Grundbegriffe der mathematischen Logik

Vorlesung WS 2005/2006

Jakob Kellner
http://www.logic.univie.ac.at/~kellner

Kurt Gödel Research Center for Mathematical Logic

1. Vorlesung, 2005-10-05

Diagonalisierung: $2^{\aleph_0} > \aleph_0$

Satz (Georg Cantor Bild, 1874)

Es gibt mehr reelle Zahlen (0-1 Folgen) als natürliche Zahlen.

Das war eines der ersten Resultate der (naiven) Mengenlehre, eines der vier Gebiete der Mathematischen Logik.

Naheliegende Fragen: (ev. Übung)

- Wie ist "gleichviel", "mehr" etc. überhaupt definiert?
- Wieso gibt es gleichviel reelle Zahlen wie 0-1 Folgen?

2 zentrale Methoden der (elementaren) Logik:

- Diagonalisierung
- Kodierung, Gödel Nummerierung

Was ist Mathematische Logik?

- Mathematische Logik ist die mathematische (nicht philosophische) Disziplin der Metamathematik.
- In der naiven Mathematik
  beweist man Sätze, definiert Algorithmen, berechnet Funktionen.
  In der Logik fragt man:
  Was ist ein Beweis? Gibt es unbeweisbare Sätze?
  Welche Funktionen sind berechenbar? etc.
- Logische Methoden liefern auch neue naiv-mathematische Resultate.

Hilberts Programm

Eine Quelle der modernen mathematischen Logik:

Problem (besonders ab Jahrhundertwende bis ca 1940)

Beweise für finite Tatsachen (z.B. über natürliche Zahlen) verwenden "fragwürdige" Methoden: nicht-konstruktive/indirekte (Brouwer Bild), unendliche Mengen, AC...

Beispiel

- "Tertium non datur" $A \vee \neg A$: Es gibt irrationale r, s s.d. r^s rational.
  (Betrachte $x = \sqrt{2}$, x^x und $(x^x)^x$.)
- Kardinalität: Es gibt nicht-algebraische reelle Zahlen.
  (Algebraische Zahlen sind abzählbar.)
- Auswahlaxiom:
  (Banach Tarski) Zerlege Ball mit Radius 1 in 3 Teile, rotiere und verschiebe jeden Teil, resultiert in Ball mit Radius 2.

Hilberts Programm (Forts.)

Lösungsvorschlag von David Hilbert Bild, 1920

Suche Beweissystem T s.d.:

1. T formalisiert gesamte Mathematik,
2. es lässt sich mit finiten Mitteln beweisen:
   T ist konsistent (widerspruchsfrei), und T ist vollständig.

Ergebnis (Gödel Bild, 1929, 1931, Frege, Zermelo Bild u.v.A.)

1. ist bedingt gelungen (Vollständigkeitssatz, ZFC),
2. ist unmöglich (Unvollständigkeitssatz).

Zu 2: Im Hinblick auf Konsistenz ist aber das tertium non datur und AC "harmlos".

Die vier klassischen Gebiete der klassischen Logik

- Beweistheorie
- Rekursionstheorie
- Modelltheorie
- Mengenlehre.

1. Beweistheorie

Disziplin der Logik die am weitesten von naiver Mathematik entfernt und der Philosophie am nächsten ist, aus dem Hilbert-Programm entstanden. Untersucht Beweissysteme und Beweise.

Ursprüngliche Fragen:

- Wie kann man Konsistenz eines Beweissystems zeigen?
  Wie seine Stärke definieren bzw. berechnen?
- Wie kann man Beweise normalisieren, vergleichen, automatisieren?
- Was ist konstruktiver Gehalt eines Beweises, wie kann man ihn extrahieren?
- Alternative Logiken (Modallogik, Intuitionismus,...).

Ein Beispiel für naive Anwendung (eines der wenigen):

- Zahlentheorie: Schranke des Satzes von Roth (Luckhardt).

2. Mengenlehre

Eine universelle Theorie aller mathematischen Objekte, insbesondere die Theorie der Unendlichkeit.

Sie bietet:

- eine universelle Sprache für die Mathematik,
- eine (praktisch) universelle Axiomatisierung der Mathematik (ZFC),
- daher: besonders relevante Unabhängigkeitsresultate,
- allgemein mathematische Methoden und Konstruktionen (naive Mengenlehre, z.B. AC, unendliche Kombinatorik).

Ursprüngliche Fragen:

- Welche Gesetze (Axiome) gelten für allgemeine Mengen?
- Welche Struktur hat die Hierarchie der Unendlichkeiten?
- Welche Eigenschaften hat die Potenzmenge von $\mathbb{N}$?
- Wie sieht die Kombinatorik unendlicher Mengen aus?

2. Mengenlehre (Fortsetzung)

Beispiele für naive Resultate:

- Gowers' Dichotomie für Banachräume.
- Goodstein's Theorem
- Viele weitere Anwendungen in Ergodentheorie, Dynamische Systeme, Algebra, ...

Unabhängigkeitsresultate: Beispiele für unentscheidbare Sätze:

- Kontinuumshypothese (CH).
- Sei A_0 Borel, A_1 stetiges Bild von A_0, A_2 stetiges Bild vom Komplement von A_1. Dann ist A_2 Lebesgue-messbar.
- Set $f : \mathbb{R} \to \mathbb{R}$ beliebig. Dann gibt es eine nicht-Lebesgue-Null-Menge $A \subseteq \mathbb{R}$ und eine stetige Funktion $g : \mathbb{R} \to \mathbb{R}$ sodaß $f \restriction A = g \restriction A$.
- Hunderte weitere aus Algebra, Analysis, Maßtheorie, Dynamik,... (nicht z.B. Zahlentheorie).

3. Modelltheorie

Untersucht den Zusammenhang zwischen Theorien und ihren Modellen.

Ursprüngliche Fragen:

- Wie viele Modelle von T (einer bestimmten Größe) gibt es?
- Welche Modelle können in andere eingebettet werden?
- Welche spezielle Modelle (minimale, maximale,...) gibt es?

Besonders stark für bestimmte Theorien, z.B. algebraisch abg. Körper (vollständig), daher (→algebraischer Geometrie).

Beispiele für naive Resultate:

- Sei V ein komplexe algebraische Varietät, $f : V \to V$ injektiver Morphismus. Dann ist f surjektiv.
- p-adische Version von Manin-Mumford (Hrushovski).
- Viele weitere Resultate aus algebraischer Geometrie, Algebra,...

4. Rekursionstheorie

Die Theorie der Berechenbarkeit.
Dementsprechend nicht nur in Mathematik wichtig, sondern v.a. in theoretischer Informatik zentral.

Ursprüngliche Fragen:

- Was ist ein Algorithmus, ein Entscheidungsverfahren, ein (idealisierter) Computer?
- Äquivalent: Welche Probleme sind (effektiv) entscheidbar? Welche Funktionen berechenbar?
- Welche entscheidbaren Probleme sind schwer (langsam), welche, leicht (schnell) zu lösen?

Algorithmus, Berechenbarkeit, Entscheidbarkeit

Frage: Welche Funktionen sind berechenbar,
welche Probleme effektiv entscheidbar?

Einschränkung:

Wir beschäftigen uns vorerst nur mit natürlichen Zahlen,d.h.
mit Funktionen von $\mathbb{N}$ nach $\mathbb{N}$ (oder $\mathbb{N}^n$ nach $\mathbb{N}$),
und Problemen der Form

"Hat die natürliche Zahl n die Eigenschaft P?", bzw.
"Hat $\vec{n} \in \mathbb{N}^n$ die Eigenschaft P?"

Grund: Formal einfacher.
Äquivalent (leicht codierbar): $\mathbb{Q}$, beliebig lange Zeichenketten (Strings), etc.
$F : \mathbb{N}^n \to \mathbb{N}^m$ berechenbar, wenn jede Komponente $\vec{x} \mapsto [F(\vec{x})]_i$ berechenbar ist.
(Stetige) Funktionen $f : \mathbb{R} \to \mathbb{R}$ können auch mit Rekursionstheorie behandelt werden (→ deskriptive Mengenlehre), nicht in dieser Vorlesung.

Algorithmus im Mathematischen Alltag

Beispiele für Algorithmen in der Mathematik

- Euklidischer Algorithmus, berechnet ggT : $\mathbb{N}^2 \to \mathbb{N}$.
  Bsp: ggT$(6, 9) = 3$.
  (Problem: Einzige Pointe: Effizienz)
- Gauß Elimination (oder Determinante) entscheidet effektiv:
  Ist $n \times n$-Matrix M (mit natürlichen Koeffizienten) in $\mathbb{Q}$ invertierbar?
  Bsp: $M = \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix}$ ist invertierbar, weil $\det(M) = 2 \neq 0$.

$n \times n$-Matrix M entspricht $M \in \mathbb{N}^{n^2}$, d.h. "$(2, 0, 0, 1)$ ist invertierbar".
Natürlich auch möglich: Matrizen mit Koeffizienten aus $\mathbb{Q}$ (z.B.: rationale Zahlen als Paare von natürlichen Zahlen kodieren).

Eine informelle Definition der Berechenbarkeit

Definition (der Berechenbarkeit)

Eine Funktion $F : \mathbb{N} \to \mathbb{N}$ ist berechenbar, wenn es ein Computerprogramm gibt, das auf Input n den Output $F(n)$ liefert.

Unpräzise!? (Computer, Programmiersprache?)
Pointe: Robust! (Natürlich: beliebig großer Speicher etc.)

Gleicher Berechenbarkeits-Begriff für:

- (prädikativ:) C, Fortran, Pascal, (objektor.:) C++, Java, Smalltalk, (Interpret.:) Basic, Perl, (funktional:) Lisp, (logisch:) Prolog, ...
- (idealisiertes, 1-dim.) Papier, Bleistift, Radiergummi und einige primitive Anweisungen (Turingmaschine ▸ Bild),
- μ-Rekursion
  (induktive Definition der berechenbaren Funktionen),
- in Robinson's Q repräsentierbare Funktionen.

Ein Problem P ist also effektiv entscheidbar, wenn ein Computerprogramm die richtige Antwort auf das Problem geben kann.

Wichtig:

- Das macht nur Sinn wenn P eine "freie Variable" hat, formaler:
- P ist Eigenschaft von (Tupeln von) natürlichen Zahlen.

Bsp: Ist Matrix M invertierbar?
Sinnlose Frage:
"Ist Riemannsche Vermutung effektiv entscheidbar?"
Es gibt ja jedenfalls Computerprogramm das die R.V. richtig entscheidet, nämlich enweder `Print Ja` oder `Print Nein`. Wir wissen halt nicht welches Programm das richtige ist.
Sehr wohl sinnvoll ist die Frage:
"Ist R.V. in ZFC entscheidbar (d.h. beweisbar oder widerlegbar), oder nicht (wie ja z.B. CH)?"

Computermodell: Unbeschränkte Registermaschine

Unbeschränkte Registermaschine (URM):

- Hardware: Unendlich viele Register (Variablen) $R_0, R_1, \ldots$,
  jede kann eine (beliebig große) natürliche Zahl speichern.
- Programmiersprache: Basic-artig, nummerierte Programm-Zeilen,
  jede enthält eines von:

addiere 1 zu R_i	$R_i := R_i + 1$,
subtrahiere 1 (wenn möglich, sonst 0)	$R_i := R_i - 1$,
eine Sprung-Anweisung	`goto` m,
teste auf $= 0$	`if` $R_i = 0$ `goto` m,
liefere R_0 als Output	`return`

Ein Beispiel für ein URM Programm: Addition

Beispiel (Addition zweier Zahlen)

```
0  if R1 = 0 goto 4
1  R1 := R1 − 1
2  R0 := R0 + 1
3  goto 0
4  return R0
```

Berechnung bei Input (3, 2)

Zeit:	0	Zeile:	0	R_0 :	3	R_1 :	2	Input $(3,2)$ in R_0, R_1
Zeit:	1	Zeile:	1	R_0 :	3	R_1 :	2	
Zeit:	S	Zeile:	S	R_0 :	5	R_1 :	0	Output (R_0): 5

Satz

Addition ist berechenbar.

Beliebig großer Speicher, endliche Programme

Endlichkeit:

- Der Speicher ist nur potentiell unendlich (oder: beliebig groß). D.h. in einem Register kann eine beliebig große Zahl stehen.
  (→ Unterschied zu wirklichem Computer.)
- Programme sind (beliebig groß, aber) endlich! (Sonst wäre jede Funktion trivialerweise berechenbar.)

Primitiv rekursive Funktionen

Auch berechenbar: alle primitiv rekursive (prim.r.) Funktionen:

Grundfunktionen:

- Projektionen: $(x_1, \dots, x_i, \dots, x_n) \mapsto x_i$,
- Konstante Nullfunktionen: $(x_1, \dots, x_n) \mapsto 0$,
- Plus 1: $(x_1, \dots, x_i, \dots, x_n) \mapsto x_i + 1$.

Zusammensetzung/Einsetzung:

Wenn $f_1, \dots f_n : \mathbb{N}^m \to \mathbb{N}$ und $g : \mathbb{N}^n \to \mathbb{N}$ prim.r., dann ist $g(f_1, \dots, f_n)$ prim.r.

Primitive Rekursion:

Wenn $g : \mathbb{N}^n \to \mathbb{N}$ und $h : \mathbb{N}^{n+1} \to \mathbb{N}$ prim.r., dann f, wobei

$$f(n, \vec{p}) := \begin{cases} g(\vec{p}) & \text{wenn } n = 0, \\ h(f(n-1), \vec{p}) & \text{sonst.} \end{cases}$$

Primitiv rekursive Funktionen (Forts.)

Beispiel (für prim.r. Funktionen:)

- Konstante Funktion c: $f(n) = (0+1) + (0+1) + \dots + (0+1)$,
- Addition: $f(0, m) = m$, $f(n+1, m) = (f(n, m)) + 1$,
- Subtraktion: $f(0, m) = 0$, $f(n+1, m) = n - (m-1) = f(n, f(m, 1))$,
- Multiplikation: $f(0, m) = 0$, $f(n+1, m) = f(n, m) + m$,
- Charakteristische Funktion $\chi_{\{0\}}$: $f(0) = 1$, $f(n+1) = 0$,
- $(m, n) \mapsto m^n$: $f(0, 0) = \chi_{\{0\}}$, $f(n+1, m) = f(n, m) \cdot m$,
- Charakteristische Funktion der Primzahlen (Übung),
- n wird auf die n-te Primzahl abgebildet (Übung).

Satz

Alle prim.r. Funktionen sind URM berechenbar.

Diagonalisierung: Berechenbare, nicht-prim.r. Funktionen

Nicht alle berechenbaren Funktionen sind prim.r.
Grund:

Diagonalisierung

Gegeben eine effektive Aufzählung berechenbarer Funktionen.
Dann gibt es berechenbare Funktion nicht in diese Aufzählung.

Beispiel: Ackermann Funktion (berechenbar, aber nicht prim.r.)

$$A(m, n) := \begin{cases} n+1 & \text{wenn } m = 0, \\ A(m-1, 1) & \text{wenn } m > 0 \text{ und } n = 0, \\ A(m-1, A(m, n-1)) & \text{sonst.} \end{cases}$$

Partiell rekursive (p.r.) Funktionen: Endlosschleifen

Beispiel

```
0  if R1 = 0 goto 2
1  goto 1
2  return R1
```

Liefert Output nur bei Input 0.

Definition

- F ist partielle Funktion von $\mathbb{N}^n$ nach $\mathbb{N}$, wenn $F : A \to \mathbb{N}$ für ein $A \subseteq \mathbb{N}^N$.
- F ist partiell rekursiv (p.r.), wenn es ein URM Programm gibt daß auf Input $\vec{x}$ genau dann einen output liefert wenn $\vec{x} \in \mathrm{dom}(F)$ und in diesem Fall wird der Output $F(x)$ geliefert.

μ-rekursive Funktionen

Definition (der μ-rekursive Funktionen)

μ-rekursive (μ-r.) Funktion ist eine partielle Funktion $\mathbb{N}^n \to \mathbb{N}$

- Jede prim.r. Funktion ist μ-r.
- Sei $f : \mathbb{N}^{n+1} \to \mathbb{N}$ μ-r. Dann ist μf ebenfalls μ-r., wobei

$$\mu f(x_0, \dots, x_n) := \min\{m : f(x_0, \dots, x_n, m) = 0\}.$$

Satz

f ist p.r. $\leftrightarrow$ f ist μ-r.

Church'sche Bild "These"

Genau die URM-berechenbaren Funktionen sind (im "natürlichen Sinn") berechenbar.