Theoretische Informatik für Wirtschaftsinformatik und Lehramt

Universelle Turingmaschinen und Church'sche These

Priv.-Doz. Dr. Stefan Milius
stefan.milius@fau.de

Theoretische Informatik
Friedrich-Alexander Universität Erlangen-Nürnberg

SS 2016

Gliederung

Worum geht es in diesem Abschnitt? (I)

Bislang: Berechnungfähigkeit arithmetischer Funktionen

$$f : \mathbb{N}^k \to \mathbb{N}$$

durch:

- Turing-Maschinen
- LOOP- und WHILE-Programme

Hauptergebnis jetzt:

Berechenbarkeit durch folgende Algorithmenmodelle ist gleichwertig

- Turing-Maschinen,
- WHILE-Programme und
- μ-rekursive Funktionen

$\rightsquigarrow$ Church'sche These

Alonzo Church

Amerikanische Logiker und Mathematiker

Wichtige Beiträge zu:

- mathematischer Logik
- Grundlagen der theoretischen Informatik

Bekannte Entdeckungen:

- λ-Kalkül,
- Unentscheidbarkeit von Hilberts *Entscheidungsproblem*,
- Church-Rosser-Satz

Alonzo Church (1903–1995)
Quelle: Wikipedia

Church'sche These (auch Church-Turing-These)

Die Klasse der intuitiv berechenbaren Funktionen stimmt mit der Klasse der Turing-berechenbaren Funktionen überein.

Lernziele

- universelle Turingmaschinen kennen und erklären
- die unberechenbare Busy-Beaver-Funktion kennen und erklären
- die Church'sche These kennen und erklären

Universelle Turingmaschine

Turing-Maschinen versus WHILE-Programme

Wiederholung:

Satz 7.6

Jede WHILE-berechenbare Funktion ist Turing-berechenbar.

Jetzt:

Satz 8.1

Jeder Turing-berechenbare Funktion ist WHILE-berechenbar.

- Schlüssel zum Beweis: Turing-Maschinen und ihre Rechnungen als natürliche Zahlen kodieren
- Kodierungstechnik bekannt als **Gödelisierung** (nach dem österreichischen Mathematiker Kurt Gödel)

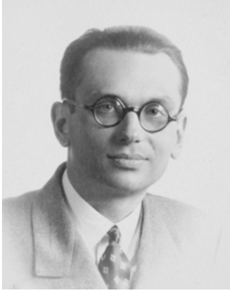
Kurt Gödel

Österreichischer Logiker, Mathematiker und Philosoph

- Einer der wichtigsten Logiker überhaupt.
- großer Einfluss auf wissenschaftlich-philosophisches Denken.

Bekannte Entdeckungen:

- Unvollständigkeitssätze,
- Konsistenz des Auswahlaxioms und der Kontinuumshypothese mit den Axiomen der Mengenlehre



Kurt Gödel (1906–1978)
Quelle: Wikipedia

Gödels 1. Unvollständigkeitssatz

Jedes konsistente formale System, dass genug Aussagen über natürliche Zahlen ermöglicht ist *unvollständig*, d.h. es gibt wahre Aussagen, die in der Sprache des Systems ausdrückbar aber **nicht** beweisbar sind.

Gödelnummer

Betrachte nur TM mit $\Sigma = \{0, 1\}$:

$$M = (Z, \{0, 1\}, \Gamma, \delta, z_0)$$

wobei OBdA $\quad Z \subseteq \{z_0, z_1, z_2, \ldots\}, \qquad \Gamma \subseteq \{a_0, a_1, a_2, \ldots\}$
(Mengen $\{z_0, z_1, z_2, \ldots\}$ und $\{a_0, a_1, a_2, \ldots\}$ fest vorgegeben)

Elemente von Z und Γ sind genau die in δ vorkommenden z_i bzw. a_j.

Jedem Übergang in δ

$$(z_i, a_j) \mapsto (z_{i'}, a_{j'}, X_s) \qquad (\text{wobei } X_0 = L, X_1 = R, X_2 = N)$$

wird der Code $\qquad \text{code}_r = 0^i 1 0^j 1 0^{i'} 1 0^{j'} 1 0^s \qquad$ zugeordnet

$\text{code}_1, \ldots, \text{code}_n$ seien alle derartigen Codes. Dann ist

$$c(M) = 111\text{code}_1 11\text{code}_2 11 \cdots \text{code}_n 111$$

die Gödelnummer von M. (Beachte: $c(M)$ Binärzahl.)

Universelle Turing-Maschine

Satz 8.2

Es gibt eine Turing-Maschine M_u über $\Sigma = \{0, 1\}$ mit
Eingabe: TM-Code und Eingabewort $c(M)w$
Verhalten: M_u verhält sich wie M auf Eingabe w

Man nennt M_u die universelle Turing-Maschine.

Bemerkungen:

- es gibt auch ein universelles WHILE-Program
  (für Eingaben über $\Sigma = \{|, \circ\}$; eine Art TM-Simulator)
- dieses Programm liefert dann den Beweis von Satz 8.1.
- Sätze 7.6 und 8.1 liefern: eine Funktion ist Turing-berechenbar genau dann wenn sie WHILE-berechenbar ist.

Eine unberechenbare Funktion

Fleißige Biber

Betrachte die Funktion $\beta : \mathbb{N} \to \mathbb{N}$ mit

$$\begin{array}{rcl} \beta(0) & = & 0 \\ \beta(n) & = & \text{größte Zahl } k\text{, die eine TM mit } n \text{ Zuständen auf} \\ & & \text{ein leeres Band schreiben und danach halten kann} \end{array}$$

TM beginnt mit leerem Band, schreibt $|^k$ und hält am Ende
Solche TM heißen fleißige Biber.

Einige (Schranken für) Werte von $\beta(n)$:

n	1	2	3	4	5	6
$\beta(n)$	1	4	6	13	≥ 4098	$> 3{,}514 \cdot 10^{18276}$

Satz 8.3 (Tibor Radó, 1962)

Die Busy-Beaver-Funktion β ist *nicht* Turing-berechenbar.

Eine unberechenbare Funktion

Church'sche These

- **Entwicklung der Berechenbarkeitstheorie:** Andere Ansätze zur Formalisierung des Berechenbarkeitsbegriffs wurden untersucht (einschließlich Chomsky-Grammatiken und RAM)
- alle diese Ansätze haben sich als gleichwertig erwiesen
- Rechfertigung für:

Church'sche These (auch Church-Turing-These)

Die Klasse der intuitiv berechenbaren Funktionen stimmt mit der Klasse der Turing-berechenbaren Funktionen überein.

- deshalb spricht man daher meist nur von *berechenbaren* statt von Turing-berechenbaren, WHILE-berechenbaren, μ-rekursiven, ...(usw.) Funktionen
- **Fazit:** Berechenbarkeit durch Turing-Maschinen umfasst alle algorithmischen Berechnungsmöglichkeiten

Konsequenz

Was ist zu tun, wenn gezeigt werden soll, dass eine Funktion f berechenbar ist?

Wir konstruieren (je nachdem, was für die Aufgabe am geeignetsten ist)

- eine Turing-Maschine, die f berechnet, oder
- ein WHILE-Programm, das f berechnet, oder
- eine μ-rekursive-Funktion für f.

Aus der Turing-Berechenbarkeit, der WHILE-Berechenbarkeit oder der μ-Rekursivität von f folgt dann die Berechenbarkeit von f.

Zusammenfassung