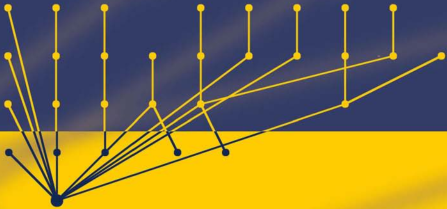
Winfried Hochstättler

Algorithmische Mathematik



Springer

Springer-Lehrbuch

Winfried Hochstättler

Algorithmische Mathematik

Springer

Winfried Hochstättler
Fernuniversität in Hagen
Fachbereich Mathematik
Lützowstr. 125
58095 Hagen
Winfried.Hochstaettler@FernUni-Hagen.de

ISSN 0937-7433
ISBN 978-3-642-05421-1 e-ISBN 978-3-642-05422-8
DOI 10.1007/978-3-642-05422-8
Springer Heidelberg Dordrecht London New York

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.d-nb.de abrufbar.

Mathematics Subject Classification (2000): 68-01

© Springer-Verlag Berlin Heidelberg 2010
Dieses Werk ist urheberrechtlich geschützt. Die dadurch begründeten Rechte, insbesondere die der Übersetzung, des Nachdrucks, des Vortrags, der Entnahme von Abbildungen und Tabellen, der Funksendung, der Mikroverfilmung oder der Vervielfältigung auf anderen Wegen und der Speicherung in Datenverarbeitungsanlagen, bleiben, auch bei nur auszugsweiser Verwertung, vorbehalten. Eine Vervielfältigung dieses Werkes oder von Teilen dieses Werkes ist auch im Einzelfall nur in den Grenzen der gesetzlichen Bestimmungen des Urheberrechtsgesetzes der Bundesrepublik Deutschland vom 9. September 1965 in der jeweils geltenden Fassung zulässig. Sie ist grundsätzlich vergütungspflichtig. Zuwiderhandlungen unterliegen den Strafbestimmungen des Urheberrechtsgesetzes.
Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Werk berechtigt auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften.

Einbandentwurf: WMXDesign GmbH, Heidelberg

Gedruckt auf säurefreiem Papier

Springer ist Teil der Fachverlagsgruppe Springer Science+Business Media (www.springer.com)

... und da ich bedachte, dass unter allen, die sonst nach Wahrheit in den Wissenschaften geforscht, die Mathematiker allein einige Beweise, das heißt einige sichere und einleuchtende Gründe hatten finden können, so war ich gewiß, dass ich mit diesen bewährten Begriffen anfangen müsse ...

René Descartes (1637)

Vorwort

Obwohl jeder schon in der Grundschule die ersten Algorithmen kennen lernt, nämlich die Addition, Subtraktion, Multiplikation und Division im Zehnersystem, ist der Begriff „Algorithmus“ kein Allgemeingut, und Mathematik wird in den weiterführenden Schulen und oft auch an Hochschulen wenig aus algorithmischer Sichtweise betrachtet. In den Anfangszeiten der Informatikausbildung wurde Mathematik vor allem aus klassischer Grundlagensicht gelehrt, die Algorithmenausbildung als Teil der Theoretischen Informatik betrachtet. In letzter Zeit ist allerdings der Anteil der Mathematik und der Theoretischen Informatik in der Informatikausbildung deutlich zurückgegangen. Ursache ist einerseits die zunehmende Bedeutung des Engineering in der Softwareentwicklung und andererseits die Verkürzung der Grundlagenausbildung im Rahmen des Bolognaprozesses.

Mit diesem Buch wollen wir verschiedene Teilgebiete der Mathematik aus algorithmischer Perspektive vorstellen und dabei auch Implementierungs- und Laufzeitaspekte diskutieren. Gleichzeitig möchten wir, bei einer verkürzten Grundausbildung in Mathematik in naturwissenschaftlichen und informatischen Studiengängen, möglichst viele Teilaspekte der Mathematik vorstellen und vielleicht zu einer vertiefenden Beschäftigung mit dem einen oder anderen Aspekt anregen. Unser Ziel ist es dabei nicht, den Leser zu einem versierten Anwender der besprochenen Algorithmen auszubilden, sondern wir wollen, immer ausgehend von konkreten Problemen, Analyse- und Lösungsstrategien in den Mittelpunkt stellen. Hierbei spielen insbesondere Beweise und Beweistechniken eine zentrale Rolle.

Bevor wir uns konkreten algorithmischen Fragestellungen zuwenden, widmen wir uns der Kombinatorik und dem elementaren Abzählen. Hier kann man, ohne ausgefeiltes Theoriegebäude, sehr schön mathematische Argumentations- und Schlussweisen vorstellen und etwa darlegen, wie man längere Rechnungen durch geschickte Argumentation vermeiden kann. Gleichzeitig dient dieses Kapitel auch der Vorbereitung von Abschätzungen und Laufzeitanalysen.

In den nächsten beiden Kapiteln stellen wir einige algorithmische Probleme auf Graphen und Digraphen vor. Wir diskutieren Baumsuche, Valenzsequenzen, Eulertouren, minimale aufspannende Bäume, das Isomorphieproblem bei Bäumen, maximale bipartite Matchings und stabile Hochzeiten. Bei der Auswahl haben wir uns eher an der Breite der angesprochenen Themen als an der Relevanz der Aufgabenstellungen orientiert.

Mit dem folgenden Kapitel verlassen wir die diskrete Mathematik und wollen zunächst Problembewusstsein für die Schwierigkeiten beim Rechnen mit Fließkommazahlen wecken. Wir stellen beispielhaft Auslöschung und Fehlerfortpflanzung vor. Daneben diskutieren wir Grundalgorithmen der Linearen Algebra, wie LU-Zerlegung und Choleskyfaktorisierung aus numerischer Sicht.

Die Kapitel 6 und 7 sind der Nichtlinearen Optimierung als algorithmischer Anwendung der Analysis gewidmet. In dem ersten dieser beiden Kapitel diskutieren wir vor allem notwendige und hinreichende Bedingungen für Extremwerte. Einen Beweis für die Lagrangebedingungen oder der Kuhn-Tucker-Bedingungen müssen wir im Rahmen dieses Buches schuldig bleiben, da wir den dafür benötigten „Satz über implizit definierte Funktionen" nicht voraussetzen wollen. Statt dessen versuchen wir, die Aussagen anschaulich geometrisch plausibel zu machen. Die geometrische Sichtweise halten wir auch bei der Diskussion numerischer Verfahren zur Lösung von nichtlinearen Optimierungsproblemen in Kapitel 7 bei. Schlüsselwörter sind hier Abstiegsrichtung und Schrittweite.

Das abschließende Kapitel zur Linearen Optimierung haben wir hinten angestellt, da wir den Dualitätssatz der Linearen Optimierung aus den Kuhn-Tucker Bedingungen ableiten. Darüber hinaus diskutieren wir den Simplexalgorithmus aus geometrischer Sicht und wie sich die geometrischen Ideen effizient im Tableau umsetzen lassen.

Wir setzen an einigen Stellen Kenntnisse in Linearer Algebra und Analysis voraus, wie sie in einführenden Büchern und Veranstaltungen der Mathematik für Ingenieure und Natur- oder Wirtschaftswissenschaftler vermittelt werden.

Dieses Buch ist aus einem Fernstudienkurs der FernUniversität in Hagen hervorgegangen, der Teil der mathematischen Grundausbildung in den Bachelorstudiengängen Informatik und Wirtschaftsinformatik im zweiten Semester ist, und den wir auch in der Lehrerfortbildung einsetzen.

Hagen, im September 2009 *Winfried Hochstättler*

Danksagung

Bei der Wahl der Themen und des Titels dieses Buches habe ich mich von meinen akademischen Lehrern Achim Bachem und Jaroslav Nešetřil anregen lassen. Dem diskreten Teil in den Kapiteln 2–4 merkt man wahrscheinlich noch die Vorbildfunktion an, die das Buch „Diskrete Mathematik – Eine Entdeckungsreise“ von Matoušek und Nešetřil hier für mich hatte. Die Wahl der Themen der übrigen Kapitel lehnt sich an die Vorlesung “Algorithmische Mathematik – für Wirtschaftsinformatiker“ an, die ich als Assistent von Achim Bachem im Wintersemester 1991/92 betreuen durfte.

Darüber hinaus gilt mein Dank allen, die mir bei der Entwicklung dieses Manuskripts behilflich waren, meinem ehemaligen Mitarbeiter Robert Nickel, insbesondere für seine Beiträge zu weiterführenden Literaturhinweisen, meinen Mitarbeitern Dr. Manfred Schulte und Dr. Dominique Andres, Herrn Klaus Kuzyk, Frau Heidrun Krimmel sowie zahllosen aufmerksamen FernStudierenden der FernUniversität in Hagen.

Inhaltsverzeichnis

Kapitel 1
Notation und Grundstrukturen

1.1 Gliederung und Motivation

Schwerpunkt dieses Buches ist vor allem die Schulung des folgerichtigen und algorithmischen Denkens. Wenn Sie Informatiker oder Ökonom sind, werden Sie oft in „Wenn-Dann“-Situationen sein, ob Sie nun Produktionsabläufe oder die Struktur einer komplexen Hard- oder Softwareumgebung analysieren wollen. Diese Analysekompetenz trainiert man unseres Erachtens am besten, indem man abstrakte Strukturen analysiert, die von störendem Ballast befreit sind. Diese Art von Analysen ist natürlicherweise Bestandteil von mathematischen „Wenn-Dann“-Aussagen, wenn diese nämlich bewiesen werden. Folglich werden wir verstärkt Wert auf Beweise legen. Ein Beweis eines Satzes ist nichts anderes als eine folgerichtige, vollständige Schlusskette, mit der man aus der Gültigkeit einer Reihe von Voraussetzungen die Aussage des Satzes herleitet.

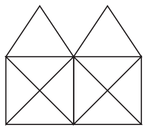
Wir werden in diesem Kapitel noch verschiedene Besonderheiten von Beweisen vorstellen. Dabei kommt dem Induktionsbeweis im Rahmen der Algorithmischen Mathematik eine besondere Bedeutung zu. Induktionsbeweise sind in der Regel konstruktiv und führen häufig zu Algorithmen, mit denen man zum Beispiel eine Struktur, deren Existenz die Induktion beweist, auch algorithmisch auffinden kann.

Nachdem wir in diesem Kapitel etwas Notation einführen, werden wir uns zunächst mit Zählproblemen beschäftigen. Dort werden Sie z. B. lernen, folgendes Problem zu lösen:

Problem 1.1. Wie groß ist die Chance, mit einem Lotto-Tipp fünf Richtige mit Zusatzzahl zu bekommen?

Im dritten Kapitel werden wir Graphen kennenlernen und das Haus vom Nikolaus ohne abzusetzen zeichnen. Ferner werden wir ein Kriterium kennenlernen, das es uns erlaubt, auch das Doppelhaus vom Nikolaus zu betrachten.

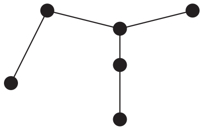
Problem 1.2. Kann man nebenstehende Figur ohne abzusetzen zeichnen?



DOI 10.1007/978-3-642-05422-8_1, © Springer-Verlag Berlin Heidelberg 2010

Im vierten Kapitel lernen wir Bäume kennen und lösen algorithmisch effizient folgendes Problem:

Problem 1.3. Gegeben sind n Stationen und Kosten für eine paarweise Verbindung von je zwei Stationen. Installiere möglichst kostengünstig Verbindungen so, dass jede Station von jeder anderen Station aus (evtl. über Zwischenstationen) erreichbar ist.



Bis hierhin konnten wir von allen Zahlenwerten annehmen, dass sie ganz oder zumindest rational sind. Im zweiten Teil des Buches untersuchen wir Probleme, bei denen dies nicht immer der Fall ist. Allerdings können wir nicht einmal theoretisch die Menge der reellen Zahlen im Computer darstellen. Die Speicherzellen im Computer sind nummeriert, also kann man nur abzählbare Mengen darstellen. Die Menge der reellen Zahlen ist aber nicht abzählbar. Hingegen kann man die ganzen und die rationalen Zahlen *abzählen*. Aber auch beim Rechnen mit rationalen Zahlen können wir im Allgemeinen nicht davon ausgehen, dass wir mit beliebiger *Genauigkeit* rechnen können, da wir nur mit endlichem Speicherplatz rechnen können. Dies führt zu *Rundungsfehlern*, die sich in Rechnungen *verstärken* und *fortpflanzen* können. Nach der Diskussion dieser allgemeinen Problematik diskutieren wir Verfahren zur Optimierung linearer und nicht-linearer Modelle.

1.2 Notation

Zunächst wiederholen wir Symbole aus der Mengenlehre, die aus der Schule bekannt sein sollten:

Wir bezeichnen mit

$\mathbb{N}$ die Menge der *natürlichen Zahlen*, die nach DIN-Norm 5473 die Null beinhaltet $\mathbb{N} := \{0,1,2,3,4,5,6,\ldots\}$.

$\mathbb{Z}$ die Menge der *ganzen Zahlen* $\mathbb{Z} := \{\ldots,-4,-3,-2,-1,0,1,2,3,4,\ldots\}$.

$\mathbb{Q}$ die Menge der *rationalen Zahlen* $\mathbb{Q} := \{\frac{p}{q} \mid p \in \mathbb{Z}, q \in \mathbb{N} \setminus \{0\}\}$.

$\mathbb{R}$ die Menge der *reellen Zahlen*, dies sind alle Zahlen, die sich als nicht notwendig abbrechende Dezimalbrüche darstellen lassen. Dazu gehören zusätzlich zu den rationalen Zahlen *irrationale, algebraische Zahlen* wie etwa $\sqrt{2}$, die Nullstelle von $x^2 - 2$ ist, aber auch *irrationale, transzendente Zahlen*, die nicht Nullstelle eines Polynoms mit rationalen Koeffizienten $\neq 0$ sind, wie etwa π. Anstatt eines Dezimalkommas, benutzen wir die internationale Schreibweise mit Dezimalpunkt.

Die meisten Operationen, die wir mit Zahlen durchführen, wie Summe, Produkt, Differenz, Quotient, Potenz etc. setzen wir als allgemein bekannt voraus. Ist x eine reelle Zahl, so bezeichnen wir mit

$\lfloor x \rfloor$ die nächstkleinere ganze Zahl, also etwa $\lfloor 1.99 \rfloor = 1$, $\lfloor 2.01 \rfloor = 2$, $\lfloor 2 \rfloor = 2$, $\lfloor -1.99 \rfloor = -2$ und mit

$\lceil x \rceil$ die nächstgrößere ganze Zahl, also etwa $\lceil 1.99 \rceil = 2$, $\lceil 2.01 \rceil = 3$, $\lceil 2 \rceil = 2$, $\lceil -1.99 \rceil = -1$.

Summen und Produkte mehrerer Elemente kürzen wir mit dem *Summationszeichen* Σ und dem *Produktzeichen* Π ab.

$$\sum_{i=1}^{n} a_i := a_1 + a_2 + \ldots + a_n, \quad \prod_{i=1}^{n} a_i := a_1 \cdot a_2 \cdot \ldots \cdot a_n.$$

Also zum Beispiel $\sum_{i=1}^{5} i^2 = 1+4+9+16+25 = 55$.

Die leere Summe setzen wir auf 0 und das leere Produkt auf 1, also z. B.

$$\sum_{i=1}^{0} 3 = 0, \quad \prod_{i=1}^{0} 3 = 1.$$

Allgemeine Mengen bezeichnen wir meist mit Großbuchstaben. Wenn x in M liegt, schreiben wir $x \in M$, ansonsten $x \notin M$. Falls M aus endlich vielen Elementen besteht, so bezeichnet $|M|$ die *Kardinalität* (oder *Mächtigkeit*) von M, also die Anzahl der Elemente, die in M liegen.

Sind M,N zwei Mengen, so ist M eine Teilmenge von N, in Zeichen $M \subseteq N$, wenn

$$\forall x : (x \in M \Rightarrow x \in N), \tag{1.1}$$

in Worten, „wenn x in M liegt, so liegt es auch in N“.

Wir haben dabei soeben den *Allquantor* $\forall$ benutzt, um zu betonen, dass die Bedingung stets erfüllt sein muss. Dieser Allquantor ist eine Abkürzung für „für alle“. Also bedeutet (1.1) wörtlich „Für alle x gilt: wenn x in M liegt, so liegt es auch in N.“ Daneben benutzen wir auch noch den *Existenzquantor* $\exists$, der bedeutet „Es gibt ein ...“. Dabei ist „Es gibt ein“ immer als „Es gibt mindestens ein ...“ zu verstehen.

Zwei Mengen M,N sind *gleich* ($M = N$), wenn $M \subseteq N$ und $N \subseteq M$ ist.

Vereinigung und *Schnitt* von Mengen sind definiert als

$$M \cup N := \{x \mid x \in M \text{ oder } x \in N\}, \qquad M \cap N := \{x \mid x \in M \text{ und } x \in N\}.$$

Man beachte, dass im Falle der Vereinigung das „oder“ nicht exklusiv ist, d. h. x darf auch in beiden Mengen liegen. $M \dot\cup N$ schreiben wir für die Vereinigung $M \cup N$ nur, wenn zusätzlich gilt, dass $M \cap N = \emptyset$. Wir sagen dann auch, M und N *partitionieren* $M \cup N$. Allgemeiner ist $M = A_1 \dot\cup A_2 \dot\cup \ldots \dot\cup A_k$ eine *Partition* von M, wenn für alle $i, j \in \{1, \ldots, k\}$ mit $i \neq j$ gilt: $A_i \cap A_j = \emptyset$. Die A_i bezeichnen wir dann als *Klassen von M*.

Die *Differenzmenge* $M \setminus N$ ist definiert als $M \setminus N := \{x \mid x \in M \text{ und } x \notin N\}$.

Betrachtet man Mengen bezüglich einer gegebenen Grundmenge X, die wir als *Universum* bezeichnen, so ist für $M \subseteq X$ das *Komplement* $\overline{M}$ von M definiert als $X \setminus M$.

Das *kartesische Produkt* zweier Mengen M und N, symbolisch $M \times N$, ist erklärt als die Menge der geordneten Paare (x,y) mit $x \in M$ und $y \in N$. Wir nennen ein solches Paar *Tupel*. Elemente von kartesischen Produkten von n Mengen, also $(x_1, \ldots, x_n)$ nennen wir auch *n-Tupel*.

Ist X eine Menge, so bezeichnen wir mit

$$2^X = \{Y \mid Y \subseteq X\}$$

die *Potenzmenge* von X, das ist die Menge aller Teilmengen von X.

1.3 Abbildungen

Eine Abbildung ordnet jedem Element aus einer *Urbildmenge* ein Element aus einer *Wertemenge* zu. Formal:

Eine *Abbildung* $f : M \to N$ aus einer Menge M in eine Menge N ist eine Menge von geordneten Paaren $(x,y) \in M \times N$ mit der Eigenschaft, dass es für jedes $x \in M$ genau ein Paar in dieser Menge gibt, das x als erste Komponente hat. Wir schreiben dann auch $x \mapsto y$.

Statt $(x,y) \in f$ schreiben wir üblicherweise $f(x) = y$. Ist $A \subseteq M$, so bezeichnen wir mit $f(A) := \{f(a) \mid a \in A\} \subseteq N$ die Menge aller Bilder von Elementen in A.

Sind $f : M \to N$ und $g : Y \to M$ Abbildungen, so definieren wir die *Komposition* oder *Hintereinanderausführung* $h := f \circ g$ der Abbildungen durch $h(x) := f(g(x))$.

Eine Abbildung $f : M \to N$ heißt

injektiv, wenn verschiedene Urbilder verschiedene Bilder haben, also

$$x \neq y \Rightarrow f(x) \neq f(y),$$

surjektiv, wenn jedes Element in der Wertemenge getroffen wird, also $f(M) = N$,
bijektiv, wenn sie injektiv und surjektiv ist.

Eine bijektive Abbildung $\sigma : M \to M$, bei der Urbildmenge und Wertemenge übereinstimmen, nennen wir auch eine *Permutation*.

Definition 1.1. Zwei Mengen A,B heißen *gleichmächtig*, wenn es eine bijektive Abbildung $f : A \to B$ gibt.

Mit diesen Begrifflichkeiten beweisen wir einige erste kleine Aussagen.

Proposition 1.1. *a) Die Hintereinanderausführung injektiver Abbildungen ist injektiv.*
b) Die Hintereinanderausführung surjektiver Abbildungen ist surjektiv.
c) Die Hintereinanderausführung bijektiver Abbildungen ist bijektiv.

Beweis.

a) Seien also f,g zwei injektive Abbildungen und die Wertemenge von g sei identisch mit dem Definitionsbereich (Urbildmenge) von f. Wir haben zu zeigen, dass $x \neq y \Rightarrow (f \circ g)(x) \neq (f \circ g)(y)$. Seien also $x \neq y$ zwei verschiedene Elemente aus dem Definitionsbereich von g. Da g injektiv ist, sind $g(x) \neq g(y)$ zwei verschiedene Elemente aus dem Definitionsbereich von f. Da f injektiv ist, folgt nun $(f \circ g)(x) = f(g(x)) \neq f(g(y)) = (f \circ g)(y)$.
b) Hier müssen wir zeigen, dass jedes Element aus der Wertemenge N von $f \circ g$ als Bild angenommen wird. Sei also x ein solches Element. Da f surjektiv ist, gibt es ein y aus dem Definitionsbereich von f mit $f(y) = x$, analog gibt es ein z mit $g(z) = y$. Also ist $(f \circ g)(z) = x$.
c) Dies folgt aus den beiden vorhergehenden Aussagen. □

Als Zeichen, dass der Beweis fertig ist, haben wir rechts ein offenes Quadrat gesetzt.

Bei einer bijektiven Abbildung $f : M \to N$ hat jedes Element in der Wertemenge ein Urbild, und dieses ist eindeutig. Also können wir die *Umkehrabbildung* $g : N \to M$ definieren durch $g(y) = x \iff f(x) = y$. Wir bezeichnen ein solches g auch mit f^{-1}.

Ist $f : M \to N$ eine Abbildung und $L \subseteq M$, so bezeichnen wir mit $f_{|L} : L \to N$, definiert durch $f_{|L}(x) := f(x)$, die *Einschränkung von f auf L*. Damit können wir auch zwei Abbildungen f, g verketten, wenn der Bildbereich von der ersten (g) nur eine Teilmenge des Definitionsbereichs der zweiten (f) Funktion ist, indem wir für $g : K \to L$ definieren:

$$f \circ g := f_{|L} \circ g.$$

Aufgabe 1.1. Bezeichne $2\mathbb{Z}$ die Menge der geraden ganzen Zahlen. Bestimmen Sie bei den folgenden Zuordnungsvorschriften f_i, ob es Abbildungen sind. Wenn ja, stellen Sie fest, ob diese injektiv, surjektiv oder bijektiv sind.

$$\begin{array}{lll} f_1 : \mathbb{Z} \to \mathbb{N} & f_2 : \mathbb{Z} \to \mathbb{Z} & f_3 : \mathbb{Z} \to 2\mathbb{Z} \\ \quad k \mapsto 2k & \quad k \mapsto 2k & \quad k \mapsto 2k \\ f_4 : \mathbb{Z} \to \mathbb{Z} & f_5 : 2\mathbb{Z} \to \mathbb{Z} & \\ \quad k \mapsto \frac{1}{2}k & \quad k \mapsto \frac{1}{2}k & \end{array}$$

Lösung siehe Lösung 9.1.

Aufgabe 1.2. Seien $f : M \to N$ und $g : L \to M$ Abbildungen. Zeigen Sie:

a) Ist $f \circ g$ surjektiv, so ist auch f surjektiv.
b) Ist $f \circ g$ injektiv, so ist auch g injektiv.
c) Geben Sie jeweils ein Beispiel für (f, g) an, bei dem $f \circ g$ surjektiv, aber g nicht surjektiv, bzw. $f \circ g$ injektiv, aber f nicht injektiv ist.

Lösung siehe Lösung 9.2.

1.4 Beweismethoden und das Prinzip der vollständigen Induktion

Wie Sie in den bisherigen Abschnitten bereits gesehen haben, besteht ein mathematischer Text zumeist aus Definitionen, Sätzen und Beweisen. Dabei ist eine *Definition* eine sprachliche Vereinbarung, die jeweils einer gewissen Struktur einen Namen gibt. Ein *Satz* besteht zumeist aus einigen Voraussetzungen und einer Behauptung. In dem zugehörigen *Beweis* wird schlüssig Schritt für Schritt dargelegt, warum unter Annahme der Gültigkeit der Voraussetzungen die Behauptung notwendig auch gelten muss. Jeder einzelne Schritt des Beweises muss logisch nachvollziehbar sein.

Neben solchen direkten Beweisen, wollen wir hier noch drei weitere Vorgehensweisen vorstellen. Zunächst den

1.4.1 Beweis durch Kontraposition

Betrachten wir hierzu den „Satz“:

Wer einkaufen geht und bar bezahlt, hat danach weniger Geld in der Brieftasche.

Diese Aussage hat die Form

$$(A \text{ und } B) \Rightarrow C.$$

Logisch gleichwertig ist die umgekehrte Implikation der Negationen, nämlich

$$\text{nicht } C \Rightarrow \text{ nicht } (A \text{ und } B),$$

wobei die rechte Seite der Implikation wiederum gleichwertig ist mit

$$(\text{nicht } A) \text{ oder } (\text{nicht } B).$$

Insgesamt können wir statt obiger Aussage also genau so gut zeigen:

Wer danach nicht weniger Geld in der Brieftasche hat, war nicht einkaufen oder hat nicht bar bezahlt.

Wir stellen einen solchen Beweis an einem geometrischen Beispiel vor. Ein *Geradenarrangement* ist eine endliche Menge von (paarweise verschiedenen) Geraden in der Ebene. Diese zerteilt die Ebene in (beschränkte und unbeschränkte) *Zellen*, das sind die zusammenhängenden Gebiete, die entstehen, wenn man alle Geraden entfernt. Die Schnittpunkte der Geraden nennen wir *Ecken* des Arrangements. Beachte, zwei Geraden schneiden sich in genau einer Ecke oder sie sind parallel.

Ein *Dreieck* in einem Geradenarrangement ist eine beschränkte Zelle, die von genau drei Geraden *berandet* wird.

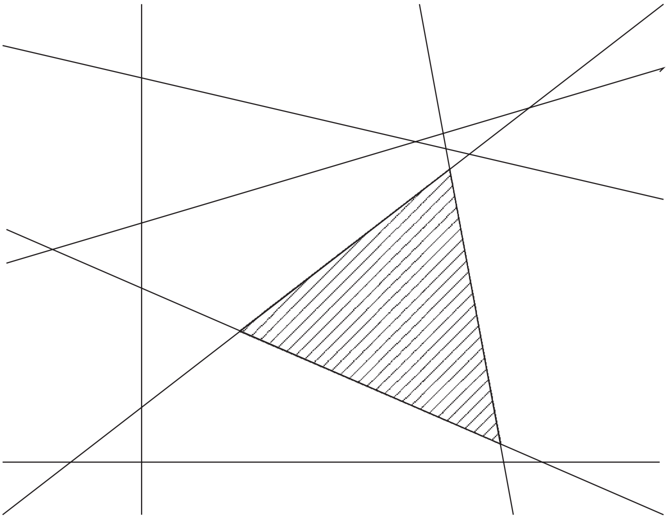


Abb. 1.1 Ein Geradenarrangement und eines seiner acht Dreiecke

Wir zeigen nun

Sei $\mathscr{G}$ ein Geradenarrangement, bei dem alle bis auf eine Gerade paarweise parallel sind oder es eine Ecke gibt, die auf allen Geraden liegt. Dann enthält $\mathscr{G}$ kein Dreieck.

Beweis. Man kann diese Aussage selbstverständlich direkt beweisen. Mit Kontraposition wird es aber einfacher. Die Kontraposition der Aussage ist: