

Vorwort

Die Fähigkeit, mathematische Beweise zu verstehen und insbesondere auch selbst zu führen, wird nicht nur in der Mathematik benötigt, sondern auch in der Informatik und in vielen anderen Disziplinen. Die Informatik untersucht zum Beispiel Datenmodelle, Berechnungsmodelle, mathematisch fundierte Programmierparadigmen wie die funktionale und die Logikprogrammierung, kryptographische Verfahren, statistische Ansätze für *Data mining* oder für Optimierungen (etwa von Lastverteilungen), Grundlagen von Suchmaschinen wie *PageRank*, unzählige verschiedenartige Algorithmen und Programme und viele andere Strukturen mehr, deren Analyse oft mathematische Methoden und Beweise erfordert.

Leider sehen Studiengänge der Informatik und wohl auch anderer Wissenschaften in der Regel nicht vor, dass mathematisches Beweisen explizit gelehrt wird. Beweise werden zwar in Vorlesungen präsentiert und zum Teil in Übungen anhand von konkreten Beispielen vertieft, aber wie ein Beweis überhaupt aufgebaut sein kann, und in welchen Fällen welche Beweismuster in Frage kommen, können Studierende höchstens indirekt anhand der Beweise erschließen, die zufällig bisher in ihrem Studium vorgekommen sind. Daher fällt es vielen Studierenden so schwer, selbständig Beweise auszuarbeiten, die nicht Varianten von vorgeführten Beispielen sind.

Um dem abzuhelpen, versucht der vorliegende Leitfaden, verbreitete Beweismuster systematisch vorzustellen und anhand von allgemein verständlichen Beispielen aus dem Alltag, der Mathematik und der Informatik zu verdeutlichen. Er beschränkt sich dabei bewusst auf *universelle* Muster, die unabhängig vom jeweiligen Teilgebiet anwendbar sind, behandelt aber keine teilgebietsspezifischen Beweismuster wie spezielle Muster für Stetigkeitsbeweise.

Ein mathematischer Beweis erfordert natürlich zunächst eine hinreichende Einsicht in das jeweilige Problem, um inhaltliche Ideen für den Beweis entwickeln zu können. Diese bekommt man nur, wenn man sich intensiv mit dem Problem selbst befasst. Für die Umsetzung der Ideen, also Aufbau und Strukturierung des Beweises, gibt es dagegen eine Reihe von allgemeinen Mustern, die sich bewährt haben, und an die man sich halten kann (und auch sollte). Diese Muster zu beherrschen ist zum Einen notwendig für das Verständnis fremder Beweise. Wenn darin zum Beispiel die Floskel „durch Kontraposition“ vorkommt, sollte man wissen, was das heißt. Zum Anderen sind die Muster enorm hilfreich für die Entwicklung eigener Beweise. Denn eine saubere Strukturierung ist nicht nur beim Lesen wichtig, sondern hilft auch, beim Ausarbeiten der Beweise die Übersicht zu behalten und nichts zu vergessen.

Bewährte Muster, an die man sich halten kann, gibt es auch in der Programmierung. Sie sind dort unter dem Begriff *Design Patterns* bekannt. Design Patterns helfen, Programme zu strukturieren und für andere Personen verständlich zu machen. Das brachte die Autoren dieses Leitfadens auf den Gedanken, gängige Muster für mathematische Beweise zusammenzustellen und Studierenden der Informatik in einem kurzen Skript mit dem Titel *Design Patterns für mathematische Beweise* als „Leitfaden“ an die Hand zu geben.

Wie alle wissenschaftlichen Arbeiten entwickelte sich der Leitfaden im Verlauf seiner Entstehung. Das „kurze Skript“ wuchs und gedieh, ja wucherte geradezu. Irgendwann sprachen praktische, aber auch inhaltliche Gründe dafür, den Leitfaden in zwei Teile aufzuteilen.

Teil I des Leitfadens behandelt einfache Beweismuster wie Fallunterscheidung, Allbeweis, Implikationsbeweis, komplexere Beweismuster wie Kontraposition, Widerspruchsbeweis, Diagonalisierung, und endet mit einem umfangreichen Kapitel über die verschiedenen Varianten der vollständigen Induktion. Gemeinsam ist diesen Varianten, dass sie sich auf jeweils unendlich viele Objekte beziehen, die zwar größer sein können als jede endliche Schranke, aber nicht unendlich groß. Die bekanntesten derartigen Objekte sind die natürlichen Zahlen.

Teil II gibt einen Einblick in die Welt jenseits der Unendlichkeit, von der die natürlichen Zahlen begrenzt werden. In dieser Welt können Objekte auch unendlich groß werden und trotzdem noch eine Form der vollständigen Induktion ermöglichen, die transfinite Induktion. Ein Kapitel im Teil II des Leitfadens stellt die Ordinalzahlen vor. Es enthält auch ein Beispiel, in dem transfinite Ordinalzahlen für den Terminierungsbeweis eines Algorithmus unabdingbar sind. Dieses Beispiel animiert vielleicht einige Leser, ähnliche Probleme auf diese Weise anzugehen. Ein anderes Kapitel behandelt das Beweismuster der transfiniten Induktion und illustriert es mit Beispielen unterschiedlicher Komplexität.

Der Leitfaden ist als studienbegleitende Lektüre für Studierende insbesondere in Informatik-Studiengängen gedacht. Daher orientieren sich sowohl seine Gliederung als auch der Umfang der jeweils mitgelieferten Hintergrundinformation in groben Zügen am durchschnittlichen Kenntnisstand im Verlauf eines Studiums. Für viele Leser wird es somit kaum sinnvoll sein, den gesamten Leitfaden ein Mal komplett am Stück durchzuarbeiten. Er kann auch wie ein Nachschlagewerk benutzt werden, welches man bei Bedarf zu Einzelthemen konsultiert.

Im Teil I sollten alle einfachen Beweismuster, mindestens die erste Hälfte der komplexen Beweismuster sowie der erste Abschnitt zur vollständigen Induktion bereits in den ersten Fachsemestern nachvollziehbar und nützlich sein. Andere Themen richten sich dagegen eher an Leser in höheren Fachsemestern, weil der jeweilige Stoff erstens mehr Hintergrundwissen voraussetzt und zweitens auch erst in fortgeschritteneren Lehrveranstaltungen relevant wird. Dazu gehören insbesondere die vertiefenden Abschnitte, die als „Exkurs“ gekennzeichnet sind, sowie der gesamte Teil II.

Als Ergänzung zu diesem Leitfaden sind einige Bücher empfehlenswert, die ähnliche Ziele, aber andere Schwerpunkte haben. George Pólyas Klassiker *„Schule des Denkens. Vom Lösen mathematischer Probleme“* [Pól95] gliedert den Lösungsprozess mathematischer (und anderer) Probleme in vier Phasen: Verstehen der Aufgabe, Ausdenken eines Plans, Ausführen des Plans und Rückschau/Überprüfung/Vertiefung. Von Daniel Grieser stammt *„Mathematisches Problemlösen und Beweisen: Eine Entdeckungsreise in die Mathematik“* [Gri13]. Das Buch entwickelt einen „Werkzeugkasten“ mathematischer Methoden, die helfen können, auf die oben angesprochenen inhaltlichen Ideen zu kommen, die man für einen Beweis braucht. Das Bändchen *„Das ist o.B.d.A. trivial!“* [Beu09] von Albrecht Beutelspacher gibt viele Tipps zu typischen mathematischen Formulierungen und ihrer Bedeutung.

Der vorliegende Leitfaden konzentriert sich dagegen vor allem auf die *Struktur* von Beweisen. Dabei strebt er an, abstrakte Beweismuster in möglichst vielen Fällen durch Beispiele aus dem Alltag zu verdeutlichen. Dieser Aspekt könnte den Leitfaden auch für Dozenten interessant machen, denn solche Beispiele sind nicht immer so naheliegend wie die „Flüsterpost“ oder ein Menschenturm von Akrobaten zur Veranschaulichung der vollständigen Induktion.

Illustrationen wie (im Teil I) ein unfehlbarer Personalmanager mit Assistent für das Diagonalargument in Turings Halteproblem-Beweis oder (im Teil II) ein Aufzug mit transfiniten Stockwerksnummerierung für die Wohlfundiertheit der $<$ -Beziehung von Ordinalzahlen sind weniger offensichtlich. Dozenten können gerne die Beispiele aus diesem Leitfaden für ihren Unterricht verwenden.

Der Leitfaden wurde von Informatikern in erster Linie für Informatiker geschrieben. Der allergrößte Teil des Inhalts ist allerdings so allgemein, dass er auch für andere Disziplinen interessant sein dürfte, in denen mathematische Beweise erforderlich sind.

Die Autoren hoffen, mit diesem Leitfaden einem verbreiteten Bedarf der Studierenden entgegenzukommen und sind dankbar für Hinweise auf Fehler und Verbesserungsmöglichkeiten.

München, Juni 2016

Hans Jürgen Ohlbach, Norbert Eisinger

Design Patterns für mathematische Beweise

Ein Leitfaden insbesondere für Informatiker

Ohlbach, H.J.; Eisinger, N.

2017, XI, 184 S. 2 Abb., Softcover

ISBN: 978-3-662-55651-1