

Gödels Unvollständigkeitssätze einfach erklärt

In den ersten Ausgaben von MATHEMATIQ habe ich über die Theorie der formalen Sprachen und Gödels Unvollständigkeitssätze geschrieben. Dabei habe ich gezeigt, wie Gödels Sätze logisch aus der Theorie der formalen Sprachen hergeleitet werden können. Wenngleich die Gedankengänge richtig sind, ist die Formulierung vielleicht für manche Leser zu knapp geraten, um von ihnen nachvollzogen zu werden. Deswegen versuche ich es noch einmal in leichter verständlicher Sprache.

Der große Mathematiker David Hilbert hatte in einem Mathematikerkongress zu Beginn des 20. Jahrhunderts gefordert, die mathematische Zunft möge sich bemühen, ein logisch konsistentes und vollständiges Axiomensystem zu entwickeln, aus dem die gesamte Mathematik hergeleitet werden könne. Kurt Gödels Leistung bestand hauptsächlich darin zu zeigen, dass es gar nicht möglich ist, dieses Vorhaben zu realisieren, weil ein formales System nicht logisch konsistent und zugleich vollständig sein kann (Erster Unvollständigkeitssatz).

Computerprogramme, die Entscheidungsprobleme lösen, sind formale Systeme im Gödelschen Sinn, und jedes formale System im Gödelschen Sinn kann durch ein Computerprogramm beschrieben werden. Für einen modernen Menschen mag Gödels Satz daher leichter verständlich sein, wenn er sich ein Computerprogramm denkt, das man mit mathematischen Aussagen füttern kann. Dieses Computerprogramm kann nur zwei mögliche Ausgaben liefern: "wahr" oder "falsch". Ein logisch konsistentes Computerprogramm dürfte nur dann "wahr" ausgeben, wenn die mathematische Aussage, die ihm als Parameter übergeben worden ist, tatsächlich wahr ist, und "falsch" auch nur, wenn die Aussage tatsächlich falsch ist. Wenn die mathematische Aussage aber weder wahr noch falsch, sondern paradox ist, dürfte das logisch konsistente Computerprogramm gar nichts ausgeben. Es würde also gar nicht terminieren, sondern in eine Endlosschleife geraten, sich "aufhängen", wie man umgangssprachlich sagt. Damit wäre dieses Computerprogramm aber nicht vollständig. Ein vollständiges Computerprogramm würde immer terminieren, egal mit welcher Eingabe man es fütterte. Es müsste also auch paradoxe Aussagen entweder als "wahr" oder als "falsch" klassifizieren, obwohl keine der beiden Alternativen zutrifft. Ein vollständiges Computerprogramm wäre also gezwungen, in einigen Fällen zu "lügen" - und wäre damit nicht logisch konsistent.

Ein Beispiel für eine paradoxe Aussage lautet: "Dieser Satz ist falsch." Wenn man annimmt, dass der Satz wahr ist, dann sagt er über sich selbst aus, dass er falsch ist. Somit kann er nicht wahr sein. Nimmt man aber an, dass er falsch ist, dann folgt, dass er wahr sein muss. Somit kann er nicht falsch sein. Nun mag man vielleicht sagen, das sei eine sprachliche Aussage, keine mathematische, aber im Prinzip sind auch sprachliche Aussagen dieser Art mathematische Aussagen. Außerdem hat Kurt Gödel einen Formalismus entwickelt, mit dem er zeigen konnte, dass auch in der Mathematik im engeren Sinn solch paradoxe Aussagen auftreten können.

Das wäre der erste Unvollständigkeitssatz gewesen. Der zweite besagt, dass ein

logisch konsistentes System nicht in der Lage ist, seine eigene logische Konsistenz zu beweisen. Wie beweist man diese Aussage? Man stelle sich ein logisch konsistentes Computerprogramm vor, also eines, das in dem Fall, dass es nicht imstande ist zu sagen, ob die Aussage, die ihm als Parameter übergeben worden ist, wahr oder falsch sei, "ehrlich" ist und die Konsequenzen zieht, also sich "aufhängt". Wie sollte ein solches Programm beweisen, dass es "ehrlich" ist? Es gibt nur eine einzige Möglichkeit: sich selbst mit einem Parameter aufzurufen, der dazu führt, dass sich das Programm "aufhängt". Das Programm ist jedoch nicht in der Lage festzustellen, dass sich die von ihm aufgerufene Instanz seiner selbst aufgehängt hat! Somit kann es in diesem Fall auch selbst nicht terminieren.

Ich hoffe, die Gödelschen Unvollständigkeitssätze nun nachvollziehbar erklärt zu haben.

Claus D. Volko, cdvolko@gmail.com