Beweisen ist Programmieren – und umgekehrt

Helmut Schwichtenberg

Mathematisches Institut, Universität München

1. Minimallogik

Hermann Weyl (Über die neue Grundlagenkrise der Mathematik, Mathematische Zeitschrift 1921):

Ein Existentialsatz – etwa „es gibt eine gerade Zahl“ – ist überhaupt kein Urteil im eigentlichen Sinne, das einen Sachverhalt behauptet; Existentialsachverhalte sind eine leere Erfindung der Logiker.

„2 ist eine gerade Zahl“, das ist ein wirkliches, einem Sachverhalt Ausdruck gebendes Urteil; „es gibt eine gerade Zahl“ ist nur ein aus diesem Urteil gewonnenes Urteilsabstrakt.

Weyl 1921 (Forts.)

Bezeichne ich Erkenntnis als einen wertvollen Schatz, so ist das Urteilsabstrakt ein Papier, welches das Vorhandensein eines Schatzes anzeigt, ohne jedoch zu verraten, an welchem Ort. Sein einziger Wert kann darin liegen, daß es mich antreibt, nach dem Schatze zu suchen.

Minimallogik

Formale Sprache: aus atomaren Aussagen (zum Beispiel Gleichungen, Ungleichungen) mit

$\wedge$	und
$\rightarrow$	wenn – so
$\forall$	für alle
$\exists$	es gibt

Gentzens Kalkül des natürlichen Schließens, 1934

$u : A$ Annahme

$$\frac{\begin{array}{cc}\mathcal{D}_0 & \mathcal{D}_1 \\ A & B\end{array}}{A \wedge B}\wedge^+ \qquad \frac{\mathcal{D}}{A \wedge B}\wedge_0^- \qquad \frac{\mathcal{D}}{A \wedge B}\wedge_1^-$$

$$\frac{\begin{array}{c}[u : A] \\ \mathcal{D} \\ B\end{array}}{A \to B}\to^+ u \qquad \frac{\begin{array}{cc}\mathcal{D}_0 & \mathcal{D}_1 \\ A \to B & A\end{array}}{B}\to^-$$

$$\frac{\begin{array}{c}\mathcal{D} \\ A\end{array}}{\forall x A}\forall^+ \text{ (mit Variablenbed.)} \qquad \frac{\begin{array}{cc}\mathcal{D} & \\ \forall x A & t\end{array}}{A[x := t]}\forall^-$$

Beispiel

$$\dfrac{\dfrac{u\colon A \to (B \to C) \qquad \dfrac{v\colon A \wedge B}{A}\,\wedge_0^- \qquad \dfrac{v\colon A \wedge B}{B}\,\wedge_1^-}{B \to C}\,\to^- \qquad \dfrac{}{B}}{C}\,\to^-$$

$$\dfrac{\dfrac{}{A \wedge B \to C}\,\to^+ v}{(A \to (B \to C)) \to (A \wedge B \to C)}\,\to^+ u$$

Minimallogik (Johansson 1937): $\Gamma \vdash A$ „Γ beweist A“ ($\Gamma = \{ A_1, \ldots, A_n \}$ Menge von Annahmen).

Intuitionistische Logik (Heyting 1930): spezielles Aussagensymbol $\bot$ „falsum“. Negation definiert: $\neg A := A \to \bot$. Zusätzliche Annahmen: Ex-Falso-Quodlibet-Formeln

$$\forall \vec{x}. \bot \to R(\vec{x}) \qquad (\mathsf{Efq}_R)$$

Klassische Logik: Mit Prinzip des indirekten Beweisens

$$\forall \vec{x}. \neg\neg R(\vec{x}) \to R(\vec{x}) \qquad (\mathsf{Stab}_R)$$

Minimallogik, intuitionistische und klassische Logik

$$\Gamma \vdash_i A \quad :\Longleftrightarrow \quad \Gamma \cup \mathsf{Efq} \vdash A,$$
$$\Gamma \vdash_c A \quad :\Longleftrightarrow \quad \Gamma \cup \mathsf{Stab} \vdash A.$$

Lemma.

$$\Gamma \vdash A \Rightarrow \Gamma \vdash_i A \Rightarrow \Gamma \vdash_c A.$$

Die Umkehrungen gelten nicht. Beispiel: die Peirce-Formel $((P \to Q) \to P) \to P$ ist klassisch, aber nicht intuitionistisch herleitbar.

Einbettung in die Minimallogik

Negative Übersetzung ^{k} nach Kolmogorov 1925:

$$
\begin{aligned}
R(\vec{r})^k &:= \neg\neg R(\vec{r}) \quad \text{für } R \neq \bot, \\
\bot^k &:= \bot, \\
(A \wedge B)^k &:= A^k \wedge B^k, \\
(A \to B)^k &:= A^k \to B^k, \\
(\forall x A)^k &:= \forall x A^k.
\end{aligned}
$$

Lemma. a. $\vdash_c A \leftrightarrow A^k$,

b. $\Gamma \vdash_c A \iff \Gamma^k \vdash A^k$, wobei $\Gamma^k := \{ B^k \mid B \in \Gamma \}$.

Der starke Existenzquantor $\exists$

In der klassischen Logik fehlt $\exists$: verwendet wird nur der schwache Existenzquantor $\exists^{\mathsf{cl}}$, der definiert ist durch

$$\exists^{\mathsf{cl}} x A := \neg\forall x \neg A.$$

Beispiel: „es gibt eine gerade Zahl“ meint „die Annahme, alle Zahlen seien ungerade, führt auf einen Widerspruch“.

In diesem Sinn ist die klassische Logik echt enthalten in der Minimallogik.

2. Konstruktive Mathematik

Weyl, Bishop: ohne Formalisierung.

Bishop/Bridges: Constructive Analysis, Springer 1985.

Feferman, Beeson, Troelstra:

formale Systeme der konstruktiven Mathematik.

Kreisel: Was weiß man mehr, wenn ein Satz konstruktiv bewiesen ist, als wenn man nur weiß, daß er wahr ist?

Beispiel eines nicht-konstruktiven Beweises

Es gibt irrationale Zahlen a, b mit a^b rational.

Beweis durch Fallunterscheidung.

Fall 1: $\sqrt{2}^{\sqrt{2}}$ ist rational. Wähle $a = \sqrt{2}$ und $b = \sqrt{2}$.
Dann sind a, b irrational, und nach Annahme ist a^b rational.

Fall 2: $\sqrt{2}^{\sqrt{2}}$ ist irrational. Wähle $a = \sqrt{2}^{\sqrt{2}}$ und $b = \sqrt{2}$.
Dann sind nach Annahme a, b irrational, und

$$a^b = \left(\sqrt{2}^{\sqrt{2}}\right)^{\sqrt{2}} = \left(\sqrt{2}\right)^2 = 2 \quad \text{ist rational.}$$

Fundamentalsatz der Algebra

Reelle Zahlen: gegeben durch Cauchy-Folge und Modul.

Komplexe Zahlen: Paare reeller Zahlen.

Fundamentalsatz. Jedes nicht konstante Polynom mit komplexen Koeffizienten hat eine Nullstelle.

Beweis. Martin Kneser (Ergänzung zu einer Arbeit von Hellmuth Kneser über den Fundamentalsatz der Algebra, Mathematische Zeitschrift 1981), enthält Algorithmus.

3. Programme aus konstruktiven Beweisen

Unentscheidbar, ob Programm seine Spezifikation erfüllt.

Formaler Beweis: Korrektheit leicht zu prüfen.

Beweis = Programm mit hinreichend vielen Kommentaren (genauer: Programm extrahierbar).

Vision: Verwende mathematische Kultur zum Organisieren komplexer Strukturen, zwecks Programmextraktion

(Zum Beispiel: Steuerprogramme im Anlagenbau oder in der Telekommunikation).

Einwände

1. Eine Algorithmus-Idee ist schon vor einem konstruktiven Beweis vorhanden.

(Richtig, aber: Programmentwicklung wird möglich).

2. Komplexität des extrahierten Programms.

(Man erhält in polynomialer Zeit berechenbare Funktionen, bei geeigneter Einschränkung der Beweismittel).

3. Klassische Beweise verwendbar?

(Ja, aber man muß genauer hinsehen).

Beispiel Analysis: Nullstellen

Verwendet: exakte reelle Zahlen.

Zwischenwertsatz für monotone Funktionen. Seien $a, b \in \mathbb{Q}$, $f : [a, b] \to \mathbb{R}$ stetig, $a < b$, und es gelte $f(a) < 0 < f(b)$. Ferner sei f streng monoton, d.h.

$$\forall x, y \in [a, b].\ x < y \to f(x) < f(y).$$

Dann findet man eine Nullstelle c von f in $[a, b]$.

Beweis

Reelle Zahl := Cauchyfolge von rationalen Zahlen.

Hilfsmittel: Vergleich einer reellen Zahl mit einem Intervall.

Gegeben reelle Zahlen $x < y$.

Dann kann man für jede reelle Zahl z entscheiden, ob

$$z < y \text{ oder } x < z.$$

Grund: Wegen $x < y$ kennt man ein $\varepsilon \in \mathbb{Q}$ mit $0 < \varepsilon < y - x$.

Approximiere x, y, z hinreichend genau (bis auf $\frac{1}{3}\varepsilon$) durch rationale Zahlen.

Beweis des Zwischenwertsatzes

Wir konstruieren Folgen $(c_m)_{m\in\mathbb{N}}$, $(d_m)_{m\in\mathbb{N}}$ in $\mathbb{Q}$ so, daß

$$a = c_0 \le c_1 \le \cdots \le c_m < d_m \le \cdots \le d_1 \le d_0 = b \qquad (1)$$

$$f(c_m) < 0 < f(d_m) \qquad (2)$$

$$d_m - c_m = \left(\frac{2}{3}\right)^m (b-a). \qquad (3)$$

Seien $c_0, \ldots, c_m, d_0, \ldots, d_m$ bereits konstruiert, so daß (1)–(3) gelten. Sei $x := c_m + \frac{1}{3}(d_m - c_m)$, $y := c_m + \frac{2}{3}(d_m - c_m)$. Nach Voraussetzung ist $f(x) < f(y)$.

Fall 1: $0 < f(y)$. Setze $c_{m+1} := c_m$, $d_{m+1} := y$.

Fall 2: $f(x) < 0$. Setze $c_{m+1} := x$, $d_{m+1} := d_m$.

Offenbar gelten dann wieder (1)–(3).

Beispiel: Das Maximalsegmentproblem

Ziel: Programmentwicklung durch Beweistransformation.

Gegeben: Folge $x_0, x_1, \ldots, x_n$ von ganzen Zahlen x_i.

Gesucht: maximales Segment $\underbrace{x_i + \cdots + x_k}_{\mathsf{S}(i,k)}$ mit $i, k \leq n$.

Beispiel: x_i = Änderung des Aktienkurses am Tag i. Dann ist $x_i + \cdots + x_k$ der Gewinn bei Kauf am Tag i und Verkauf am Tag $k+1$.

Das Maximalsegmentproblem (Forts.)

Spezifikation. Für alle n gilt (mit $i, k, i', k' \leq n$)

$$\exists i, k \forall i', k' \; \mathsf{S}(i', k') \leq \mathsf{S}(i, k).$$

Erweiterte Spezifikation. Für alle n gilt $(i, j, k, i', j', k' \leq n)$

$$\exists i, k \forall i', k' \; \mathsf{S}(i', k') \leq \mathsf{S}(i, k) \tag{4}$$

$$\exists j \forall j' \; \mathsf{S}(j', n) \leq \mathsf{S}(j, n) \tag{5}$$

Beweis der erweiterten Spezifikation

Induktion über n. Basis $n = 0$. Wähle $i = k = 0$ und $j = 0$.

Schritt $n \mapsto n + 1$. Nach IH: i_n, k_n sowie j_n.

Wir wollen j_{n+1} konstruieren, haben also zu zeigen

$$\exists j{\leq}n{+}1 \forall j'{\leq}n{+}1\ \mathsf{S}(j', n+1) \leq \mathsf{S}(j, n+1). \qquad (6)$$

Da j_n nicht weiterhilft, verallgemeinern wir die Behauptung (3) so, daß sie durch Nebeninduktion beweisbar wird:

$$\forall m{\leq}n{+}1 \exists l{\leq}m \forall l'{\leq}m\ \mathsf{S}(l', n+1) \leq \mathsf{S}(l, n+1). \qquad (7)$$

Annahme: S ist monoton

$$x_i + \cdots + x_k \leq x_j + \cdots + x_k$$
$$\rightarrow \quad x_i + \cdots + x_k + x_{k+1} \leq x_k + \cdots + x_k + x_{k+1}$$

Deshalb betrachten wir den Spezialfall

$$\forall i, j, k.\ \mathsf{S}(i,k) \leq \mathsf{S}(j,k) \rightarrow \mathsf{S}(i,k+1) \leq \mathsf{S}(j,k+1). \qquad (8)$$

Wegen (8): Existenz (6) eines maximalen Endsegments direkter beweisbar, und zwar

- ohne (7) (also auch ohne Nebeninduktion), aber
- mit Hilfe der IH – also j_n – und der Fallunterscheidung $\mathsf{S}(j_n, n+1) < \mathsf{S}(n+1, n+1)$ bzw. $\geq$.

Algorithmus zum ersten Existenzbeweis

$n = 0$: $\quad i_0 = k_0 = j_0 = 0$

$n{+}1$: $\quad$ Gegeben: i_n, k_n, j_n

$$j_{n+1} := l_{n+1} \text{ mit}$$

$$\begin{cases} l_0 := 0 \\ l_{m+1} := \begin{cases} m{+}1 & \text{falls } \mathsf{S}(l_m, n{+}1) < \mathsf{S}(m{+}1, n{+}1) \\ l_m & \text{sonst} \end{cases} \end{cases}$$

$$(i_{n+1}, k_{n+1}) := \begin{cases} (i_n, k_n) & \text{f. } \mathsf{S}(j_{n+1}, n{+}1) < \mathsf{S}(i_n, k_n) \\ (j_{n+1}, n{+}1) & \text{sonst} \end{cases}$$

Algorithmus zum vereinfachten Existenzbeweis

$n = 0$: $\quad i_0 = k_0 = j_0 = 0$

$n + 1$: $\quad$ Gegeben: i_n, k_n, j_n

$$j_{n+1} := \begin{cases} n+1 & \text{falls } \mathrm{S}(j_n, n+1) < \mathrm{S}(n+1, n+1) \\ j_n & \text{sonst} \end{cases}$$

(i_{n+1}, k_{n+1}) wie eben

Linearer (statt quadratischer) Algorithmus.

Das vom verbesserten Algorithmus gelieferte maximale Segment kann verschieden sein vom durch den ursprünglichen Algorithmus gefundenen.

4. Programme aus klassischen Beweisen

Zu jedem Beweis von $\forall x \exists^{\mathsf{cl}} y A(x,y)$ mit $A(x,y)$ quantorenfrei findet man einen Beweis von $\forall x \exists y A(x,y)$.

Eine Methode: Friedmans A-Übersetzung.

Aus einem Beweis von $\forall x \exists y A(x,y)$ kann man ein Programm zur Berechnung eines y in Abhängigkeit von x extrahieren (Realisierbarkeit).

Kreisels Gegenbeispiel

Ein Beweis von $\forall x \exists^{\mathsf{cl}} y B(x,y)$ liefert i.a. kein Programm.

$T(x,z)$: „z ist ein terminierender Lauf des Automaten x.“

Halteproblem: $\exists^{\mathsf{cl}} z T(x,z)$ ist unentscheidbar.

$$\vdash \forall x. \exists^{\mathsf{cl}} z T(x,z) \to \exists^{\mathsf{cl}} y T(x,y)$$

$$\vdash_c \forall x \exists^{\mathsf{cl}} y \forall z. T(x,z) \to T(x,y).$$

Aber: es gibt kein berechenbares f mit

$$\forall x, z. T(x,z) \to T(x, f(x)),$$

denn dann wäre $\exists^{\mathsf{cl}} z\, T(x,z)$ entscheidbar:

es wäre wahr genau dann, wenn $T(x, f(x))$ gilt

Beispiel

Gegeben Zahlenfolgen $f, g\colon \mathbb{N} \to \mathbb{N}$. Gesucht Indizes $i < j$ so daß $f(i) \leq f(j)$ und $g(i) \leq g(j)$.

f: 1 2 1 4 3 2 1 8 7 6 5 4 3 2 1 16 ...

g: 4 3 3 2 2 2 2 1 1 1 1 1 1 1 1 0 ...

Beweis mit Minimumprinzip:

$$\exists^{\mathsf{cl}} x Q(x) \to \exists^{\mathsf{cl}} x.Q(x) \wedge \forall y.y < x \to \neg Q(y),$$

also klassisch: kein offensichtliches Berechnungsverfahren (etwa wenn $Q(x)$ unentscheidbar ist).

Allgemeiner: Dicksons Lemma

Für alle k, l

$$\forall f_1, \ldots, f_k \exists^{\mathsf{cl}} i_0, \ldots, i_l \mathop{\mathbb{M}}_{n<l} . i_n < i_{n+1} \wedge \mathop{\mathbb{M}}_{m=1}^{k} f_m(i_n) \leq f_m(i_{n+1}).$$

Beweis aus dem Minimumprinzip bzgl. einer Maßfunktion.

Dicksons Lemma: Extrahiertes Programm

$\varphi(0)$, mit

$$\varphi(i) = \psi(i, \varphi)$$

$$\psi(i, h) = \xi_{i,h}(i+1)$$

$$\xi_{i,h}(j) = \begin{cases} \psi(j, \xi_{i,h}) & \text{falls } g(j) < g(i) \\ h(j) & \text{falls nicht, aber } f(j) < f(i) \\ (i, j) & \text{sonst} \end{cases}$$

Unerwartetes funktionales Programm; es verwendet ein „allgemeines“ Rekursionsprinzip.

Konklusion

- Beweise sind der Kern der Mathematik. Dadurch unterscheidet sie sich von allen anderen Wissenschaften.

- Formalisierung der Logik ist einfach, aber auch wesentlich für Studien, in denen Beweise der Gegenstand sind (z.B. Programmextraktion).

- Formale Beweise kann man auch als eine Programmierumgebung ansehen, in der die Korrektheit von Programmen einsehbar und maschinell überprüfbar ist.