UNIVERSITÄT
BERN

Fachschaftstagung Gymnasium Köniz

LOGIK IM INTERDISZIPLINÄREN SPANNUNGSFELD ZWISCHEN INFORMATIK, MATHEMATIK UND PHILOSOPHIE

Thomas Strahm

Institut für Informatik und angewandte Mathematik
Universität Bern

4. November 2004

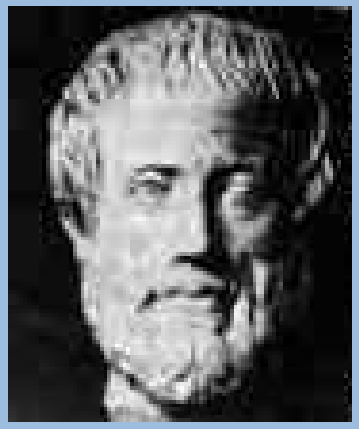
Zu den Ursprüngen der Logik

Aristoteles: Syllogismen

Objektive Gesetze des menschlichen Denkens.

Schemata für **gültige Schlüsse:**

$$\frac{\text{Prämissen}}{\text{Konklusion}}$$



z.B.

$$\frac{\text{Alle } P \text{ sind } Q,\ a \text{ ist } P}{a \text{ ist } Q}$$

Instanz:

$$\frac{\text{Alle Menschen sind sterblich}\quad \text{Sokrates ist ein Mensch}}{\text{Sokrates ist sterblich}}$$

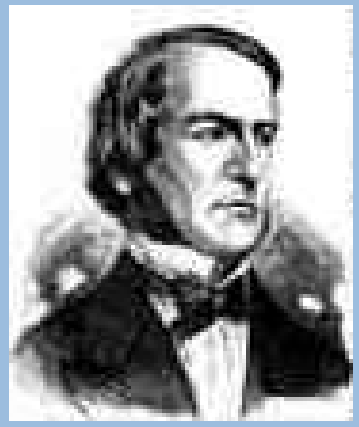
Zu den Ursprüngen der modernen Logik

UNIVERSITÄT BERN

Symbolisch-mathematische Logik
Kalkülisierung der Logik

Wichtige Vorläufer: Lullus (13. Jh.) Leibniz (17. Jh.)

George Boole: „The mathematical analysis of logic“ (1847)
Begründung der Aussagenlogik.



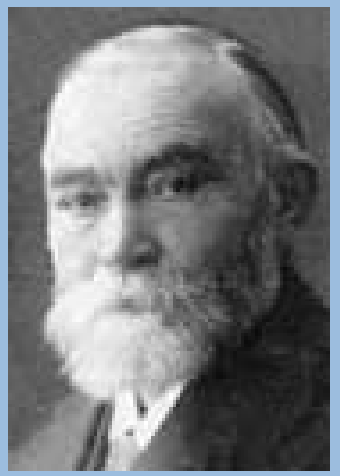
$P, Q, ...$	atomare Aussagen
$\neg A$	„nicht A“
$A \wedge B$	„A und B“
$A \vee B$	„A oder B“
$A \rightarrow B$	„A impliziert B“

Endlich viele Axiomenschemata und Schlussregeln.

Zu den Ursprüngen der modernen Logik

z.B. $A \vee \neg A$ $\qquad \dfrac{A \;\; A \to B}{B}$

Gottlob Frege: „Begriffsschrift“ (1879)
Begründung der Prädikatenlogik.



$x, y, z, \ldots$	Individuenvariabeln
$P(x), Q(x,y,z), \ldots$	mehrstellige Prädikate
$\neg, \wedge, \vee, \to$	
$\forall x A(x)$	„$A(x)$ trifft auf alle x zu“
$\exists x A(x)$	„$A(x)$ trifft auf ein x zu“

Zusätzliche Axiome und Schlussregeln, z.B.:

$A(y) \to \exists x A(x)$ $\qquad \dfrac{A(x)}{\forall x A(x)}$

Vollständigkeit der Prädikatenlogik

UNIVERSITÄT BERN

Unterscheidung Syntax - Semantik

Syntax: formale Zeichenreihen, Axiome und Regeln zum Ableiten von neuen Ausdrücken.

Semantik: Interpretation/Bedeutung von syntaktischen Ausdrücken

---

Logisch gültige Aussagen

Aussagen, die unter beliebigen Interpretationen wahr sind, z.B.

$$A \wedge B \rightarrow A, \quad \exists x \forall y Q(x,y) \rightarrow \forall y \exists x Q(x,y)$$

Vollständigkeit der Prädikatenlogik (Gödel)
Alle logisch gültigen Aussagen sind herleitbar.

Erweiterung auf „nicht-logische“ Theorien.

Die axiomatische Methode und das Hilbertsche Programm

UNIVERSITÄT BERN

1878 – 1897: Georg Cantor schafft das „Paradies“der unendlichen Mengen. Grundlegende Wandlung der Beweismethoden in der Mathematik.

$\Rightarrow$ Rechtfertigungsbedarf der verwendeten Prinzipien.

Hilbertsches Programm

Die gesamte Mathematik wird in einem „grossen“ Axiomensystem M repräsentiert. Die Widerspruchsfreiheit von M wird in einem kleinen finiten Teil F von M nachgewiesen.

Gödelsche Unvollständigkeitssätze (1931)

UNIVERSITÄT BERN

In jedem widerspruchsfreien Axiomensystem, welches die elementare Arithmetik enthält, gibt es arithmethische Aussagen, die wahr aber nicht beweisbar sind. Die Widerspruchsfreiheit des Axiomensystems selbst ist eine solche Aussage.

„Es gibt kein Axiomensystem, das alle Fragen beantwortet“.

⇒ **ab 1936:** Erweitertes, modifiziertes Hilbertsches Programm (Gentzen).

Die klassischen Gebiete der mathematischen Logik

Beweistheorie — Modelltheorie

Mengenlehre — Rekursionstheorie

- Klassifikation von Axiomensystemen aufgrund ihrer beweistheoretischen Stärke.
  Messen des nicht-finiten, d.h. transfiniten Gehaltes: Ordinalzahlen
- Entwurf und Analyse von Axiomensystemen für die mathematische Praxis
- Konstruktive Mathematik
  - Intuitionismus (Brouwer)
  - Explizite Mathematik (Feferman)

Intuitionistische Logik (Brouwer, Heyting)

UNIVERSITÄT BERN

Kritik am sog. tertium non datur:

$A \vee \neg A$: „A oder nicht A“

Konstruktive Existenzbeweise erlauben die Extraktion von Zeugen.

Ein nicht-konstruktiver Existenzbeweis:

Theorem Es gibt irrationale Zahlen a,b, so dass a^b rational ist.

1. Fall: $\sqrt{2}^{\sqrt{2}}$ ist rational. Wähle $a = b = \sqrt{2}$

2. Fall: $\sqrt{2}^{\sqrt{2}}$ ist irrational. Wähle $a = \sqrt{2}^{\sqrt{2}}$ und $b = \sqrt{2}$

Verwendung des tertium non datur.

Rekursionstheorie / Berechenbarkeitstheorie

UNIVERSITÄT BERN

Algorithmus (Al Chwarismi 825)

Endliche, eindeutige Beschreibung eines effektiven Verfahrens zur Lösung eines Problems

Beispiele: *Addition, Multiplikation, n-te Primzahl, Euklid*

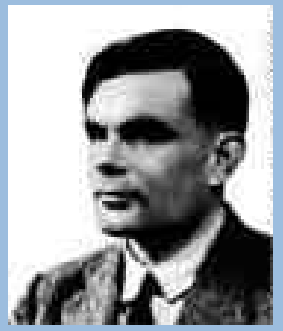
Intuitiver Berechenbarkeitsbegriff.

Was ein Computer im Prinzip berechnen kann.

Formal-mathematische Berechenbarkeitsmodelle

Gödel, Turing, Church, Kleene, ... (ab 1930)

Turing-Maschinen (Turing 1936)



Die These von Church

UNIVERSITÄT BERN

Weitere Modelle für Berechenbarkeit

>Gödel-Herbrand Gleichungskalküle

> Ungetypter λ-Kalkül (Church)

> Partiell-rekursive Funktion (Kleene)

Alle diese Modelle definieren dieselbe Klasse von Funktionen!

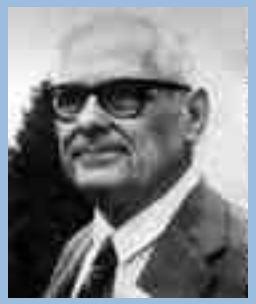
Churchsche These

Jedes dieser formalen Berechenbarkeitsmodelle stimmt mit dem intuitiven Algorithmusbegriff überein.

Unentscheidbare Probleme

> Halteproblem (Turing)

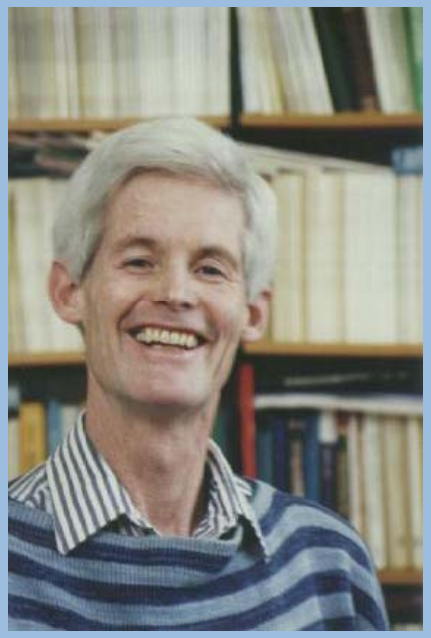
> Gültigkeit in der Prädikatenlogik (Church)



Komplexitätstheorie

UNIVERSITÄT BERN

Klassifikation von Algorithmen nach ihrem Ressourcenverbrauch (Zeit, Speicher)



Ein 1-Millionen-Dollar-Problem

P = Probleme mit polynomialem Algorithmus

NP = Probleme mit polynomialem Verifikationsalgorithmus

$\Rightarrow$ **Vermutung: P ist verschieden von NP** (S. Cook)

Logik und Komplexitätstheorie

- Logische und intrinsische Charakterisierung von Komplexitätsklassen, v.a. P
- Logische Ansätze zur P / NP – Frage
- Spannungsfeld „klassisch-konstruktiv-feasible“

Anwendungen der Logik in der Informatik

UNIVERSITÄT BERN

- Semantik von Programmiersprachen
- Entwicklung von Programmiersprachen:
  LISP (λ-Kalkül)
  PROLOG (Prädikatenlogik)
- Objektorientierte Programmierung
- Spezifikation und Verifikation von Hardware und Software
- ★ Extraktion von Programmen aus konstruktiven Beweisen
- ★ Logik und Wissensrepräsentation

Extraktion von Programmen aus konstruktiven Beweisen

UNIVERSITÄT BERN

πt $\forall x\, \exists y\, Spec(x,y)$ Spezifikation

$\Downarrow$

Algorithmus f_π, so dass $\forall x\, Spec(x, f_\pi(x))$.

Formulas-as-Types (Curry-Howard)

$[A]$: = {Beweise von A}

$[A \to B]$: = {Operationen g: g transformiert Beweis π von A in Beweis $g(\pi)$ von B}

Konstruktiver Beweis, dass der Typ einer Spezifikation nicht leer ist, liefert Algorithmus, der die Spezifikation erfüllt.

Logik und Wissensrepräsentation

UNIVERSITÄT BERN

- klassische Prädikatenlogik erster Stufe (PL1) als allgemeiner Rahmen für Wissensrepräsentation
- Gewinnung von „neuem“ Wissen = Ableiten von neuen Fakten aus der Wissensbasis mit Hilfe von PL1 (z.B. Expertensysteme)
- Einschränkung bei der Automatisierung
  - PL1 „nur“ semientscheidbar
  - Effizienz
- Unvollständigkeit (Gödelscher Satz)

Nicht-monotone Logiken

UNIVERSITÄT BERN

Wesentliche Merkmale von „Schliessen“

> unvollständige Information

> allgemeine Regeln mit Ausnahmen

> Rückgängigmachen von Schlüssen

Beispiel

„Alle Vögel fliegen“ vs „Alle Menschen sind sterblich“

Schliessen in der Prädikatenlogik ist monoton.

Schliessen „per Default“ ist nicht monoton.

Problem der Formalisierung von Regeln mit Ausnahmen in PL1.

Interesse/Relevanz von nicht-monotonen Logiken für die Philosophie.

Modale Logiksysteme

Logische Analyse der Modalitäten „notwendig“ und „möglich“ (Aristoteles)

$\Box A$: A ist notwendig wahr.

$\Diamond A$: A ist möglicherweise wahr.

z.B. $\Box(A \rightarrow B) \rightarrow (\Box A \rightarrow \Box B)$.

Semantik möglicher Welten (Kripke)

Mögliche Welten W mit Erreichbarkeitsrelation $R \subseteq W \times W$

$\Box A$ ist wahr in $w \in W$, falls A wahr ist in allen Welten w’ mit wRw’.

$\Diamond A$ ist wahr in $w \in W$, falls A wahr ist in einer Welt w’ mit wRw’.

Verifikation von Programmen

UNIVERSITÄT BERN

Beispiel Programm „Primzahl drucken“

$y_1 := 2$

λ_0 : PRINT(y_1)

λ_1 : $y_1 := y_1 + 1$

λ_2 : $y_2 := 2$

λ_3 : if $y_2^2 > y_1$ then goto λ_0

λ_4 : if (y_1 mod y_2) = 0 then goto λ_1

λ_5 : $y_2 := y_2 + 1$

λ_6 : goto λ_3.

Mögliche Zustände: (λ, y_1, y_2)

Erreichbarkeitsrelation: Zustandsübergänge

Programmkorrektheit: $\Box$(at $\lambda_0 \rightarrow$ Primzahl (y_1))

Weitere Anwendungen der Modallogik

UNIVERSITÄT BERN

- **Wissenslogiken**

  K_iA: „i weiss A“

  z.B.

  $K_iA \rightarrow K_iK_iA$ (positive Introspektion)

  $\neg K_iA \rightarrow K_i \neg K_iA$ (negative Introspektion)

  „Reasoning about distributed systems“

- **Beweisbarkeitslogiken**

  $\Box A$: “A ist beweisbar in der Zahlentheorie”

  Formaler Rahmen zum Studium der Metamathematik der Arithmetik.

Temporale Logiksysteme

UNIVERSITÄT BERN

- Einbezug der Zeit in die Logik
- Ursprüngliche philosophische/linguistische Motivation (natürliche Sprachen; tenses)
- Informatik: Spezifikation und Verifikation von verteilten Systemen

GA : A in der Zukunft immer wahr

HA : A in der Vergangenheit immer wahr

FA : = $\neg$G $\neg$A : A in der Zukunft manchmal wahr

PA : = $\neg$H $\neg$A : A in der Vergangenheit manchmal wahr