Technische
Universität
Dresden

Theoretische Informatik und Logik

1. Vorlesung: Willkommen/Einleitung/Übersicht

Markus Krötzsch

Professur Wissensbasierte Systeme

TU Dresden, 13. April 2026

Willkommen zur Vorlesung Theoretische Informatik und Logik

(ab Oktober 2026: “Logik und Komplexität”)

Raum, Zeit, URL

- **Vorlesungen:**
  montags DS3 (11:10–12:40) in HÜL/S386
  donnerstags DS4 (13:00–14:30) in HSZ/0002
- **Keine Vorlesungen:**
  Feiertag: 14.5. (Himmelfahrt)
  Pfingstferien: Mo 25.5., Do 28.5.
- **Vorlesungswebseite**:
  `https://iccl.inf.tu-dresden.de/web/TheoLog2026`
  (Folien, Übungsblätter, Termine, etc.)
- **Quellcode**:
  `https://github.com/knowsys/TheoLog`
  (Fragen, Bug-Reports, Pull-Requests sind willkommen)
- **Matrix (Chat)**:
  `https://matrix.tu-dresden.de/#/room/#theolog:tu-dresden.de`
  (Support bei allen Fragen zur Vorlesung)

Übungen

- **Anmeldung** zu den Übungen über OPAL
  (Link siehe Vorlesungswebseite; Beginn Einschreibung 13.4. um 15:00 Uhr)
- **Übungsblätter** jeweils freitags für übernächste Woche
- **Beginn der Übungen:** 20. April 2026
- **Übungsablauf**, vereinfacht, idealisiert:
  Aufgaben werden zu Hause bearbeitet so gut es geht;
  in der Übung helfen Gruppenleiter bei Fragen und Problemen und zeigen Beispiellösungen

Prüfung und Prüfungsvorbereitung

- schritliche Prüfung (90min) am Ende des Sommersemesters
- kann für alte und neue Studienordnungen belegt werden (INF-B-290, INF-D-330, INF-LE-EUI, INF-25-BA-LUK)
- prüfungsrelevant:
  kompletter Stoff aus Vorlesung und Übung;
  Wiedergeben (Definieren), Anwenden (Rechnen) und Erklären (Beweisen)
- Modulnote ergibt sich je nach Studiengang
- zur zusätzlichen Vorbereitung gibt es mehrere Konsultationen und eine Probeklausur

2. Wiederholungsprüfung TheoLog?

ACHTUNG!

- Im Oktober 2026 findet ein automatischer Übertritt in die neuen Studienordnungen statt
- TheoLog bestanden? $\leadsto$ Modulnote wird für INF-25-BA-LUK übernommen
- TheoLog nicht bestanden? $\leadsto$ Fehlversuche werden nicht übernommen
- Wer Sommer eine 2. Wiederholungsprüfung antritt und nicht besteht, beendet sein Studium vor dem Übertritt. Der zweite Fehlversuch kann nicht gelöscht werden.
- Für die 2. Wiederholung müssen Sie sich zum nächstmöglichen Termin (SoSe'26) anmelden, wenn (1) der schriftliche Bescheid zum "wiederholten Nichtbestehen" rechtzeitig zugestellt wurde und (2) Sie nicht in einem Urlaubssemester sind. Sie müssen die Prüfung dann auch ablegen (falls Sie nicht am Prüfungstag krank sind).

$\leadsto$ Informieren Sie sich über die für Sie geltenden Fristen!

Analoge Bemerkungen gelten für viele andere Prüfungen im SoSe 2026.
Siehe `https://tu-dresden.de/ing/informatik/studium/examination-office/ueberleitung-zwangsuebertritt`

Übersicht „TheoLog“ (vorläufige Planung)

Teil 1: Aussagenlogik

- Syntax und Semantik
- logisches Schließen (Resolution)
- Horn-Logik als Vereinfachung
- Praktische Anwendungen, SMT-Solver

Teil 2: Komplexität

- Zeit und Raum
- P und NP
- PSpace

Teil 3: Prädikatenlogik

- Syntax und Semantik
- Modell-Checking (Anwendung: Datenbanken)
- Logisches Schließen mit Resolution

Teil 4: Weitere Themen

- Datalog
- Modallogiken
- Gödel (wenn die Zeit ausreicht)

Literatur: Lehrbücher

Der Vorlesungsstoff gehört zu fast jeder Informatikausbildung. Es gibt viele Lehrmaterialien und eine weitgehend einheitliche Notation.

- Uwe Schöning: **Theoretische Informatik – kurz gefasst.** Spektrum Akademischer Verlag
  klassischer deutschsprachiger Text
- Michael Sipser: **Introduction to the Theory of Computation.** Cengage Learning
  Standardtext zu Sprachen und Berechnungskomplexität
- Christopher Moore, Stephan Meterns: **The Nature of Computation.** Oxford University Press
  moderner Text zu Komplexität und Berechnung, weniger formell

Zu speziellen Themen der Logik werden wir bei Gelegenheit noch Literaturangaben machen.

Literatur: Mehr Bücher

Teile des Stoffs der Vorlesung werden auch in vielen, zum Teil recht kurzweiligen Büchern weniger formal behandelt, z.B.:

- Apostolos Doxiadis, Christos Papadimitriou:
  Logicomix: An Epic Search for Truth. Bloomsbury
  Graphic Novel, inspiriert von Russels Leben und der Geschichte der Logik
- Scott Aaronson: **Quantum Computing Since Democritus.** Cambridge
  eher informeller Text mit interessanten Denkanstößen
- Douglas Hofstadter:
  Gödel, Escher, Bach: An Eternal Golden Braid. Basic Books
  der Klassiker

Warum?

Logik für InformatikerInnen: Darum (1)

$$\frac{\text{Logik}}{\text{Informatik}} = \frac{\text{Analysis}}{\text{Ingenieurwesen}}$$

„Wer rechnende Systeme verstehen und konstruieren will, der benötigt passende mathematische Modelle.
Dieser Weg führt oftmals zur Logik.“

- Modellierung von Programmlogik und logischen Schaltungen
- Berechnung praktisch relevanter Eigenschaften durch logisches Schließen
- Hauptanwendung: Verifikation von Systemen

Logik = Wissenschaft vom folgerichtigen Denken

„Wer intelligente Software entwickeln will, der muss logische Schlussfolgerungen algorithmisch umsetzen.
Die Logik liefert die nötigen Methoden.“

- Kodierung von gültigen Zusammenhängen und Regeln
- Logisches Schließen als Simulation von intelligentem Denken
- Hauptanwendung: Künstliche Intelligenz

Logisches Schließen = Problemlösen

„Bedeutende Klassen von (schweren) Problemen lassen sich
durch logische Schlussfolgerung lösen.
Algorithmen aus der Logik sind in vielen anderen Bereichen anwendbar.“

- Logik als Spezifikationssprache für komplexe Probleme
- Logisches Schließen als Suche nach zulässigen Lösungen
- Anwendungen: Constraint-Satisfaction-Probleme und verwandte Optimierungsaufgaben

Logik für InformatikerInnen: Darum (4)

Logiken = Beschreibungssprachen

„Überall wo Informationen maschinell kodiert werden und wo exakt
spezifiziert ist, was eine Anwendung
aus dieser Kodierung ableiten darf,
hat man es mit einer Art Logik zu tun.“

- Logik als Oberbegriff exakt spezifizierter Datenformate
- Schlussfolgerung zur Interpretation/Analyse/Optimierung
- Anwendungen: Wissensrepräsentation und Datenbanken

Was ist Logik?

„Logik“ ist ein allgemeiner Oberbegriff für viele mathematische und technische Formalismen, gekennzeichnet durch:

- Syntax: Sprache einer Logik (normalerweise Formeln mit logischen Operatoren)
- Semantik: Definition der Bedeutung (Worauf beziehen sich die Formeln? Wann ist eine Formel wahr oder falsch?)

Typische Zielstellung: Logische Schlussfolgerung

- Welche Schlüsse kann man aus einer gegebenen (Menge von) Formel(n) ziehen?
- Spezifikation der korrekten Schlussfolgerungen sollte sich aus Semantik ergeben
- Praktische Berechnung von Schlussfolgerungen ist oft kompliziert

Viele Logiken

Es gibt sehr viele Logiken, z.B.

- Aussagenlogik
- Prädikatenlogik (erster Stufe)
- Prädikatenlogik zweiter Stufe
- Beschreibungslogiken (Wissensrepräsentation und KI, z.B. Modallogik $\mathcal{ALC}$)
- Temporallogiken (z.B. Modallogik LTL)
- Logikprogramme (Datalog, Answer Set Programming, Prolog, . . .)
- Nichtklassische Logiken (z.B. intuitionistische Logik)
- Mehrwertige Logiken (z.B. probabilistische Logik)
- . . . und viele andere mehr

$\leadsto$ In dieser Vorlesung lernen wir zunächst Aussagenlogik kennen,
später dann Prädikatenlogik, Modallogiken und Datalog

Aussagenlogik

Aussagenlogik

Die Aussagenlogik untersucht logische Verknüpfungen von atomaren Aussagen.

Atomare Aussagen sind Behauptungen, die wahr oder falsch sein können, z.B.:

- A1 „Morgen schneit es.“
- A2 „Wir werden einen Schneemann bauen.“
- B1 „Die Vorstellung von der globalen Erderwärmung wurde von den Chinesen erfunden.“
- B2 „Die Temperatur der Ozeane ist seit 1998 in unverminderter Geschwindigkeit angestiegen.“
- C1 „Typ-2-Sprachen sind unter Vereinigung abgeschlossen.“
- C2 „Typ-2-Sprachen sind unter Komplement abgeschlossen.“
- C3 „Typ-2-Sprachen sind unter Schnitt abgeschlossen.“

Aussagen verknüpfen

Atomare Aussagen können mithilfe logischer Junktoren (oder Operatoren) verknüpft werden:

- $A1 \to A2$: „Wenn es morgen schneit, dann werden wir einen Schneemann bauen."
- $A1 \vee \neg A1$: „Entweder schneit es morgen oder nicht."
- $B1 \to \neg B2$: „Falls die Chinesen die globale Erwärmung erfunden haben, dann steigt die Temperatur der Ozeane nicht unvermindert an."
- $(C1 \wedge C2) \to C3$: „Sind Typ-2-Sprachen unter Vereinigung und Komplement abgeschlossen, dann sind sie auch unter Schnitt abgeschlossen."

Logisches Schließen

Wenn einige Formeln als wahr angenommen werden, dann kann die Wahrheit anderer Formeln daraus abgeleitet werden.

Beispiele:

- Aus B2 und B1 $\to$ ¬B2 folgt ¬B1
- Aus C1, ¬C3 und $(\text{C1} \wedge \text{C2}) \to \text{C3}$ folgt ¬C2
- Aus A1 $\to$ A2 und ¬A1 folgt nicht ¬A2

Die Gültigkeit bestimmter Schlussfolgerungen hat nichts mit Schneemännern, Erderhitzung oder Typ-2-Sprachen zu tun!
Sie ist eine rein logische Konsequenz.

Beispiel: Logelei

Anna behauptet: „Barbara lügt!“

Barbara behauptet: „Chris lügt!“

Chris behauptet: „Anna und Barbara lügen!“

Wer lügt?

(Und wie kann man das beweisen?)

Aussagenlogik: Syntax

Wir betrachten eine abzählbar unendliche Menge **P** von atomaren Aussagen (auch bekannt als: aussagenlogische Variablen, Propositionen oder schlicht Atome)

Die Menge der aussagenlogischen Formeln ist induktiv[a] definiert:

- Jedes Atom $p \in \mathbf{P}$ ist eine aussagenlogische Formel
- Wenn F und G aussagenlogische Formeln sind, so auch:
  - $\neg F$: Negation, „nicht F“
  - $(F \wedge G)$: Konjunktion, „F und G“
  - $(F \vee G)$: Disjunktion, „F oder G“
  - $(F \to G)$: Implikation, „F impliziert G“
  - $(F \leftrightarrow G)$: Äquivalenz, „F ist äquivalent zu G“

[a] Das bedeutet: Die Definition ist selbstbezüglich und soll die kleinste Menge an Formeln beschreiben, die alle Bedingungen erfüllen.

Wir verzichten hier oft auf „aussagenlogisch“ und sprechen z.B. einfach von „Formeln“.

Beispiele

Die folgenden Ausdrücke sind aussagenlogische Formeln:

- p
- $((p \to q) \to p) \to p$
- $\neg\neg\neg p$

Die folgenden Ausdrücke sind keine aussagenlogischen Formeln:

- $p \wedge q \vee r$ (fehlende Klammern)

  Vereinfachung: Äußere Klammern dürfen wegfallen, d.h. wir erlauben z.B. $p \to q$ anstatt auf $(p \to q)$ zu bestehen.

- $(p \leftarrow q)$ (Operator $\leftarrow$ undefiniert)
- $\overline{p \wedge q}$ (wir verwenden $\bar{\ }$ nicht als Negation)

Teilformeln

Wir können Formeln als Wörter über (endlichen Teilmengen aus) dem unendlichen Alphabet $\mathbf{P} \cup \{\neg, \wedge, \vee, \rightarrow, \leftrightarrow, (,)\}$ sehen:

$$\mathsf{F} \rightarrow \mathbf{P} \mid \neg\mathsf{F} \mid (\mathsf{F}\wedge\mathsf{F}) \mid (\mathsf{F}\vee\mathsf{F}) \mid (\mathsf{F}\rightarrow\mathsf{F}) \mid (\mathsf{F}\leftrightarrow\mathsf{F})$$

Eine Teilformel ist ein Teilwort (Infix) einer Formel, welches selbst eine Formel ist. Teilformeln werden auch Unterformeln genannt.

Alternativ kann man Teilformeln auch rekursiv definieren:

Die Menge $\mathsf{Sub}(F)$ der Teilformeln einer Formel F ist definiert als:

$$\mathsf{Sub}(F) = \begin{cases} \{F\} & \text{falls } F \in \mathbf{P} \\ \{\neg G\} \cup \mathsf{Sub}(G) & \text{falls } F = \neg G \\ \{(G_1 \wedge G_2)\} \cup \mathsf{Sub}(G_1) \cup \mathsf{Sub}(G_2) & \text{falls } F = (G_1 \wedge G_2) \\ \{(G_1 \vee G_2)\} \cup \mathsf{Sub}(G_1) \cup \mathsf{Sub}(G_2) & \text{falls } F = (G_1 \vee G_2) \\ \{(G_1 \rightarrow G_2)\} \cup \mathsf{Sub}(G_1) \cup \mathsf{Sub}(G_2) & \text{falls } F = (G_1 \rightarrow G_2) \\ \{(G_1 \leftrightarrow G_2)\} \cup \mathsf{Sub}(G_1) \cup \mathsf{Sub}(G_2) & \text{falls } F = (G_1 \leftrightarrow G_2) \end{cases}$$

Semantik

Was bedeutet eine aussagenlogische Formel?

- Atome an sich bedeuten zunächst nichts
  $\leadsto$ sie können einfach wahr oder falsch sein
- Je nachdem, welche Atome wahr sind und welche falsch, ergeben sich verschiedene „Interpretationen“
  $\leadsto$ dargestellt durch Wertzuweisungen
  (Funktionen von **P** nach $\{1, 0\}$; 1=„wahr“ und 0=„falsch“)
- Die Wahrheit (oder Falschheit) einer Formel ergibt sich aus dem Wahrheitswert der in ihr vorkommenden Atome
  $\leadsto$ Wertzuweisungen machen Formeln wahr oder falsch

Die Bedeutung einer Formel besteht darin, dass sie uns Informationen darüber liefert, welche Wertzuweisungen möglich sind, falls die Formel wahr sein soll.

Wertzuweisungen können Formeln erfüllen

Eine Wertzuweisung ist eine Funktion $w : \mathbf{P} \to \{1, 0\}$

Eine Wertzuweisung w erfüllt eine Formel F, in Symbolen $w \models F$, wenn eine der folgenden rekursiven Bedingungen gilt:

Form von F	$w \models F$ wenn:	$w \not\models F$ wenn:
$F \in \mathbf{P}$:	$w(F) = 1$	$w(F) = 0$
$F = \neg G$	$w \not\models G$	$w \models G$
$F = (G_1 \wedge G_2)$	$w \models G_1$ und $w \models G_2$	$w \not\models G_1$ oder $w \not\models G_2$
$F = (G_1 \vee G_2)$	$w \models G_1$ oder $w \models G_2$	$w \not\models G_1$ und $w \not\models G_2$
$F = (G_1 \to G_2)$	$w \not\models G_1$ oder $w \models G_2$	$w \models G_1$ und $w \not\models G_2$
$F = (G_1 \leftrightarrow G_2)$	$w \models G_1$ und $w \models G_2$ oder $w \not\models G_1$ und $w \not\models G_2$	$w \models G_1$ und $w \not\models G_2$ oder $w \not\models G_1$ und $w \models G_2$

Dabei bedeutet „A oder B“ immer „A oder B oder beides“.

Formeln Wahrheitswerte zuweisen

Wir können Wertzuweisungen von Atomen auf Formeln erweitern:

$$w(F) = \begin{cases} 1 & \text{falls } w \models F \\ 0 & \text{falls } w \not\models F \end{cases}$$

Wahrheitswertetabellen illustrieren die Semantik der Junktoren:

$w(F)$	$w(\neg F)$
0	1
1	0

$w(F)$	$w(G)$	$w(F \land G)$	$w(F \lor G)$	$w(F \to G)$	$w(F \leftrightarrow G)$
0	0	0	0	1	1
1	0	0	1	0	0
0	1	0	1	1	0
1	1	1	1	1	1

Wahrheitswerte von Formeln bestimmen

- Der Wahrheitswert einer Formel hängt nur vom Wahrheitswert der (endlich vielen) Atome ab, die in ihr vorkommen.
  $\rightsquigarrow$ Wir geben oft nur diese an.
- Der Wahrheitswert einer Formel ergibt sich rekursiv aus dem Wahrheitswert ihrer Teilformeln.
  $\rightsquigarrow$ Darstellung in Wahrheitswertetabelle

Beispiel: Für die Formel $F = ((p \to q) \leftrightarrow (\neg q \to \neg p))$ und Wertzuweisung w mit $w(p) = 0$ und $w(q) = 1$ ergibt sich die folgende Tabelle:

$w(p)$	$w(q)$	$w(p \to q)$	$w(\neg p)$	$w(\neg q)$	$w(\neg q \to \neg p)$	$w(F)$
0	1	1	1	0	1	1

Beispiel: Logelei

Anna behauptet: „Barbara lügt!“ $A \leftrightarrow \neg B$

Barbara behauptet: „Chris lügt!“ $B \leftrightarrow \neg C$

Chris behauptet: „Anna und Barbara lügen!“ $C \leftrightarrow (\neg A \wedge \neg B)$

Wer lügt?

(Und wie kann man das beweisen?)

Zusammenfassung und Ausblick

Logik, logisches Modellieren und logisches Schlussfolgern hat in der Informatik viele direkte und indirekte Anwendungen

Mithilfe der Aussagenlogik kann man logische Beziehungen atomarer Aussagen spezifizieren

Wertzuweisungen können eine aussagenlogische Formel erfüllen – dann nennt man sie Modell – oder widerlegen

Offene Fragen:

- Geht logisches Schließen auch ohne Wahrheitswertetabellen?
- Wie (in)effizient ist logisches Schließen?
- Was hat das mit Komplexität zu tun?