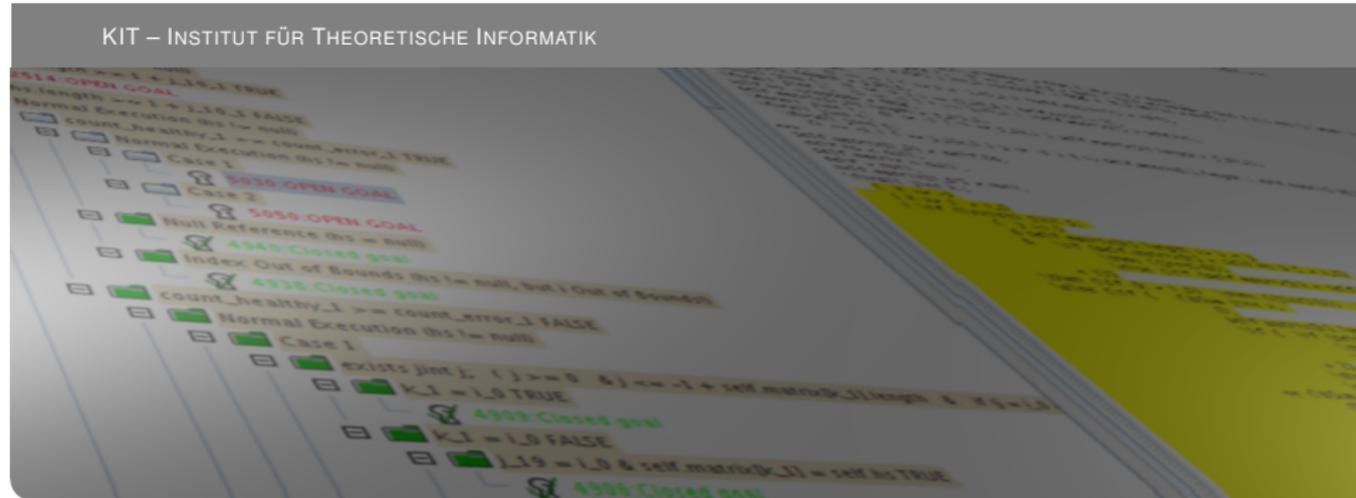


Formale Systeme

Prof. Dr. Bernhard Beckert, WS 2019/2020

Organisatorisches

KIT – INSTITUT FÜR THEORETISCHE INFORMATIK





Dozent: Prof. Dr. Bernhard Beckert



Vertretung: Dr. Mattias Ulbrich



Betreuung der Vorlesung: Mihai Herda

Übungen

- ▶ Große Übungen alle zwei Wochen freitags;
erste Übung am Fr., 25.10.19

Übungsblätter

- ▶ Übungsaufgaben aus Aufgabensammlung
(Bekanntgabe in Vorlesungen)
- ▶ Lösungen werden nicht korrigiert
- ▶ Erläuterung der Lösungen in den großen Übungen

Übungsschein: Zwischentests und Praxisaufgaben

- ▶ ein Zwischentests (10 Punkte)
- ▶ drei Praxisaufgaben (je 10 Punkte)
- ▶ Teilnahme freiwillig
- ▶ Übungspunkte werden im Verhältnis 1:10 als Bonuspunkte auf die bestandene Abschlussklausur angerechnet (max. 4 Punkte, max. ein Notenschritt)

Zwischentest

- ▶ „Leichtgewichtig“
- ▶ Am Anfang einer Übung, 15–20min
- ▶ Stoff aus der ersten Vorlesungshälfte
- ▶ Termin:
vermutlich Freitag, 13.12.19

Achtung:

Termin steht nicht endgültig fest, Webseite beachten!

Praxisaufgaben machen mit konkreten Systemen vertraut

- ▶ **minisat**

Ein System zu Erfüllbarkeitsprüfung aussagenlogischer Formeln (SAT Solver)

- ▶ **z3**

Ein System zu Erfüllbarkeitsprüfung prädikatenlogischer Formeln mit Theorien (SMT Solver)

- ▶ **KeY**

Ein interaktives Beweissystem für Prädikaten- und Programmlogik

Webseite zur Vorlesung

`http://formal.itl.kit.edu/teaching/
FormSysWS1920/`

Enthält alle für die Vorlesung relevanten Informationen und Materialien:

- ▶ Termine, Ankündigungen
- ▶ Vorlesungsskriptum
- ▶ Folien
- ▶ Aufgabensammlung, Praxisaufgaben etc.

ILIAS-Kurs zur Vorlesung

- ▶ Link liegt auf der Seite zur Vorlesung.
- ▶ Ankündigungen
- ▶ Foren: Inhaltlich, Organisatorisch, . . .
- ▶ Praxisaufgaben – Abgabe und Bewertung

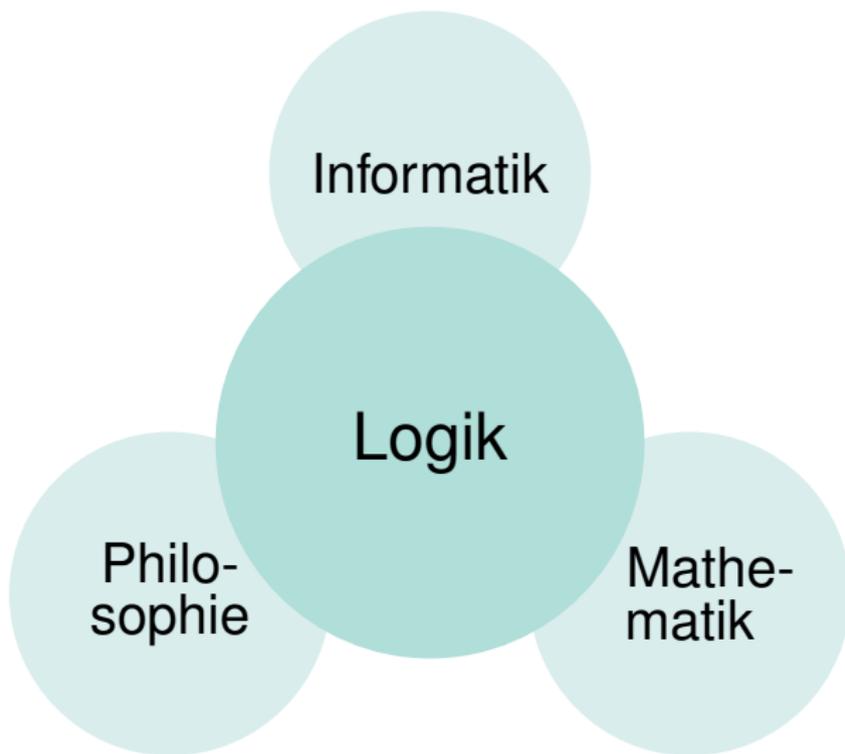
1. Klausurtermin

Freitag, 28.02.2020, 10:00 Uhr

2. Klausurtermin

Nach dem Sommersemester 2020

Eine sehr kurze Geschichte der Logik



Was macht eine Logik aus?

1. **Syntax**

ergibt die Sprache der Aussagen der Logik

2. **Semantik**

Bedeutungsstrukturen und welche Strukturen machen welche Sätze wahr (Modelle)

3. **Kalkül(e)**

Regelsätze/Algorithmen, um den “Wahrheitsgehalt” eines Satzes zu prüfen

(verschiedene Fragestellungen: Erfüllbarkeit, Allgemeingültigkeit, . . .)

Kompetenzen (s. auch Modulhandbuch)

Nach erfolgreichem Abschluss des Moduls

- ▶ verstehen Sie logische Grundbegriffe anhand verschiedener Logiken,
- ▶ können Sie deutsche Sätze als logische Aussagen formulieren,
- ▶ können Sie logische Fragestellungen mittels der vorgestellten Lösungsverfahren lösen,
- ▶ können Programmeigenschaften in formalen Spezifikationssprachen formulieren und kleine Beispiele verifizieren,
- ▶ können Sie beurteilen, welcher logische Formalismus und welcher Kalkül sich zur Formalisierung und zum Beweis eines Sachverhalts eignet.

Logik über atomare Wahrheitsaussagen

- ▶ Syntax/Semantik der AL
- ▶ Hilbert: Theoretischer (mathematischer) Kalkül für Beweise über Logik
- ▶ SAT-Solver: erfolgreiches praktisches Entscheidungsverfahren

Logik, die über „Objekte“ spricht.

Nachts sind alle Katzen grau.

Miez ist eine Katze.

Miez ist rot.

Es ist nicht nachts.

(Das konnten die alten Griechen schon.)

- ▶ Syntax/Semantik der PL
- ▶ Kalküle
- ▶ Gleichheit (Termersetzung)
- ▶ Prädikatenlogik über natürlichen Zahlen (Peano-Arithmetik)

3. Block: Formale Spezifikation

Anwendung der Prädikatenlogik um Eigenschaften von Programmen zu beweisen.

```
/** Der Rückgabewert von m ist eine Zahl, die  
 * größer als die Eingabe x ist.  
 */
```

```
/*@ ensures \result > x; */
```

```
public int m(int x) { ... }
```

- ▶ Java Modeling Language (Spezifikationsprache)
- ▶ KeY (Verifikationswerkzeug)

4. Block: Modale und Temporale Logik

Logik, die mehrere Zustände kennt

Beispiel

„Nachdem das Fenster geöffnet worden ist, ist die Heizung abgeschaltet.“

- ▶ Syntax/Semantik Modallogik
- ▶ zeitlicher Verlauf: Temporallogik
- ▶ Erfüllt ein „Automat“ eine temporale Spezifikation (Modellprüfung)?

Inhaltsübersicht

- ▶ Aussagenlogik: Syntax und Semantik
- ▶ Erfüllbarkeitstester (SAT Solver)
- ▶ Prädikatenlogik: Syntax und Semantik
- ▶ Kalküle
Aussagenlogik + Prädikatenlogik
- ▶ Peano-Arithmetik
- ▶ Gleichheit
- ▶ Java Modeling Language (JML)
- ▶ Modale Aussagenlogik
- ▶ Temporale Logik (LTL)
- ▶ Endliche Automaten (Wiederholung)
- ▶ Büchi-Automaten
- ▶ Modellprüfung

PETER H. SCHMITT: Formale Systeme. Skriptum zur Vorlesung.

MELVIN FITTING: First Order Logic and Automated Theorem Proving, 1990

U. SCHÖNING: Logik für Informatiker, 2000.

V. SPERSCHNEIDER/G. ANTONIOU: Logic: a Foundation for Computer Science, 1991.

ALONZO CHURCH: Introduction to Mathematical Logic, 1956.

EBBINGHAUS/FLUM/THOMAS: Mathematische Logik, 1992.

LOVELAND: Automated Theorem Proving: a Logical Basis, 1978.

SALLY POPKORN: First Steps in Modal Logic, 1994.

M. R. HUTH AND M. D. RYAN: Logic in Computer Science. Modelling and reasoning about systems, 2000.

J. E. HOPCROFT AND J. D. ULLMANN: Introduction to Automata Theory, 1979.

JAN VAN LEEUWEN (ED.): Handbook of Theoretical Computer Science. Vol. B : Formal Models and Semantics, 1990.

M. HUISMAN ET AL. Formal Specification with the Java Modeling Language, Chapter 7 in Deductive Software Verification—The KeY book. Springer, Ende 2016