

Formale Systeme

Prof. Dr. Bernhard Beckert, WS 2017/2018

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., 03.11.17

Übungen

- ▶ Große Übungen alle zwei Wochen freitags;
erste Übung am Fr., 03.11.17

Übungsblätter

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

Übungsschein: Zwischentests und Praxisaufgaben

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

Zwei Zwischentests

- ▶ „Leichtgewichtig“

Zwei Zwischentests

- ▶ „Leichtgewichtig“
- ▶ Jeweils am Anfang einer Übung, 15–20min

Zwei Zwischentests

- ▶ „Leichtgewichtig“
- ▶ Jeweils am Anfang einer Übung, 15–20min
- ▶ Stoff aus ca. 6 Wochen

Zwei Zwischentests

- ▶ „Leichtgewichtig“
- ▶ Jeweils am Anfang einer Übung, 15–20min
- ▶ Stoff aus ca. 6 Wochen
- ▶ Termine:
 1. am Do 07.12.17 oder Fr 08.12.17
 2. am Do 25.01.18 oder Fr 26.01.18

Praxisaufgaben machen mit konkreten Systemen vertraut

- ▶ **minisat**

Ein System zur Erfüllbarkeitsprüfung
aussagenlogischer Formeln (SAT Solver).

Praxisaufgaben machen mit konkreten Systemen vertraut

- ▶ **minisat**

Ein System zu Erfüllbarkeitsprüfung
aussagenlogischer Formeln (SAT Solver).

- ▶ **KeY**

Ein interaktives Beweissystem für Prädikaten-
und Programmlogik

Webseite zur Vorlesung

`http://formal.iti.kit.edu/teaching/
FormSysWS1718/`

Webseite zur Vorlesung

`http://formal.itk.kit.edu/teaching/
FormSysWS1718/`

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.

ILIAS-Kurs zur Vorlesung

- ▶ Link liegt auf der Seite zur Vorlesung.
- ▶ Ankündigungen

ILIAS-Kurs zur Vorlesung

- ▶ Link liegt auf der Seite zur Vorlesung.
- ▶ Ankündigungen
- ▶ Foren: Inhaltlich, Organisatorisch, ...

ILIAS-Kurs zur Vorlesung

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

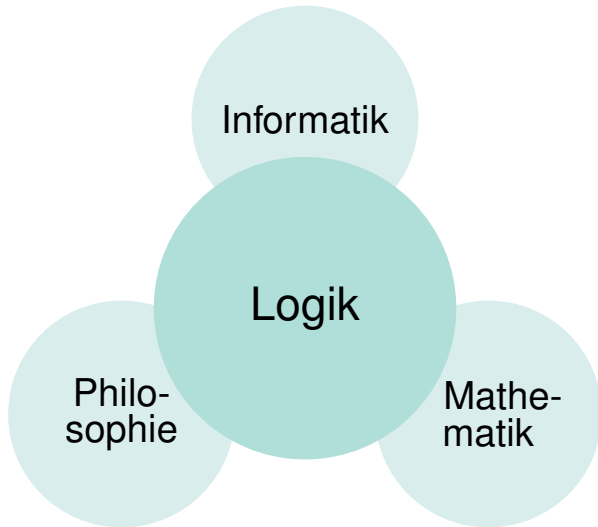
1. Klausurtermin

Mittwoch, 28.02.18, 11:00 Uhr

2. Klausurtermin

Nach dem Sommersemester 2018

Eine sehr kurze Geschichte der Logik



Was macht eine Logik aus?

Was macht eine Logik aus?

1. **Syntax**

ergibt die Sprache der Aussagen 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)

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,

Kompetenzen (s. auch Modulhandbuch)

Nach erfolgreichem Abschluss des Moduls

- ▶ verstehen Sie logische Grundbegriffe anhand verschiedener Logiken,
- ▶ können Sie deutsche Sätze als logische Aussagen formulieren,

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,

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,

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

2. Block: Prädikatenlogik

Logik, die über „Objekte“ spricht.

Logik, die über „Objekte“ spricht.

Nachts sind alle Katzen grau.

Miez ist eine Katze.

Miez ist rot.

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.)

2. Block: Prädikatenlogik

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.

Anwendung der Prädikatenlogik um Eigenschaften von Programmen zu beweisen.

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

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

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 (Spezifikationssprache)
- ▶ 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.

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