Formale Logik in der Informatik

Einführung, Aussagenlogik

050026 VO Theoretische Informatik

Wolfgang Dvořák

Theory and Applications of Algorithms Gruppe,
Fakultät für Informatik, Universität Wien

Sommersemester 2016

Formale Logik in der Informatik

Das Gebiet der Logik

- beschäftigt sich mit den Prinzipien des korrekten Schließens.
- hat Überschneidungen mit vielen Wissenschaftsdisziplinen: Philosophie, Mathematik, Rechtswissenschaften, . . . , und Informatik.

Wir interessieren uns für formale Logik (auch mathematische Logik)

- Nutzt Formale Sprachen.
- Studiert die Gültigkeit von Ableitungs- und Folgerungsbeziehungen.

Ergänzende/Weiterführende Literatur

Die Vorlesung folgt dem Buch von Kreuzer & Kühling

Martin Kreuzer, Stefan Kühling (2006).
Logik für Informatiker.
Addison-Wesley Verlag, ISBN-13: 978-3827372154

Freie Ressourcen:

Wikibooks
Logic for Computer Science.
http://en.wikibooks.org/wiki/Logic_for_Computer_Science

Mandana Eibegger (2007).
Logik E-Book logic-rulez.net: Logik Einführung und Aussagenlogik.
http://www.logic-rulez.net/

In der Fachbereichsbibliothek:

Jürgen Dassow (2005).
Logik für Informatiker.
Vieweg+Teubner Verlag, ISBN-13: 978-3519005186

Hartmut Ehrig et al. (2001).
Mathematisch-strukturelle Grundlagen der Informatik. (Teil III & IV)
Springer Verlag, ISBN-13: 978-3540419235

Worum geht es in der Logik?

Es geht um

- das Ziehen von **Schlussfolgerungen**
- die Gültigkeit von **Begründungen**
- die **Widerspruchsfreiheit** zwischen Aussagen

Die formale Logik befasst sich **nicht**

mit dem Wahrheitsgehalt/Inhalt einer Aussage

an sich, sondern mit Regeln die es erlauben

aus wahren Aussagen richtige Schlüsse zu ziehen.

Worum geht es in der Logik?

Beispiel

Aussage 1: Jeder Mensch hat 2 Augen.
Aussage 2: John ist ein Mensch.
Schlussfolgerung: Also hat John 2 Augen.

Beispiel

Aussage 1: Jeder Mensch hat 4 Augen.
Aussage 2: John ist ein Mensch.
Schlussfolgerung: Also hat John 4 Augen.

Beobachtungen:

1. Wir haben die gleiche (gültige) Regel zum Schlussfolgern verwendet.
2. Aber nur wenn die Prämissen wahr sind stimmt auch die Folgerung.

Worum geht es in der Logik?

Die formale Logik beschäftigt sich damit unter welchen Bedingungen man aus der

Gültigkeit von Voraussetzungen

auf die

Gültigkeit von Folgerungen

schließen kann.

Dazu nutzen wir in der Informatik unterschiedliche logische Systeme (oft auch als unterschiedliche "Logiken" bezeichnet).

Logische Systeme

Bestandteile eines logischen Systems:

1. **Syntax**: Legt fest welche formalen Ausdrücke als Formeln des logischen Systems gelten. Üblicher Weise startet man von sogenannten atomaren Formeln und legt Regeln fest wie aus diesen weitere Formeln zusammengesetzt werden dürfen.
2. **Semantik**: Regeln die festlegen wie Formeln Wahrheitswerte zugeordnet werden können. Dem syntaktischen Element einer Formel wird damit eine Bedeutung gegeben.
3. **Kalkül**: Ein System von "mechanisch" anwendbaren Regeln zum Umformen von Formeln, die es erlauben aus gegebenen Formeln (maschinell) weitere Formeln abzuleiten.

Logische Systeme, Beispiele

In der Vorlesung werden wir zwei logische Systeme kennen lernen

- Aussagenlogik
- Prädikatenlogik

Es gibt aber noch unzählige andere logische Systeme (mit Bedeutung in der Informatik)

- Mehrwertige Logiken und Fuzzylogik
- Nichtmonotone Logiken (z.B.: Reitersche Default-Logik)
- Modallogik
- ...

Anwendungen in der Informatik

- Boolesche Ausdrücke in Programmiersprachen (if Anweisungen)
- Formale Spezifikation
- Verifikation von Programmen
- Logische Programmierung
- Datenbanksysteme
- Wissensrepräsentation und Künstliche Intelligenz
- Berechenbarkeits- und Komplexitätstheorie
- ...

Aussagenlogik - Einleitung

Eine **Aussage** ist

- Ein Satz in der natürlichen Sprache (z.B. “Das Auto ist rot.”)
- Ist entweder wahr (W) oder falsch (F)

Beispiel

„4 ist größer als 10“	F
„101 ist eine Primzahl“	W
„Junge Pferde nennt man Welpen“	F
„4 ist größer als 10, oder 10 ist größer als 4“	W
„4 ist größer als 10, und 10 ist größer als 4“	F

Aussagenlogik - Einleitung

Es gibt Aussagen die aus der logischen Verknüpfung von mehreren einfacheren Aussagen bestehen. Wir wollen diese mit atomaren Aussagen und logischen Operatoren darstellen.

Eine **einfache/atomare Aussage**

- ist eine Aussage
- die nur einen Sachverhalt enthält (unterteilbare Aussage).

Insbesondere enthält sie keine aussagenlogischen Verknüpfungen wie: nicht, und, oder, wenn ... dann, genau dann wenn

Als **logische Operatoren / Verknüpfungen** betrachten wir

- die Negation $\neg A$ (NOT),
- die Konjunktion $A \wedge B$ (AND),
- die Disjunktion $A \vee B$ (OR), und
- die Folgerung/Implikation $A \rightarrow B$.

Aussagenlogik - Einleitung

Beispiel

Die Aussage „Das Auto ist rot und das Fahrrad ist grün“ besteht aus zwei atomare Aussagen:

1. „Das Auto ist rot“
2. „Das Fahrrad ist grün“

Mit einer Konjunktion können wir die ursprüngliche Aussage aus den Atomen aufbauen.

$$\text{„Das Auto ist rot“} \wedge \text{„Das Fahrrad ist grün“}$$

Beispiel

Aussage: „Gestern hat es in Wien geregnet“

Die Aussage ist atomar. Sie enthält zwar mehrere Informationen (Zeit, Ort, Wetter) beschreibt aber nur einen Sachverhalt und kann nicht in mehrere Aussagen aufgespalten werden.

Aussagenlogik - Einleitung

Beispiele:

Beispiel

„4 ist größer als 10, oder 10 ist größer als 4“
„4 ist größer als 10” $\vee$ “10 ist größer als 4“

Beispiel

„Wenn es regnet ist die Straße nass.“
„Es regnet” $\rightarrow$ “Die Straße ist nass.“

Beispiel

„Wenn ich nicht lerne schaffe ich die Prüfung nicht.“
($\neg$ “Ich lerne.”) $\rightarrow$ ($\neg$ „Ich schaffe die Prüfung.“)
(„Ich lerne.”) $\vee$ ($\neg$ „Ich schaffe die Prüfung.“)

Aussagenlogik - Formale Logik

Zur Erinnerung: In der formalen Logik müssen wir alle Bestandteile eines logischen Systems genau definieren.

Auf den folgenden Folien:

- **Formale Syntax**: Wir definieren was genau eine Aussagenlogische Formel ist.
- **Formale Semantik**: Wir geben Aussagenlogischen Formeln eine Bedeutung.

Später in der Vorlesung:

- **Kalkül & Schlussregeln**

Aussagenlogische Formeln - Syntax

Nun wollen wir formal definieren was wir unter Aussagenlogischen Formeln (auch Boolesche Ausdrücke, nach George Boole) verstehen.

Definition (Aussagenlogische Formel)

Wir betrachten eine Menge von **atomaren Formeln** (Variablen) *Var*

- Jede atomare Formel aus *Var* ist eine Formel
- Sind F und G Formeln dann sind auch

$$\neg F, \quad (F \wedge G), \quad (F \vee G), \quad \text{und} \quad (F \rightarrow G)$$

  Formeln.

$\mathcal{F}_{Var}$ bezeichnet die Menge der aussagenlogischen Formeln die aus den Atomen *Var* gebildet werden können.

Aussagenlogische Formeln - Syntax

Beispiel

Für die atomaren Formeln $Var = \{x, y, z\}$ können wir z.B. die folgende Formel bilden:

$$((x \to y) \wedge (z \vee \neg x))$$

Zunächst können wir Formeln $F_1 = x$ und $F_2 = y$ bilden und dann

$$F_3 = (F_1 \to F_2) = (x \to y)$$

Mit $F_4 = z$ und $F_5 = \neg F_1 = \neg x$ bilden wir dann

$$F_6 = (F_4 \vee F_5) = (z \vee \neg x)$$

Schlussendlich erhalten wir

$$(F_3 \wedge F_6) = ((x \to y) \wedge (z \vee \neg x))$$

Semantik der Aussagenlogik

Die Semantik der Aussagenlogik wird über Wahrheitswerte "wahr" (W) und "falsch" (F) definiert.

Ein $x \in Var$ steht für irgendeine Aussage. Der Wahrheitswert von x hängt also davon ab für welche konkrete Aussage x steht.

- Der Wahrheitswert einer Formel wird in Abhängigkeit von den Wahrheitswerten der atomaren Formeln berechnet.
- Jeder Formel wird dann entweder der Wert W oder F zugeordnet.

Definition

Eine **Belegung** ist eine Funktion $\alpha : Var \to \{1, 0\}$ die

- jeder atomaren Formel entweder den Wert 1 (W) oder 0 (F) zuweist.

Diese Wahrheitswerte setzten sich auf beliebige Formeln fort.

Semantik der Aussagenlogik

Definition

Sei α eine Belegung für Atome in *Var*. Wir definieren die Erweiterung $\widehat{\alpha}$ der Belegung α auf alle Formeln in $\mathcal{F}_{Var}$ wie folgt:

- für atomare Formeln $F \in Var$: $\widehat{\alpha}(F) = \alpha(F)$
- für Formeln $F, G \in \mathcal{F}_{Var}$:

$$\widehat{\alpha}(F \wedge G) = \begin{cases} 1 & \text{Wenn } \widehat{\alpha}(F) = 1 \text{ und } \widehat{\alpha}(G) = 1 \\ 0 & \text{sonst} \end{cases}$$

$$\widehat{\alpha}(F \vee G) = \begin{cases} 1 & \text{Wenn } \widehat{\alpha}(F) = 1 \text{ oder } \widehat{\alpha}(G) = 1 \\ 0 & \text{sonst} \end{cases}$$

$$\widehat{\alpha}(\neg F) = \begin{cases} 1 & \text{Wenn } \widehat{\alpha}(F) = 0 \\ 0 & \text{sonst} \end{cases}$$

- für Formeln $F, G \in \mathcal{F}_{Var}$: $\widehat{\alpha}(F \rightarrow G) = \widehat{\alpha}(\neg F \vee G)$

Semantik der Aussagenlogik - Wahrheitstafeln

Alternativ kann man die Operationen auch über Wahrheitstafeln definieren. Für Formeln $F, G \in \mathcal{F}_{Var}$:

F	G	$F \wedge G$
0	0	0
0	1	0
1	0	0
1	1	1

F	G	$F \vee G$
0	0	0
0	1	1
1	0	1
1	1	1

F	$\neg F$
0	1
1	0

F	G	$F \rightarrow G$
0	0	1
0	1	1
1	0	0
1	1	1

Da alle Formeln nur aus solchen Verknüpfungen aufgebaut sind ist damit $\widehat{\alpha}$ für alle Formeln definiert.

Semantik der Aussagenlogik - Äquivalenz Operator

Wir können jetzt noch einen zusätzlichen Operator definieren.

Die **Äquivalenz**: $(F \leftrightarrow G)$

F	G	$F \leftrightarrow G$
0	0	1
0	1	0
1	0	0
1	1	1

$F \leftrightarrow G$ ist also eine Kurzschreibweise von $(F \rightarrow G) \wedge (G \rightarrow F)$.

Semantik der Aussagenlogik - Modelle

Definition

Eine Belegung α für *Var* ist ein **Modell einer Formel** F über *Var* wenn $\widehat{\alpha}(F) = 1$, wir schreiben auch $\alpha \models F$.

Die Semantik eine Formel ergibt sich aus der Menge ihrer Modelle.

Definition

Eine Belegung α für *Var* ist ein **Modell einer Menge von Formeln** $\mathcal{F}$ wenn $\widehat{\alpha}(G) = 1$ für jede Formel $G \in \mathcal{F}$, wir schreiben auch $\alpha \models \mathcal{F}$.

Semantik der Aussagenlogik - Grundlegende Begriffe

α: Belegung für Var,
F,G: Formeln über Var
$\mathcal{F}$: Eine Menge von Formeln über Var

Definition

- $\mathcal{F}$ ist **konsistent** (**widerspruchsfrei**) wenn es ein Modell für $\mathcal{F}$ gibt. $\mathcal{F}$ ist **inkonsistent**(**widersprüchlich**) wenn es kein Modell für $\mathcal{F}$ gibt.
- F ist **erfüllbar** wenn F mind. ein Modell hat (sonst **unerfüllbar**).
- F ist **allgemein gültig** / eine **Tautologie** wenn jede Belegung auch ein Modell ist.
- F ist **semantisch äquivalent** zu G wenn $\widehat{\alpha}(F) = \widehat{\alpha}(G)$ für alle Belegungen α. Wir schreiben $F \equiv G$.
- G **folgt** aus $F/\mathcal{F}$ wenn jedes Modell von $F/\mathcal{F}$ auch Modell von G ist. Wir schreiben $F \models G$ bzw. $\mathcal{F} \models G$.

Semantik der Aussagenlogik - Beispiele

Beispiel

Belegung α mit $\alpha(a) = 1$, $\alpha(b) = 1$, $\alpha(c) = 0$

- α ist ein Modell von $(a \vee c) \wedge (b \vee c)$ aber nicht von $(a \wedge c \wedge \neg c)$
- $(a \vee c) \wedge (b \vee c)$ ist daher erfüllbar
- $(a \wedge c \wedge \neg c)$ ist unerfüllbar da auch keine andere Belegung die Formel wahr macht

Beispiel

- $(a \vee c \vee \neg c)$ und $(c \vee \neg c)$ sind Tautologien
- $(a \vee b) \wedge (c \vee \neg c)$ ist semantisch äquivalent zu $(a \vee b)$
- Aus $(\neg a \wedge b)$ folgt $(a \vee b)$

Fundamentale Sätze

Satz

Ein Formel F ist genau dann eine Tautologie wenn $\neg F$ unerfüllbar ist.

Beispiel: Tautologie: $(c \vee \neg c)$
unerfüllbar: $\neg(c \vee \neg c) \equiv (\neg c \wedge c)$

Satz (Ersetzungssatz)

Betrachtet man eine Formel G mit einer Teilformel F_1 und bildet eine neue Formel $G[F_1/F_2]$ indem man (in G) die Teilformel F_1 durch eine semantisch äquivalente Formel F_2 ersetzt, dann gilt $G \equiv G[F_1/F_2]$.

Jede Teilformel kann durch eine semantisch äquivalente Formel ersetzt werden.

Beispiel: $G = (\neg a \vee b \vee c)$
$F_1 = (\neg a \vee b)$, $F_2 = (a \rightarrow b)$ (es gilt $F_1 \equiv F_2$)
$G[F_1/F_2] = ((a \rightarrow b) \vee c)$
Ersetzungssatz: $(\neg a \vee b \vee c) \equiv ((a \rightarrow b) \vee c)$

Äquivalenzen der Aussagenlogik

Mit dem Ersetzungssatz kann man äquivalente Formeln verwenden um eine Formel zu vereinfachen. Einige hilfreiche Äquivalenzen:

- Idempotenz $(F \land F) \equiv F$ und $(F \lor F) \equiv F$
- Kommutativität $(F \land G) \equiv (G \land F)$ und $(F \lor G) \equiv (G \lor F)$
- Assoziativität $((F \land G) \land H) \equiv (F \land (G \land H))$ und
  $((F \lor G) \lor H) \equiv (F \lor (G \lor H))$

Daher können wir statt $((F \land G) \land H)$ auch $(F \land G \land H)$ schreiben.

- Absorption $(F \land (F \lor G)) \equiv F$ und $(F \lor (F \land G)) \equiv F$
- Distributivität $(F \land (G \lor H)) \equiv ((F \land G) \lor (F \land H))$
  $(F \lor (G \land H)) \equiv ((F \lor G) \land (F \lor H))$
- Doppelnegation $\neg\neg F \equiv F$
- de Morgansche Regeln $\neg(F \land G) \equiv (\neg F \lor \neg G)$
  $\neg(F \lor G) \equiv (\neg F \land \neg G)$

Konjunktive Normalform

Eine **Normalform** ist eine Einschränkung auf der Syntax sodass jede beliebige Formel in eine semantisch äquivalente Formel in Normalform umgewandelt werden kann.

- Vereinfacht die maschinelle Verarbeitung von logischen Formeln. (Viele Algorithmen verarbeiten nur eine bestimmte Normalform)

Unter **Literalen** verstehen wir Atome $x \in Var$ und negierte Atome $\neg x$ für $x \in Var$.

Definition

Eine Formel ist in **konjunktiver Normalform (KNF)**, wenn sie eine Konjunktion von Disjunktionen von Literalen ist. F ist also von folgender Form ($L_{i,j}$ Literale):

$$F = (L_{1,1} \vee \cdots \vee L_{1,m_1}) \wedge \cdots \wedge (L_{n,1} \vee \cdots \vee L_{n,m_n})$$

Beispiel: $(\neg a \vee b) \wedge (a \vee \neg c \vee d) \wedge \neg b$

Disjunktive Normalform

Definition

Eine Formel ist in **disjunktiver Normalform (DNF)**, wenn sie eine Disjunktion von Konjunktionen von Literalen ist. F ist also von folgender Form ($L_{i,j}$ Literale):

$$F = (L_{1,1} \wedge \cdots \wedge L_{1,m_1}) \vee \cdots \vee (L_{n,1} \wedge \cdots \wedge L_{n,m_n})$$

Beispiele:

- $(\neg a \vee b) \wedge (a \vee \neg c \vee d) \wedge \neg b$ (KNF)
- $(\neg a \wedge b \wedge c) \vee (a \wedge \neg c \wedge d)$ (DNF)
- $\neg a \wedge b \wedge c$ (DNF, KNF)
- $\neg a \vee b \vee c$ (DNF, KNF)
- $(\neg a \vee b \vee c) \wedge (a \vee (b \wedge d))$ ()

Jede Formel kann mit Hilfe der präsentierten Äquivalenzumformungen in eine semantisch äquivalente KNF (DNF) transformiert werden.

Konjunktive und Disjunktive Normalform - Beispiele

Zwei Beispiele wie man eine Formel in KNF umwandeln kann:

$$
\begin{aligned}
&(A \land B) \to (C) && \\
&(\neg(A \land B) \lor C) && \text{(Definition von } \to\text{)} \\
&((\neg A \lor \neg B) \lor C) && \text{(de Morgan)} \\
&(\neg A \lor \neg B \lor C) && \text{(Assoziativität)}
\end{aligned}
$$

$$
\begin{aligned}
&C \leftrightarrow (A \land B) && \\
&(\neg C \lor (A \land B)) \land (C \lor \neg(A \land B)) && \text{(Definition von } \leftrightarrow\text{)} \\
&(\neg C \lor (A \land B)) \land (C \lor (\neg A \lor \neg B)) && \text{(de Morgan)} \\
&(\neg C \lor (A \land B)) \land (C \lor \neg A \lor \neg B) && \text{(Assoziativität)} \\
&((\neg C \lor A) \land (\neg C \lor B)) \land (C \lor \neg A \lor \neg B) && \text{(Distributivität)} \\
&(\neg C \lor A) \land (\neg C \lor B) \land (C \lor \neg A \lor \neg B) && \text{(Assoziativität)}
\end{aligned}
$$

Konjunktive und Disjunktive Normalform - Algorithmus

Diese Schritte lassen sich auch in einen Algorithmus fassen der eine Formel F in eine semantisch äquivalente KNF umzuwandeln.

Algorithmus

1. Ersetze $G \leftrightarrow H$ durch $(G \rightarrow H) \wedge (H \rightarrow G)$
2. Ersetze $G \rightarrow H$ durch $(\neg G \vee H)$
3. Iteriere das Folgende solange wir möglich
    1. Ersetze $\neg\neg G$ durch G
    2. Ersetze $\neg(G \wedge H)$ durch $(\neg G \vee \neg H)$
    3. Ersetze $\neg(G \vee H)$ durch $(\neg G \wedge \neg H)$
4. Iteriere das Folgende solange wir möglich
    1. Ersetze $(G \vee (H \wedge I))$ und $((H \wedge I) \vee G)$ durch $((G \vee H) \wedge (G \vee I))$

„Ersetze“ liest sich als „Ersetze alle Teilformeln der Form“

Formalisieren in Aussagenlogik

Unser ursprüngliches Ziel ist es (natürlich sprachliche) Aussagen zu formalisieren und logische Zusammenhänge zu Identifizieren.

Faustregel zur Formalisierung von Deutsch:

Identifizieren von atomaren Aussagen:

- die kürzeste Aussage, in der keine logische Verknüpfung vorkommt (nicht, und, oder, wenn ... dann usw.), und
- die entweder wahr oder falsch sein kann

Dann ersetzt man atomare Aussagen mit Aussagenvariablen

Wörter/Phrasen nicht, und, oder, wenn ... dann, usw.

- werden durch die entsprechenden logischen Operationen ersetzt.

Formalisieren in Aussagenlogik - Beispiel 1

Aussage 1: Der Verkauf von Häusern sinkt, wenn die Zinsen steigen.
Aussage 2: Auktionäre sind nicht glücklich, wenn der Verkauf von Häusern sinkt.
Aussage 3: Die Zinsen steigen.
Aussage 4: Auktionäre sind glücklich.

Atomare Aussagen:
S: der Verkauf von Häusern sinkt
R: die Zinsen steigen
H: Auktionäre sind glücklich

Aussage 1: $R \rightarrow S$
Aussage 2: $S \rightarrow \neg H$
Aussage 3: R
Aussage 4: H

Da wir wissen wollen ob die Aussagen miteinander konsistent sind betrachten wir deren Konjunktion $(R \rightarrow S) \wedge (S \rightarrow \neg H) \wedge R \wedge H$.

Formalisieren in Aussagenlogik - Beispiel 1

Da wir wissen wollen ob die Aussagen miteinander konsistent sind betrachten wir deren Konjunktion $(R \to S) \land (S \to \neg H) \land R \land H$.

Wir testen ob die Formel erfüllbar ist und nutzen dazu Wahrheitstafeln.

S	R	H	$(R \to S)$	$(S \to \neg H)$	$(R \to S) \land (S \to \neg H) \land R \land H$
0	0	0	1	1	0
0	0	1	1	1	0
0	1	0	0	1	0
0	1	1	0	1	0
1	0	0	1	1	0
1	0	1	1	0	0
1	1	0	1	1	0
1	1	1	1	0	0

Die Formel hat kein Model (ist unerfüllbar) und daher sind die Aussagen widersprüchlich.

Formalisieren in Aussagenlogik - Beispiel 2

Aussage 1: Wenn der Geiger das Konzert gibt, werden viele kommen, wenn die Preise nicht zu hoch sind.
Aussage 2: Wenn der Geiger das Konzert gibt, werden die Preise nicht zu hoch sein.
Schluss: Daher werden, falls der Geiger das Konzert bestreitet, viele kommen.

Wir wollen wissen ob der Schluss korrekt ist.

Atomare Aussagen:
P: „der Geiger gibt das Konzert"
C: „viele werden kommen"
H: „die Preise sind zu hoch"

Aussage 1: $P \to (\neg H \to C)$
Aussage 2: $P \to \neg H$
Schluss: $P \to C$

Wir wollen wissen ob $P \to C$ aus $P \to (\neg H \to C)$ und $P \to \neg H$ folgt.

Formalisieren in Aussagenlogik - Beispiel 2

Wir wollen wissen ob $P \to C$ aus $P \to (\neg H \to C)$ und $P \to \neg H$ folgt.
(Wir schreiben auch $\{\neg H \to C, P \to \neg H\} \models P \to C$.)

Widerspruchsmethode:

Um $A, B \models C$ zu zeigen, zeigen wir dass $A \wedge B \wedge \neg C$ widersprüchlich ist.

Wir betrachten also die Formel:

$$(P \to (\neg H \to C)) \wedge (P \to \neg H) \wedge \neg(P \to C)$$

Mittels Wahrheitstafel (oder dem später diskutierten Resolutionskalkül) stellen wir fest dass die Formel unerfüllbar ist.

Daher ist der ursprüngliche Schluss auf $P \to C$ korrekt.

Formalisieren in Aussagenlogik - Beispiel 3

(zum Selbststudium)

Aussage 1: Tom kann am Wochenende nicht beides eine Wanderung und eine Radtour machen.
Aussage 2: Wenn es am Wochenende regnet macht Tom eine Radtour.
Aussage 3: Tom geht am Wochenende Wandern.
Aussage 4 Es regnet nicht am Wochenende.

Aufgabe (Teil 1): Formalisieren Sie die Aussagen mittels aussagenlogischer Formeln. Geben Sie die Bedeutung jeder Aussagenvariablen an.

Atomare Aussagen:
R: „Es regnet am WE“ T: „Tom macht eine Radtour am WE“
W: „Tom macht eine Wanderung am WE“

Aussage 1: $\neg(T \wedge W)$ oder $(\neg T \vee \neg W)$
Aussage 2: $R \rightarrow T$
Aussage 3: W
Aussage 4: $\neg R$

Formalisieren in Aussagenlogik - Beispiel 3

(zum Selbststudium)

Aufgabe (Teil 2): Kann man aus den ersten drei Aussagen auf Aussage 4 schließen? Beweisen Sie Ihre Vermutung mit einer geeigneten Methode aus der Vorlesung (geben Sie auch an welche Methode Sie verwenden).

Wir verwenden die Widerspruchsmethode und betrachten die Konjunktion der drei Aussagen und des negierten Schluss:

$$(\neg T \vee \neg W) \wedge (R \rightarrow T) \wedge W \wedge R$$

Im nächsten Foliensatz werden wir zeigen, dass diese Formel unerfüllbar ist. Daher ist der Schluss aus den drei Aussagen korrekt.

Zusammenfassung & Ausblick

Bis jetzt haben wir Folgendes behandelt:

- Worum geht es in der Logik
- Einführung Aussagenlogik
- Formale Definition von Syntax und Semantik der Aussagenlogik
- Formalisieren in Aussagenlogik

Nach der Pause:

- Schlussregeln
- Kalküle in der Aussagenlogik
- Einführung Prädikatenlogik

Pause

Schlussregeln

Es gibt verschiedene syntaktische Methoden, d.h. ohne alle Belegungen zu testen, um von einer Menge von (wahren) Formeln auf weitere wahre Formeln zu schließen.

- Modus ponens
- Modus tollens
- Resolution
- Einheitsresolution
- ...

Modus ponens

Modus ponens

Aus den Prämissen A und $A \to B$ folgt die Conclusio B.

Oft wird modus ponens auch wie folgt notiert:

$$\frac{A, A \to B}{B}$$

Beispiel

Aus den Prämissen

- „Es regnet“ und
- „Wenn es regnet, wird die Straße nass“

folgt durch Modus ponens logisch „Die Straße wird nass“.

Modus tollens

Modus tollens

Aus den Prämissen $A \to B$ und $\neg B$ folgt die Conclusio $\neg A$.

Oft wird modus tollens auch wie folgt notiert:

$$\frac{\neg B, A \to B}{\neg A}$$

Beispiel

Aus den Prämissen

- „Die Straße wird nicht nass" und
- „Wenn es regnet, wird die Straße nass"

folgt durch Modus tollens logisch „es regnet nicht".

Es gibt aber viele Formeln die **nicht** mir Modus ponens und Modus tollens allein abgeleitet werden können.

Der Resolutionskalkül der Aussagenlogik

- Resolution ist ein Verfahren der formalen Logik, um zu Testen ob eine Formel / eine Menge von Formeln widersprüchlich/unerfüllbar ist.
- Diese Herleitung geschieht mittels eines Algorithmus
  $\hookrightarrow$ Maschinengestütztes Beweisen.
- Der Resolutionskalkül arbeitet auf Formeln in KNF (Konjunktiver Normalform).
  $\hookrightarrow$ Der erste Schritt ist die gegebenen Formeln in KNF zu bringen.

Der Resolutionskalkül der Aussagenlogik

Der Resolutionskalkül erlaubt die

Unerfüllbarkeit von einer Menge aussagenlogischer Formeln

zu beweisen.

Viele andere Aufgaben können hierauf zurückgeführt werden:

- Testen ob eine Formel F aus einer Formelmenge $\mathcal{F}$ folgt.
  Man testet $\mathcal{F} \cup \{\neg F\}$ auf Unerfüllbarkeit.
- Testen ob eine Formel F eine Tautologie ist.
  Man testet $\neg F$ auf Unerfüllbarkeit.
- Testen ob eine Menge von Formeln $\mathcal{F}$ widersprüchlich ist.
  Man testet die Konjunktion der Formeln auf Unerfüllbarkeit.

Klauseln von Formeln

Gegeben sei eine Formel F in KNF, also

$$F = (L_{1,1} \vee \cdots \vee L_{1,m_1}) \wedge \cdots \wedge (L_{n,1} \vee \cdots \vee L_{n,m_n})$$

wobei die $L_{i,j}$ Literale sind.

Konjunktion und Disjunktion sind idempotent, kommutativ und assoziativ. Daher können wir eine Formel in KNF auch als Menge von Mengen anschreiben.

Die F zugeordnete **Klauselmenge** $K(F)$ ist

$$K(F) = \{\{L_{1,1}, \ldots, L_{1,m_1}\}, \ldots, \{L_{n,1}, \ldots, L_{n,m_N}\}\}$$

Eine einzelne Menge $\{L_{i,1}, \ldots, L_{i,m_i}\}$ heißt **Klausel** von F.

- Eine Klausel entspricht einer Disjunktion von Literalen

Klauseln von Formeln

- Verschiedene Formeln können dieselbe Klauselmenge besitzen

Beispiel

Die Formeln

$F_1 = (A \vee B) \wedge (C)$,
$F_2 = (C \vee C) \wedge (A \vee B) \wedge (C)$ und
$F_3 = C \wedge (B \vee A)$

besitzen alle die Klauselmenge $\{\{A, B\}, \{C\}\}$.

- Eine leere Klausel entspricht dem Wahrheitswert 0 (falsch).
  (Eine leere Disjunktion ist falsch)
- Eine Klauselmenge die eine leere Klausel enthält hat den Wahrheitswert 0.
  (Eine Konjunktion ist falsch wenn ein Eintrag falsch ist)
- Eine leere Klauselmenge entspricht dem Wahrheitswert 1.
  (Eine leere Konjunktion ist wahr)

Resolvente von Klauseln

Idee: Wenn wir Klauseln $\{A, B\}$ und $\{\neg A, C\}$ haben dann kann nur entweder A oder $\neg A$ wahr sein. Es folgt, dass B oder C wahr sein muss, d.h. die Klausel {B,C}.

Definition

Eine Klausel K_3 heißt **Resolvente** der Klauseln K_1, K_2 wenn es ein Literal L gibt sodass

- $L \in K_1$, $\neg L \in K_2$
- $K_3 = (K_1 \setminus \{L\}) \cup (K_2 \setminus \{\neg L\})$

Wenn K_3 Resolvente von K_1 und K_2 ist verwenden wir auch folgende graphische Darstellung.

$$\begin{tikzcd}
K_1 \arrow[dr, dash] & & K_2 \arrow[dl, dash] \\
& K_3 &
\end{tikzcd}$$

Resolvente von Klauseln - Beispiel

Gegeben sei die Klauselmenge $\{\{A, B, \neg C\}, \{A, \neg B, C\}\}$. Zu den Klauseln dieser Klauselmenge gibt es die folgenden Resolventen:

$$\begin{array}{ccc} \{A, B, \neg C\} & & \{A, \neg B, C\} \\ & \{A, C, \neg C\} & \end{array} \qquad \begin{array}{ccc} \{A, B, \neg C\} & & \{A, \neg B, C\} \\ & \{A, B, \neg B\} & \end{array}$$

Folgendes ist aber FALSCH:

$$\begin{array}{ccc} \{A, B, \neg C\} & & \{A, \neg B, C\} \\ & \{A\} & \end{array}$$

Resolutionslemma

Satz (Resolutionslemma)

Für jede Formel F mit Klauselmenge $K(F)$: Ist K_3 eine Resolvente zweier Klauseln $K_1, K_2 \in K(F)$ so ist F semantisch äquivalent zu allen Formeln mit Klauselmenge $K(F) \cup \{K_3\}$.

- Wir können also neue Klauseln hinzufügen ohne die Semantik zu verändern.
- Weiters können wir Klauseln die Tautologien entsprechen (Klauseln die x und $\neg x$ enthalten) weglassen ohne die Semantik zu verändern.

Idee des Resolutionskalkül: Solange Resolventen bilden bis man entweder die leere Klausel enthält oder alle möglichen Resolventen gebildet wurden.

Resolutionskalkül

Sei $K = K(F)$ die Klauselmenge einer Formel:

- $Res(K) = K \cup \{R \mid R \text{ ist Resolvente zweier Klauseln in } K\}$
- $Res^0(K) = K$ und für $n \geq 1$ sei

$$Res^n(K) = Res(Res^{n-1}(K))$$

- $Res^\infty(K) = \bigcup_{n \geq 0} Res^n(K)$

Satz

Eine Formel F ist genau dann unerfüllbar wenn $\emptyset \in Res^\infty(K(F))$.

Hat F nur n Variablen dann hat $Res^\infty(K)$ höchsten 4^n Klauseln.

$\hookrightarrow$ $Res^\infty(K)$ kann mit (maximal) 4^n Iterationen berechnet werden.

Resolutionskalkül - Erfüllbarkeitstest

Gegeben sei eine Formel in KNF:

1. Bilde Klauselmenge $K(F)$
2. Berechne $Res^1(K), Res^2(K), \ldots, Res^n(K)$ bis
$$\emptyset \in Res^n(K) \text{ oder } Res^{n-1}(K) = Res^n(K).$$
3. Wenn $\emptyset \in Res^n(K)$ „F unerfüllbar“

   anderenfalls „F erfüllbar“.

Resolutionskalkül - Beispiel

Wir betrachten die Formel aus dem Geiger-Beispiel:

$$(P \to (\neg H \to C)) \land (P \to \neg H) \land \neg(P \to C)$$

1.Schritt: Wir bringen die Formel in KNF.

$$\begin{array}{rl} (\neg P \lor (H \lor C)) \land (\neg P \lor \neg H) \land \neg(\neg P \lor C) & \text{Definition von } \to \\ (\neg P \lor H \lor C) \land (\neg P \lor \neg H) \land P \land \neg C & \text{de Morgan, Assoziativität} \end{array}$$

2.Schritt: Bilden der Klauselmengen

$$K(F) = \{\{\neg P, H, C\}, \{\neg P, \neg H\}, \{P\}, \{\neg C\}\}$$

Resolutionskalkül - Beispiel

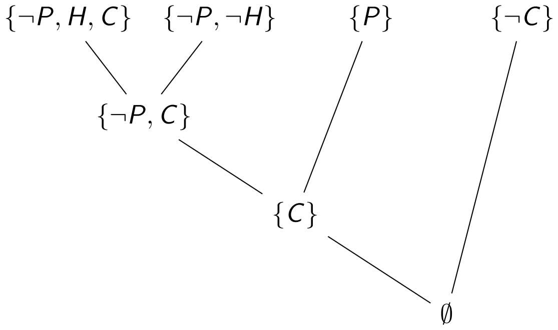
3.Schritt: Resolventen berechnen

$$Res^0(K) = \{\{\neg P, H, C\}, \{\neg P, \neg H\}, \{P\}, \{\neg C\}\}$$
$$Res^1(K) = Res^0(K) \cup \{\{\neg P, C\}, \{H, C\}, \{\neg P, H\}, \{\neg H\}\}$$
$$Res^2(K) = Res^1(K) \cup \{\{C\}, \{\neg P\}, \{H\}\}$$
$$Res^3(K) = Res^2(K) \cup \{\{\}\}$$

Da wir die leere Klausel erreicht haben ist die Formel unerfüllbar.

Resolutionskalkül - Beispiel

Oft müssen wir nicht alle Resolventen berechnen um einen Widerspruch (die leere Klausel) zu finden:



Aber: Um zu zeigen dass eine Formel widerspruchsfrei ist muss man alle Resolventen berechnen!!!

Hornlogik

Die Hornlogik ist eine Einschränkung der Aussagenlogik, d.h. es sind nur Formeln einer speziellen Form erlaubt.

- Benannt nach dem Logiker Alfred Horn.
- Kann mit speziellen Resolutions-Varianten sehr effizient ausgewertet werden.
  $\hookrightarrow$ Einheitsresolution, SLD-Resolution
- Dient als Grundlage der logischen Programmierung (Prolog).
- Bedeutung in der Komplexitätstheorie.

Hornlogik

Definition

Eine Klausel heißt **Hornklausel** wenn sie höchstens ein positives Literal enthält. Eine Formel F in KNF heißt **Hornformel** wenn $K(F)$ nur aus Hornklauseln besteht.

Beispiel für eine Hornformel:

- $(a \vee \neg b \vee \neg c) \wedge (\neg b \vee \neg c) \wedge (c)$

Unterschiedliche Arten von Hornklauseln:

- Tatsachenklauseln/Fakten: nur ein positives Literal, z.B. (c)
- Regeln: ein positives Literal, z.B. $(a \vee \neg b \vee \neg c)$ $\big(\equiv (b \wedge c) \rightarrow a\big)$
- Zielklausel: nur negative Literale, z.B. $(\neg b \vee \neg c)$

Einheitsresolution

Eine **Einheitsklausel** ist eine Klausel die nur aus einem (positiven oder negativen) Literal besteht.

Beobachtung: Enthält die Klauselmenge eine Einheitsklausel so muss dieses Literal jedenfalls auf 1 gesetzt werden um die Formel zu erfüllen.

Einheitsresolution (unit propagation)

Formel F in KNF, $K = K(F)$

- Solange die Klauselmenge K eine Einheitsklausel enthält wähle eine Einheitsklausel $\{L\}$
  - Entferne alle Klauseln in K die L enthalten
  - Lösche $\neg L$ aus allen Klauseln in K
- Wenn $\emptyset \in K$ dann ist die Formel F unerfüllbar.

Einheitsresolution

Sei K eine Klauselmenge dann schreiben wir $ERes(K)$ für die vereinfachte Klauselmenge die der Einheitsresolution-Algorithmus erzeugt.

Satz (Erfüllbarkeitsäquivalenz)

Eine Klauselmenge K ist genau dann erfüllbar wenn ihre Vereinfachung $ERes(K)$ durch Einheitsresolution erfüllbar ist.

- Wenn $\emptyset \in ERes(K)$ dann ist die Formel F unerfüllbar.
- **Für Horn-Formeln** gilt auch: Wenn $\emptyset \notin ERes(K)$ dann ist die Formel erfüllbar.

Satz

Eine Horn-Formel ist genau dann erfüllbar wenn in $\emptyset \notin ERes(K)$.

Hornlogik/Einheitsresolution - Beispiel 1

Beispiel

Formel $F = a \wedge b \wedge ((a \wedge b) \rightarrow c) \wedge (d \rightarrow e)$
Klauselmenge $K(F) = \{\{a\}, \{b\}, \{\neg a, \neg b, c\}, \{\neg d, e\}\}$
mit zwei Einheitsklausel $\{a\}, \{b\}$

$$\begin{array}{rl}
\{\{a\}, \{b\}, \{\neg a, \neg b, c\}, \{\neg d, e\}\} & \text{Resolviere } \{a\} \\
\{\{b\}, \{\neg b, c\}, \{\neg d, e\}\} & \text{Resolviere } \{b\} \\
\{\{c\}, \{\neg d, e\}\} & \text{Resolviere } \{c\} \\
\{\{\neg d, e\}\} &
\end{array}$$

Die Formel F ist erfüllbar.

Hornlogik/Einheitsresolution - Beispiel 2

Beispiel

Formel $F = a \wedge b \wedge ((a \wedge b) \rightarrow c) \wedge (\neg a \vee \neg b) \wedge (d \rightarrow e)$
Klauselmenge $K(F) = \{\{a\}, \{b\}, \{\neg a, \neg b, c\}, \{\neg a, \neg b\}, \{\neg d, e\}\}$
Zwei Einheitsklausel $\{a\}, \{b\}$

$$\begin{array}{rl} \{\{a\}, \{b\}, \{\neg a, \neg b, c\}, \{\neg a, \neg b\}, \{\neg d, e\}\} & \text{Resolviere } \{a\} \\ \{\{b\}, \{\neg b, c\}, \{\neg b\}, \{\neg d, e\}\} & \text{Resolviere } \{b\} \\ \{\{c\}, \{\}, \{\neg d, e\}\} & \end{array}$$

Da wir die leere Klausel erhalten haben ist die Formel F unerfüllbar.

Hornlogik/Einheitsresolution - Beispiel 3

(zum Selbststudium)

Betrachte die Formel

$$F = (\neg T \vee \neg W) \wedge (R \to T) \wedge W \wedge R$$

Durch Auflösen der Inklusion $(R \to T)$ zu $(\neg R \vee T)$ erhalten wir

$$K(F) = \{\{\neg T, \neg W\}, \{\neg R, T\}, \{W\}, \{R\}\}$$

$$\begin{array}{rl}
\{\{\neg T, \neg W\}, \{\neg R, T\}, \{W\}, \{R\}\} & \text{Resolviere } \{W\} \\
\{\{\neg T\}, \{\neg R, T\}, \{R\} & \text{Resolviere } \{\neg T\} \\
\{\{\neg R\}, \{R\}\} & \text{Resolviere } \{\neg R\} \\
\{\{\}\} &
\end{array}$$

Da wir die leere Klausel erhalten haben ist die Formel F unerfüllbar.

Einheitsresolution und allgemeine Klauselmengen

Gegeben: Klauselmenge $K(F)$ die nicht in Horn-Form ist.

Ziel: Testen ob die Klauselmenge $K(F)$ erfüllbar ist.

Idee: Kombiniere Resolution mit Einheitsresolution

Verfahren: Wandle den Erfüllbarkeitstest mittels Resolution wie folgt ab

- Zu Beginn starte Einheitsresolution um die Klauselmenge zu verkleinern.
- Wurde die leere Klausel berechnet wird ist $K(F)$ unerfüllbar; sonst starte Resolution auf der verkleinerten Klauselmenge.
- Wann immer Resolution eine Einheitsklausel hinzugefügt starte Einheitsresolution um die Klauselmenge zu verkleinern bevor mit Resolution fortgesetzt wird.

Einheitsresolution und allgemeine Klauselmengen - Beispiel

Beispiel

Formel $F = a \wedge (\neg a \vee b \vee c) \wedge (b \leftrightarrow c) \wedge (\neg a \vee \neg b \vee \neg c) \wedge (a \vee b \vee c)$
Klauselmenge
$K(F) = \{\{a\}, \{\neg a, b, c\}, \{b, \neg c\}, \{\neg b, c\}, \{\neg a, \neg b, \neg c\}, \{a, b, c\}\}$
mit Einheitsklausel $\{a\}$

Einheits-Resolution mit $\{a\}$ ergibt: $\{\{b, c\}, \{b, \neg c\}, \{\neg b, c\}, \{\neg b, \neg c\}\}$

Ein Resolution-Schritt ergibt: $\{\{b, c\}, \{b, \neg c\}, \{\neg b, c\}, \{\neg b, \neg c\}, \{b\}\}$

Einheits-Resolution mit $\{b\}$ ergibt: $\{\{c\}, \{\neg c\}\}$

Einheits-Resolution mit $\{\neg c\}$ ergibt: $\{\{\}\}$

Die Formel F ist also unerfüllbar.

Limitierungen von Aussagenlogik

Es gibt logische Zusammenhänge die sich mit Aussagenlogik nicht abbilden lassen.

Beispiel

Aussage 1: Fritz fährt Ski
Aussage 2: Uli fährt Ski

Obwohl beide Aussagen augenscheinlich von gleicher Natur sind, kann Aussagenlogik die gemeinsame Struktur nicht modellieren.

Beispiel

Aussage 1: Jeder Mensch ist wertvoll.
Aussage 2: Hans ist ein Mensch.

Wir würden also gerne schließen „Hans ist wertvoll“. Das ist aber in Aussagenlogik nicht möglich.

Limitierungen von Aussagenlogik

Limitierungen von Aussagenlogik

- Aussagen werden als Atome genutzt und nicht weiter analysiert
  $\hookrightarrow$ oft stecken mehrere Informationen in einer atomaren Aussage
- Die innere Struktur einer Aussage geht verloren.
- Es ist unmöglich/schwer auszudrücken, dass
  - gewisse Beziehungen zwischen Objekten gelten;
  - etwas für alle Objekte gilt; oder
  - es ein Objekt mit einer Eigenschaft geben muss.

Wir benötigen also eine **ausdrucksstärkere Logik**.

Prädikatenlogik

Prädikatenlogik – Prädikate

Wir wollen Objekte von ihren Eigenschaften und Beziehungen zwischen einander trennen. Dazu verwenden wir **Prädikate**.

Beispiel

Um die Aussage „Hans ist ein Mensch“ zu formalisieren verwenden wir ein (einstelliges) Prädikat *Mensch*(.) und ein Objekt *hans* und bekommen

$$Mensch(hans)$$

Mit dem gleichen Prädikat können wir auch ausdrücken dass Barbara ein Mensch ist

$$Mensch(barbara)$$

Wir können auch ausdrücken das der Kater Findus kein Mensch ist

$$\neg Mensch(findus)$$

Ein Prädikat kann für manche Objekte wahr aber für andere falsch sein.

Prädikatenlogik – Quantoren

Prädikate erlauben es mittels Quantoren Aussagen über die Gesamtheit der Objekte die eine Eigenschaft erfüllen zu Treffen.

Wir betrachten zwei **Quantoren**

- **Allquantor** $\forall$: etwas gilt für alle Objekte
- **Existenzquantor** $\exists$: etwas gilt für mindestens ein Objekt

Beispiel

Die Aussage „Jeder Mensch ist wertvoll“ können wir jetzt wie folgt formalisieren:

$$\forall x\ (Mensch(x) \to Wertvoll(x))$$

Gemeinsam mit $Mensch(hans)$ schließen wir

$$Wertvoll(hans)$$

Prädikatenlogik – 2-stellige Prädikate

Beispiel

Aussage 1: Fritz fährt Ski
Aussage 2: Uli fährt Ski
Aussage 3: Uli fährt Snowboard

Wir könnten jetzt zwei Prädikate *FahrtSki*, und *FahrtSnowboard* einführen. Damit würden wir aber wieder etwas von der gemeinsamen Struktur der Aussagen verlieren.

Besser ist wir verwenden ein 2-stelliges Prädikat *Fahrt*

$$Fahrt(fritz, ski) \land Fahrt(uli, ski) \land Fahrt(uli, snowboard)$$

Jetzt können wir auch sagen das jeder Wintersportler mindestens ein Sportgerät fährt (das muss aber nicht Ski oder Snowboard sein)

$$\forall x \; (Wintersportler(x) \rightarrow \exists y \; \; Fahrt(x, y))$$

Prädikatenlogik – Funktionen

Mache Beziehungen sind von der Form dass einem Objekt ein anderes eindeutig zugeordnet wird, z.B. jeder hat nur eine (biologische) Mutter. In solchen Fällen verwendet man statt einer Relation besser eine Funktion.

Beispiel

Aussage: „Jo Anns Mutter liebt Musik“

Ohne Funktionen: $\exists x(Mutter(x, joAnn) \wedge Liebt(x, musik))$.

Das liest sich als: „Jo Ann hat mindestens eine Mutter die Musik liebt“

Mit Funktionen

Wir verwenden: Objekte *joAnn*, *musik*,
das 2-stellige Prädikat *Liebt*,
die 1-stellige Funktion *mutter*

$$Liebt(mutter(joAnn), musik)$$

Prädikatenlogik – Funktionen

Unterschied Relation – Funktion (in der Prädikatenlogik):

- Relationen ordnen Objekten Wahrheitswerte zu.
- Funktionen ordnen Objekten wieder Objekte zu.

Prädikatenlogik – Funktionen

Beispiel

Aussage: Wenn a größer als b ist dann ist für alle x auch $a + x$ größer als $b + x$

$$Großer(a, b) \rightarrow \forall x \, Großer(plus(a, x), plus(b, x))$$

Beispiel

Die Multiplikation kann mittels der Addition wie folgt definiert werden:

- x mal 0 ist 0
- x mal $(y + 1)$ ist x mal y plus x

In der Prädikatenlogik erhalten wir:

$$\forall x \forall y \big(IstGleich(mal(x, 0), 0) \wedge IstGleich(mal(x, plus(y, 1)), plus(mal(x, y), x)) \big)$$

Das entspricht $\forall x \forall y \big(x \cdot 0 = 0 \wedge x \cdot (y + 1) = (x \cdot y) + x \big)$.

Zusammenfassung & Ausblick

In dieser Einheit:

- Worum geht es in der Logik
- Aussagenlogik
- Formalisieren und Schließen in der Aussagenlogik
- Resolutionskalkül der Aussagenlogik
- Hornlogik und Einheitsresolution
- Ausblick Prädikatenlogik

Nächste Einheit:

- Einführung Prädikatenlogik
- Formale Syntax der Prädikatenlogik
- Formale Semantik der Prädikatenlogik
- Formalisieren in Prädikatenlogik