Turing, Church, Gödel und Rechenmaschinen

Ege Girit

Diese Arbeit behandelt die theoretischen Grundlagen heutiger Computer, die von Alan Turing, Alonzo Church und Kurt Gödel entwickelt wurden. Im Fokus stehen die Turingmaschine, die Church-Turing-Hypothese, die Gödelsche Unvollständigkeit und das Halteproblem. Es werden die Gemeinsamkeiten und Beiträge dieser drei Wissenschaftler untersucht und ihre Beziehungen zueinander erklärt.

Alan Turing, Alonzo Church und Kurt Gödel waren bedeutende Mathematiker und Logiker des 20. Jahrhunderts. Obwohl sie unterschiedliche Ansätze verfolgten, hatten sie gemeinsam, dass sie sich intensiv mit den Grundlagen der Berechenbarkeit und Logik auseinandersetzten. Ihre Arbeiten legten den Grundstein für das Verständnis von Rechenmaschinen und formalen Systemen.

Alonzo Church war ein Mathematiker und er ist bekannt geworden für seine Entwicklung des Lambda-Kalküls, welches eine formale Sprache ist, mit der man Funktionen untersuchen kann. Er hat ausserdem die Existenz von unentscheidbaren Problemen gezeigt, die mathematisch nicht berechenbar sind. Church hatte ein Student namens Alan Turing, und die Erfindungen von Church hat Turing zum Überlegen gebracht, ob das Halteproblem auch unentscheidbar sei.

Die Turingmaschine, von Alan Turing im Jahr 1936 eingeführt, ist ein abstraktes mathematisches Modell einer Rechenmaschine. Sie diente als theoretisches Konzept für das Verständnis von Berechenbarkeit.

Church und Turing haben gemeinsam herausgefunden, dass die Turingmaschine von Turing und das Lambda-Kalkül von Church die gleiche Ausdruckskraft haben. Sie haben bemerkt, dass es noch weitere äquivalente Modelle zum Berechnen von Funktionen gibt. Ein weiteres äquivalentes Beispielmodell ist das Konzept der GOTO-Programme. GOTO-Programme können als abstrakte Repräsentation von Turingmaschinen dienen. Jedes GOTO-Programm kann in eine äquivalente Turingmaschine umgewandelt werden und umgekehrt. Die Äquivalenz von GOTO-Programmen zu Turingmaschinen kann durch den Satz "Jede WHILE-berechenbare Funktion ist auch Turing-berechenbar" (1) bewiesen werden.

Im Laufe der Zeit wurden unterschiedliche Berechnungsmodelle entwickelt, man hat aber festgestellt, dass sie entweder weniger stark als die Turingmaschinen sind oder gleich stark sind. Daher besagt die Church-Turing-These, dass die Turingmaschinen den Begriff der Berechenbarkeit schildern.

Mit Jacques Herbrand erstellte Kurt Gödel die formale Definition der Klasse der allgemein rekursiven Funktionen, was wiederum gleichmächtig zu Turingmachinen ist. Er hat 1929 die Vollständigkeit des Funktionenkalküls bewiesen und damit war es möglich, ein formales System zu konstruieren, in dem jede allgemeingültige prädikatenlogische Formel erster Stufe in endlich vielen Schritten aus den Axiomen ableitbar sind. Mit den rekursiven Funktionen beschreibt man nicht die einzelnen Schritte des Rechenverfahrens, sondern die Lösungen von Problemen. Dieses Berechnungsmodell ist daher ähnlich wie die funktionalen Programmiersprachen. Die Gödel'schen Unvollständigkeitssätze sind sehr wichtig für die Beweistheorie. Aus seinen Unvollständigkeitsergebnissen hatte dann Gödel Interesse an einer genauen Definition der Berechenbarkeit, um die Allgemeinheit seiner Ergebnisse festzustellen. Alonzo Church hatte Gödel vorgeschlagen, die berechenbaren Funktionen als die Lambda-Funktionen zu definieren. Gödel war nicht überzeugt von Church's Idee und bezeichnete den Vorschlag als völlig unbefriedigend (2). Er hat aber nach wenigen Jahren diesen Vorschlag doch akzeptiert, weil die Arbeit von Turing seine Meinung geändert habe. Laut Gödel hat Turing eine genaue Definition eines formalen Systems geliefert. B. Jack Copeland et. al (2) sehen die Ansichten von Turing und Gödel normalerweise als Kontrast zueinander. Die Diskussion zwischen Turing und Gödel war es, ob Maschinen niemals Mathematiker ersetzen können, oder allgemeiner, ob der Mensch jede Maschine übertrifft. Im Allgemeinen nimmt B. Jack Copeland et. al (2) an, dass Gödel der Meinung ist, dass der Mensch mächtiger als die Maschine sei, während Turing die gegenteilige Position eingenommen haben soll.

Das Halteproblem beschäftigt sich mit den Terminierungseigenschaften von Turing-Maschinen. Es geht um die Frage, ob auf algorithmischem Wege entschieden werden kann, ob eine Turing-Maschine unter bestimmten Eingaben terminiert oder endlos weiter rechnet. Das erste Hauptresultat von Turing stellt im Grunde die Unlösbarkeit des Halteproblems dar. Das Halteproblem ist unlösbar, es kann also keinen Algorithmus geben, der dieses Problem entscheidet. Alan Turing bewies diese Unentscheidbarkeit im Jahr 1936. Der Beweis basiert auf der Annahme, dass es eine Turingmaschine geben würde, die das Halteproblem entscheidet, und zeigt, dass dies zu einem Widerspruch führen würde (3).

Zusammenfassend haben Alan Turing, Alonzo Church und Kurt Gödel mit ihren Arbeiten die theoretischen Grundlagen der heutigen Computer gelegt. Die Turingmaschine stellt ein fundamentales Modell für Berechnungen dar, während die Church-Turing-Hypothese besagt, dass jede berechenbare Funktion von einer Turingmaschine berechnet werden kann. Die Gödelsche Unvollständigkeit zeigt, dass es in einem hinreichend mächtigen formalen System Aussagen gibt, die weder bewiesen noch widerlegt werden können. Das Halteproblem verdeutlicht die Grenzen der Berechenbarkeit, da es unentscheidbar ist, ob eine Turingmaschine bei einer gegebenen Eingabe anhalten wird oder nicht.

Literatur

1. Markus Nebel. Formale Grundlagen der Programmierung. 2012. ISSN 2522-0640. doi: doi.org/10.1007/978-3-8348-2296-3.
2. B. Jack Copeland, Carl J. Posy, and Oron Shagrir. Computability: Turing, Gödel, Church, and Beyond. *Project MUSE*, 2016.
3. Dirk W. Hoffmann. Grenzen der Mathematik - Eine Reise durch die Kerngebiete der mathematischen Logik. 2013. doi: doi.org/10.1007/978-3-642-34720-7.