

# Logik

Prof. Dr. Madlener

TU Kaiserslautern

SS 2010

Studiengang „Informatik“, „Technoinformatik“ und „ WiWi/Inf“  
SS'10

Prof. Dr. Madlener TU - Kaiserslautern

Vorlesung:

Mi 11.45-13.15 52/207

- ▶ Informationen  
<http://www-madlener.informatik.uni-kl.de/teaching/ss2010/logik/logik.html>
- ▶ Grundlage der Vorlesung: Skript  
Einführung in die Logik und Korrektheit von Programmen.

- ▶ **Bewertungsverfahren:**  
Zulassungsvoraussetzungen zu Abschlussklausur:  
Übungen: mind. 50 %  
Aufsichtsarbeit: mind. 50 %
- ▶ **Aufsichtsarbeit:** Vor. Freitag 21/05/10
- ▶ **Abschlussklausur:** Dienstag, 2010/08/10
- ▶ **Übungen:** Gruppen  
Einschreiben, Sprechzeiten siehe Homepage

## Grundlagen der Aussagenlogik

Syntax

Semantik

Deduktiver Aufbau der Aussagenlogik

Natürliche Kalküle

## Algorithmischer Aufbau der Aussagenlogik

Semantische Tableaux

Normalformen

Davis-Putnam-Algorithmen

Resolutions-Verfahren

## Grundlagen der Prädikatenlogik

Beziehungen zwischen Eigenschaften von Elementen

Semantik der P-Logik 2-Stufe – Interpretationen, Belegungen, Bewert

Transformationen von Termen und Formeln

## Entscheidbarkeit in der Prädikatenlogik

Unentscheidbarkeit der Allgemeingültigkeit

Hauptsätze der Prädikatenlogik erster Stufe

Theorien erster Stufe

## Algorithmen der Prädikatenlogik

Aufzählungsverfahren für PL-1

Resolventenmethode – (Allg. Resolutionsverfahren)

Logisches Programmieren und Prolog

# Einleitung

## Methoden zur Lösung von Problemen mit Hilfe von Rechnern Formalisierung ( $\equiv$ Festlegung)

- ▶ **Logik::** „Lehre vom folgenrichtigen Schließen“ bzw. „Lehre von formalen Beziehungen zwischen Denkinhalten“

## Zentrale Fragen: Wahrheit und Beweisbarkeit von Aussagen $\rightsquigarrow$ **Mathematische Logik.**

- ▶ **Logik in der Informatik:**
  - ▶ **Aussagenlogik:** Boolesche Algebra. Logische Schaltkreise (Kontrollsystemen), Schaltungen, Optimierung.
  - ▶ **Prädikatenlogik:** Spezifikation und Verifikation von Softwaresystemen.
  - ▶ **Modal- und Temporallogik:** Spezifikation und Verifikation reaktiver Systeme.

# Logik in der Informatik

1. Semantik von Programmiersprachen (Hoarscher Kalkül).
2. Spezifikation von funktionalen Eigenschaften.
3. Verifikationsprozess bei der SW-Entwicklung.  
Beweise von Programmeigenschaften.
4. Spezielle Programmiersprachen (z.B. PROLOG)

## ▶ **Automatisierung des logischen Schließens**

1. Automatisches Beweisen (Verfahren,...)
2. Grundlagen von Informationssystemen (Verarbeitung von Wissen, Reasoning,...)

# Voraussetzungen

1. **Mathematische Grundlagen.** Mengen, Relationen, Funktionen. Übliche Formalisierungen: „Mathematische Beweise“, Mathematische Sprache, d.h. Gebrauch und Bedeutung der üblichen Operatoren der naiven Logik. Also Bedeutung von **nicht, und, oder, impliziert, äquivalent, es gibt, für alle.**
2. **Grundlagen zur Beschreibung formaler Sprachen.** Grammatiken oder allgemeiner **Kalküle** (Objektmenge und Regeln zur Erzeugung neuer Objekte aus bereits konstruierter Objekte), Erzeugung von Mengen, Relationen und Funktionen, Hüllenoperatoren (Abschluss von Mengen bzgl. Relationen).
3. **Vorstellung von Berechenbarkeit**, d.h. entscheidbare und rek.aufzählbare Mengen, Existenz nicht entscheidbarer Mengen und nicht berechenbarer Funktionen.

# Berechnungsmodelle/Programmiersprachen

Algorithmische Unlösbarkeit?

prinzipielle Lösbarkeit



effiziente Lösbarkeit



algorithmischer Entwurf



$P$ : Programm in einer HPS



Problem  
Spezifikation

(Formalisiert)

# Syntaktische und semantische Verifikation von $P$ .

- ▶ **Syntaxanalyse**

- Sprachen Chomski-Hierarchie
  - Kontext freie Sprachen
  - Grammatiken/Erzeugungsprozess

- ▶ **Programmverifikation**

- Tut  $P$  auch was erwartet wird.
  - Gilt  $P \rightsquigarrow$  Problem Spezifikation

## Typische Ausdrücke

- ▶  $(x + 1)(y - 2)/5$  Terme als Bezeichner von Objekten.
- ▶  $3 + 2 = 5$  Gleichungen als spezielle Formeln.
- ▶ „29 ist (k)eine Primzahl “ Aussagen.
- ▶ „ $3 + 2 = 5$  und 29 ist keine Primzahl “ Aussage.
- ▶ „wenn 29 keine Primzahl ist, dann ist  $0 = 1$  “ Aussage.
- ▶ „jede gerade Zahl, die größer als 2 ist, ist die Summe zweier Primzahlen “ Aussage.
- ▶  $2 \leq x$  und  $(\forall y \in \mathbb{N})$   
 $((2 \leq y$  und  $y + 1 \leq x) \rightarrow$  nicht  $(\exists z \in \mathbb{N})y * z = x)$   
Aussage.

## Typische Ausdrücke (Fort.)

- ▶  $(\forall X \subseteq \mathbb{N})(0 \in X \wedge (\forall x \in \mathbb{N})(x \in X \rightarrow x + 1 \in X) \rightarrow X = \mathbb{N})$   
**Induktionsprinzip.**
- ▶  $(\forall X \subseteq \mathbb{N})(X \neq \emptyset \rightarrow X \text{ hat ein kleinstes Element})$   
Jede nichtleere Menge natürlicher Zahlen enthält ein minimales Element.

**Zweiwertige Logik** Jede Aussage ist entweder **wahr** oder **falsch**.

- ▶ Es gibt auch andere Möglichkeiten (Mehrwertige Logik).
- ▶ Prädikatenlogik erster Stufe (PL1): Nur Eigenschaften von Elementen und Quantifizierung von Elementvariablen erlaubt.



# Kapitel I

## Grundlagen der Aussagenlogik

























