Martin Goldstern

Der logische Denker Kurt Gödel und sein Unvollständigkeitssatz

http://www.tuwien.ac.at/goldstern/
6.September 2006

Kurt Gödel, 1906-1978

1906: geboren am 28.April in Brünn (heute Brno)

1924: Studium in Wien
- Studium der Mathematik
- auch Physikvorlesungen
- Interesse für Philosophie; Wiener Kreis

1930: Dissertation — **Vollständigkeitssatz**

1931: Habilitation — **Unvollständigkeitssatz**

1933– : Besuche in Princeton, *Institute for Advanced Study.*
Freundschaft mit Albert Einstein

Kurt Gödel, 1906-1978

1938: Heirat

1940: Mengenlehre; konstruktibles Universum, Auswahlaxiom, Kontinuumshypothese

1940: Emigration in die USA

1949: Gödels rotierendes Universum, Zeitreisen

1954: permanente Stelle am IAS

1978: an Unterernährung gestorben

Formale Systeme

Wenn $x + 2 = 9$, dann $x = 7$.

Schematisch:

$$\frac{x + 2 \quad = \quad 9}{x \quad = \quad 7}$$

Deutlicher:

$$\frac{x + 2 \quad = \quad 7 + 2}{x \qquad = \quad 7}$$

Formale Systeme

Allgemeine Regel:

$$\frac{x+2 \quad = \quad a+2}{x \quad = \quad a}$$

Diese Regel lässt sich aber durch die folgende einfachere begründen:

$$\frac{x+1 \quad = \quad a+1}{x \quad = \quad a}$$

$$\begin{array}{rlcl}
 & x+2 & = & 9 \\ \hline
 & x+1+1 & = & 7+1+1 \\ \hline
\text{Regel} \rightarrow & x+1 & = & 7+1 \\ \hline
\text{Regel} \rightarrow & x & = & 7
\end{array}$$

Rechenregeln

- $x + 1 \neq 0$
- Wenn $x + 1 = y + 1$, dann $x = y$.
- $x + 0 = x$
- $x + (y + 1) = (x + y) + 1$.
- $x \cdot 0 = 0$
- $x \cdot (y + 1) = (x \cdot y) + x$.
- Wenn $x \neq 0$, dann $\exists y : x = y + 1$.

Die angeführten Rechenregeln (oder auch „Axiome für die natürlichen Zahlen“) haben die folgenden Eigenschaften:

- Sie sind von besonders einfacher Form.
- Sie sind alle offensichtlich wahr (in Bezug auf die natürlichen Zahlen)
- Durch Verknüpfung dieser Regeln kann man auch kompliziertere Regeln gewinnen (die dann natürlich auch wahr sind)
- Durch Hinzufügen einer weiteren Regel („Induktionsaxiome“) kann man „fast alle“ wahren Sätze über die natürlichen Zahlen beweisen.

Wahre und falsche Sätze

$\forall x$ ... für alle natürlichen Zahlen x

$\exists x$ ... es gibt (mindestens) eine nat. Zahl x

$:$ ... sodass gilt:

$$\forall x\, \forall y : x + y = y + x.$$

$$\forall x\, \exists y : x = y$$

$$\exists y\, \forall x : x = y$$

$$\exists x, y, z > 0 : x^2 + y^2 = z^2$$

$$\exists x, y, z > 0 : x^3 + y^3 = z^3$$

$$\exists x, y, z > 0 : x^3 + y^5 = z^2$$

$$\exists x, y, z > 0 : x^3 + y^3 = 2z^3$$

In allen diesen Fällen ist „ja" leichter zu zeigen als „nein".

Entscheidungsproblem: Gibt es eine Maschine (bzw: ein Computerprogramm; bzw: ein Axiomensystem), die/das von jedem arithmetischen Satz entscheiden kann, ob er wahr ist?

(So eine Maschine könnte Mathematiker *im Prinzip* überflüssig machen.)

Bescheidenere Variante: können wir zumindest entscheiden, ob ein Satz der Form

$$\exists x \ldots \exists z : (\text{Gleichungen mit } x, \ldots, z)$$

wahr oder falsch ist?

(Dass er **wahr** ist, können wir im Prinzip immer wissen. Aber wie können wir sicher sein, dass er **falsch** ist?)

Der 1.Gödelsche Satz

Sei M eine Maschine (ein Programm, ein Axiomensystem, ein Kalkül), die nur wahre arithmetische Sätze produziert. Dann gibt es einen arithmetischen Satz S_M, der zwar wahr ist, aber von M niemals produziert werden kann.

S kann aus einer Beschreibung von M explizit definiert werden.

Der 1.Gödelsche Satz – Gödelisierung

Gödels Idee: Wir können sowohl Maschinen M wie auch Sätze S wie auch einen Produktionsvorgang P durch natürliche Zahlen m, s und p codieren. (Bitfolgen!)

Die Aussage „M produziert den Satz S durch den Programmablauf P" lässt sich dann als Gleichung

$$\big(m - \cdots + m^{p-s} - \cdots\big) \cdot \big(s \cdot m - \cdots\big) = 0$$

kodieren.

Der Satz: „M kann den Satz S niemals generieren" lässt sich durch den folgenden Satz $G_{m,s}$ beschreiben:

$$\forall p \, \big(m - \cdots + m^{p-s} - \cdots\big) \cdot \big(s \cdot m - \cdots\big) \neq 0$$

$$G_{m,s} \ : \quad \forall p\,\big(m-\cdots+m^{p-s}-\cdots\big)\cdot\big(s\cdot m-\cdots\big) \neq 0$$

„Die durch m kodierte Maschine M kann den durch s kodierten Satz S nicht erzeugen."

Sei M eine fest vorgegebene Maschine (oder ein fest vorgegebenes Axiomensystem), kodiert durch die Zahl m. Dann können wir jeden Satz S (kodiert durch eine Zahl s) einen neuen Satz $G_{m,s}$ zuordnen. $G_{m,s}$ sieht im Allgemeinen viel komplizierter aus als der Satz S.

Gödel konnte nun zeigen, dass man für jede Zahl m (bzw jede Maschine M) eine Zahl s^* (bzw einen Satz S^*) finden kann, sodass $S^* = G_{m,s^*}$ ist.

Der Satz S^* besagt also nun:

> „Die durch m kodierte Maschine M kann mich (d.h., den Satz S^*) nicht erzeugen."

S^* muss wahr sein, kann also von M nicht erzeugt werden.

Zusammenfassung: Zu jeder Maschine M, die nur wahre arithmetische Sätze generiert, gibt es einen wahren Satz S_M, der von M nie generiert werden kann.

(Matijasewitsch) Überdies kann S_M eine sehr einfache Form haben:

$$\forall x \ldots \forall z : (\text{Gleichungen mit } x, \ldots, z)$$

Schlussfolgerung:

- Die wahren Sätze der Arithmetik (und erst recht die der Mathematik) lassen sich nicht automatisch generieren
- Gödel weiß mehr, als jede Maschine je wissen kann. (?)
- Der Mensch (das Hirn) weiß mehr, als jede Maschine je wissen kann. (?)

Wirklich?

Woher wissen wir, dass eine vorgegebene Maschine M nur wahre Sätze produziert?

Umformulierung: Zu jeder Maschine M kann man explizit einen Satz S_M mit der folgenden Eigenschaft angeben:

> Wenn M nur wahre arithmetische Sätze generiert,
> dann ist S_M wahr, kann aber von M nie generiert werden.

(Dieser Satz S_M kann aus der Beschreibung von M mechanisch generiert werden.)

Rezeption

- Zunächst ignoriert von vielen Mathematikern
- **Bourbaki** (französisches Autorenteam, „Enzyklopädie der Mathematik"), sprechen über logische Grundlagen der Mathematik, axiomatische Methode — erwähnen Gödel nicht
  Dieudonne 1939, A.Weil 1948
- ignoriert in Nazi-Deutschland (Logik=jüdisch)
- größere Bekanntheit in 1950er Jahren
- populärwissenschaftlich: *Gödel, Escher, Bach* 1979, Bestseller.

Gegenwartsbezug

Codierung von Algorithmen/Programmen mit Zahlen ist heute selbstverständlich. Damals war das nicht so.

Gödels Satz:
Zeigt Grenzen der Berechenbarkeit.

Gödels Methode:
Definition der Berechenbarkeit. =
Grundlage der theoretischen Informatik

Gödel, Alan Turing, Stephen Kleene:
Computerprogramme, bevor es noch Computer gab.

Erkenntnisvorlauf der Mathematik.

Philosophische Konsequenzen

Ist das Gehirn eine „endliche Maschine“?

- endlich viele Atome, Beziehungen zwischen ihnen
- (Gödel:) Gehirn ändert sich, kann lernen.
- (Turing:) Kein mathematisches Modell für „Intuition“
- (Penrose:) Quantenmechanische Vorgänge im Gehirn. „Wir sind noch nicht so weit.“
- *idealer Mathematiker* vs. *wirklicher Mathematiker* — Wahrheit als Annäherung

The modern development of the foundations of mathematics in the light of philosophy (Gödel 1961)

Es zeigt sich nämlich, dass bei einem systematischen Aufstellen der Axiome der Mathematik immer wieder neu und neuere Axiome evident werden, die nicht formallogisch aus den bisher aufgestellten folgen.

Es ist durch [den Unvollständigkeitssatz] gar nicht ausgeschlossen, dass trotzdem auf diese Weise jede klar gestellte mathematische Ja- oder Nein-Frage lösbar ist,
denn eben dieses Evidentwerden immer neuerer Axiome auf Grund des **Sinnes** der Grundbegriffe ist etwas, was eine Maschine nicht nachahmen kann.

Die theologische Weltanschauung ist mit allen bekannten Tatsachen vereinbar. Das heißt, die Vorstellung, dass alles auf der Erde Sinn und Vernunft hat.

Da unser Erdendasein aber nur zweifelhaften Sinn hat, kann es nur Mittel zum Zweck für eine andere Existenz sein.

Weiterführende Informationen

Bücher:

- Ernest Nagel, James R. Newman:
  Der Gödelsche Beweis.
- Douglas Hofstadter: *Gödel, Escher, Bach.*
- *und viele andere...*