionsverfahrens, das in [Bib87] mit CP_1^0 bezeichnet wurde, kann auf die Rücksetztechnik ohne Verlust der Vollständigkeit verzichten. Sie ist in Abbildung 2.10 in Form eines Struktogrammes genau beschrieben. F bezeichnet darin die gegebene Matrix, *WAIT* die bisher durch schräge Pfeile illustrierte Menge offener Literale und p den aktuellen Pfad. Alle drei Varianten des allgemeinen Extensionsverfahrens sind korrekt und vollständig. Für die zuletzt genannte Variante findet sich der Beweis unter II.6.13 in [Bib87] zu dem folgenden Satz.

Theorem 2.7.1
Das Extensionsverfahren CP_1^0 ist korrekt und vollständig für die Aussagenlogik. Zugleich ist es ein Entscheidungsverfahren.

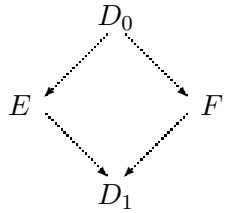
Definition 2.7.1

Beschränktheit und Konfluenz

Ein Beweisverfahren heißt beschränkt*, wenn es eine Funktion f gibt, so daß für jede Formel F mit ℓ Literalen das Verfahren nach höchstens $f(\ell)$ Schritten zum Stillstand kommt.*

Ein Beweisverfahren heißt konfluent*, wenn das folgende gilt: erreicht das Verfahren bei zwei unabhängigen Anwendungen, ausgehend von einer gemeinsamen Konfiguration D_0, die jeweiligen Konfigurationen E und F, so gibt es in beiden Fälle Fortsetzungen des Verfahrens hin zu irgendeiner gemeinsamen Konfiguration D_1.*

Die Eigenschaft der Konfluenz wird durch folgende Skizze illustriert.



Das allgemeine Extensionverfahren ist nicht konfluent. Genau deshalb ist es bei ihm erforderlich, die Rücksetztechnik einzusetzen. Das

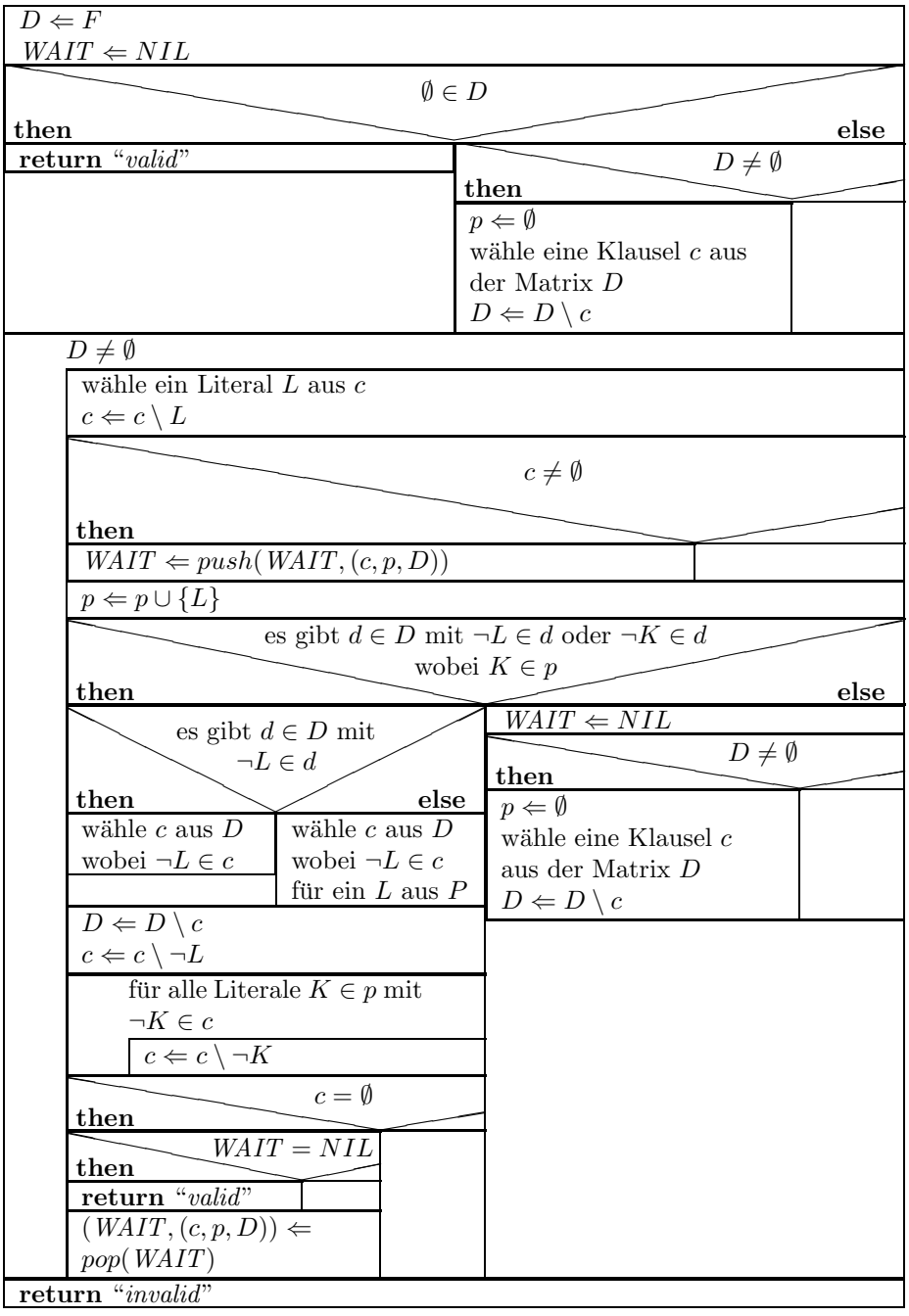


Abbildung 2.10: Das Extensionsverfahren CP_1^0

Verfahren CP_1^0 dagegen ist konfluent. Bei ihm spielt es für den endgültigen Erfolg der Beweissuche keine Rolle, welche Auswahl das Verfahren im Falle alternativer Extensionsmöglichkeiten trifft. Rücksetztechnik ist bei ihm daher nicht erforderlich. Wir wollen die Vor– und Nachteile an dieser Stelle nicht abwägen, sondern nur darauf hinweisen, daß in der Verallgemeinerung auf die Prädikatenlogik erster Stufe auch CP_1^0 wieder die Rücksetztechnik benötigt. Innerhalb der Aussagenlogik sind jedoch alle Verfahren beschränkt und genau aus diesem Grunde Entscheidungsverfahren.

Für die im Abschnitt 2.6.2 behandelten Varianten gilt das dort Gesagte auch für ihre Beziehung zum allgemeinen Extensionsverfahren. Insbesondere behandeln sowohl das Tableau–Verfahren als auch das Modell–Eliminations–Verfahren die hier besprochenen Schleifen. Im letzteren Fall werden sie durch einen sogenannten *Reduktionsschritt* realisiert. Aus unserer Sicht ist diese Bezeichnung irreführend, da eine Reduktion nicht stattfindent, und wird daher in diesem Zusammenhang nicht weiter gebraucht.

2.7.2 Komplexitätsbetrachtungen

Komplexitätstheorie

Beweisverfahren für die Aussagenlogik testen, ob ein Element aus der Menge aller Formeln zugleich ein Element aus der Teilmenge der Tautologien ist (das Tautologieproblem). Eine abgeschwächte Fragestellung ist, ob eine solche Formel zugleich ein Element aus der (umfassenderen) Teilmenge der erfüllbaren Formeln ist (das Erfüllbarkeitsproblem). Beide Fragestellungen spielen eine wichtige Rolle in der Komplexitätstheorie. Umgekehrt haben die Ergebnisse der Komplexitätstheorie eine wichtige Bedeutung für die Entwicklung von Beweisverfahren. Es wird dem Leser daher sehr empfohlen, sich mit grundlegenden Begriffen daraus vertraut zu machen (siehe zB. [GJ79]). Hier können wir nur die wichtigsten Punkte ohne weitere Erläuterung erwähnen.

$\mathcal{NP}$–vollständig

Das Erfüllbarkeitsproblem ist *$\mathcal{NP}$–vollständig*. Das Tautologieproblem ist *co–$\mathcal{NP}$–vollständig*. Dies hat zur Folge, daß bis heute keine polynomiellen Algorithmen für eines dieser beiden Probleme bekannt sind. Von allen Verfahren, die wir bisher genannt haben (also Resolution, Extensionsverfahren, Tableau–Verfahren usw.), wissen wir vielmehr, daß sie *exponentiell* sind. Dies bedeutet, daß es eine exponentielle Funktion f und Formeln mit n Literalen gibt, zu deren Beweis das Verfahren mindestens $f(n)$ Schritte benötigt. Ein solches exponentielles Verfahren erreicht mit wachsender Formelgröße schnell die Leistungsgrenze auch des größten verfügbaren Supercomputers.

Man könnte daraus den Schluß ziehen, daß schon aus Komplexitäts-

gründen der Versuch der Automatisierung des logischen Schließens vergeblich sein muß. Die folgenden Gründe lassen es dennoch sinnvoll erscheinen, an diesem Versuch weiterzuarbeiten.

Ein erster Grund ist durch die Tatsache gegeben, daß alle entscheidenden Fragen der Komplexitätstheorie ungelöst sind. Ob die Klasse $\mathcal{P}$ der polynomiell lösbaren Probleme gleich oder ungleich $\mathcal{NP}$ ist, ist eine völlig offene Frage, auch wenn die Mehrzahl der Forscher derzeit für "ungleich" plädiert, so als ob sie dies schon wüßten, nur halt noch nicht beweisen könnten. Jede Verbesserung eines Beweisverfahrens ist daher zugleich ein Beitrag zur Lösung dieses $\mathcal{P}$–$\mathcal{NP}$–Problems. Unabhängig davon ist die ebenfalls völlig offene Frage, ob $\mathcal{NP}$ gleich oder ungleich co–$\mathcal{NP}$ ist.

Polynomielle Fälle

Der Mensch ist offenbar in der Lage, auch schwierige logische Zusammenhänge zu erkennen. Es erscheint unwahrscheinlich, daß er nicht den gleichen Gesetzen unterliegt. Wäre also $\mathcal{P}$ ungleich $\mathcal{NP}$, so würde man daraus schließen, daß es trotzdem möglich ist, interessante Theoreme zu beweisen, nur eben nicht alle, genauso wie der Mensch. Insbesondere lassen sich Teilklassen von Formeln angeben, innerhalb derer das Tautologieproblem polynomiell lösbar ist. Zum Beispiel sind die Klasse der Hornklauseln (siehe Abschnitt 2.7.3), die Klasse der Formeln, die nur Äquivalenzen als logische Zeichen enthalten, und die Klasse der Klauselmengen mit maximal zwei Literalen je Klausel linear lösbar (bei drei Literalen dagegen ist man schon wieder bei einem $\mathcal{NP}$–vollständigen Fall).

Nicht zuletzt liegt ein guter Grund für die Forschung auf dem Gebiet der Deduktion im Erfolg der bestehenden Systeme, mit denen erstaunlich komplexe Beweise entdeckt werden konnten.

2.7.3 Reduktionen

Kriterien

Da kein Algorithmus existiert, der in allen Fällen in akzeptabler Zeit eine Lösung eines gegebenen Beweisproblems findet, ist es von großer Bedeutung, ein solches Problem vor Anwendung eines allgemeinen Beweisverfahrens nach Möglichkeit zu reduzieren. Hierfür stehen eine Reihe von Reduktionsmechanismen zur Verfügung, die wir im folgenden kurz zusammenfassen wollen. Von einer solchen Reduktion verlangen wir, daß

- sie das Problem echt in ein "kleineres" Problem überführt;
- die Gültigkeit des reduzierten Problems die des ursprünglichen Problems impliziert und möglichst, aber nicht notwendigerweise, auch umgekehrt;
- ihre Anwendbarkeit "schnell" getestet werden kann; und
- ihre Ausführung ebenfalls "schnell" erfolgen kann.

Statt "schnell" kann man zB. auch "polynomiell" im komplexitätstheoretischen Sinne lesen, jedoch ist in der Regel sogar darunter "linear" oder höchstens "quadratisch" zu verstehen. Auch das "kleiner" läßt sich in einem gegebenen Kontext exakt fassen. Wenn nichts anderes ausdrücklich erwähnt wird, dann sind alle nachfolgenden Mechanismen Reduktionen in diesem Sinne.

Reduktionen

MULT Wenn das gleiche Literal in ein und derselben Klausel mehrfach (multipel) auftritt, kann das Auftreten auf eines reduziert werden.

PURE Wenn zu einem Literal nirgends in der Matrix ein komplementäres auftritt (engl. "pure"), kann die Klausel, in der es auftritt, entfernt werden.

TAUT Wenn in einer Klausel ein Literal und sein Komplement gleichzeitig auftreten, kann die (tautologische) Klausel entfernt werden.

SUBS Eine Klausel, die Obermenge einer anderen Klausel in der Matrix ist, kann entfernt werden (da sie von der kleineren Klausel "subsumiert" wird).

UNIT Ein Literal in einer Klausel, zu dem es ein komplementäres Literal in einer Einerklausel (englisch unit clause) gibt, kann aus der Klausel entfernt werden.

ISOL Wenn die Literale einer Konnektion nirgends sonst in der Matrix auftreten (die Konnektion also "isoliert" ist), dann können die hierdurch konnektierten Klauseln durch die Resolvente ersetzt werden.

CIRC Unter bestimmten Umständen können tautologische Zyklen (englisch circuits) von Konnektionen gestrichen werden (siehe [Bib81] für die etwas subtilen Details).

FACTOR Haben zwei Klauseln je ein Vorkommen ein und desselben Literals, so lassen sie sich durch eine (Nicht–Normalform–) Klausel mit nur einem Vorkommen dieses Literals ersetzen, indem man es mittels des Assoziativitäts– und Distributivitätsgesetzes ausfaktorisiert.

MONOT Ein Pfad, der eine Obermenge eines anderen Pfades in der Matrix darstellt, kann ignoriert werden. Deshalb kann zB. eine Klauselmenge in der Matrix (nicht notwendig in Normalform) dann entfernt werden, wenn noch eine Untermenge von ihr in der Restmatrix enthalten ist (Monotonieeigenschaft).

POLYN Zusätzlich lassen sich alle polynomen Verfahren, die für Spezialformelklassen bekannt sind (siehe den vorangegangenen Unterabschnitt), als Reduktionen im obigen Sinne deuten.

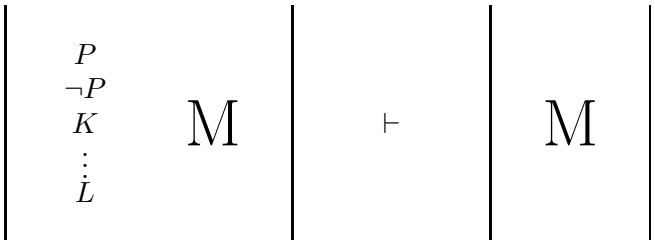
Wir benutzen in dieser Aufzählung den Begriff der Faktorisierung im üblichen mathematischen Sinne. In der Resolutionsliteratur wird er anders gebraucht, obwohl dieser andere Gebrauch sich unter dem hier definierten

Begriff in einem gewissen Sinne subsumieren läßt, was in Abschnitt 2.6.4 kurz erläutert ist.

Die Bedeutung dieser Reduktionen, die in Tabelle A4.2 nochmal zusammengefaßt sind, kann nicht hoch genug eingeschätzt werden. Zum Beispiel ist in [Bib90] gezeigt worden, wie ein exponentieller Beweis (der "Taubenschlag"–Formeln (englisch pigeonhole formulas)) durch Anwendung von $MONOT$ in Verbindung mit der Umbenennung von Variablen zu einem linearen Beweis reduziert werden kann, worauf wir in Abschnitt 2.7.6 kurz eingehen werden. Überdies dürfte die Liste wohl nicht vollständig sein. So läßt sich zB. ISOL wesentlich verallgemeinern [LSBB92] (vgl. auch Abschnitt 4.4.4). Unbefriedigend ist die Situation insofern, als man sich eher ein einheitliches Reduktionsprinzip anstelle der vielen unterschiedlichen Reduktionsregeln wünschen würde. Hierzu gibt es derzeit Ansätze (in denen die Subsumtion als vereinheitlichendes Prinzip eine Rolle spielt), aber keine vollständigen Lösungen.

Nachweis der Gültigkeit

Eine der an Reduktionen geknüpften Bedingungen war die Erhaltung der Eigenschaft der Gültigkeit. Dies muß natürlich für jede einzelne Reduktion nachgewiesen werden (größtenteils in Abschnitt II.6 von [Bib87]). Die hierbei verwendbare Beweistechnik ist von so allgemeiner Bedeutung, daß wir sie wenigstens an einem Fall, nämlich für $TAUT$, illustrieren wollen. Schematisch läßt sich die Reduktion wie folgt darstellen.



Auf der linken Seite ist eine tautologische Klausel und die Restmatrix M, auf der rechten Seite nur die Restmatrix nach Entfernen der Klausel dargestellt, wie es die Reduktion verlangt. Zu beweisen ist, daß die linke Seite genau dann gültig ist, wenn die rechte Seite gültig ist. Nach dem Charakterisierungstheorem 2.4.1 müssen wir also zeigen, daß die Komplementarität aller Pfade der linken Seite die aller Pfade der rechten Seite impliziert und umgekehrt. Die Umkehrung ist trivial, da jeder Pfad durch die rechte Seite nur eine Erweiterung eines (dann nach Voraussetzung komplementären) Pfades der linken Seite ist.

Sei also die Komplementarität aller Pfade der linken Seite vorausgesetzt und sei p irgendein Pfad durch die rechte Seite. Sowohl $\{P\} \cup p$ als auch $\{\neg P\} \cup p$ sind Pfade durch die linke Seite, also komplementär nach

Voraussetzung. Wäre in beiden Fällen das P bzw. $\neg P$ der tautologischen Klausel in der Konnektion enthalten, die die beiden Pfade jeweils als komplementär erweist (im anderen Fall ist ohnehin nichts zu zeigen), so hieße das, daß p sowohl ein komplementäres Literal zu P, also $\neg P$, als auch eines zu $\neg P$, also P, enthält; beide bilden aber zusammen wieder eine Konnektion, so daß p in jedem Fall auch komplementär ist. Da p beliebig gewählt war, ist der Beweis also erbracht.

Beweisprinzip

Die Komplementarität zweier (im allgemeinen mehrerer) Pfade der einen Matrix hat uns die eines Pfades in der anderen garantiert. Dieses Beweisprinzip ist bei allen Reduktionen anwendbar. Hiermit läßt sich auch ein eleganter Beweis für die Korrektheit und Vollständigkeit der Resolution führen (siehe Abschnitt IV.1 in [Bib87]).

Hiermit haben wir zugleich *TAUT* illustriert. *MULT* und *PURE* sollten ohne weitere Illustration verständlich sein. Bei *SUBS* ist nur zu beachten, daß für den Vergleich der betreffenden Klauseln hier natürlich nur die Literale, nicht deren Vorkommen beachtet werden müssen.

Linearer Aufwand bei Horn

Eine mehrfache Anwendung von *UNIT* unter Einsatz immer der gleichen Einerklausel, sagen wir $\{L\}$, wird schließlich dazu führen, daß das Literal $\neg L$ nirgends mehr in der Matrix auftritt. Mittels *SUBS* läßt sich zudem jede von $\{L\}$ verschiedene Klausel aus der Matrix eliminieren, die das Literal L enthält. Auf $\{L\}$ schließlich ist *PURE* anwendbar, so daß L völlig aus der Matrix eliminiert ist. Hiermit läßt sich leicht einsehen, daß Hornklauseln in linearer Zeit beweisbar sind, wie in Abschnitt 2.7.2 erwähnt. Da es nämlich nach Lemma 2.6.1 in einer gültigen Matrix mindestens eine rein negative Klausel gibt, muß jede gültige Hornformel mindestens eine Einerklausel enthalten, da Hornklauseln ja überhaupt höchstens ein negatives Literal enthalten. Mit dieser läßt sich mit dem eben beschriebenen Verfahren jedes Auftreten der Variablen dieser Einerklausel in der Matrix eliminieren. Das Ergebnis dieses Reduktionsprozesses bleibt eine Hornformel, da ja nur Literale oder Klauseln entfernt werden. Also muß auch sie wieder eine Einerklausel enthalten, mit der genau das gleiche wiederholt werden kann und so fort. In jedem Schritt wird die Matrix mindestens um ein Literal kleiner, also ist man nach spätestens n Schritten fertig, wobei n die Anzahl der Literale in der ursprünglichen Matrix ist.

Faktorisierung

Unter den verbleibenden Reduktionen wollen wir nur noch *FACTOR* am Beispiel der *vollen* Matrizen $\mathcal{M}(n)$ über n Variablen illustrieren. Sie bestehen aus 2^n Klauseln mit je n Literalen, so daß alle kombinatorischen Möglichkeiten hinsichtlich des Vorzeichens ausgeschöpft werden. Zum Beispiel ist $\mathcal{M}(3)$ die folgende Matrix.

$$\left|\begin{array}{cccccccc} P & \neg P & P & \neg P & P & \neg P & P & \neg P \\ Q & Q & \neg Q & \neg Q & Q & Q & \neg Q & \neg Q \\ R & R & R & R & \neg R & \neg R & \neg R & \neg R \end{array}\right|$$

Wenden wir nun $FACTOR$ auf die R–Literale an, so erhalten wir die folgende Matrix.

$$\left|\begin{array}{cc} \mathcal{M}(2) & \mathcal{M}(2) \\ R & \neg R \end{array}\right|$$

Hierbei ist die Untermatrix $\mathcal{M}(2)$ jeweils durch ihren Namen repräsentiert. Wie diese Darstellung klar illustriert, müssen zum Beweis der Gültigkeit dieser Matrix drei verschiedene Arten von Pfadmengen (dh. Teilmatrizen) auf Komplementarität untersucht werden, nämlich $\{R\} \cup p, p \cup \{\neg R\}, p \cup p'$, wobei p, p' jeweils beliebige Pfade durch $\mathcal{M}(2)$ sind. Nach Konstruktion führt die Anwendung von $PURE$ auf die ersten beiden Mengen (Teilmatrizen) zur Elimination von $R, \neg R$, während $MONOT$ die dritte Pfadmenge (Teilmatrix) auf (die Pfade durch) $\mathcal{M}(2)$ reduziert. Da offensichtlich dieser Reduktionsprozeß für beliebiges n ganz genauso durchführbar ist, sehen wir, daß $\mathcal{M}(n)$ auf $\mathcal{M}(n-1)$ in 2^n+1 Schritten reduziert werden kann, ein Schritt mehr, als es Klauseln in der Matrix gibt. Iteriert man diesen Prozeß, so ergibt sich nach bekannten Gesetzen ein quadratischer Beweis der gesamten Matrix, mit praktisch keinerlei Suchaufwand.

Volle (und "ziemlich" volle) Matrizen lassen sich also schnell beweisen. Dasselbe gilt für "ziemlich" leere Matrizen, wie man sich leicht überlegt. Die wirklich schwierigen Matrizen sind sozusagen die "halbvollen".

Die Faktorisierung kann in gewissem Sinne als Umkehrung des in Abschnitt 2.5 beschriebenen Distributionsverfahren zur Transformation in die Normalform angesehen werden. Der Nachteil des Distributionsverfahrens wird durch die soeben gezeigten Beweise nochmals demonstriert, denn das Extensionsverfahren würde zum Beweis der vollen Matrix exponentiell viele Schritte benötigen (wovon sich der Leser am Beispiel von $\mathcal{M}(3)$ überzeugen kann). Wir erinnern jedoch auch an das dort besprochene definitorische Verfahren, das es erlaubt, die faktorisierte Nicht-Normalform–Matrix wieder in eine optimalere Normalform zurückzutransformieren. Dieser Weg bietet aber keinen Ersatz für die Anwendung von $MONOT$. Wir fügen hier noch die Bemerkung an, daß

die Faktorisierung auch den Nebeneffekt der Schleifenelimination hat und daher den Beweisprozess hin zu den linearen Verkettungsbeweisen vereinfacht.

Dynamische Reduktion

Unsere Diskussion von Reduktion war bis hierher von der Vorstellung einer statischen Reduktion einer gegebenen Problemstellung geprägt. Reduktionen spielen jedoch auch während eines Beweisprozesses mit irgendeinem Verfahren eine wichtige Rolle. Hat man im Verlauf eines solchen Prozesses zB. die Konnektion in den beiden Klauseln

$$\{P, Q\},\ \{\neg P, R\}$$

bearbeitet, so verbleiben die beiden Unterziele $\{Q, R\}$ entweder in Form einer Klausel (im Falle von Resolution) oder als markierte Unterziele (in manchen Varianten der Konnektionsmethode). Das Gesamtproblem mag nun aber eine weitere Klausel enthalten, die diese beiden Literale enthält. Eine solche Klausel kann dann aber zu diesem Beweisprozeß nichts beitragen, da sie von den genannten Unterzielen subsumiert wird. Diese Subsumtion wurde aber erst nach Abarbeitung der genannten Konnektion sichtbar, weshalb wir von einer dynamischen Reduktion sprechen. Das Gebiet der dynamischen Reduktion ist äußerst komplex, wenn auch für die Praxis sehr wichtig. Wir müssen daher diesbezüglich auf die Spezialliteratur verweisen [EOP91, SA90, Bib87].

Wir schließen diese Ausführungen mit dem Hinweis auf eine Form der Reduktion besonderer Art. Hat man die volle Matrix über n Variablen, so lassen sich alle Tautologien über n Variablen (modulo der Anwendung von Reduktionen und Variablenumbennungen) durch Entfernung von bis zu $n-1$ Literalen aus jeder Klausel generieren [Bib79].

2.7.4 Varianten des allgemeinen Extensionsverfahrens

Matrixreduktion

Die in Abschnitt 2.7.1 beschriebenen Varianten müssen als noch ziemlich primitive Verfahren eingestuft werden. In diesem Abschnitt wollen wir noch einige weitere Varianten und Verbesserungen kurz andeuten. Wir beginnen mit der Matrixreduktion von Prawitz. Gegeben sei eine Matrix der folgenden Gestalt.

$$\left|\begin{array}{cc} & \begin{array}{cc} K_1 & L_1 \\ \vdots & \vdots \\ K_m & L_n \\ P & \neg P \end{array} \\ \mathrm{M} & \end{array}\right|$$

Sie ist komplementär genau dann, wenn die beiden Matrizen in Abbildung 2.11 komplementär sind. Man beachte, daß diese beiden Matrizen

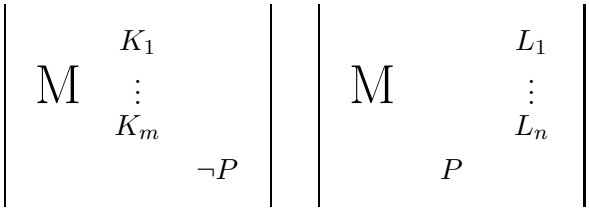


Abbildung 2.11: Prawitz-reduzierte Matrizen

aufeinandergelegt wieder die Ausgangsmatrix ergeben. In Pfaden ausgedrückt heißt dies, daß alle Pfade, die in den beiden explizit gezeigten Klauseln weder durch P noch durch $\neg P$ gehen, nicht mehr getestet werden müssen. Auch diese keineswegs offensichtliche Einsicht läßt sich leicht mit dem im letzten Abschnitt besprochenen Beweisprinzip beweisen. Hieraus ergibt sich die folgende Formulierung der Matrixreduktion.

Matrixreduktion
Wähle aus der zu beweisenden Matrix eine Klausel mit einem Literal P und eine zweite mit dem komplementären Literal $\neg P$. Zum Nachweis der Komplementarität aller Pfade durch die gegebene Matrix genügt es, die folgenden beiden Teilmengen dieser Pfade (dh. die reduzierten Matrizen) als komplementär nachzuweisen:
(i) die Menge aller Pfade, die dieses Auftreten von P, aber nicht von $\neg P$ enthalten;
(ii) die Menge aller Pfade, die dieses Auftreten von $\neg P$, aber nicht von P enthalten.

Da die reduzierten Matrizen kleiner sind als die Ausgangsmatrix, führt die wiederholte Anwendung der Matrixreduktion bei einer gültigen Matrix schließlich zu Matrizen, die eine leere Klausel enthalten. Das Verfahren ist also korrekt und vollständig. Es handelt sich aber hier nicht um eine Reduktion im Sinne des vorangegangenen Abschnitts, denn, wenn sich auch die Anzahl der Pfade verringert, so sind die beiden entstehenden Matrizen zusammen im allgemeinen doch größer als die Ausgangsmatrix.

Davis–Putnam Verfahren

Das Davis–Putnam Verfahren [DP60] besteht aus der Anwendung der Reduktionsregeln TAUT, PURE, UNIT und SUBS des vorangegangenen Unterabschnitts sowie einer Spaltungsregel, die als die folgende Kombination der Matrixreduktion mit FACTOR und PURE aus dem vorangegangenen Unterabschnitt verstanden werden kann.

Spaltungsregel, 1. Fassung
Wähle aus der zu beweisenden Matrix ein Literal P. Ziehe den gemeinsamen Faktor P aus allen Klauseln, die P enthalten, und analog für $\neg P$. Wende auf die so erhaltene (nicht-Normalform) Matrix die Matrixreduktion an. Aus den beiden reduzierten Matrizen (die nun wieder in Normalform sind) kann das jeweilige Auftreten von $\neg P$ und P mit PURE eliminiert werden.

Ohne Regreß auf die Matrixreduktion lautet die Spaltungsregel wie folgt, wobei wir bei der Formulierung auf die bestehende Analogie zu der Matrixreduktion geachtet haben.

Spaltungsregel, 2. Fassung
Wähle aus der zu beweisenden Matrix ein Literal P, das auch in negierter Form auftritt. Ersetze die Menge der Klauseln durch die folgenden beiden reduzierten Klauselmengen:
(i) die Menge der Ausgangsklauseln, jedoch ohne all diejenigen Klauseln, die P enthalten, und ohne die Literale $\neg P$;
(ii) die Menge der Ausgangsklauseln, jedoch ohne all diejenigen Klauseln, die $\neg P$ enthalten, und ohne die Literale P.

Verallgemeinertes Prawitz Verfahren

Das Prawitzsche Verfahren ist in [Bib79] so erweitert worden, daß die Reduktion nicht durch ein komplementäres Literalpaar, das man auch als die volle Matrix über einer Variablen auffassen kann, sondern durch die volle Matrix über beliebig vielen Variablen bewerkstelligt werden kann, was zu wesentlich kürzeren Beweisen führt, weil in jedem Schritt eine große Reduktion stattfinden kann.

Dissolution

Eine weitere Verallgemeinerung ist unter dem Namen *Dissolution* in [MR87], hier auf Nicht-Normalform–Matrizen, gegeben worden. Dieses erlaubt somit, vollen Gebrauch von der Reduktion *FACTOR* zu machen, ohne eine Rücktransformation in die Normalform der Art zwischenschieben zu müssen, wie sie im letzten Abschnitt erläutert wurde.

Verbesserungen

Wegen der großen Bedeutung einer effizienten Behandlung von Faktoren innerhalb der gegebenen Matrix, dh. von Literalen, die in der Matrix mehrfach auftreten (solche mehrfach auftretenden Literale lassen sich in der Konnektionsmatrix durch zusätzliche Faktorisierungskonnektionen kennzeichnen), ist auch ihr Einbau direkt in den Extensionsprozeß zusammen mit einer Reihe weiterer Verbesserungen bewerkstelligt worden. Darunter befinden sich die Vermeidung tautologischer Zyklen, die Entfernung redundanter Teilziele, die Vermeidung mehrfacher Lösungen des gleichen Teilziels und andere. Die Arbeiten [BH82, Fro85, Joh89] sowie Abschnitt IV.6 in [Bib87] befassen sich mit diesen Fragestellungen.

Die soeben beschriebene Zielrichtung der Forschung ist noch bei weitem nicht voll ausgeschöpft. Selbstverständlich gehen wir jetzt davon aus, daß die eingangs gegebene Matrix mittels der Reduktionen so weit wie möglich vereinfacht wird, bevor der Extensionsprozeß einsetzt. Nun haben wir aber am Beispiel des Beweises der vollen Matrizen schon gesehen, daß sich weitere Möglichkeiten zur Reduktion erst im Verlauf des Beweises herausstellen können. Zu solch *dynamischen* Reduktionen gibt es bislang wenig Ansätze.

2.7.5 Resolutionsvarianten

Konsensus-Regel

Durch unsere mengenorientierte Darstellung hat es sich bei der Einführung des Resolutionsprinzips im Abschnitt 2.6.4 erübrigt, eigens darauf hinzuweisen, daß in der Resolution heutzutage ausschließlich die negative Darstellung bevorzugt wird. Dies war keineswegs immer so. So hat W. v. O. Quine [vOQ55] die Resolution in der positiven Darstellung unter dem Namen *consensus rule* bzw. Konsensus–Regel eingeführt. (Genaugenommen hat Quine die Tautologiereduktion (siehe Abschnitt 2.7.3) mit in die Konsensus–Regel integriert, dh. er hat als Konklusion keine Klausel mit komplementären Literalen zugelassen.) In diesem Abschnitt wollen wir nun noch einige Verfeinerungen der Resolution kurz beschreiben.

Stützmenge

Wir haben bereits im Abschnitt 2.6.5 bei der Aufwärtsstrategie darauf hingewiesen, daß Beweisketten nicht notwendig mit der eigentlich zu beweisenden Aussage verknüpft sein müssen. Diese Beobachtung ist natürlich verfahrensunabhängig und bedarf zum Beispiel auch bei der Resolution Abhilfemaßnahmen. Sie rangieren hier unter dem Stichwort "Stützmenge" (englisch set-of-support). Dazu wird aus der Menge der Klauseln eine Teilmenge T (zB. die das zu beweisende Theorem repräsentierende Klauselmenge) als Stützmenge ausgewählt. Dann sind nur solche Resolutionsschlüsse erlaubt, bei denen mindestens eine Elternklausel von T unterstützt wird, dh. aus T selbst ist oder einen "Vorfahren" darin hat. Diese *Stützstrategie* ist vollständig genau dann, wenn die Restklauselmenge erfüllbar ist (siehe [Sti87], wie diese Bedingung zur Behandlung von Inkonsistenzen ausgenützt werden kann). Sie ist als eine Variante der Abwärtsstrategie aufzufassen; es gilt also das zu dieser (insbesondere hinsichtlich Mischstrategien) Gesagte.

Einerresolution

Wir haben bereits die *Einerresolution* in Form einer Reduktion kennengelernt. Allgemein wird bei ihr gefordert, daß eine Elternklausel nur ein Element enthält. Sie ist vollständig für den Fall der Hornklauseln (was wir implizit im Abschnitt 2.7.3 mit gezeigt haben), nicht aber im allgemeinen Fall. In der *Einer-Resolventen Resolution* (englisch unit-resulting resolution) geht man von einer Elternklausel mit $n > 1$ Li-

teralen und $n-1$ (nicht notwendig verschiedenen) Einerklauseln aus, so daß durch $n-1$ Einerresolutionsschritten, dh. *UNIT* Reduktionen, wieder eine Einerklausel entsteht. Dieser Prozeß wird aber als ein einziger Einer-Resolventen Schritt aufgefaßt, was den Vorteil hat, daß die Zwischenresolventen nicht gespeichert werden müssen.

Hyper-resolution

In der P_1 *Resolution* [Rob79] muß eine Elternklausel positiv, in der N_1 *Resolution* negativ sein. Beide sind vollständig im allgemeinen Fall. In der (ebenfalls vollständigen) *Hyperresolution* [Rob65a] werden mehrere P_1 Resolutionsschritte gleichzeitig ausgeführt. Man wählt dazu eine Klausel mit mindestens einem negierten Atom, den *Nukleus*, und zu jedem negativen Atom darin eine positive Klausel (den *Elektronen*), die dieses Atom enthält. Mit diesen Elternklauseln bildet man die *Hyperresolvente* genauso, als wären die entsprechenden P_1 Resolutionsschritte einzeln ausgeführt worden. Die *negative Hyperresolution* ist analog definiert, nur sind "positiv" und "negativ" vertauscht. Mit der Hyperresolution sind sehr komplexe Beweise gefunden worden.

Konnektionsgraphen Resolution

Beim Vergleich des Extensionsverfahrens mit der linearen Resolution (Abschnitt 2.6.4) konnten wir sehen, daß auch die Resolution Pfade durch die Matrix auf Konnektionen testet. Sie tut dies aber in einer komplexeren Weise, was Vor– und Nachteile mit sich bringt. Eine der möglichen Redundanzen liegt darin, daß ohne weiteres nicht ausgeschlossen ist, daß über eine Konnektion mehrfach resolviert wird. Dies auszuschließen ist die Motivation hinter der *Konnektionsgraphen Resolution* [Kow75]. In ihr werden solche Konnektionen, über die bereits resolviert wurde, eigens markiert und so von den anderen unterschieden. Da ja aber bei jedem Resolutionsschritt neue Konnektionen entstehen können, muß auch genau mitangegeben werden, wie diese Markierungen sich vererben. Technisch wird die Markierung dadurch bewerkstelligt, daß alle unmarkierten, nicht aber die markierten, Konnektionen explizit im sogenannten *Klauselgraphen* (das ist eine Matrix mit Konnektionen) gespeichert werden. Vererbt werden dann nur die gespeicherten in einer Weise, wie es die Darstellung in Abbildung 2.12 eines solchen Konnektionsgraphen-Resolutionsschrittes illustriert.

Die Konnektionsgraphen–Resolution ist in ihren technischen Details hochgradig kompliziert. Dies zeigt sich zB. darin, daß wichtige Fragen, wie die nach einer allgemeinen, konfluenten Strategie zur Konnektionenauswahl, bis heute ungelöst sind (siehe [Bib80b] bezüglich einer "Beinahe"–Lösung. Der dort gegebene Beweis enthält jedoch eine Lücke, deren Schließung eine erneute Anstrengung erfordern würde, zu der der Autor angesichts aus seiner Sicht wichtigerer Probleme bislang keine Gelegenheit fand). Die explizite Behandlung von Konnektionen

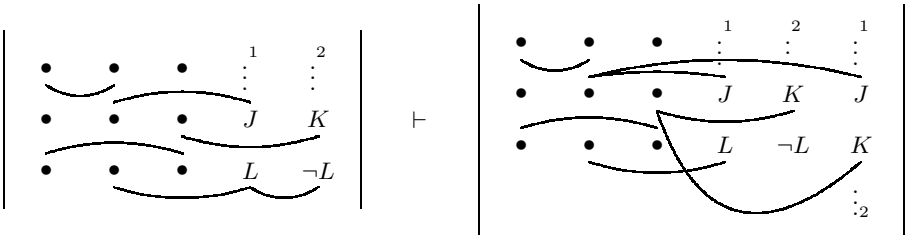


Abbildung 2.12: Ein Konnektionsgraphen-Resolutionsschritt

wirkt sich auch auf die Reduktionen aus, die entsprechend modifiziert werden müssen. Zum Beispiel ist hier *PURE* dann anwendbar, wenn das Literal in keiner gespeicherten Konnektion auftritt (obwohl möglicherweise noch komplementäre Literale dazu vorhanden sind). Hinsichtlich der subtilen technischen Details hierzu siehe [Bib81, EO91].

2.7.6 Lemmata

Lemmata

Eine der Standardtechniken in der Mathematik beruht auf der Verwendung von Lemmata und definitorischen Abkürzungen. Betrachten wir zur Illustration der zugrundeliegenden Idee die folgende Formel.

$$(P \vee Q \vee \neg P \wedge \neg Q) \wedge R \vee (S \vee T \vee \neg S \wedge \neg T) \wedge \neg R$$

Die Unterformel $P \vee Q \vee \neg P \wedge \neg Q$ in dieser Formel erweist sich als Tautologie; man kann sie daher im Hinblick auf den weiteren Tautologietest für die Gesamtformel ignorieren. Wichtiger in unserem Zusammenhang ist jedoch die Beobachtung, daß die gleiche Formel mit unterschiedlichen Benennungen (S und T anstelle von P und Q) nochmals auftritt. Erkennt man dies vor Beginn der Beweissuche, so genügen drei (statt fünf) Konnektionen zur Erbringung des Beweises der gesamten Formel. Denn entweder betrachtet man die Teilformel als Lemma, das einmal bewiesen und dann nur noch angewendet wird, oder man faktorisiert diese Teilformel aus und erhält die einfachere Formel $(P \vee Q \vee \neg P \wedge \neg Q) \wedge (R \vee \neg R)$; beide Alternativen sind nur verschiedene Sichtweisen der gleichen Technik.

Taubenschlagformeln

Der durch die Lemma–Anwendung erzeugte Effekt kann beliebig groß werden. So ist in [Bib90] anhand der sogenannten "Taubenschlag"–Formeln demonstriert worden, daß durch diese Technik exponentiell lange Beweise zu linearen Beweisen zusammenschrumpfen können. Da schon bei relativ kleinen Formeln exponentiell lange Beweise wegen ihrer Länge nicht mehr aufgeschrieben, geschweige denn gefunden werden können, er-

weist sich für diese und ähnliche Formeln die Lemma–Technik als unabdingbar. Sie ist allerdings bis heute in keinem der bekannten Systeme realisiert. Der Grund dürfte in der Tatsache liegen, daß das Erkennen solcher Lemmata selbst ein kostspieliger Prozeß ist, für den es keinen polynomen Algorithmus gibt. Da der gesamte Beweisprozeß im allgemeinen nicht polynom ist, erweist sich dieser Grund jedoch als nicht stichhaltig. Bei entsprechend vorsichtigem Einsatz kann die Lemmatechnik durchaus die bestehenden Verfahren verbessern.

Resolution mit Erweiterungen

Zur Resolution gibt es eine entsprechende Erweiterung, die sogenannte *Resolution mit Erweiterungen* (englisch resolution with extensions) [Tse68], wobei es sich bei den Erweiterungen hier um Namen für die Lemmata handelt.

2.7.7 Weitere Verfahren

Darstellungsvarianten

Schon die Aussagenlogik läßt sich in einer Reihe verschiedener Darstellungen behandeln; für ausdrucksstärkere Logiken gilt das dann noch umso mehr. Zum Beispiel sind wir von der Darstellung mit den üblichen logischen Junktoren (wie $\wedge, \vee, \rightarrow,$ usw.) ausgegangen. Schon hier besteht aber ein große Vielfalt in der Auswahl der Junktoren, da es bekanntlich eine ganze Reihe verschiedener vollständiger Junktorensysteme gibt. Dann haben wir aber zusätzlich die Matrixrepräsentation, die Klauselform und die PROLOG–Form kennengelernt.

Für jede dieser und weiterer Repräsentationsformen läßt sich das Tautologieproblem studieren. Es sollte daher den Leser nicht überraschen, daß die bisher genannten Verfahren zur Lösung des Tautologieproblems bei weitem noch nicht die Fülle bekannter Verfahren ausschöpfen. Andererseits ist in den weiteren Verfahren keine Technik erkennbar, die für eine wesentliche Verbesserung des Verhaltens signifikant und in irgendeiner Form nicht schon in den bisherigen Verfahren enthalten wäre. Ungeachtet dessen wollen wir einige dieser weiteren Verfahren kurz nennen.

Wahrheitstafel

Das bekannteste Verfahren überhaupt ist wahrscheinlich das mittels der sogenannten Wahrheitstafel (oder –tabelle) [Koh70], das wir bereits in Abschnitt 2.3 kurz erwähnten. Hierzu legt man für jede in der gegebenen Formel auftretenden Variable je eine Spalte an, in der die Wahrheitswerte $0, 1$ für die Variablen so eingetragen sind, daß jedes mögliche Tupel genau einmal auftritt. Weitere Spalten enthalten dann den aufgrund der bekannten Gesetze resultierenden Wahrheitswert für die gesamte Formel bzw. für ihre Teilformeln. Für die Praxis erweist sich dieses Verfahren als untauglich.

Schaltwerkstheorie

Die booleschen Funktionen, also die Aussagenlogik, spielen in der

Schaltwerkstheorie eine ebenso wichtige Rolle wie in vielen anderen Anwendungen. Leider hat dies zu einer außerordentlichen Redundanz der Forschungen geführt, da die Forscher der Schaltwerkstheorie das Tautologieproblem ohne Bezugnahme auf die in dem Gebiet der Deduktion erzielten Resultate untersucht haben und umgekehrt. Dies betrifft auch die Terminologie. So wird die Negation einer Variablen x (wofür wir bisher Zeichen wie P verwendeten) als x' (oder auch als $1 - x$), die Disjunktion mit dem Symbol $+$ und die Konjunktion mit dem Symbol $\cdot$ bezeichnet [Koh70]. Auch spricht man in der Schaltwerkstheorie von *Absorption* anstelle von (einem Spezialfall der) *Subsumtion*, um nur eines der vielen Beispiele zu nennen.

Shannonsches Expansionstheorem

Das grundlegende Verfahren in der Schaltwerkstheorie beruht auf dem *Shannonschen Expansionstheorem*. Dieses besagt, daß für jede boolesche Funktion $f(x_1, \ldots, x_n)$, das heißt in unserer bisherigen Sprechweise für jede aussagenlogische Formel über diesen n Variablen, die folgende Gleichheit gilt.

$$f(x_1, \ldots, x_n) = x_1 \cdot f(1, x_2, \ldots, x_n) + x_1' \cdot f(0, x_2, \ldots, x_n)$$

Mehrfache Anwendung dieser Expansionsregel führt zu der *Shannonschen Normalform* [Sha38, Koh70, Mor82]. Sie ist eine Variante dessen, was wir in Abschnitt 2.6.4 einen semantischen Baum genannt haben. Wie dort bildet sie auch hier die Grundlage für ein weiteres Verfahren. Es führt zur disjunktiven Normalform, deren Gültigkeit unmittelbar erkennbar ist. Auch dieses Verfahren ist den im vorangegangenen behandelten Verfahren weit unterlegen.

Boolesche Unifikation

Die dem Shannonschen Expansionstheorem zugrundeliegende Gleichung bildet auch das Fundament der in modernen Beweissystemen integrierten *booleschen Unifikation* [DSv90].

In jüngster Zeit hat man auch bei den Schaltwerkstheoretikern die Bedeutung von faktorisierten Formen erkannt, was auch dort zu wesentlich verbesserten Verfahren geführt hat [MB88]. Es wäre endlich an der Zeit, daß diese Mehrfachentwicklung vermieden würde und die Kräfte sich auf die schwierigen noch ausstehenden Probleme gemeinsam konzentrieren würden.

2.7.8 Primimplikanten

Prim-implikanten

Wie im Vorangegangenen erwähnt, sind Tautologien auch für die Entwicklung boolescher Schaltungen von großer Wichtigkeit. In dieser Anwendung haben bei der Minimierung solcher Schaltungen besonders auch die sogenannten Primimplikanten eine große Bedeutung. Diese ha-

ben jüngst eine weitere wichtige Rolle bei den sogenannten Begründungsverwaltungssystemen (englisch truth maintenance systems) erhalten.

Definition 2.7.2
Eine Klausel C heißt Implikant *einer Matrix F, wenn $C \to F$ gültig ist. C heißt* Primimplikant*, wenn C Implikant ist und es keinen davon verschiedenen Implikanten C' von F gibt, so daß $C' \to C$ gültig ist (dh. C' auch ein Implikant von C ist).*

Berechnung von Primimplikanten

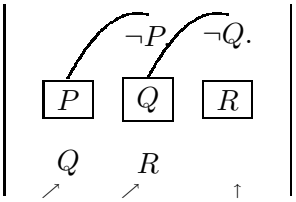
Wie immer gehen wir hier von der positiven Repräsentation von Matrizen aus. Für den negativen Fall gibt es den analogen Begriff eines (Prim-) Implikanten.

Die Resolvente bei der Resolution (wegen der positiven Repräsentation sollte man genauer sagen, bei der Konsensus–Regel) ist ein Implikant der Matrix, die die Elternklauseln enthält. Auch die beim Extensionsverfahren in einem gewissen Status noch nicht als komplementär erwiesenen Pfade durch eine gegebene Matrix F bilden einen Implikanten C dieser Matrix. Der im hierauf folgenden Extensionsschritt entstehende Implikant C' dieser Art von F ist auch ein Implikant von C. In diesem Sinne ist also C' "mehr prim als" C. Das Extensionsverfahren (und in ähnlicher Weise auch die Resolution) läßt sich in dieser Weise also auch zur Berechnung von Primimplikanten heranziehen [JP90].

2.7.9 Gegenbeispiele

Generierung von Gegenbeispielen

Wir haben in diesem Abschnitt Verfahren zum Test der Tautologieeigenschaft kennengelernt. Sie lassen sich auch dazu einsetzen, Gegenbeispiele zu konstruieren, wenn sich die zu beweisende Formel als nicht gültig herausstellt. Wir wollen uns das am Beispiel des allgemeinen Extensionsverfahrens klarmachen. Betrachten wir dazu das in Abschnitt 2.7.1 betrachtete Beispiel, dem wir jedoch dadurch die Eigenschaft der Tautologie entziehen, indem wir die letzte Klausel weglassen. Der Versuch eines Extensionsbeweises bleibt dann beim folgenden Zustand stecken.



In dieser Matrix gibt es nun zu keinem der Literale $\{P, Q, R\}$ des aktiven Pfades ein komplementäres Literal in der Restmatrix, da diese leer ist. Da wir wissen, daß unser Verfahren ein Entscheidungsverfahren ist,

ist an dieser Stelle bewiesen, daß die Matrix nicht gültig ist. Wir sind nun aber sogar in der Lage, eine Interpretation anzugeben, bei der die Formel falsch wird. Man gebe nämlich allen Literalen des aktiven Pfades den Wert "falsch". Mit jeder Interpretation, die diese Bedingung in einem solchen Fall erfüllt, erhält dann auch die Gesamtmatrix den Wert "falsch". Jede solche Interpretation läßt sich offensichtlich als *Gegenbeispiel* für die Formel auffassen. Wählt man umgekehrt den Wert "wahr", so ergibt sich ein *Beispiel* für die Formel.

Das so illustrierte Verfahren, Gegenbeispiele zu generieren, läßt sich allgemein auf jede beliebige nichtgültige Formel anwenden. Die etwas subtilen Details des Verfahrens können wir hier aus Platzgründen jedoch nicht wiedergeben (siehe [Hör81]).