Logik und Logikprogrammierung

Dietrich Kuske

FG Automaten und Logik, TU Ilmenau

Sommersemester 2022

Organisatorisches zum Modul

zwei Teile:

- „Logik“ (Prof. Kuske, 15 Vorlesungen)
- „Logikprogrammierung“ (Prof. Knauf, 6 Vorlesungen)

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

Organisatorisches zur Vorlesung im Logikteil

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.

Organisatorisches zur Übung im Logikteil

- Die Übungen werden von Herrn Schwarz 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** am Beginn der Übung oder im Briefkasten vor dem Büro Z1047 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“) herausschreiben 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!

Ressourcen

- 5 Leistungspunkte ergeben 150 h Arbeitsaufwand, d.h. 10 h / Woche
- $1^1/_2$ Vorlesungen, eine Übung: $3^3/_4$ h / Woche
- es bleiben also

$6^1/_4$ h / Woche für das Selbststudium

= Vorlesungs- und Übungsvor- und -nachbereitung

Die Logik

- versucht, gültige Argumentationen von ungültigen zu unterscheiden,

Syllogismen

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:

$$\begin{array}{l} \text{Alle Dackel sind Hunde} \\ \text{Alle Hunde sind Tiere} \\ \hline \text{Dann sind alle Dackel Tiere} \end{array} \qquad \begin{array}{l} \text{Alle P sind M} \\ \text{Alle M sind S} \\ \hline \text{Alle P sind S} \end{array} \qquad \text{(Barbara)}$$

$$\begin{array}{l} \text{Keine Blume ist ein Tier} \\ \text{Alle Hunde sind Tiere} \\ \hline \text{Dann ist keine Blume ein Hund} \end{array} \qquad \begin{array}{l} \text{Kein P ist M} \\ \text{Alle S sind M} \\ \hline \text{Kein P ist S} \end{array} \qquad \text{(Cesare)}$$

$$\begin{array}{l} \text{Alle Delfine leben im Meer} \\ \text{Alle Delfine sind Säugetiere} \\ \hline \text{Dann leben einige Säugetiere im Meer} \end{array} \qquad \begin{array}{l} \text{Alle M sind P} \\ \text{Alle M sind S} \\ \hline \text{Einige S sind P} \end{array} \qquad \text{(Darapti)}$$

Kalküle

Gottfried Wilhelm Leibniz (1646-1716) wollte korrekte von inkorrekten Argumentations*ketten* 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 Bertand 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,

Logik 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 $\rightsquigarrow$ 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 $\rightsquigarrow$ Abfragesprache SQL (Structured Query Language)
- (klassische) Künstliche Intelligenz:
    - Planung
    - Mensch-Maschine Kommunikation
    - Theorembeweiser: Der Computer beweist mathematische Sätze $\rightsquigarrow$ 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

Probleme mit natürlicher Sprache

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.

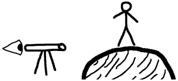
Der Beweis: Denn wenn M zu keinem N gehört, aber zu einem O, ist es notwendig, dass N zu einem O nicht gehört. Denn da die negative Aussage umsetzbar ist, wird N zu keinem M gehören: Aber es war erlaubt, dass M zu einem O gehört: Deshalb wird N zu einem O nicht gehören: Denn das Ergebnis wird durch die 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 vom 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

  $\Longrightarrow$ 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

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

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

Den Beweis, daß das Teil A heil ist, werden wir als „Beweisbaum“ formalisieren:

$$\dfrac{\dfrac{\dfrac{AK \lor BK \qquad \dfrac{[AK] \quad AK \to BK}{BK} \qquad [BK]}{BK} \qquad RL}{BK \land RL} \qquad (BK \land RL) \to \neg AK}{\neg AK}$$

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.
- Das Glas ist halb voll.
- Albert Camus était un écrivain français.
- In theory, practically everything is possible.

Für diese Aussagen verwenden wir die atomaren Formeln p, q, r bzw. $p_0, p_1, \ldots$.

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 $\bot$ sind Formeln.
2. Falls φ und ψ Formeln sind, sind auch $(\varphi \land \psi)$, $(\varphi \lor \psi)$, $(\varphi \to \psi)$ und $\neg\varphi$ Formeln.
3. Nichts ist Formel, was sich nicht mittels der obigen Regeln erzeugen läßt.

Beispielformel: $\neg\big((\neg p_4 \lor p_1) \land \bot\big)$

Bezeichnungen:

- Falsum: $\bot$
- Konjunktion: $\land$
- Disjunktion: $\lor$
- Implikation: $\to$
- Negation: $\neg$

Abkürzungen

$$
\begin{array}{lcl}
p, q, r\ \ldots & \text{statt} & p_0, p_1, p_2 \ldots \\[1em]
\left(\bigvee_{i=1}^{n} \varphi_i\right) & \text{statt} & \Big(\ldots\big((\varphi_1 \vee \varphi_2) \vee \varphi_3\big) \vee \ldots \vee \varphi_n\Big) \\
\left(\bigwedge_{i=1}^{n} \varphi_i\right) & \text{statt} & \Big(\ldots\big((\varphi_1 \wedge \varphi_2) \wedge \varphi_3\big) \wedge \ldots \wedge \varphi_n\Big) \\
(\varphi \leftrightarrow \psi) & \text{statt} & \big((\varphi \to \psi) \wedge (\psi \to \varphi)\big)
\end{array}
$$

Präzedenz der Operatoren:

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

Es gilt also z.B.:

$$(\alpha \leftrightarrow \beta \vee \neg\gamma \rightarrow \delta \wedge \neg\eta) = \Big(\alpha \leftrightarrow \big((\beta \vee \neg\gamma) \rightarrow (\delta \wedge \neg\eta)\big)\Big)$$

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

Zusammenfassung 1. Vorlesung

in dieser Vorlesung neu

- Geschichte, Sinn und Anspruch der Logik
- Syntax (d.h. Formeln) der Aussagenlogik als Formalisierung gewisser sprachlicher Ausdrücke

kommende Vorlesung

- Formalisierung von Argumentationen mittels Formeln der Aussagenlogik