

Logik und Logikprogrammierung

Dietrich Kuske

FG Automaten und Logik, TU Ilmenau

Sommersemester 2021

Organisatorisches zum Modul

zwei Teile:

- „Logik“ (Prof. Kuske, 15 Vorlesungen)
- „Logikprogrammierung“ (Prof. Knauf, 7 Vorlesungen)

Bewertung: eine gemeinsame Klausur aus zwei Teilen im Verhältnis 2:1

Informationen finden Sie über

www.tu-ilmenau.de/al

Literaturempfehlung:

- M. Huth und M. Ryan „Logic in Computer Science“, Cambridge 2010.
- S. Hölldobler „Logik und Logikprogrammierung“, Krottenmühl 2009.

Ablauf

- auf der genannten Seite und im Moodle werden die aufgezeichneten Vorlesungen des SS20 veröffentlicht
- am Montag (17 Uhr) gibt es die Möglichkeit, Ihre Fragen online mit Herrn Köcher zu besprechen

Organisatorisches zur Übung im Logikteil

- Die Übungen werden von Herrn Hugenroth organisiert und gehalten.
- Übungsaufgaben finden Sie ca. eine Woche vor der Übung im Netz.
- Sie bereiten diese **schriftlich** vor und geben Ihre Lösungen bis **Montag 13 Uhr** ab.
- Die Aufgaben werden korrigiert und dafür Punkte vergeben.
- Diese Übungspunkte werden dann als **Bonus** auf die Klausur angerechnet.
- In den Übungen werden mögliche Lösungen und Probleme besprochen.

Arbeitsweise

- 1 Sie verfolgen natürlich jede Vorlesung aktiv.
- 2 Aber der Stoff ist zu anspruchsvoll, um durch alleiniges Hören verstanden zu werden.
- 3 Daher werden Sie den Vorlesungsstoff semesterbegleitend nacharbeiten: Definitionen („Konzepte“) und Sätze („Sachverhalte“) herschreiben und auswendig lernen, Beweise („Begründungen“) verstehen (= wiedergeben können), weitere Literatur zu Rate ziehen
- 4 Sie drucken die Übungsblätter lange vor dem Übungstermin aus, lesen sie genau, überlegen eine Lösung, schreiben diese auf und arbeiten an Lösungen in den Übungen mit.
- 5 Auch Übungen werden semesterbegleitend nachgearbeitet.
- 6 Bei jeder Veranstaltung haben Sie sämtliche Unterlagen griffbereit zum Nachschlagen.
- 7 **Bei Verständnisproblemen fragen Sie bitte frühzeitig!**

- 5 Leistungspunkte ergeben **150 h** Arbeitsaufwand, d.h. 10 h / Woche
- 1¹/₂ Vorlesungen, eine Übung: 3³/₄ h / Woche
- es bleiben also

6¹/₄ h / Woche für das Selbststudium

= Vorlesungs- und Übungsvor- und -nachbereitung

Die Logik

- versucht, gültige Argumentationen von ungültigen zu unterscheiden,

Aristoteles (384–322 v.Chr.) untersuchte das Wesen der **Argumentation** und des **logischen Schließens** mit dem Ziel, korrekte von inkorrekten Argumenten zu unterscheiden.

Verschiedene Werke, u.a. *Analytica priora*, *Analytica posteriora*.

Aristoteles nennt die logischen Schlußfolgerungen **Syllogismen** (griechisch: „Zusammenrechnung“).

*Ein Syllogismus ist eine Aussage, in der bestimmte Dinge [die **Prämissen**] behauptet werden und in der etwas anderes [die **Konsequenz**], unumgänglich aus dem Behaupteten folgt. Mit dem letzten Satz meine ich, daß die Prämissen die Konsequenz zum Resultat haben, und damit meine ich, daß keine weitere Prämisse erforderlich ist, um die Konsequenz unumgänglich zu machen.*

(Übersetzung aus der englischen Übersetzung des Electronic Text Center, University of Virginia Library)

Beispiele

*Wenn alle Menschen sterblich sind und
Sokrates ein Mensch ist,
dann ist Sokrates sterblich.*

*Wenn eine Zahl gerade und größer als zwei ist,
dann ist sie keine Primzahl.*

*Wenn die Leitzinsen hoch sind,
dann sind die Börsianer unzufrieden.*

Aristoteles identifizierte einige zulässige Syllogismen, die Scholastiker fügten weitere hinzu:

Alle Dackel sind Hunde	Alle P sind M	(Barbara)
Alle Hunde sind Tiere	Alle M sind S	
<hr/> Dann sind alle Dackel Tiere	<hr/> Alle P sind S	

Keine Blume ist ein Tier	Kein P ist M	(Cesare)
Alle Hunde sind Tiere	Alle S sind M	
<hr/> Dann ist keine Blume ein Hund	<hr/> Kein P ist S	

Alle Delfine leben im Meer	Alle M sind P	(Darapti)
Alle Delfine sind Säugetiere	Alle M sind S	
<hr/> Dann leben einige Säugetiere im Meer	<hr/> Einige S sind P	

Gottfried Wilhelm Leibniz (1646-1716) wollte korrekte von inkorrekten Argumentationsketten unterscheiden. Hierzu sollte ein Kalkül entwickelt werden, in dem alle korrekten Argumentationsketten ermöglicht sind (und keine inkorrekten).

David Hilbert (1862-1943) entwickelte solche Kalküle.

Diese „Hilbertkalküle“ sind sehr verschieden von üblichen Argumentationsmustern.

Gerhard Gentzen (1909-1945) entwickelte Kalküle des „natürlichen Schließens“, die übliche Argumentationsmuster formalisieren.

Die Aussagenlogik

George Boole (1815 – 1864) entwickelte einen Kalkül zum Rechnen mit **atomare Aussagen**, die entweder **wahr** oder **falsch** sein können.

Verknüpfung durch **Operatoren** (und; oder; nicht; wenn-dann ...).

Die Prädikatenlogik (Ende des 19. Jahrhunderts)

Gottlob Frege (1848-1925), Giuseppe Peano (1858-1932) und Bertrand Russell (1872-1970) entwickelten die Logik zur Grundlage der Mathematik, als formale Basis für die Vermeidung von Widersprüchen.

Entwicklung der Prädikatenlogik, die erlaubt:

- **Beziehungen** zwischen „Objekten“ zu beschreiben
- **existentielle Aussagen** zu treffen: „es gibt ein x , so daß ...“
- **universelle Aussage** zu treffen: „für jedes x gilt, daß ...“

Die Logik

- versucht, gültige Argumentationen von ungültigen zu unterscheiden,
- hat Anwendungen in der Informatik,

Claude Shannon (1916 – 2001) benutzt die Aussagenlogik 1937, um elektromechanische Schaltkreise zu beschreiben und zu optimieren.

Allen Newell (1927-1992), Herbert Simon (1916-2001) und Alan Robinson (1930-2016) entwickelten 1950-1960 die ersten Systeme für die Automatisierung des logischen Schließens als Werkzeug der Künstlichen Intelligenz.

- **Schaltkreisentwurf:** Schaltkreise lassen sich durch logische Formeln darstellen \leadsto Entwurf und Optimierung von Schaltungen
- **Modellierung und Spezifikation:** Eindeutige Beschreibung von komplexen Systemen
- **Verifikation:** Beweisen, daß ein Programm das gewünschte Verhalten zeigt
- **Datenbanken:** Formulierung von Anfragen an Datenbanken
 \leadsto Abfragesprache SQL (Structured Query Language)
- (klassische) **Künstliche Intelligenz:**
 - Planung
 - Mensch-Maschine Kommunikation
 - Theorembeweiser: Der Computer beweist mathematische Sätze \leadsto automatischer Beweis von wichtigen Sätzen im Bereich der Booleschen Algebren
- **Logische Programmiersprachen:** Prolog

Außerdem: Logik ist ein Paradebeispiel für **Syntax** und formale **Semantik**

Edsger W. Dijkstra (1920-2002):

Informatik = VLSAL (Very large scale application of logics)

Die Logik

- versucht, gültige Argumentationen von ungültigen zu unterscheiden,
- hat Anwendungen in der Informatik,
- formalisiert die zu untersuchenden Aussagen und

1. Problem: Zuordnung von Wahrheitswerten zu natürlichsprachigen Aussagen ist problematisch.

Beispiele:

- Ich habe nur ein bißchen getrunken.
- Sie hat sich in Rauch aufgelöst.
- Das gibt es doch nicht!
- Rache ist süß.

2. Problem: Natürliche Sprache ist oft schwer verständlich.

Beispiel: Auszug aus der „Analytica Priora“ von Aristoteles

Die Aussage: Wenn der Mittelbegriff sich universell auf Ober- oder Untersatz bezieht, muss ein bestimmter negativer Syllogismus resultieren, immer wenn der Mittelbegriff sich universell auf den Obersatz bezieht, sei es positiv oder negativ, und besonders wenn er sich auf den Untersatz bezieht und umgekehrt zur universellen Aussage.

Der Beweis: Denn wenn M zu keinem N gehört, aber zu einem O, ist es notwendig, dass N zu einem O nicht gehört. Denn da die negative Aussage umsetzbar ist, wird N zu keinem M gehören: Aber es war erlaubt, dass M zu einem O gehört: Deshalb wird N zu einem O nicht gehören: Denn das Ergebnis wird durch die erste Figur erreicht. Noch einmal: Wenn M zu allen N gehört, aber nicht zu einem O, ist es notwendig, dass N nicht zu einem O gehört: Denn wenn N zu allen O gehört und M auch alle N-Eigenschaften zugeschrieben werden, muss M zu allen O gehören: Aber wir haben angenommen, dass M zu einem O nicht gehört. Und wenn M zu allen N gehört, aber nicht zu allen O, können wir folgern, dass N nicht zu allen O gehört: Der Beweis ist der gleiche wie der obige. Aber wenn M alle O-Eigenschaften zugeschrieben werden, aber nicht alle N-Eigenschaften, wird es keinen Syllogismus geben.

3. Problem: Natürliche Sprache ist mehrdeutig.

Beispiel:

Ich sah den Mann auf dem Berg mit dem Fernrohr.



((((Ich sah den Mann) auf dem Berg) mit dem Fernrohr)



((Ich sah (den Mann auf dem Berg)) mit dem Fernrohr)



((Ich sah den Mann) (auf dem Berg mit dem Fernrohr))



(Ich sah ((den Mann auf dem Berg) mit dem Fernrohr))



(Ich sah (den Mann (auf dem Berg mit dem Fernrohr)))

4. Problem: Natürliche Sprache hängt von Kontext ab.

Die Beatles sind Musiker
Paul McCartney ist ein Beatle

Paul McCartney ist ein Musiker

Die Beatles sind vier
Paul McCartney ist ein Beatle

Paul McCartney ist vier

Die Logik

- versucht, gültige Argumentationen von ungültigen zu unterscheiden,
- hat Anwendungen in der Informatik,
- formalisiert die zu untersuchenden Aussagen und
- beschränkt sich auf einen wohldefinierten Teil der möglichen Aussagen
⇒ es gibt verschiedene „Logiken“, z.B. „Aussagen-“ und „Prädikatenlogik“ (in dieser Vorlesung)

daneben: temporale Logiken (Mastervorlesung „Verifikation“)

modale Logiken

epistemische Logiken

...

Kapitel 1: Aussagenlogik

Beispiel

Ein Gerät besteht aus einem Bauteil *A*, einem Bauteil *B* und einem roten Licht. Folgendes ist bekannt:

- 1) Bauteil *A* oder Bauteil *B* (oder beide) sind kaputt.
- 2) Wenn Bauteil *A* kaputt ist, dann ist auch Bauteil *B* kaputt.
- 3) Wenn Bauteil *B* kaputt ist und das rote Licht leuchtet, dann ist Bauteil *A* nicht kaputt.
- 4) Das rote Licht leuchtet.

Zur Formalisierung verwenden wir folgende Abkürzungen: RL (rotes Licht leuchtet), AK (Bauteil A kaputt), BK (Bauteil B kaputt), \vee (oder), \rightarrow (wenn, dann), \wedge (und) und \neg (nicht).

Damit können wir unser Wissen kompakter hinschreiben:

- 1) $AK \vee BK$
- 2) $AK \rightarrow BK$
- 3) $(BK \wedge RL) \rightarrow \neg AK$
- 4) RL

- 6) Falls AK gilt, so folgt aus $AK \rightarrow BK$, daß BK gilt.
- 7) Falls BK gilt, so gilt natürlich BK .
- 8) Da $AK \vee BK$ gilt, folgt aus (5) und (6), daß BK in jedem Fall gilt.
- 9) Es gilt auch RL .
- 10) Also gilt $BK \wedge RL$ (aus (7) und (8)).
- 11) Es gilt auch $(BK \wedge RL) \rightarrow \neg AK$.
- 12) Also gilt $\neg AK$ (aus (9) und (10)).

Damit sind wir überzeugt, daß das Bauteil A heil ist.

In der Aussagenlogik gehen wir von „Aussagen“ aus, denen wir (zumindest prinzipiell) Wahrheitswerte zuordnen können.

Beispiele

- Die Summe von 3 und 4 ist 7.
- Jana reagierte aggressiv auf Martins Behauptungen.
- Jede gerade natürliche Zahl > 2 ist Summe zweier Primzahlen.
- Alle Marsmenschen mögen Pizza mit Pepperoni.
- Albert Camus était un écrivain français.
- In theory, practically everything is possible.

Für diese Aussagen verwenden wir die **atomaren Formeln** p, q, r bzw.

p_0, p_1, \dots

Die Aussagen werden durch „Operatoren“ verbunden.

Beispiele

- ... und ...
- ... oder ...
- nicht ...
- wenn ... dann ...
- entweder ... oder ..., aber nicht beide.
- mehr als die Hälfte der Aussagen ... gilt.

Für solche zusammengesetzten Aussagen verwenden wir φ , ψ usw.

Durch die Wahl der erlaubten Operatoren erhält man unterschiedliche „Logiken“.

Da der Wahrheitswert einer zusammengesetzten Aussage nur vom Wahrheitswert der Teilaussagen abhängen soll, sind Operatoren wie „weil“ oder „obwohl“ nicht zulässig.

Syntax der Aussagenlogik

Eine **atomare Formel** hat die Form p_i (wobei $i \in \mathbb{N} = \{0, 1, \dots\}$).

Formeln werden durch folgenden induktiven Prozeß definiert:

- 1 Alle atomaren Formeln und \perp sind Formeln.
- 2 Falls φ und ψ Formeln sind, sind auch $(\varphi \wedge \psi)$, $(\varphi \vee \psi)$, $(\varphi \rightarrow \psi)$ und $\neg\varphi$ Formeln.
- 3 Nichts ist Formel, was sich nicht mittels der obigen Regeln erzeugen läßt.

Beispielformel: $\neg((\neg p_4 \vee p_1) \wedge \perp)$

Bezeichnungen:

- Falsum: \perp
- Konjunktion: \wedge
- Disjunktion: \vee
- Implikation: \rightarrow
- Negation: \neg

Abkürzungen

$p, q, r \dots$ statt $p_0, p_1, p_2 \dots$

$\left(\bigvee_{i=1}^n \varphi_i \right)$ statt $\left(\dots \left((\varphi_1 \vee \varphi_2) \vee \varphi_3 \right) \vee \dots \vee \varphi_n \right)$

$\left(\bigwedge_{i=1}^n \varphi_i \right)$ statt $\left(\dots \left((\varphi_1 \wedge \varphi_2) \wedge \varphi_3 \right) \wedge \dots \wedge \varphi_n \right)$

$(\varphi \leftrightarrow \psi)$ statt $((\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi))$

Präzedenz der Operatoren:

- \leftrightarrow bindet am schwächsten
- \rightarrow ...
- \vee ...
- \wedge ...
- \neg bindet am stärksten

Es gilt also z.B.:

$$(\alpha \leftrightarrow \beta \vee \neg \gamma \rightarrow \delta \wedge \neg \eta) = \left(\alpha \leftrightarrow \left((\beta \vee \neg \gamma) \rightarrow (\delta \wedge \neg \eta) \right) \right)$$

Dennoch: Zu viele Klammern schaden i.A. nicht.