

Logik

WiSe 2016/17

Prof. Dr. Thomas Schneider
 AG Theorie der künstlichen Intelligenz
 Cartesium, Raum 2.56
ts@informatik.uni-bremen.de

Homepage der Vorlesung:
<http://tinyurl.com/ws16-logik>

Zeit und Ort:

Di 8–10 MZH 1460
 Mi 16–18 MZH 1460

Vortragender:

Prof. Dr. Thomas Schneider
 Raum Cartesium 2.56
 Tel. (218)-64432
ts@informatik.uni-bremen.de

Position im Curriculum:

Wahlbereich Bachelor-Basis
 Modulbereich Theorie
 Master-Profile SQ und KIKR

Organisatorisches

Voraussetzungen:

Grundvorlesung Theoretische Informatik

Form:

K4
 3 Termine Vorlesung, 1 Termin Übung (also 26.10., 15.11., 29.11. usw.)
 Diskussion in Vorlesung **jederzeit erwünscht!**

Vorlesungsmaterial:

- Folien und Aufgabenblätter auf:
<http://tinyurl.com/ws16-logik> (s. auch Webseite der AG TDKI)
- Beispiele, Beweise etc. an der Tafel (**mitschreiben!**)

Literatur

Große Teile aus:

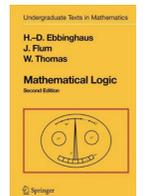
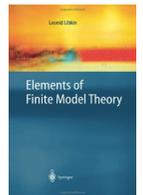
- Erich Grädel. Mathematische Logik I. Vorlesungsskript, RWTH Aachen, Verfügbar in Stud.IP

Logik zweiter Stufe:

- Leonid Libkin. Elements of Finite Model Theory. Springer Verlag, 2004

Weitere Referenzen:

- Uwe Schöning. Logik für Informatiker. Spektrum akademischer Verlag, 2000 (5. Auflage).
- Christel Baier. Advanced Logics. VL-Skript, TU Dresden.
- Heinz-Dieter Ebbinghaus, Jörg Flum, Wolfgang Thomas. Mathematical Logic. Springer Verlag, 1994 (2. Auflage).



Übungsaufgaben & Fachgespräch:

- Übungsaufgaben jede zweite Woche (mit Zusatzaufgaben)
- Werden in Gruppen (2–3 Personen) bearbeitet, abgegeben und korrigiert – jede_r muss mindestens einmal vorrechnen
- Fachgespräche am Ende des Semesters
Voraussetzung: 50% der Punkte in Übungsaufgaben

oder

Mündliche Prüfung

Logik



Ursprünge der Logik

Traditionell ist die Logik ein Teilgebiet der Philosophie und Mathematik:

Philosophie:

Lehre des vernünftigen Schlussfolgerns, geht zurück auf Aristoteles (≈300 a. D.)

Klassisches Beispiel: **Syllogismen**

Alle Menschen sind sterblich
Sokrates ist ein Mensch

Sokrates ist sterblich

Jedes P ist auch ein Q
 x ist ein P

 x ist ein Q

Seit dem 20. Jh. ein elaboriertes und vielfältiges Teilgebiet der Philosophie
Ziel: Abstrakte und formale Behandlung philosophischer Fragestellungen

Ursprünge der Logik

Traditionell ist die Logik ein Teilgebiet der Philosophie und Mathematik:

Mathematik:

Logik spielt zentrale Rolle für die Grundlagen der Mathematik

Klassisches Beispiel: die **Peano-Axiome** für die natürlichen Zahlen
(formuliert in der Logik zweiter Stufe)

- $0 \in \mathbb{N}$
- $\forall n \in \mathbb{N} : \exists n' \in \mathbb{N} : n' = nf(n)$
- $\forall n \in \mathbb{N} : nf(n) \neq 0$
- $\forall n \forall m \in \mathbb{N} : (nf(n) = nf(m) \rightarrow n = m)$
- $\forall X : (0 \in X \wedge \forall n : (n \in X \rightarrow nf(n) \in X)) \rightarrow \mathbb{N} \subseteq X$

Aus diesen Grundannahmen lassen sich alle Eigenschaften der natürlichen Zahlen herleiten.

Logik in der Informatik

Logik ist eine der wichtigsten mathematischen Grundlagen der Informatik

Von essentieller Bedeutung z. B. für:

- Datenbanken und Semistrukturierte Daten (XML)
- Verifikation von Hard- und Software
- Programmiersprachen
- Komplexitätstheorie
- Wissensrepräsentation / Künstliche Intelligenz
- Automatisches Theorembeweisen
- etc.

Logische Methoden haben die Entwicklung der Informatik entscheidend mitbestimmt.

Umgekehrt ist heute die Informatik eine der größten Triebkräfte hinter der Weiterentwicklung der Logik.

Fallbeispiel 1: Datenbanken

SQL-Anfragebeantwortung kann als Logikproblem verstanden werden

Im Folgenden: FO = Prädikatenlogik erster Stufe

- SQL-Anfragen sind im Wesentlichen FO-**Formeln**
- SQL-Datenbankinstanzen sind FO-**Strukturen**
- SQL-Anfragebeantwortung entspricht **Modellprüfung** in FO

Slogan: **SQL ist Logik**

Diese Sichtweise hat die Entwicklung und den Erfolg von relationalen Datenbanken entscheidend mitgeprägt.

(Ted Codd, System R am IBM Almaden Research Center 1960'er–70'er)

Fallbeispiel 2: Verifikation

Verifikation: nachweisen, dass ein Chip / Programm eine gewünschte Spezifikation erfüllt (z. B. keine Division durch 0, keine Deadlocks)

Verifikation basiert in der Regel auf Logik:

- Chip / Programm kann als (endliche oder unendliche) **logische Struktur** modelliert werden
- Spezifikation kann als logische **Formel** modelliert werden, z. B. in einer Temporallogik wie LTL oder CTL
- Verifikation entspricht dann wieder **Modellprüfung**

Verifikation ist heutzutage ein zentrales Thema im Chipdesign und wird auch für Software zunehmend wichtiger.

Logik hat dieses wichtige Teilgebiet der Informatik entscheidend geprägt

Fallbeispiel 3: Komplexitätstheorie

Bekanntestes offenes Problem der theoretischen Informatik:

Ist $P \neq NP$?

Klassische Definition NP:

Menge der Probleme, die von einer nicht-deterministischen Turingmaschine in Polynomialzeit gelöst werden können.

Alternative, aber äquivalente Definition:

Menge der Probleme, die mittels einer **Formel** der existentiellen Logik zweiter Stufe **definiert** werden können.

Dies erlaubt das Studium von P und NP **mit logischen Methoden**, komplett **ohne Turingmaschinen oder andere Berechnungsmodelle** (**Deskriptive Komplexitätstheorie**)

Ziele der Vorlesung

-  **Einführung der grundlegenden logischen Formalismen**
insbesondere Aussagenlogik und Prädikatenlogik 1. und 2. Stufe
-  **Formulierung und Beweis der zentralen Resultate der Logik**
insbesondere zu Schlussfolgerungsproblemen, Ausdrucksstärke und anderen Informatik-relevanten Themen
-  **Herstellung von Querbezügen zu anderen Teilgebieten der Informatik**
insbesondere zu Datenbanken, Verifikation und Komplexitätstheorie

Übersicht Vorlesung



- Einführung
- Teil 1: Aussagenlogik
- Teil 2: Prädikatenlogik Grundlagen
- Teil 3: Mehr zur Prädikatenlogik erster Stufe
- Teil 4: Prädikatenlogik zweiter Stufe