Grundbegriffe der mathematischen Logik

Vorlesung WS 2005/2006

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

Kurt Gödel Research Center for Mathematical Logic

4. Vorlesung, 2005-11-09

Richards Paradoxon

Es gibt einen kanonischen Berechenbarkeitsbegriff.
I.A. gibt es in der Logik solche kanonischen Konzepte nicht.
Wichtiges Beispiel: Die Sprache der Mathematik.

Richard's Zahl

- Eine natürliche Zahl n ist definierbar, wenn es einen Satz gibt, der nur für diese Zahl gilt.
- Bsp: "x ist die 10^{10}-te Primzahl" definiert eine Zahl.
- Es gibt höchstens ca. 50^{300} Sätze mit weniger als 300 Zeichen.
- Sei R die kleinste natürliche Zahl die nicht mit einem Satz mit weniger als 300 Zeichen definierbar ist.

"Definierbar" ist kein absolutes Konzept.
Nur sinnvoll: Definierbar in einer bestimmten (formalen) Sprache $\mathcal{L}$.
In $\mathcal{L}$ wird i.A. Definierbarkeit in $\mathcal{L}$ (genauer: Wahrheit) nicht formulierbar sein, daher: Es gibt keine "universelle Sprache".

Objekt- und Metasprache

Um die Sprache der Mathematik zu analysieren müssen wir also i.A. Folgendes unterscheiden:

1. Objektsprache (Gebrauchssprache, hier: formale Sprache), und
2. Metasprache, in der über Objektsprache gesprochen wird.

Aussagenlogik

Die erste formale Sprache die wir verwenden ist Aussagenlogik.

Beispiele für Sätze der Aussagenlogik

1. $((\neg A_1 \to \neg A_2) \to (A_1 \to A_2))$
2. $((A_1 \to A_2) \to (\neg A_2 \to \neg A_1))$

Beispielsatz 1 ist nicht gültig, 2 schon. (Unterschiede zu Alltagssprache.)

Induktive Definition der aussagenlogischen Sätze

- A_i ist ein Satz für jedes $i \in \mathbb{N}$
- Wenn φ und ψ Sätze sind, dann auch $\neg\varphi$, $(\varphi \to \psi)$, $(\varphi \wedge \psi)$ und $(\varphi \vee \psi)$.

Belegungen

- Eine Belegung b ist eine Abbildung $\{A_i : i \in \mathbb{N}\} \to \{w, f\}$.
- Jede Belegung b kann kanonisch auf Funktion von den Sätzen nach $\{w, f\}$ fortgesetzt werden:
  - $b(\neg\varphi) = w$ gdw $b(\varphi) = f$,
  - $b(\varphi \vee \psi) = w$ gdw $b(\varphi) = w$ oder $b(\psi) = w$,
  - $b(\varphi \wedge \psi) = w$ gdw $b(\varphi) = w$ und $b(\psi) = w$,
  - $b(\varphi \to \psi) = f$ gdw $b(\varphi) = w$ und $b(\psi) = f$.
- Ein Satz φ ist erfüllbar, wenn $b(\varphi) = w$ für eine Belegung.
- Ein Satz φ ist gültig (Tautologie), wenn $b(\varphi) = w$ für alle Belegungen.
- Ein Satz φ ist unerfüllbar (Kontradiktion), wenn $b(\varphi) = w$ für keine Belegung.
- Eine Menge Σ von Sätzen ist erfüllbar, wenn es eine Belegung b gibt so daß $b(\varphi) = w$ für alle $\varphi \in \Sigma$.

Wiederholung: Berechenbarkeit, URM

- Frage: Welche Funktionen sind berechenbar? Welche Probleme effektiv entscheidbar?
- 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 (beliebig großer Speicher, endliche Programme).
- Konkretes Computermodell: URM.
- Ein URM Programm hält (oder: terminiert, liefert Output) im allgemeinen nicht auf jedem Input.
- Partielle Funktionen: Def.Bereich nur Teilmenge von $\mathbb{N}^n$.
- Partiell rekursive (p.r.) Funktionen.
  $f(x) = y$ gdw Programm mit Input x hält und Output y liefert.
  Total rekursiv (t.r.): totale, partiell rekursive Funktion.
- Gödelnummerierung: Ordne jeder Zahl ein Programm (bzw. die entsprechende Funktion) zu.

Gödelnummerierung

Beispiel: Programm P (Addition):

$$
\begin{array}{llllll}
1 & \texttt{if } R_1 = 0 \texttt{ goto } 5 & \rightarrow & (1,4,1,5) & \rightarrow & c_1 = 2^1 \cdot 3^4 \cdot 5^1 \cdot 7^5 \\
2 & R_1 := R_1 - 1 & \rightarrow & (2,2,1,0) & \rightarrow & c_2 = 2^2 \cdot 3^2 \cdot 5^1 \cdot 7^0 \\
3 & R_0 := R_0 + 1 & \rightarrow & (3,1,0,0) & \rightarrow & c_3 = 2^3 \cdot 3^1 \cdot 5^0 \cdot 7^0 \\
4 & \texttt{goto } 1 & \rightarrow & (4,3,0,1) & \rightarrow & c_4 = 2^4 \cdot 3^3 \cdot 5^0 \cdot 7^1 \\
5 & \texttt{return} & \rightarrow & (5,0,0,0) & \rightarrow & c_5 = 2^5 \cdot 3^0 \cdot 5^0 \cdot 7^0 \\
 & & & & \rightarrow & e = 2^{c_1} \cdot 3^{c_2} \cdot 5^{c_3} \cdot 7^{c_4} \cdot 11^{c_5}
\end{array}
$$

Das Programm P hat also den Code e, oder umgekehrt:
Die Gödel-Nummer e entspricht dem Programm P.
Aus e kann man P rekonstruieren!

Der Zustand einer URM

Beispiel: Das Programm Nr. e (Addition)

```
1  if R1 = 0 goto 5
2  R1 := R1 − 1
3  R0 := R0 + 1
4  goto 1
5  return
```

Berechnung bei Input $(3, 1)$

Zeit:	0	Zeile:	1	R_0 :	3	R_1 :	1	Input $(3, 1)$ in R_0, R_1
Zeit:	1	Zeile:	2	R_0:	3	R_1:	1	
Zeit:	6	Zeile:	0	R_0 :	4	R_1 :	0	Output (R_0): 4

Zustand der URM e zu bestimmten Zeitpunkten t:

$\varphi^2_{e,0}(3,1) = (1,3,1) = 2^1 \cdot 3^3 \cdot 5^1$, $\varphi^2_{e,1}(3,1) = (2,3,1) = 2^2 \cdot 3^3 \cdot 5^1$, ...

$\varphi^2_{e,6}(3,1) = (0,4,0) = 2^0 \cdot 3^4 \cdot 5^0$ $\qquad$ $\varphi^2_e(3,1) = 4.$

Die Funktion $(e, t, x_0, x_1) \to \varphi^2_{e,t}(x_0, x_1)$ ist t.r.! (sogar prim.r.)
(Mühselig zu beweisen.
Alle anderen rek.th. Tatsachen der Vorlesung folgen leicht daraus, insbesondere:)
Die Funktion $(e, x_0, x_1) \to \varphi^2_e(x_0, x_1)$ ist p.r. (*mu*-r.).

Gödel-Nummerierung

Grundlegende Eigenschaften der Gödelnummern

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

Syntaktische Modifikation von Programmen

Bsp: Wenn P die Funktion f berechnet, können wir Q finden das $f + 1$ berechent (ohne P simulieren zu müssen).

$e = 2^{c_1} \cdot 3^{c_2} \cdot 5^{c_3} \cdot 7^{c_4} \cdot 11^{c_5}$

1	`if` $R_1 = 0$ `goto 5`	$\rightarrow$	c_1
2	$R_1 := R_1 - 1$	$\rightarrow$	c_2
3	$R_0 := R_0 + 1$	$\rightarrow$	c_3
4	`goto 1`	$\rightarrow$	c_4
5	~~`return`~~ `goto 6`	$\rightarrow$	c_5~~c'_5~~
6	$R_0 := R_0 + 1$	$\rightarrow$	c_6
7	`return`	$\rightarrow$	c_7

$e' = 2^{c_1} \cdot 3^{c_2} \cdot 5^{c_3} \cdot 7^{c_4} \cdot 11^{c'_5} \cdot 13^{c_7} \cdot 17^{c_8}$

Die Abbildung $e \rightarrow e'$ is t.r.!
(Entspricht: Programm P wird syntaktisch zu P' modifiziert)
Auch definiert z.B. für e die niemals haltendes Programm definieren.

Unberechenbare Funktionen

Verschiedene Arten, unberechenbare Funktionen zu finden/konstruieren:

Kardinalität:

Es gibt nur abzählbar viele Programme, aber überabzählbar viele Funktionen $\mathbb{N} \to \mathbb{N}$. Daher gibt es unberechenbare Funktionen.

klassische Diagonalisierung:

$$g(n) := \begin{cases} \varphi_n(n) + 1 & \text{falls } \varphi_n(n) \text{ definiert,} \\ 0 & \text{sonst} \end{cases}$$ ist nicht rekursiv.

(Beweis: Sonst wäre $g = \varphi_e$, d.h. $g(e) = \varphi_e(e)$.)
g mod 2 ist eine nicht rekursive 0-1-Folge.

Halteproblem

Die Frage "ist $\varphi_n(n)$ definiert" ist nicht effektiv entscheidbar.

(Beweis: Sonst wäre das obige g berechenbar.)

Rekursiv aufzählbar (r.e.)

Das Halteproblem $H = \{e : \varphi_e(0) \text{ definiert}\}$ ist nicht entscheidbar (oder: die charakteristische Funktion von H ist nicht berechenbar.)
Man kann also nicht entscheiden ob ein Programm hält.
Aber wenn ein Programm hält, dann kann man natürlich feststellen dass das Programm hält.
Man sagt: H ist rekursiv aufzählbar (r.e.), d.h. man kann ein Programm angeben das alle (auf Input 0) haltenden Programm-Nummern ausgibt / auflistet.

Definition

- $A \subseteq \mathbb{N}$ ist rekursiv aufzählbar (r.e.), wenn $A = \varphi_e''\mathbb{N}$ für ein e.
- $A \subseteq \mathbb{N}$ ist rekursiv, wenn die charakteristische Funktion von A berechenbar (t.r.) ist.

Dabei verwenden wir die Notation $f''A := \{f(x) : x \in A \cap \text{dom}(f)\}$.

Rekursiv aufzählbar (r.e.)

"Ist Problem P entscheidbar (rekursiv)" ist äquivalent zu: "ist die charakteristische Funktion von P berechenbar/p.r./t.r."
A ist rekursiv heißt: Es gibt ein Programm dass bei Input n entscheidet ob $n \in A$ oder nicht (und das Programm terminiert immer).
Rekursiv aufzählbar ist so etwas wie "halb rekursiv".

Satz

Äquivalent sind ($A \subseteq \mathbb{N}$):

- A r.e., d.h. A ist Bild einer p.r. Funktion,
- $A = \emptyset$ oder A Bild einer t.r. Funktion,
- $A = \mathrm{dom}(\varphi_e)$ für ein e, d.h. A domain einer p.r. Funktion,

Eigenschaften von r.e. Mengen

Wenn man die Begriffe r.e. und rekursiv einmal verstanden hat, sind die folgenden Eigenschaften völlig klar:

- Jede endliche Menge ist rekursiv.
- Wenn A und B r.e., dann auch $A \cup B$ und $A \cap B$.
- Wenn A r.e. und unendlich dann gibt es ein injektives t.r. f mit $f''\mathbb{N} = A$.
- A ist rekursiv genau dann wenn A r.e. und $\mathbb{N} \setminus A$ r.e.
- Wenn A r.e. dann ist $\mathbb{N} \setminus A$ i.a. nicht r.e.
- r.e. bleibt erhalten unter p.r. Bildern und Urbildern.
- Wenn A und B r.e., dann gibt es disjunkte r.e. $A' \subseteq A$ und $B' \subseteq B$ s.d. $A' \cup B' = A \cup B$.

(Sei d_1, d_2 t.r. s.d. $x \mapsto (d_1(x), d_2(x))$ bijektiv.)

Satz

A r.e. $\rightarrow$ A ist Definitionsbereich einer p.r. Funktion.

Beweisskizze:

A is Bild von φ_e.

D.h. $a \in A$ ist äquivalent zu: Es gibt $n, t \in \mathbb{N}$ s.d.

$\varphi_{e,t}(n) = (0, a, \dots) = 2^0 \cdot 3^a \cdots$.

Definiere $f(a, x) = \begin{cases} 0 & \text{wenn } \varphi_{e,d_1(x)}(d_2(x)) = (0, a, \dots), \\ 1 & \text{sonst.} \end{cases}$

f ist t.r. (sogar prim.r.).

Sei $g(a) = \min(x : \{f(a, x) = 0\})$. Dann ist g p.r.

A ist Definitionsbereich von g.

(Sei d_1, d_2 t.r. s.d. $x \mapsto (d_1(x), d_2(x))$ bijektiv.)

Satz

$A \neq \emptyset$ ist Definitionsbereich einer p.r. Funktion $\to A$ ist Bild einer t.r. Funktion.

Beweisskizze:

A is Definitionsbereich von φ_e und $a_0 \in A$.

D.h. $a \in A$ ist äquivalent zu: Es gibt $t \in \mathbb{N}$ s.d. $\varphi_{e,t}(a) = (0, \dots) = 2^0 \cdots$.

Definiere $f(x) = \begin{cases} d_1(x) & \text{wenn } \varphi_{e,d_2(x)}(d_1(x)) = (0, \dots), \\ a_0 & \text{sonst.} \end{cases}$

f ist t.r. (sogar prim.r.).

A ist Bild von f.

(Sei d_1, d_2, d_3 t.r. s.d. $x \mapsto (d_1(x), d_2(x), d_3(x))$ bijektiv.)

Satz

Wenn A und B r.e., dann auch $A \cup B$.

Beweisskizze:

A is Bild von φ_a und B von φ_b.

Definiere $f(x) = \begin{cases} \varphi_a(x/2) & \text{wenn } x \text{ gerade,} \\ \varphi_b((x-1)/2) & \text{sonst.} \end{cases}$

$A \cup B$ ist Bild von f.

Satz

Wenn A und B r.e., dann auch $A \cap B$.

Beweisskizze:

A is Definitionsbereich von φ_a und B von φ_b.

Definiere $f(x) = \varphi_a(x) + \varphi_b(x)$.

Definitionsbereich von f ist $A \cap B$.

Beispiele für nicht-rekursive r.e. Mengen

Man kann zeigen, dass bestimmte Probleme nicht effektiv entscheidbar sind, indem man zeigt dass sie genauso schwer sind wie das Halteproblem.

Definition

A lässt sich auf B reduzieren ($A \leq_m B$),
wenn es ein t.r. $f : \mathbb{N} \to \mathbb{N}$ gibt s.d. $n \in A$ gdw $f(n) \in B$.

Wenn also H (die Haltemenge) auf eine Menge A reduzierbar ist, dann kann A nicht rekursiv sein (sonst wäre H rekursiv). Damit kann man die effektive Unlösbarkeit vieler Problem aus der Mathematik zeigen.
Beispiele:

nicht-rekursive r.e. Mengen

- Ableitbarkeit in Prädikatenlogik.
- 10. Hilbert'sches Problem.
- Wortproblem in Gruppen.

10. Hilbertsches Problem

Sei $f(X_1, \ldots, X_n) \in \mathbb{Z}[X_1, \ldots, X_n]$ ein Polynom mit ganzzahligen Koeffizienten.
Hat f eine ganzzahlige Nullstelle?
D.h. gibt es $X_1, \ldots, X_n$ aus $\mathbb{Z}$ s.d. $f(X_1, \ldots, X_n) = 0$?
Das ist also die Frage nach der Lösbarkeit allgemeiner Diophantischer Gleichungen, ein zentrales Problem der Zahlentheorie.
Offenbar r.e.
Man kann jedem URM Programm effektiv ein Polynom f (in 7 Variablen) zuordnen, so dass das Programm auf Input 0 hält genau dann wenn f eine ganzzahlige Nullstelle hat.