Gödels Unvollständigkeits-Theorem

Helmut Eberharter
Matr.# 9926931
Kennzahl 535
Institut für Technische Informatik
Technische Universität Wien
e9926931@student.tuwien.ac.at

20. Mai 2008

"*Epimenides, der Kreter, sagte alle Kreter wären Lügner und alle sonst von den Kretern aufgestellten Behauptungen wären gewiß Lügen. War das eine Lüge?*" [1]

Abstract — *Das folgende Essay behandelt die Formalisierung der Mathematik, die Zerstörung der Hilbertschen Vision durch Gödel und das Halteproblem als praktisches, an den Gödelschen Beweis angelehntes Beispiel. Die wichtigsten Begriffe wie formales System, Widerspruchsfreiheit und Vollständigkeit werden eingeführt. Abschließend wird Gödels Unvollständigkeits-Theorem nah am Originalartikel erklärt.*

1 Einleitung

Diese Arbeit führt zunächst in Zeit und Leute rund um den Gödelschen Unvollständigkeitssatz ein. Nach der Betrachtung der Hintergründe der Hilbertschen Forderung nach einer Formalisierung der Mathematik und einer Einführung in formale Systeme in Abschnitt 2, behandelt Abschnitt 3 die Turing-Maschine, die formale Systeme "automatisch" bearbeiten kann. Der Beweis für die Unentscheidbarkeit des Halteproblems und somit des Entscheidungsproblems wird nachvollzogen. Der anschließende Teil diskutiert den Zusammenhang mit Gödels Unvollständigkeitssatz. Im letzten Abschnitt 4.1 wird der Beweis der Unvollständigkeit, wie in Gödel geführt hat, nah am Originialdokument, nachvollzogen, bevor die Conclusio in Abschnitt 5 den Artikel abschließt und kurz die Konsequenzen des Unvollständigkeitssatzes für die Mathematik beleuchtet.

2 Die Formalisierung der Mathematik

2.1 Millennium Probleme

Im Jahr 2000, veröffentlichte das Clay Mathematics Institute (CMI, Cambridge) eine Liste von sieben ungelösten Problemen der Mathematik und setzte für deren Lösung jeweils eine Million Dollar Preisgeld aus. Motivation war nicht nur die Jahrhundertwende und damit die Standortbestimmung der modernen Mathematik, sondern wesentlich auch ein Vortrag von David Hilbert im Jahr 1900 [2], bei dem er zehn ungelöste mathematische Probleme formulierte, um einen Ausblick auf die im neuen Jahrhundert wichtigsten Forschungsgebiete der Mathematik zu geben und zur Auseinandersetzung mit diesen Problemen zu animieren. Dies ist ihm, als einem der größten Mathematiker seiner Zeit, auch gelungen.

Die zehn Probleme Hilberts (die noch im Jahr 1900 auf 23 ausgeweitet wurden) sind berühmt geworden, mit ihnen die Mathematiker, die eines der Probleme lösten. Hilberts zweites Problem lautet: "*Sind die arithmetischen Axiome widerspruchsfrei?*"[3]. Die *arithmetischen Axiome* sind die Peanoschen Axiome, welche die Grundlage für die Mathematik bilden sollten und heute noch als (eine mögliche) didaktische Grundlage für das Studium der Mathematik dienen.

Das CMI bemühte Experten aus den einzelnen mathematischen Fachgebieten, welche die mathematischen Fragestellungen exakt ausformulieren sollten. Hilberts Probleme waren viel einfacher formuliert, da zu seiner Zeit die Mathematik noch nicht formalisiert war und noch nicht auf axiomatischen Fundamenten stand. Dieses Manko war Hilbert bewusst, und er war überzeugt, dass eine Formalisierung der Mathematik nötig und möglich ist.

2.2 Formalisierung der Mathematik

Die Hoffnung und Forderung Hilberts, dass die Mathematik vollständig formalisiert würde, entstand aus einer tiefen Krise: Innerhalb der Logik wurden vorerst unlösbare Paradoxa gefunden. Eines der Bekanntesten ist die Russellsche Antinomie, die von Bertrand Russell und Ernst Zermelo entdeckt wurde. Russell argumentierte mit Klassen und Prädikaten, folgende Formulierung ist aber einfacher zu verstehen. Durch die Einfachheit wird klar, warum diese Antinomie die Mathematik so schwer erschüttert hat.

Als Russellsche Klasse definieren wir die Menge M, welche die Menge aller Mengen ist, die sich nicht selbst als Element enthalten. Die Gretchenfrage lautet nun: Enthält M sich selbst?

Angenommen M enthält sich selbst, dann gilt aufgrund der Definition, dass sich M nicht selbst enthält. Die

Annahme führt also zu einem Widerspruch zur Definition. Wenn M sich aber nicht selbst enthält, dann gehört M zur Menge derjenigen Mengen die sich nicht enthalten, d.h. M enthält sich selbst. Russel ist also auf eine Antinomie gestoßen[1], auf zwei Aussagen, die sich widersprechen, aber beide beweisbar sind.

Das eingangs stehende Zitat über Epimenides, den Kreter, basiert auf dem über 2000 Jahre alten Paradoxon des Epimenides und ist von ähnlicher Form: eine Aussage wird solcher Art formuliert, dass sie weder bewiesen noch widerlegt werden kann.

Als Ausweg aus diesen und anderen Paradoxa sollte die Mathematik formalisiert werden. Eine Formalisierung geht von klar definierten Grundaussagen (Axiomen) aus, von welchen sich die ganze Mathematik ableiten solle. Damit wäre kein Spielraum mehr aufgrund unpräzise formulierter Begriffe. Die formale Mathematik würde sich darauf beschränken, aus den Axiomen schrittweise die Aussagen der Mathematik abzuleiten. Und solange die Axiome und Schlußregeln einfach (intuitiv) wären, könnten keine Antinomien oder Paradoxa formuliert werden. Ein schrittweiser, mechanischer Vorgang, nach welchem Aussagen gewonnen werden, wird heute Algorithmus genannt. Hier finden wir einen Hinweis auf die enge Verwandtschaft von Informatik und mathematischer Logik - davon später mehr, wenn wir Turings Beweis kennen lernen.

Nicht nur Hilbert forderte diese Formalisierung (Algorithmisierung), sondern Russell selbst versuchte mit der Principia Mathematica[2], die Mathematik auf die Grundlage eines formalen Systems zu stellen und somit widerspruchsfrei zu machen.

2.3 Formale Systeme, Widerspruchsfreiheit und Vollständigkeit

Die Principia Mathematica, geschrieben von Bertrand Russell und Alfred Whitehead, ist eines der wichtigsten mathematischen Werke überhaupt und baut ein formales System auf. Ein formales System besteht aus Symbolen, Ableitungsregeln und (wenigen) Axiomen. Die **Symbole** tauchen nahezu immer als Symbolketten auf, vergleichbar mit den ganzen Zahlen, die aus Ziffern zusammengesetzt sind. Die **Ableitungsregeln** bestimmen, wie und unter welchen Bedingungen eine Symbolkette in eine andere umgewandelt werden kann; sie legen Operationen fest. Die **Axiome** sind vorgegebene Formeln, welche die Ausgangssätze für das formale System bilden, diese *initialisieren* das formale System.

Eine gute Einführung in formale Systeme gibt Douglas R. Hofstadter in seinem monumentalen Werk "*Gödel, Escher, Bach*" [4]. Er stellt dort das *MU-Rätsel* [4, S.37ff] vor. Zugrunde liegt das MIU-System, welches als Symbole die drei Buchstaben M, I und U verwendet (wie für alle formalen Systeme ist es wichtig, die Symbole von ihrer eigentlichen Bedeutung zu lösen, speziell bei Zahlzeichen ist dies essentiell!) Weiters gehören zum MIU-System vier (Ableitungs)Regeln:

1. Ist das letzte Symbol einer Kette ein I, kann ein U angefügt werden.
2. Aus Mx kann Mxx erzeugt werden. (z.B: aus MUM wird MUMUM)
3. Die Symbolkette III kann durch U ersetzt werden.
4. Die Symbolkette UU kann gestrichen werden.

Das MU-Rätsel lautet nun: Kann aus der Kette (dem Axiom) MI die Kette MU erzeugt werden?[3] Wer sich mit diesem Rätsel beschäftigt, hantiert mit einem formalen System. Das MIU-System ist ein ganz einfaches formales System, in welchem logische Aussagen nicht formuliert werden können. Hilbert fordert daher auch den Beweis der Widerspruchsfreiheit für das formale System, das auf den Peano-Axiomen aufbaut. Wie aber ist die Widerspruchsfreiheit definiert?

Ein formales System ist **widerspruchsfrei**, wenn von den Aussagen *A* und *nicht A*[4], das ist die Verneinung von A, nur eine abgeleitet werden kann. Der Bruder der Widerspruchsfreiheit ist die Vollständigkeit: Ein formales System ist **vollständig**, wenn entweder A oder $\overline{A}$ abgeleitet werden kann. Warum ist aber Vollständigkeit so wichtig? Weil die Paradoxa der Logik, und somit die Krise der Mathematik nur gelöst werden können, wenn jede wahre Aussage[5] abgeleitet werden kann. Nur dann ist die Mathematik formalisiert und alle Aussagen können aufgelöst und aus den Axiomen entwickelt werden. Zusammen mit der Widerspruchsfreiheit bedeutet dies, dass entweder *A* oder *nicht A* auf jeden Fall abgeleitet werden können, aber nicht beide gleichzeitig.

3 Algorithmisierung der Mathematik und die Turing-Maschine

Bevor wir im letzten Abschnitt Gödels Beweis genauer analysieren, springen wir ins Jahr 1936, als Alan Turing das Halteproblem löste [5]. Bei den Versuchen, die Mathematik zu formalisierten, setzte Alan Turing einen ganz entscheidenden Schritt. Er löste den schrittweisen Vorgang des Schließens (Beweisens) aus der Mathematik selbst heraus und erdachte eine Maschine, die dies - theoretisch - alleine, ohne menschliche Hilfe, leisten konnte. Er beschrieb die Turing-Maschine, welche in der

[1] er fand mit der Typentheorie auch eine Form der Mengenlehre, die dieses Problem durch Einschränkung umgeht: Einfache Mengen dürfen keine Mengen als Elemente enthalten.

[2] und der ihr zugrunde liegendenTypentheorie

[3] Nein

[4] eine Verneinung wird durch Überstreichung gekennzeichnet. $\overline{A}$ bedeuted also $nicht\ A$

[5] innerhalb des Systems wahr

Lage war, die schrittweise Ableitung aus Axiomen mechanisch durchzuführen. Dazu stützte er sich auf Gödels Arbeit, im speziellen auf die Methode der Gödelisierung, der Repräsentation von Formeln durch Zahlen, und damit die Reduzierung von logischen Problemen auf arithmetische.

Die Turing-Maschine ist relativ einfach aufgebaut: Sie besteht aus:

- einem unendlichen Band mit fortlaufenden Feldern, in jedem Feld ist immer genau ein Symbol gespeichert,
- einer Maschine, die von diesem Band lesen und schreiben kann und die sich schrittweise auf dem Band hin und her bewegen kann
- und einer (endlichen) Zahl von inneren Zuständen der Maschine.

Die Analogien zu formalen Systemen sind dabei: Das Band mit den Symbolen repräsentiert die Symbolketten, die inneren Zustände entsprechen den Schlußregeln und die Startposition, zusammen mit der genauen Symbolbelegung des Bandes beim Start, entspricht den Axiomen, den Ausgangsbedingungen formaler Systeme. Alle drei Elemente lassen sich (theoretisch) niederschreiben und können als Programm für die Turing-Maschine angesehen werden.

Wird nun eine Turing-Maschine mit einem gewissen Programm und einer gewissen Symbolfolge (Axiome) gestartet, arbeitet sie automatisch so lange, bis sie einen Zustand[6] erreicht hat, der als Endzustand definiert wurde. Dann bricht sie die Bearbeitung ab, das Programm endet. Die Ausgabe des Programms liegt anschließend in Form von Symbolen auf dem Band vor.

Um Hilberts Forderung nach einer strikten Formalisierung zu verwirklichen, musste auch das so genannte Entscheidungsproblem gelöst werden, das Hilbert 1928 formulierte. Genauer wollen wir das *spezielle Entscheidungsproblem*, das Entscheidungsproblem in der Prädikatenlogik und die Lösung Turings behandeln. Die Hilbertsche Fragestellung lautete: Gibt es ein (Entscheidungs)verfahren, das eine mathematische Aussage auf seine Korrektheit[7] überprüft? Mit einem solchen Verfahren wäre es möglich Aussagen, wie die des Epimenides, auf Korrektheit zu überprüfen, das Verfahren würde einfach "Ja" oder "Nein" antworten. Es sollte also eine Möglichkeit gefunden werden, für eine beliebige Aussage zu überprüfen, ob sie beweisbar ist.

[6] es können auch mehrere Zustände als Endzustände defniert werden

[7] Korrektheit kann für unsere Zwecke mit Beweisbarkeit gleichgesetzt werden.

3.1 Das Halteproblem

Turing und Alonzo Church[8] beantworteten diese Frage fünf Jahre nachdem Gödel den Unvollständigkeitssatz bewies. Für uns ist dabei zweierlei von Interesse: Zunächst die Konsequenz, die sich aus den beiden Beweisen ergibt: Wenn ein System ein Entscheidungsverfahren kennt, ist es vollständig, da ein Entscheidungsverfahren jede (wohlgeformte[9]) Aussage überprüfen können muss. Zweitens, die Ähnlichkeit der Beweise, womit wir Gödels rein formalen Beweis für Informatiker verständlicher machen.

Turing überträgt also Algorithmen auf die Turing-Maschine und behauptet in seiner Arbeit, dass die Maschine jede Berechnung durchführen kann, zu der auch ein Mensch in der Lage ist. Wenn also ein Programm geschrieben wird, welches feststellt, ob ein Aussage korrekt ist, muss dieses Programm nur noch in einen Endzustand gelangen, also anhalten, und das Entscheidungsproblem wäre gelöst. Das Halteproblem überträgt das Entscheidungsproblem auf die Turing-Maschine: Ist es möglich, ein Programm zu schreiben, welches für ein beliebiges Programm feststellt, ob dies anhält? Kann ein Haltetester kodiert werden?

Im System der Turing-Maschine kann das Halteproblem nicht entschieden werden. Turings Beweis wird oft über folgende Analogie erklärt: Gehen wir davon aus, dass ein *Haltetester* [6] geschrieben werden kann. Dieser erhält als Eingabe ein Programm, also eine Turing-Maschine mit Bandinhalt, inneren Zuständen und der Startposition und analysiert, ob das Programm anhält. Nun gibt es ein weiteres Programm, den Haltetestverderber. Auch der *Haltetestverderber* verwendet ein komplettes Programm als Eingabe und benutzt den Haltetester um zu analysieren, ob das Programm anhält. Sollte es halten, geht er in eine Endlosschleife über, sollte es nicht anhalten, so hält der Haltetestverderber an.

Ähnlich wie bei Gödel wird jetzt *eine Formel des Systems,* bei uns der Haltetestverderber, auf sich selbst angewendet: Wir geben dem Haltetestverderber sein eigenes Programm als Eingabe. Was passiert? Sollte das Programm des Haltetestverderbers anhalten, sorgt dieser dafür, dass es nicht anhält, was ein Widerspruch ist. Sollte hingegen das Programm anhalten, so geht der Haltetestverderber nicht in eine Endlosschleife, und er hält an. Wir haben ihn aber so konstruiert, dass er nie anhält, auch dies ist ein Widerspruch. Turing hat mit dieser Argumentation bewiesen, dass es keine allgemeinen Haltetester geben kann! Mit andere Worten: Der Haltetestverderber mit seinem eigenen Programm als Eingabe hält

[8] Church bewies dies zwar Monate bevor Turing seine Arbeit veröffentlichte, schlug aber den formalen Weg ein. Wir interessieren uns hier für Turings Beweis

[9] Aus den Symbolen entsprechend der Regeln des formalen System zusammengesetzt.

genau dann an, wenn er nicht anhält.

3.2 Das Halteproblem ist unentscheidbar

In Turings Beweis kommt kein Haltetester und kein Haltetestverderber vor. Bevor wir den Beweis näher in seiner Originalform betrachten, hier noch die Idee des Beweises nach Turing: "*Es wird gezeigt, dass es keine allgemeine Methode gibt die feststellt, ob eine gegebene Formel U beweisbar in Z ist, oder, was sich auf das gleiche beläuft, ob das System Z mit der hinzugefügten Formel -U konsistent ist.*" [5, S. 259][10] Es folgt also aus der Unentscheidbarkeit des Halteproblems dasselbe wie aus dem Unvollständigkeits-Theorem: Ein formales System ist entweder unvollständig oder widersprüchlich.[11]

Wir setzten die Turing-Maschine und ihre prinzipielle Möglichkeit, alle Aussagen eines formalen Systems zu berechnen als gegeben voraus[12]. Dann folgt, dass jede Aussage als Programm für eine Turin-Maschine niedergeschrieben werden kann. Alle möglichen, *korrekten* Programme[13] für die Turing-Maschine können nun in einer Liste sortiert werden. Dabei beschränken wir uns auf Programme, die genau eine Zahl (Symbolkette) als Input haben. Wohlgemerkt sind das *alle* Programme, wobei nicht jedes Programm einer Aussage entsprechen muß, aber alle Aussagen durch mindestens ein Programm abgebildet werden.

Wenn wir nun wissen wollen, ob eine bestimmte Funktion $f(n)$ für alle möglichen Eingaben n berechenbar ist, starten wir die Turing-Maschine für jedes n und warten ab, ob sie im Endzustand stehen bleibt. Was aber wenn die Turing-Maschine nicht anhält? Es lässt sich von Aussen nicht feststellen, ob eine Turing-Maschine ewig weiter läuft (in einer Schleife gefangen ist) oder ob nur noch länger zu warten ist, bis das Ergebnis feststeht. Es wird also eine Methode benötigt, die analysiert, ob ein Programm anhält, oder nicht.

Nehmen wir an, es gibt ein Verfahren, das für ein beliebiges Programm angibt ob es anhält. Dann könnte festgestellt werden, ob die zum Programm gehörende Funktion berechenbar ist, und weiters, dass (so es sie gibt) die entsprechende Aussage im formalen System ableitbar ist. Mithilfe dieses Verfahrens wäre es also möglich, alle Funktionen aus der Liste zu streichen, die nicht berechenbar sind. Es ist also möglich eine Liste *aller berechenbaren* Funktionen aufzustellen. Schreiben wir diese Liste nieder[7, 2.3 Berechenbarkeit und Turings Halteproblem]: in die erste Zeile die erste Funktion $f_1(n)$ mit allen möglichen Eingaben[14], in der zweiten Zeile die zweite Funktion f_2, etc.:

- $f_1(1),\ f_1(2),\ f_1(3),\ \ldots$
- $f_2(1),\ f_2(2),\ f_2(3),\ \ldots$
- $f_3(1),\ f_3(2),\ f_3(3),\ \ldots$
- $\ldots$

Auf diese Liste wird jetzt Cantors Diagonalverfahren angewendet: Wir wählen pro Zeile ein Element entlang der Diagonale, und addieren zu jedem Element 1: $f_1(1)+1,\ f_2(2)+1,\ f_3(3)+1, \ldots$. Diese Diagonalisierung gibt eine neue Funktion f_D an, die nicht in der Liste *aller* berechenbarer Programme vorkommt, da sich ihre Ergebnisfolge von allen anderen unterscheidet. f_D kann nicht in der ersten Zeile stehen, da sich das Ergebnis für $f_D(1)$ von $f_1(1)$ um 1 unterscheidet, auch nicht in der Zeile von f_2, usw. Es handelt sich um eine neue Funktion, die nicht in der *kompletten* Liste vorkommt. Also gibt es keine Möglichkeit eine solche Liste zu generieren, da immer die Diagonalfunktion erzeugt werden kann, die dann nicht in der Liste enthalten und somit nicht berechenbar ist.

Was aber, wenn wir eine Liste aller Funktionen aufstellen, auch solcher die nicht berechnet werden können? Wir erzeugen eine erweiterte Diagonalfunktion: Wenn das Element $f_n(n)$ aus der Liste nicht berechenbar ist, soll das erweiterte Diagonalelement gleich 1 sein. Auch die erweiterte Diagonalfunktion ist nicht in der Liste enthalten, sie unterscheidet sich ebenfalls von jeder gelisteten Funktion an mindestens einer Stelle. Entweder in der gleichen Art wie vorher, oder sie gibt 1 für undefinierte $f_n(n)$ aus. Es kann also durch das Erzeugen der Diagonalfunktion immer ein nicht berechenbares Programm erzeugt werden. Über dieses Programm kann die (hypothetische, aber allgemeine) Methode, die feststellt, ob die Maschine hält, nichts aussagen, da sie nicht in der kompletten Liste vorkommt.

Hält die Turing-Maschine für die neu erzeugte, erweiterte Diagnoalfunktion? Nein, denn um die Diagnoalfuntion zu erzeugen, müssen auch die undefinierten $f_n(n)$ analysiert werden, aber genau wenn die Funktion undefiniert ist hält die Maschine nicht. Wir haben eine korrekte Funktion definiert (sie setzt sich aus einer einfachen Abwandlung der Liste aller korrekten Funktionen zusammen), für die es keinen allgmeinen Haltetester gibt. Das Halteproblem ist unentscheidbar. Damit hat Turing die Unentscheidbarkeit des Halteproblems bewiesen. Nachdem das Halteproblem eine Variante des Entscheidungsproblems ist, führt dies zu der Konsequenz, dass ein formales System entweder unvollständig oder widersprüchlich ist.

Es ist anzumerken, dass das Halteproblem nicht mit Hilfe einer Turing-Maschine entscheidbar ist. Ebenso ist der Unvollständigkeitssatz nicht im formalen System[15]

[10] Übersetzung des Autors

[11] genau genommen nur eine schwächere Form des Unvollständigkeits-Theorems

[12] Turing zeigt in seiner Arbeit, dass beides möglich ist

[13] im formalen System wohlgeformten Aussagen

[14] Hier als natürliche Zahlen, es können aber beliebige Symbolketten als Eingabe definiert werden.

[15] der Principia Mathematica

entscheidbar. Es gibt in beiden Fällen mächtigere Systeme (bzw. Maschinen), welche die Probleme lösen können. Aber diese wären für sich genauso ungenügend, d.h. für eine mächtigere Maschine, welche das Halteproblem für Turing-Maschinen löst, kann wieder ein Haltetester und ein Haltetestverderber geschrieben werden.

Hilberts Vision einer komplett formalisierten Mathematik muss also scheitern. Turing hat dies anschaulich bewiesen und sich dabei auf die Ergebnisse Gödels gestützt. Die Beweisidee folgt in weiten Teilen der des Unvollständigkeitssatzes, der allerdings wesentlich formaler geführt wurde. Der oben skizzierte Beweis sollte aber aufgrund der Ähnlichkeit helfen, Gödels Ausführungen nachzuvollziehen.

4 GÖDELS UNVOLLSTÄNDIGKEITS-THEOREM

Kurt Gödel bewies in den "Monatsheften für Mathematik und Physik" [8], dass die Principia Mathematica nicht beides, sowohl widerspruchsfrei als auch vollständig sein kann. Gödel gelang es zu zeigen, dass der Beweis nicht nur für die Principia Mathematica gilt, sondern auch das Zermelo-Fraenkelsche (das System der Arithmetik natürlicher Zahlen mit Addition und Multiplikation) und damit dem arithmetischen System, das aus den Peano-Axiomen entsteht, und daraus abgeleiteten Systemen[16]. Zudem greift Gödel nur auf ganze Zahlen zurück, was die Auswirkung des Satzes auf die Mathematik umso verheerender macht.

4.1 ÜBER FORMAL UNENTSCHEIDBARE SÄTZE

Gödels Unentscheitbarkeits-Theorem brachte im Jahr 1931 die Bemühungen der Mathematik, logische Paradoxa durch Formalisierung auszuschließen zu Fall. Ihm gelang es, das Paradoxon zu formulieren, dass ein Satz innerhalb formaler Systeme, die mächtig genug sind grundlegende Arithmetik abzubilden, nicht beweisbar ist, obwohl er wahr ist[17]. Wir wollen in Folge näher auf den Beweis eingehen und ihn analysieren. Dazu wird er in drei Teile zerlegt: Zunächst die Gödelisierung der Formeln der Principia Mathematica, anschließend die Diagonalisierung, mittels welcher eine neue Formel erzeugt wird, und zuletzt der Beweis der Unentscheidbarkeit. Dabei stützen wir uns auf sie Beweisskizze wie sie Gödel beschreibt [8] und verwenden auch weitest gehend dieselbe Notation, sodass ein Nachvollziehen anhand der Originalschrift möglich ist.

[16] Abgeleitetes System: dem System wird eine (endliche) Zahl von Axiomen hinzugefügt

[17] abgeschwächtes Unvollständigkeits-Theorem

4.1.1 Gödelisierung

Dies ist Gödels erster Schritt, der so bedeutsam ist, dass er seinen Namen bekam: die Gödelisierung. Er bildet das gesamte System der Principia Mathematica isomorph (umkehrbar eindeutig) auf die natürlichen Zahlen ab, es werden also Aussagen (Formeln) des Systems im System selber dargestellt! Das heißt, Formeln werden durch natürliche Zahlen repräsentiert. Und da das formale System der Principia Mathematica Aussagen über natürliche Zahlen trifft, ist es jetzt in der Lage, Aussagen über sich selbst zu treffen.

75 Jahre später, aus der Sicht des Informatikers, ist dieses Verfahren beinahe schon intuitiv: Es entspricht einer Codierung der Zeichen des formalen Systems, z.B. in ASCII-Code. Jedem ASCII-Codezeichen kann eine natürliche Zahl zugeordnet werden. Eine Folge von Zahlen kann kodiert werden, sodass das Ergebnis wieder nur eine Zahl ist. Dabei muss die Kodierung immer umkehrbar bleiben!

Wir vollziehen Gödels ursprüngliches Verfahren nach: Er weißt zunächst jedem Zeichen aus dem formalen System der Principia Mathematica eine natürliche Zahl zu. Da Formeln als Symbolketten dargestellt werden, entspricht eine Formel einer Folge von natürlichen Zahlen. Auch dieser Zahlfolge ordnet Gödel wieder eine (umkehrbar) eindeutige Nummer zu: Die Gödelisierung weist der Reihe nach der n-ten Stelle der Zahlfolge die n-te Primzahl (in der Folge der Primzahlen) zu, die mit dem Zahlwert der n-ten Stelle exponiert wird: $gd(x_1 x_2 x_3 \ldots x_n) = 2^{x_1} \cdot 3^{x_2} \cdot \cdots \cdot p_n^{x_n}$. Schlussendlich ist ein Beweis nun wiederum durch eine Folge von natürlichen Zahlen darstellbar, und auch diese Folge ist wiederum gödelisierbar.

Wichtig ist dabei zu zeigen, dass die Beweise und Formeln rekursiv definiert werden können. Dies nimmt einen guten Teil in Gödels Originalbeweis ein.

Mittels Primfaktorenzerlegung lässt sich jede der gödelisierten Symbolketten wieder auflösen: aus einer Zahl wird eine Folge von Folgen von natürlichen Zahlen, die Symbolkette, die einen Beweis wiedergibt.

4.1.2 Diagonalisierung

Dieser Schritt ist das Wesentliche an Gödels Beweis. Für die weiteren Ausführungen beschränkt sich Gödel auf Formeln der Principia Mathematica mit genau einer freien Variablen aus den natürlichen Zahlen und bezeichnet diese als Klassenzeichen.

Alle Klassenzeichen werden nun (beliebig) sortiert und durchnummeriert, das n-te Klassenzeichen (die n-te Formel) wird mit $R(n)$ bezeichnet. Dabei werden alle(!) Klassenzeichen sortiert, das ist für die Diagonalisierung essentiell.

Zudem wird nun ein Klassenzeichen, welches die Beweisbarkeit ausdrückt, formuliert: $\overline{Bew}[R(x);\ y]$ bedeutet, dass das in der Liste an der Stelle x stehende Klassenzeichen, wenn die Variable y eingesetzt wird, nicht beweisbar, d.h. im System nicht ableitbar ist. Dabei ist $\overline{Bew}[R(x);\ y]$ wiederum ein Klassenzeichen des Systems, sprich: eine Formel mit einer freien Variable vom Typ der natürlichen Zahlen und damit in unserer Liste enthalten.

Nun definiert Gödel eine Klasse[18] natürlicher Zahlen: $n \in K \equiv \overline{Bew}\,[R(n);\ n]$. In Worten: wenn n zur Klasse K gehört, ist das Klassenzeichen $R(n)$ mit n als Variable nicht beweisbar. Die Klasse K macht also eine Aussage über ein Klassenzeichen, wenn die ihm zugeordnete Gödelnummer eingesetzt wird (Diagonalisierung). In gewissem Maße macht K eine Aussage über sich selbst, da die Gödelnummer nur eine andere Darstellung der Formel ist. Da die Begriffe über welche K definiert ist, alle in der Principia Mathematik definierbar sind, ist dies auch der zusammengesetzte Begriff K [8, S. 175].

Jetzt führt Gödel die Formel S ein, die nichts anderes besagt, als dass die natürliche Zahl n zu K gehört, $S(n) \equiv n \in K$[19]. S ist aber wiederum ein Klassenzeichen und kommt somit in unserer Liste aller Klassenzeichen vor, und zwar an der Stelle q: $S = R(q)$. In Worten sagt $S(n)$ aus, dass das Klassenzeichen $R(n)$, wenn dessen Gödelnummer n als Argument verwendet wird, nicht beweisbar ist.

Nun konstruiert Gödel die *Diagonalformel*: Er setzt q in S ein: $S(q) \equiv q \in K$. q ist aber gerade die Gödelnummer von S! Ausgeschrieben kann dies interpretiert werden als: Wenn $S(q)$ gilt, dann ist $S(q)$ mit q als Variable nicht beweisbar! Wenn S durch das dementsprechende $R(q)$ ersetzt wird, gelangen wir zur Diagonalformel $R(q) \equiv \overline{Bew}\,[R(q); q]$, auf deren Grundlage die Unentscheidbarkeit bewiesen wird.

4.1.3 Unentscheidbarkeit

Analysieren wir $[R(q);\ q] \equiv \overline{Bew}\,[R(q);\ q]$: Wenn $[R(q);\ q]$ gilt, ist $[R(q);\ q]$ nicht beweisbar. Dies wird oft als Aussage der Art "Ich bin nicht beweisbar" gedeutet und erinnert uns ein letztes Mal an Epimenides und die Lügen der Kreter. Exakter formuliert heißt es "Wenn ich gelte, bin ich nicht aus dem System ableitbar". Die Beweisbarkeit der Formel $[R(q);\ q]$ ist nun nicht entscheidbar, dazu untersuchen wir zwei Fälle:

1. Wenn der Satz $[R(q);\ q]$ beweisbar ist, dann wäre er auch richtig und widerspricht der Definition, dass$[R(q);\ q]$ nicht beweisbar ist. Das formale System ist also widersprüchlich.
2. Ist $[R(q);\ q]$ allerdings nicht beweisbar (ist dessen Negation beweisbar), dann gälte $Bew\,[R(q);\ q]$. $[R(q);\ q]$ wäre also ebenso wie seine Negation beweisbar, was unmöglich ist.

Wenn der Satz $[R(q);\ q]$ nicht beweisbar ist, ist es dessen Negation? Gödel führt in seinem Beweis aus, dass in einem widerspruchsfreien System[20] auch die Negation nicht bewiesen werden kann: Also ist weder $[R(q);\ q]$ noch dessen Negation beweisbar. Das System ist unvollständig.

Auch hier gilt: Die Unentscheidbarkeit kann nur im System nicht entschieden werden, durch metamathematische Überlegungen - außerhalb des Systems - ist sofort klar, dass $[R(q);\ q]$ wahr ist, da er ja seine eigene Unbeweisbarkeit behauptet.[8, p. 176]

Werden diese Ableitungen im formalen System geführt, sind sie direkt verständlich. Vorraussetzung ist ein formales System, dass gemäß der Forderung vollständig und widerspruchsfrei ist. Wir fordern auch die Korrektheit: alles was ableitbar ist, soll wahr sein.

- WSF (Widerspruchsfreiheit): $\overline{(Bew(A) \land Bew(\overline{A}))}$
- VOL (Vollständigkeit): $Bew(A) \lor Bew(\overline{A})$
- KOR (Korrektheit): $Bew(A) \rightarrow A$

Es gelang die Diagonalformel $[R(q);\, q] \equiv \overline{Bew}\,[R(q);\, q]$, herzuleiten. Bezeichen wir sie als Formel I und untersuchen sie: Aufgrund der Vollständigkeit gilt entweder $a)\quad Bew\,[R(q);\ q]$ oder $b)\ Bew\overline{[R(q);\ q]}$.

$$
\begin{array}{rll}
a)\ Ang.\ Bew\,[R(q);\ q] & \xrightarrow{KOR} & \\
{[R(q);\ q]} & \xrightarrow{I} & \\
\overline{Bew}\,[R(q);\ q] & Widerspruch & zur\,WSF
\end{array}
$$

$$
\begin{array}{rll}
b)\ Ang.\ Bew\overline{[R(q);\ q]} & \xrightarrow{KOR} & \\
\overline{[R(q);\ q]} & \xrightarrow{I} & \\
\overline{\overline{Bew}}\,[R(q);\ q] & \longrightarrow & \\
Bew\,[R(q);\ q] & Widerspruch & zur\,WSF
\end{array}
$$

Da weder $a)$ noch $b)$ gilt (beide führen zu einem Widerspruch zur Vorraussetzung) ist die Vollständigkeit verletzt. Gehen wir davon aus, daß unser System nicht Vollständig ist, ist auf jeden fall die Wiederspruchsfreiheit verletzt. So hat Gödel also seinen Unvollständigkeitssatz formuliert:

[18] Der Begriff der Klasse ist mit dem der Menge vergleichbar, für die weiteren Betrachtungen ist der genaue Unterschied nicht relevant.

[19] S(n) “*ist nur eine metamatematische Beschreibung des unentscheidbaren Satzes*”[8, S. 175] Die im diesem Text folgenden Formeln in welchen S vorkommt (ausgenommen $S = R(q)$), sind daher mathematisch nicht korrekt, dienen aber der Verständlichkeit.

[20] genauer: ω-Widerspruchsfreiheit, über welche auch Gödel den Beweis führt.

“*Jedes hinreichend mächtige formale System ist entweder widersprüchlich oder unvollständig.*”

5 Conclusio

Was beutet nun, dass ein System nicht widerspruchsfrei und vollständig gleichzeitig sein kann? Zunächst einmal, dass die Mathematik nicht so formalisiert werden kann, wie Hilbert es forderte. Das eng verwandte Entscheidungsproblem sagt zudem aus, dass Aussagen nicht mechanisch auf ihre Wahrheit überprüfbar sind. Das heißt aber nicht, dass die Mahtematik zu falschen Ergebnissen kommt. Und das Unvollständigkeits-Theorem bringt auch die Principia Mathematica oder auf die arithmetischen Axiome aufbauende Systeme nicht zu Fall, denn sie haben sich als gute Grundlage um Mathematik zu betreiben, erweisen.

Man könnte soweit gehen, und Gödels Satz als Rechtfertigung ansehen, unbewiesene Aussagen, sprich Hypothesen, so sie sich nützlich erweisen, als Grundlage für weitere mathematische Erkenntnisse zu nutzen.[7, Kapitel 5.7 Unentscheidbarkeit und Wahrheit] Mathematik ist aus philosophischer Sicht der Physik nähergerückt und nicht die Wissenschaft, die sich aus sich selber (mit Hilfe einer weniger Axiome) erschafft. Zumindest nicht Widerspruchfrei und Vollständig.

Literatur

[1] Alfred North Whitehead and Bertrand Russell. *Prinicipa Mathematica*. Suhrkamp, 1984.

[2] David Hilbert. Mathematische Probleme. *Göttinger Nachrichten*, pages 253–297, 1900.

[3] wikipedia.de. Hilberts Liste von 23 mathematischen Problemen, 14 May 2008.

[4] Douglas R. Hofstadter. *Gödel, Escher, Bach: ein Endloses Geflochtenes Band*. Klett-Cotta, 17 edition, 2006.

[5] Alan Turing. On Computable Numbers, With an Application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society, Series 2*, 42:223–265, 1936.

[6] Grogory J. Chaitin. Grenzen der Berechenbarkeit. *Spektrum der Wissenschaft*, pages 86–93, February 2004.

[7] Jörg Resag. Die Grenzen der Berechenbarkeit, 14 May 2008.

[8] Kurt Gödel. Über formal unentscheidbare Sätze der Principa Mathematica und verwandter Systeme I. *Monatshefte für Mathematik und Physik*, (38):173–198, 1931.