Mathematische Logik

Teil 3: Wo Beweise und Algorithmen an ihre Grenzen stoßen

Prof. Dr. Anton Freund

Eine zugehörige Einführungsvorlesung (Teil 1) und ein Übungsblatt (Teil 2) sind verfügbar auf https://www.mathematik.uni-wuerzburg.de/mathematicallogic/lehre/material/.

Ada Lovelace, frühes Computerprogramm, 1843

Kurt Gödel, Unvollständigkeitssätze, 1931

Alan Turing, Halteproblem, 1938

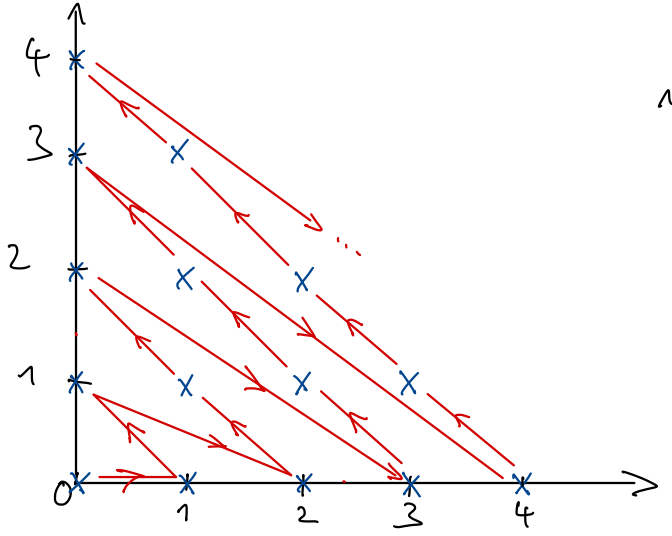
Julia Robinson, zentrale Beiträge zur Berechenbarkeitstheorie, 1940er bis 80er

Bildquellen: siehe letzte Folie

Unser erstes Ziel:

Satz: Es gibt Funktionen von $\mathbb{N}$ nach $\mathbb{N}$, die nicht durch ein Computerprogramm berechnet werden können.

Schritt 1: Man kann eine Aufzählung $(m_0,n_0),(m_1,n_1),\ldots$ finden, in der jedes Paar von natürlichen Zahlen genau einmal vorkommt (Cantorsche Paarfunktion).



$\rightsquigarrow$ $(0,0),(1,0),(0,1),(2,0),(1,1),(0,2),$
$(3,0),(2,1),(1,2),(0,3),$
$(4,0),(3,1),(2,2),(1,3),(0,4),\ldots$

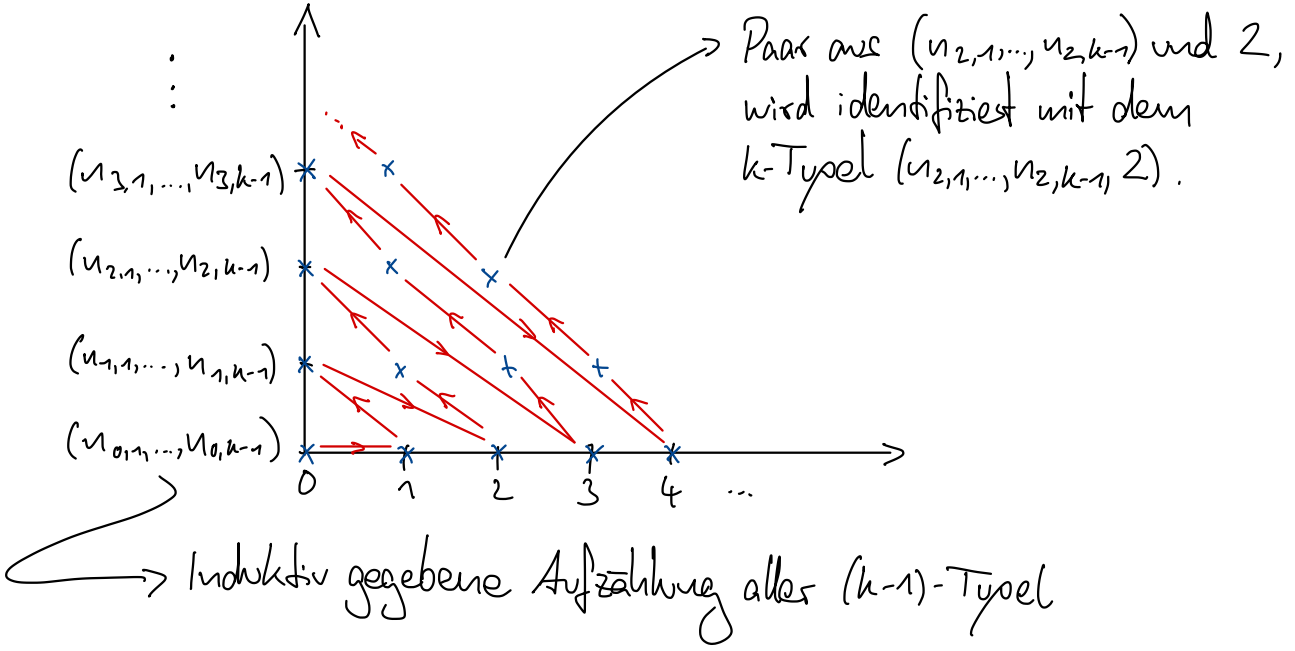
Das Paar (m,n) erscheint an Position

$$1+\ldots+(m+n)+n = \frac{(m+n)(m+n+1)}{2}+n.$$

Schritt 2: Für jedes $k \geq 1$ kann man eine Aufzählung

$$(u_{0,1}, \ldots, u_{0,k}), (u_{1,1}, \ldots, u_{1,k}), (u_{2,1}, \ldots, u_{2,k}), \ldots$$

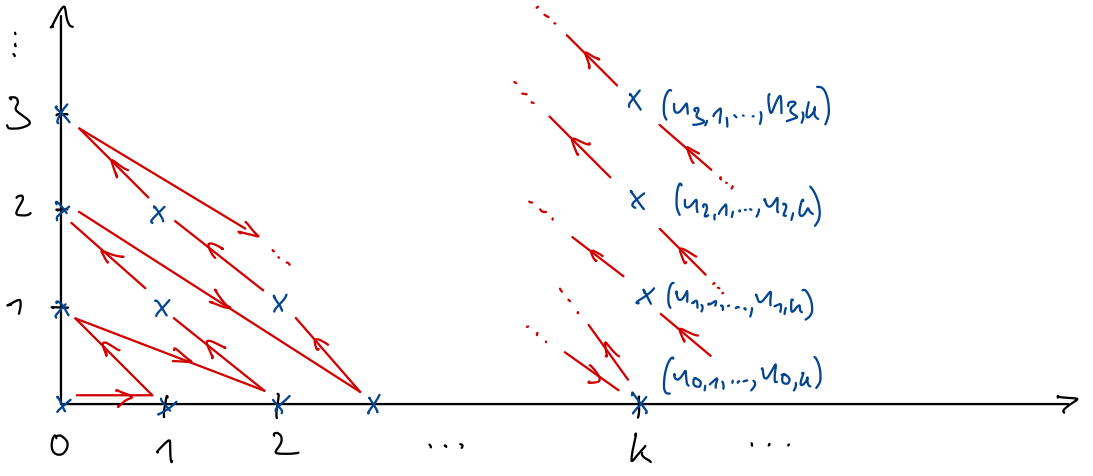
finden, in der jedes k-Tupel $(u_1, \ldots, u_k)$ genau einmal vorkommt.



Schritt 3: Es gibt eine Aufzählung

$$(u_{0,1}, \dots, u_{0,k(0)}), (u_{0,1}, \dots, u_{1,k(1)}), \dots)$$

in der jede endliche Folge beliebiger Länge genau einmal vorkommt.



→ Gemäß Schritt 2 können wir in der k-ten Spalte alle Folgen der Länge k auflisten.

Schritt 4: Es gibt eine Liste $P_0, P_1, \ldots$, in der jedes Computerprogramm genau einmal vorkommt.

Beweis: Computerprogramme sind endliche Folgen von Buchstaben aus einem endlichen Alphabet. Indem wir jedem Buchstaben eine Zahl zuordnen, können wir sie als endliche Folgen in $\mathbb{N}$ auffassen. Aus Schritt 3 haben wir eine Auflistung aller solcher Folgen. Wir lassen nun einfach die Folgen weg, die kein Programm repräsentieren. □

Computerprogramme können in unendliche Schleifen geraten. Wir schreiben $P_i(n)\downarrow$, wenn das *i*-te Programm mit Eingabe *n* nach endlich vielen Schritten anhält und eine natürliche Zahl als Ergebnis ausgibt. Diese Zahl bezeichnen wir dann mit $P_i(n)$.

Satz: Es gibt eine Funktion $f: \mathbb{N} \to \mathbb{N}$, die nicht durch ein Computerprogramm berechnet werden kann.

Beweis: Wir listen alle Computerprogramme $P_0, P_1, \ldots$ auf, die Funktionen von $\mathbb{N}$ nach $\mathbb{N}$ berechnen, für die also $P_i(n)\downarrow$ für alle $n \in \mathbb{N}$ gilt. Betrachte

$$f: \mathbb{N} \to \mathbb{N} \quad \text{mit} \quad f(n) = P_n(n) + 1.$$

Für einen Widerspruchsbeweis nehmen wir an, dass *f* durch ein Computerprogramm berechnet werden kann. Dann gibt es also ein *i* mit $f(n) = P_i(n)$ für alle $n \in \mathbb{N}$. Wir erhalten

$$P_i(i) = f(i) = P_i(i) + 1,$$

was der gewünschte Widerspruch ist. □

Fakt: Die Funktion $(i,n) \mapsto P_i(n)$ ist selbst berechenbar durch ein Computerprogramm, welches für die Eingabe *(i,n)* genau dann in endlicher Zeit anhält, wenn $P_i(n)\downarrow$ gilt (~ *code as data*).

Einwand: Es ist dann auch die Funktion *f* mit $f(n) = P_n(n) + 1$ berechenbar und also gleich P_i für ein *i*. Ergibt sich wegen

$$P_i(i) = f(i) = P_i(i) + 1$$

nicht ein Widerspruch?

Auflösung: Es muss so sein, dass das Programm P_i auf der Eingabe *i* in eine unendliche Schleife gerät, sodass $P_i(i)$ und $f(i)$ in der obigen Gleichungskette nicht definiert sind.

Moral: Dass Computerprogramme in unendliche Schleifen geraten können, ist kein Defekt, den man beheben könnte, sondern ein essenzieller Teil des Begriffs von Berechenbarkeit.

Satz (Turingsches Halteproblem, 1937): Die Funktion $h: \mathbb{N}^2 \to \mathbb{N}$ mit

$$h(i,n) = \begin{cases} 1 & \text{wenn } P_i(n)\downarrow, \\ 0 & \text{andernfalls,} \end{cases}$$

ist nicht berechenbar, d.h. man kann nicht algorithmisch entscheiden, ob ein gegebenes Computerprogramm anhalten wird.

Beweis: Wäre *h* berechenbar, so wäre es auch die Funktion $f: \mathbb{N} \to \mathbb{N}$ mit

$$f(n) = \begin{cases} P_n(n)+1 & \text{wenn } h(n,n)=1, \\ 0 & \text{sonst.} \end{cases}$$

Es wäre also f gleich P_i für ein *i*. Da *f(i)* definiert ist, gilt $P_i(i)\downarrow$ und somit $h(i,i)=1$. Man erhält

$$P_i(i) = f(i) = P_i(i)+1,$$

was der gewünschte Widerspruch ist. □

Satz (Erster Gödelscher Unvollständigkeitssatz, 1931): Es sei ein beliebiges Arsenal an Axiomen und Beweismethoden gegeben (mit gewissen Bedingungen). Dann gibt es eine wahre Aussage, die nicht bewiesen werden kann.

Beweis: Erstelle eine Liste aller Beweise $B_0, B_1, \dots$, sodass B_n eine Aussage der Form „für alle $m \in \mathbb{N}$ gilt $P_{i(n)}(m)\downarrow$" beweist. Unter milden Bedingungen an unsere Axiome und Beweismethoden gilt:

(i) Die Funktion $n \mapsto i(n)$ ist berechenbar.

(ii) Es gilt $P_{i(n)}(m)\downarrow$ für alle *i* und *m*.

Betrachte nun die Funktion $f: \mathbb{N} \to \mathbb{N}$ mit $f(n) = P_{i(n)}(n) + 1$. Wegen (i) ist *f* berechenbar und also gleich P_j für ein *j*. Wegen (ii) haben wir:

(+) Für alle $m \in \mathbb{N}$ gilt $P_j(m)\downarrow$.

Angenommen, die wahre Aussage (+) ist beweisbar. Dann taucht ihr Beweis B_n in unserer Liste auf. Es gilt also $j = i(n)$ für ein *n*. Nun folgt

$$P_j(n) = f(n) = P_j(n) + 1,$$

was ein Widerspruch ist. □

Die Aussage (+) aus dem Beweis des Gödelschen Satzes wurde speziell für diesen erdacht, ohne dass sie sonst in der Mathematik eine Rolle spielte.

Frage: Kann man in der „normalen" mathematischen Praxis Beispiele für Unbeweisbarkeit finden? Gibt es „natürliche" mathematische Ergebnisse, die nicht ohne die Verwendung von ungewöhnlich starken Axiomen beweisbar sind?

Antwort: Ja! Unverzichtbar sind sehr starke Axiome beispielsweise für den Minorensatz von Neil Robertson und Paul Seymour, wie diese zusammen mit Harvey Friedman bewiesen haben.[1] Der Minorensatz ist ein ganz zentrales Ergebnis der Graphentheorie und hat wichtige Anwendungen in der Informatik.

Im folgenden betrachten wir die Goodstein-Folgen aus der Einführungsvorlesung[2] als weniger bedeutendes aber instruktives Beispiel.

[1] H. Friedman, N. Robertson and P. Seymour, The metamathematics of the graph minor theorem, S. 229-261 in: S. Simpson (Hg.), Logic and Combinatorics, Contemporary Mathematics 65, American Mathematical Society, 1987.

[2] verfügbar über https://www.mathematik.uni-wuerzburg.de/mathematicallogic/lehre/material/

Wir rufen in Erinnerung, dass die Goodstein-Folge $G_n(0), G_n(1), \ldots$ mit Startwert n gegeben ist durch

$$G_n(0) = n, \qquad G_n(i+1) = \begin{cases} (i+3)^{e_0} \cdot c_0 + \ldots + (i+3)^{e_k} \cdot c_k - 1 \\ \quad \text{falls } 0 < G_n(i) = (i+2)^{e_0} \cdot c_0 + \ldots + (i+2)^{e_k} \cdot c_k \\ \quad \text{mit } e_0 > \ldots > e_k \text{ und } c_i < i+2, \\ 0 \quad \text{falls } G_n(i) = 0. \end{cases}$$

Betrachte nun die Funktionen $F_k : \mathbb{N} \to \mathbb{N}$, die für $k \in \mathbb{N}$ gegeben sind durch

$$F_0(n) = n+1, \qquad F_{k+1}(n) = F_k^n(n) = \underbrace{F_k(F_k(\ldots F_k(n)\ldots))}_{n \text{ Anwendungen von } F_k}.$$

Satz: Für jedes $k \in \mathbb{N}$ gibt es ein $N \in \mathbb{N}$ sodass für alle $n \geq N$ gilt: Das kleinste $i \in \mathbb{N}$ mit $G_n(i) = 0$ ist größer als $F_k(n)$. Es dauert also mehr als $F_k(n)$ Schritte, bis die Goodstein-Folge terminiert.

Fakt: Die Funktionen F_n wachsen enorm schnell. Es gilt etwa $F_2(n) = 2^n \cdot n$ und $F_3(3) > 10^N$ mit $N = 10^6$.

Fakt: Es gibt ein Axiomensystem WKL_0 mit den folgenden Eigenschaften:

(i) In WKL_0 können wichtige Teile der elementaren Analysis entwickelt werden.
(ii) Ist P_i eine berechenbare Funktion, sodass die Aussage

„für jedes n gilt $P_i(n)\downarrow$"

in WKL_0 beweisbar ist, so gibt es ein $k \in \mathbb{N}$ mit $P_i(n) \leq F_k(n)$ für alle $n \in \mathbb{N}$.

Korollar: Die Aussage, dass jede Goodstein-Folge terminiert, dass es also für jedes $n \in \mathbb{N}$ ein $i \in \mathbb{N}$ gibt mit $G_n(i) = 0$, ist nicht in WKL_0 beweisbar.

Stellen Sie sehr gern Ihre Fragen!

Jetzt oder später per Email an anton.freund@uni-wuerzburg.de.