Grundbegriffe der mathematischen Logik

Vorlesung WS 2005/2006

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

Kurt Gödel Research Center for Mathematical Logic

2. Vorlesung, 2005-10-12

Wiederholung der letzten Stunde: Einführung

- Mathematische Logik und naive Mathematik.
- Historische Quelle: Mißtrauen gegen indirekte, nichtkonstruktive Methoden, aktuell unendliche Mengen und AC.
- Das Hilbertprogramm: Finde Axiomatisierung T der Mathematik, zeige mit finiten Methoden daß T konsistent und vollständig ist.
- Erfolg: ZFC, Vollständigkeitssatz, (rel.) Konsistenz von AC und tertium non datur.
- Scheitern: Unvollständigkeitssatz.
- Die vier Gebiete der mathematischen Logik: Beweistheorie, Mengenlehre, Modelltheorie, Rekursionstheorie.

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

Satz (Georg Cantor ▸ Bild, 1874)

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

Definition (Kardinalitäten)

- $A \preceq B$ heißt: es gibt $f : A \to B$ injektiv.
- $A \cong B$ heißt: es gibt $f : A \to B$ bijektiv.
- $A \preceq^* B$ heißt: es gibt $f : B \to A$ surjektiv oder $A = \emptyset$.

Satz

- *$A \preceq B$ und $B \preceq A$ impliziert $A \cong B$ (Cantor-Schröder-Bernstein).*
- *$A \preceq^* B$ ist äquivalent zu $A \preceq B$ (AC).*
- *$A \preceq B$ oder $B \preceq A$ (AC).*

Auswahlaxiom (AC)

Definition (AC, Zermelo ▸ Bild, 1904)

Wenn $(A_i)_{i \in I}$ eine nichtleere Familie nichtleerer Mengen ist, dann gibt es eine Funktion ("Auswahlf.") f mit Definitionsbereich I s.d. $f(i) \in A_i$.

Äquivalent:

- Umformulierung: Wenn $I \neq \emptyset$, $A_i \neq \emptyset$ für $i \in I$, dann $\prod_{i \in I} A_i \neq \emptyset$.
- (Zorn'sches Lemma) Jede nichtleere Halbordnung, in der jede jede total geordnete Teilmenge eine obere Schranke hat, hat ein maximales Element.
- Wohlordnungssatz: Jedes $A \neq 0$ läßt sich wohlordnen, d.h.: es gibt lineare Ordnung $\leq$ auf A mit: jede Teilmenge von A hat kleinstes Element.

WO wichtig für Induktion!
Bsp für WO: $(\mathbb{N}, <)$. Keine WO: $(\mathbb{Q}, <)$. Aber: $\mathbb{Q}$ leicht WO-bar (wie?)
Frage: Gibt es WO auf $\mathbb{R}$?

AC: Folgerungen

Nützliche Anwendungen:

- Lin.Alg.: Jeder Vektorraum hat Basis (äquivalent, ← Zorn).
- Algebra: Jeder Ring$\neq \{1\}$ mit 1 hat maximales Ideal (← Zorn).
- Topologie: Das Produkt kompakter top.R. ist kompakt.
- Funktionalanalysis: Hahn Banach (← Zorn).
- Mengenlehre: $A \preceq B$ oder $B \preceq A$ (← Wohlordnung).
- M.L.: Abzählbare Vereinigung endlicher Mengen ist abzählbar.

(Manche konkrete Instanzen der Sätze brauchen natürlich kein AC.)

Weniger nützliche Anwendungen (Maßtheorie):

- Es gibt nicht meßbare Teilmengen von $\mathbb{R}$. Sogar:
- (Banach Tarski) Zerlege Ball mit Radius 1 in 3 Teile, rotiere und verschiebe jeden Teil, resultiert in Ball mit Radius 2.

Bsp: Jeder Ring hat max. Ideal

Satz

Sei R Ring mit 1 (neutr. El. der Mult.), $R \neq \{1\}$.
Dann gibt es ein maximales ideal $M \subsetneq R$ (d.h. einziges Ideal größer als M ist R selbst).

Beweis.

Betrachte $X := \{I \subset R \text{ Ideal} : 1 \notin I\}$, geordnet durch $\subseteq$. Das ist partielle Ordnung (Halbordnung). Jede Kette hat obere Schranke: Kette ist totalgeordnete Teilmenge von X. Sei $L \subseteq X$ total geordnet, d.h. für $I_1, I_2 \in X$: $I_1 \subseteq I_2$ oder $I_2 \subseteq I_1$. Dann ist $\bigcup L \in X$ obere Schranke von X. Zorn → es gibt maximales Element, d.h. ein größtes $M \in X$. □

Wiederholung: Rekursionstheorie

- Frage: Was ist ein Algorithmus? Welche Funktionen sind berechenbar? Welche Probleme effektiv entscheidbar?
- Einschränkung: Untersuchen nur $f : \mathbb{N}^n \to \mathbb{N}$.
  Äquivalent: $\mathbb{Q}$, Zeichenketten (strings), ...
- Beispiel 1: Die Funktion $(n, m) \mapsto$ ggT(n,m) ist berechenbar.
- Beispiel 2: Die Frage
  Ist die 17×17 Matrix M invertierbar?
  ist effektiv entscheidbar (Determinante).
- Definition: $f : \mathbb{N}^n \to \mathbb{N}$ ist berechenbar, wenn es Computerprogramm gibt daß auf Input $\vec{x}$ den Output $f(\vec{x})$ liefert.
- Begriff robust (beliebiger Speicher, endliche Programme).
- Konkretes Computermodell: URM.

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.

Exkurs: 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 $A \subseteq \mathbb{N}^n$. $A := \text{dom}(F)$.
- F total wenn $\text{dom}(F) = \mathbb{N}^n$. (D.h. wenn klassisch $F : \mathbb{N}^n \to \mathbb{N}$.)
- F ist partiell rekursiv (p.r.), wenn es ein URM Programm gibt daß auf Input $\vec{x}$ genau dann einen output liefert wenn $\vec{x} \in \text{dom}(F)$ und in diesem Fall wird der Output $F(x)$ geliefert.
- F is total rekursiv (t.r.) wenn p.r. und total (bisher: berechenbar).

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

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

Satz

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

Church'sche Bild "These"

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

Codierung/Gödel-Nummerierung:

Codierung/Gödel-Nummerierung:

Wir ordnen jedem URM-Programm "auf vernünftige Weise" eine (eindeutige) Nummer zu.

Beispiel

- "natürlich" aber sub-optimal: Schreibe das Programm als ASCII-Text auf. Das ergiebt Folge von Bytes (1 Byte=Zahl zwischen 0 und 255). Interpretiere diese Folge als Zahl zur Basis 256.
- "effizienter" aber umständlicher: Verwende formale Definition des URM-Programms (Menge von 4-Tupeln) ...

Wir können leicht (d.h. effektiv) entscheiden, ob eine Nummer e der Code eines Programms ist. Falls nicht, ordnen wir der Nummer e das Programm

```
1  goto 1
```

zu (welches nie hält, d.h. die leere partielle Funktion berechnet).
So bekommen wir eine surjektive Abbildung von $\mathbb{N}$ in Programme.

Das universelle Programm

Definition (Gödelnummer)

φ_e^n ist die n-stellige p.r. Funktion, die dem URM-Programm mit Nummer e entspricht.

D.h.: $\varphi_e^n(\vec{x}) = y$ gdw das Programm Nr. e auf Input $\vec{x}$ den Output y liefert.
Per Definition (trivial): $\varphi_e^0(x) = \varphi_e^1(0) = \varphi_e^2(0,0)$ etc.

Satz (Grundsätzliche Eigenschaften der Gödelnummern:)

- *φ^n bildet $\mathbb{N}$ surjektiv auf Menge der p.r. Funktionen $\mathbb{N}^n \to \mathbb{N}$ ab.*
- *$(e,x) \mapsto \varphi_e^1(x)$ is p.r., d.h. $\exists u \forall e\, \varphi_u^2(e,x) = \varphi_e^1(x)$* *Allgemeiner:*
- *Enumeration Thm: $\forall n \exists u \forall e\, \varphi_u^{n+1}(e,\vec{x}) = \varphi_e^n(\vec{x})$.*
- *Input kann effektiv ins Programm kodiert werden:*
  Es gibt t.r. Funktion S s.t. $\varphi_e^1(x) = \varphi_{S(e,x)}^0$. *Allgemeiner:*
- *S_m^n Thm: Für n, m gibt es t.r. Funktion S_m^n s.t.*
  $\forall e\, \varphi_e^{n+m}(\vec{x},\vec{y}) = \varphi_{S_m^n(e,\vec{x})}^m(\vec{y})$.