Formale Logik

PD Dr. Markus Junker
Abteilung für Mathematische Logik
Universität Freiburg

Wintersemester 16/17
Sitzung vom 25. Januar 2017

Gödels Unvollständigkeitssatz

Unvollständigkeit von Axiomensystemen: Für jede „aufschreibbare“ und hinreichend aussagefähige widerspruchsfreie Menge von $\mathcal{L}$-Sätzen („eine $\mathcal{L}$-Theorie“) Φ findet man einen $\mathcal{L}$-Satz F, den Φ nicht entscheidet, d. h. weder F noch $\neg F$ folgen logisch aus Φ.

(*Aufschreibbar: Es muss ein Computerprogramm geben können, das alle Axiome erzeugt. Hinreichend aussagefähig: Die Theorie muss über Zahlen sprechen können bzw. über sich selbst.*)

Unvollständigkeit von Beweissystemen:
In jedem Modell solch einer Theorie gibt es also wahre Sätze F, die nicht beweisbar sind (d. h. deren Wahrheit nicht nachweisbar ist).

(*Denn in einem Modell $\mathfrak{M}$ gilt entweder F oder $\neg F$; der Nachweis von F oder $\neg F$ kann sich aber nur auf eine „aufschreibbare“ Menge von Voraussetzungen gründen.*)

Die Menge der in $\mathfrak{M}$ gültigen Sätze ist also nicht aufschreibbar.

Das Halteproblem (1)

Ein Computerprogramm hat üblicherweise eine Eingabe und eine Ausgabe.

Zum Beispiel hat ein Routenplaner als Eingabe zwei Orte, als Ausgabe eine kürzeste oder schnellste Verbindung zwischen den beiden Orten.
Ein automatischer Übersetzer hat als Eingabe einen deutschen Text und als Ausgabe eine englische Übersetzung.
Ein Programmverifikator hat als Eingabe ein Programm und als Ausgabe „ja“, falls das Programm korrekt arbeitet, oder „nein“ andernfalls.

Die Eingabe kann auch leer sein.
(*Nichts ins Eingabefeld eingeben und Return-Taste drücken . . .*)

Manchmal liefern Programme auf einen Eingabe hin keine Ausgabe: Sie geraten in eine unendliche Schleife oder „stürzen ab“, d. h. der Computer rechnet und rechnet weiter, ohne je eine Ausgabe zu liefern. Man weiß aber zu keime Zeitpunkt, ob er nicht irgendwann eine Ausgabe liefern wird.

Das Halteproblem (2)

Halteproblem: Kann man ein Computerprogramm H schreiben, das als zweiteilige Eingabe zum einen Computerprogramme P (einer festen Programmiersprache) und zum andern Eingaben E für diese Programme bekommt und als Ausgabe die Antwort liefert, ob das Programm P mit der Eingabe E irgendwann anhält und eine Ausgabe liefert?

> Alan Turing (1937):
> *Das Haltproblem ist unentscheidbar,*
> *das heißt, es gibt kein solches „Meta-Programm".*

Anmerkung: Programme sind in einer Programmiersprache geschrieben, also Zeichenfolgen oder Texte, und können damit selbst zur Eingabe eines Programms werden. Man kann also „normale Programme" von „Meta-Programmen" nicht prinzipiell unterscheiden.

Das Halteproblem (3)

Beweis: Angenommen H wäre ein Meta-Programm für das Halteproblem und lieferte bei Eingabe (P, E) die Antwort

$H(P, E) = 1$, falls P bei Eingabe E irgendwann stoppt
$H(P, E) = 0$, falls P bei Eingabe E ohne Ende läuft

Dann gäbe es auch das Programm Q, das bei Eingabe eines Programms P folgendes tut: Berechne $H(P, P)$ und wenn

$H(P, P) = 1$:	irgendeinen Befehl, der im Kreis laufen lässt	Also $H(Q, P) = 0$
$H(P, P) = 0$:	Programm anhalten und „Heureka“ ausgeben.	Also $H(Q, P) = 1$

Mit $P = Q$ folgt nun der Widerspruch:

$$H(Q, Q) = 1 \iff H(Q, Q) = 0$$

Das Halteproblem und der Unvollständigkeitssatz

All dies kann man präzise machen mit einem festen Computermodell und einer festen Programmiersprache und man kann dann prädikatenlogische Formeln angeben, die diese Computerprogramme beschreiben und genau dann erfüllbar sind, wenn die Programme (z. B. mit leeren Eingabe) anhalten.

Wenn man also die Erfüllbarkeit prädikatenlogischer Formeln entscheiden könnte, könnte man auch das Halteproblem entscheiden.

Wichtiges Ingredienz: Selbstbezüglichkeit (hier: Ein Computerprogramm hat sich selbst als Eingabe und macht über sich eine Aussage) + Verschränkung durch Verneinung, wie im

Lügner-Paradoxon:	*Dieser Satz ist falsch.*
(Dagegen falscher Satz:	*Alle Kreter lügen, sagt der Kreter.*)

Gödel nach Smullyan (1)

Raymond Smullyan (*1919)

> *an American mathematician, concert pianist, logician, Taoist philosopher, and magician (Wikipedia)*

Viele „logische Puzzles“ im Umkreis um Gödels Unvollständigkeitssatz.

Ein paar Buchtitel:

- *What Is the Name of This Book?*
- *To Mock a Mockingbird*
- *Forever Undecided*
- *This Book Needs No Title*

Gödel nach Smullyan (2)

Eine kleine formale Theorie:

- Es gibt eine Reihe von Zeichen (Großbuchstaben).
- Jede Folge von Zeichen heißt Ausdruck.
- Manche Ausdrücke heißen Prädikate.
- Ein Satz besteht aus einem Prädikat gefolgt von einem beliebigen Ausdruck.
- Wenn P ein Prädikat ist und $A\ldots$ ein Ausdruck, dann trifft P entweder auf $A\ldots$ zu oder trifft nicht zu. Im ersten Fall heißt der Satz $PA\ldots$ wahr, im zweiten falsch.
- Eine Menge M von Ausdrücken heißt benennbar, falls es ein Prädikat P gibt, so dass M gerade aus den Ausdrücken $A\ldots$ besteht, auf die P zutrifft.
- Zwei Sätze heißen äquivalent, wenn sie den gleichen Wahrheitswert haben.

Gödel nach Smullyan (3)

Es gibt zwei besondere Zeichen N und R und dafür folgende Regeln:

Negationsregel:
Für jedes Prädikat P ist auch NP ein Prädikat und für jeden Ausdruck $A\dots$ ist $PA\dots$ genau dann wahr, wenn $NPA\dots$ falsch ist.

Repetitionsregel:
Für jedes Prädikat P ist auch RP ein Prädikat und für jedes Prädikat Q ist der Satz RPQ äquivalent zu PQQ.

Prädikate können also von Prädikaten ausgesagt werden:
z. B. *„Ist rot" ist rot.* (falsch) oder: *„Ist rot" ist nicht rot.* (wahr)

Mit P = *„ist Unsinn"* und Q = *„ist rot"* ist dann:
$RPQ = PQQ =$ *„‚Ist rot' ist rot" ist Unsinn.*

Gödel nach Smullyan (4)

Fixpunktsatz: (*Kleene für Prädikatenlogik*)

Jedes Prädikat P hat (mindestens) einen Fixpunkt,
d. h. einen Satz $S\ldots$, so dass $PS\ldots$ äquivalent zu $S\ldots$ ist.

(*Also etwa: $S\ldots$ ist falsch und $PS\ldots$ auch.*)

Beweis:
RPQ ist äquivalent zu PQQ für jedes Prädikat Q,
also auch für $Q = RP$:

$$RPRP \text{ ist äquivalent zu } PRPRP$$

Also ist $RPRP$ Fixpunkt von P.

Gödel nach Smullyan (5)

Es gibt kein Wahrheitsprädikat: (*Tarski für Prädikatenlogik*)

Es gibt kein Prädikat T, das die wahren Sätze benennt,
d. h. es gibt kein Prädikat T mit:

$TS\ldots$ ist wahr, wenn $S\ldots$ wahr ist
und falsch, wenn $S\ldots$ falsch ist.

(Dies würde bedeuten, dass alle Sätze Fixpunkte von T wären.)

Beweis:
Für jedes Prädikat T ist $RTRT$ Fixpunkt von T
NT ist ein Prädikat, also ist $X\ldots = RNTRNT$ Fixpunkt von NT
d. h. $X\ldots$ ist wahr genau dann, wenn $NTX\ldots$ wahr ist,
genau dann, wenn $TX\ldots$ falsch ist.
Also ist T kein Wahrheitsprädikat wegen $X\ldots$.

Gödel nach Smullyan (6)

Angenommen, wir haben nun die Möglichkeit, für einige Sätze ihre Wahrheit bzw. Falschheit nachzuweisen.

B sei ein Prädikat dafür: Wenn wir die Möglichkeit haben, die Wahrheit eines Satzes $PA\ldots$ nachzuweisen, soll der Satz $BPA\ldots$ wahr sein, (und wenn wir die Möglichkeit haben, die Falschheit eines Satzes $PA\ldots$ nachzuweisen, soll der Satz $BNPA\ldots$ wahr sein).

B heißt Beweisbarkeitprädikat. Ein Satz $PA\ldots$ heißt beweisbar, falls B auf ihn zutrifft (und widerlegbar, falls $NPA\ldots$ beweisbar ist).

Annahme: B soll *korrekt* sein, d. h. alle beweisbaren Sätze sind tatsächlich wahr (und alle widerlegbaren Sätze tatsächlich falsch).

Also: $BS\ldots$ wahr impliziert $S\ldots$ wahr.

Gödel nach Smullyan (7)

Unvollständigskeitssatz (*Gödel für Prädikatenlogik*)

Es gibt wahre, nicht beweisbare Sätze,
d. h. Sätze $S\ldots$, die wahr sind, aber für die $BS\ldots$ falsch ist.

Beweis:
(1) Die wahren Sätze sind nach Tarski nicht benennbar (durch ein Prädikat), die beweisbaren Säze sind aber durch B benennbar.
Jeder beweisbar Satz ist aber wahr.
Also gibt es wahre Sätze, die nicht beweisbar sind.

(2) Sei $X\ldots$ ein Fixpunkt von NB. Dann

$X\ldots$ ist genau dann wahr, wenn $NBX\ldots$ wahr ist,
genau dann, wenn $BX\ldots$ falsch ist,
genau dann, wenn $X\ldots$ nicht beweisbar ist.

Also ist $X\ldots$ entweder nicht wahr und beweisbar (was unmöglich ist) oder wahr und nicht beweisbar.

Gödel nach Smullyan (8)

Ein Fixpunkt $X\ldots$ von NB ist eine Variante des Lügner-Paradoxons.

Er sagt gewissermaßen: Ich bin nicht beweisbar.

Schwierig für die Prädikatenlogik ist es, die Repetitionsregel zu beweisen und das Beweisbarkeitsprädikat zu definieren.

Dies sind technisch äußerst anspruchsvolle Vorgänge, die tatsächlich einiges an Mathematik brauchen (im Gegensatz zu der hier präsentierten Abstraktion von Smullyan, die formal ist, aber keine Mathematik).