
Logik & Semantik

2. Vorlesung

Aussagenlogik

Grundbegriffe

Syntax

Semantik

Für den ersten Teil der Vorlesung wird im wesentlichen auf
Melvin Fitting:

First-order logic and automated Theorem proving.

New York: Springer. 1996, 2nd Edition.

zurück gegriffen. (Gegebenenfalls wird auf andere Literatur
hingewiesen.)

- Die Nummerierung von Definitionen und Sätzen (etc.) bezieht sich
auf Fitting.
Ein Stern * markiert Abweichungen von Fittingschen Formulierungen

Warum Aussagenlogik ?

Gemeinsamer Kern (fast) aller Logiken

- Konjunktion: Zusammenfassen (von Aussagen / Informationen)
- Negation: Bestreiten von Aussagen
- Disjunktion: Alternativenerfassung
- Implikation: Aussagen unter Bedingungen stellen
- Biimplikation: Definition, Gleichsetzung

Eine einfache Logik

- Modellierung auf allen Ebenen
- leicht beherrschbare semantische Strukturen (Wahrheitstabeln)
- Ausgangspunkt für die Entwicklung von Beweisverfahren

Definition eines logischen Systems: Generelles Schema

Spezifikation

- einer formalen Sprache (zur Repräsentation)
- von Evaluations- / Interpretationsprinzipien
- semantischer Kategorisierungen und Beziehungen
- Ableitungs-, Beweisverfahren

Die symbolische Logik

- Die Menge der Formeln ist eine formale Sprache

Objektsprache

- die Menge der Zeichenketten, über die wir sprechen
- auf Folien und in pdf-Dateien in [dieser Schrift](#) dargestellt
- Die Zeichen **F, G, H, X, Y** ... verwenden wir als Variablen, die Zeichenketten als Wert haben können.

Metasprache

- eine Fachsprache, mit der wir über die Objektsprachen sprechen
- Deutsch plus Fachterminologie (definierte neue Ausdrücke, wie Vokabeln zu lernen)
- dargestellt in schwarzer Schrift

- In den Anmerkungen zu den Folien ist der Farbcode nicht konsequent verwendet. Bitte lassen Sie sich dadurch nicht irritieren.

Aussagenlogik und Prädikatenlogik

Verschiedene Objektsprachen

\mathcal{L}_{AL} : Die Objektsprache der **Aussagenlogik**

- Beispiele: P , $\neg P$, $(P \wedge Q)$, $(P \supset Q)$, $(P \vee (P \wedge Q))$
- Die kleinsten Einheiten sind atomare Formeln (P , Q)
- Mit Junktoren (\neg , \wedge , \vee , \supset , \equiv) werden komplexe Formeln gebildet.

\mathcal{L}_{PL} : Die Objektsprache der **Prädikatenlogik**

- Anreicherung der Aussagenlogik
- Atomare Formeln haben eine interne Struktur: $P(a)$, $P(x)$, $R(a, b)$
gebildet aus Prädikatssymbolen (P , R) und Termen (a , b , x)
- Formeln mit Quantoren: $\forall x (R(x, a) \supset P(x))$, $\exists x (P(x) \wedge Q(x))$

Gemeinsamkeiten in der Metasprache

- Fachbegriffe werden an Hand der Aussagenlogik eingeführt.
- Sie werden an die reichere Struktur der Prädikatenlogik angepasst.

Die Sprache der Aussagenlogik: Vokabular

Das Vokabular (Alphabet)

- Verfügbare Symbole
 - Eine abzählbare Menge von Aussagensymbolen ($\mathcal{A}_S(\mathcal{L}_{AL})$):
 $P, Q, P_1, P_2, P_3, \dots$
- logische Symbole
 - logischen Konstanten: \perp, \top
 - Junktoren: einstellig (unär): \neg
zweistellig (binär): $\wedge, \vee, \supset, \subset, \equiv$
- Hilfssymbole
 - Klammern: $), ($

Definition (2.2.1*)

- Die **atomaren Formeln** ($\mathcal{A}_t(\mathcal{L}_{AL}) = \mathcal{A}_S(\mathcal{L}_{AL}) \cup \{\perp, \top\}$) sind die
- Aussagensymbole $\mathcal{A}_S(\mathcal{L}_{AL})$
 - und logischen Konstanten \perp, \top

- Das Symbolinventar (Alphabet) ist in der Literatur zur Logik sehr unterschiedlich. Die hier verwendete Notation orientiert sich an Fitting (1996).
- Die Verwendung von P, Q für Aussagensymbole ist in der angelsächsischen (englisch-sprachigen) Fachwelt weit verbreitet.
- Welche Junktoren (bei Fitting: *connectives*) verwendet – und wie sie symbolisiert – werden, ist häufig in individuellen „Vorlieben“ der AutorInnen begründet.
Wichtig für die LeserInnen ist jedoch stets, die verwendete Notation mit anderen Notationen in Verbindung bringen zu können, und gegebenenfalls die „Übersetzung“ zwischen verschiedenen Notationen leisten zu können.
- Da es keine ‘Normierung’ einer Logiksprache gibt, muss man fast davon ausgehen, dass jeder Autor seine eigenen Symbole und Schreibweisen verwendet. Lernen Sie, bei jedem Aufsatz oder Buch zu klären, was die Konventionen des Autors sind, bevor Sie sich über die Inhalte der Formeln wundern.
- Die Einbeziehung der logischen Konstanten \perp, \top in das Vokabular der Sprache ist eine – durchaus diskutierbare – Möglichkeit, die insbesondere von VertreterInnen des maschinellen Beweisens gewählt wird.
Achtung: Es handelt sich um logische Symbole, die eine spezifische Rolle in den logischen Verfahren spielen. Sie sind nicht mit den Aussagesymbolen gleichzusetzen.
- Klammern: In der Vorlesung werden z.T. weitere Klammerformen verwendet. Dies dient nur der besseren Lesbarkeit und sollte nicht

Die Sprache der Aussagenlogik



Definition (2.2.2)

Die Menge der aussagenlogischen Formeln $For(\mathcal{L}_{AL})$ ist die (bezüglich Mengeneinklusion \subseteq) kleinste Menge \mathbf{P} , so dass:

1. Falls P eine atomare Formel, so $P \in \mathbf{P}$
2. Falls $X \in \mathbf{P}$, dann $\neg X \in \mathbf{P}$
3. Falls \odot ein binärer Junktor ist und $X, Y \in \mathbf{P}$, dann ist $(X \odot Y) \in \mathbf{P}$

Beispiel: $\neg((P_1 \supset P_2) \vee ((P_3 \wedge P_2) \supset \top))$

Anmerkung: Das Symbol \odot wird im weiteren als Schema-Variable für binäre Junktoren verwendet

- Formeller geschrieben lautet die Definition wie folgt:
- Definition (2.2.2*)
 - Für die Menge der aussagenlogischen Formeln $For(\mathcal{L}_{AL})$ gilt:
 1. $At(\mathcal{L}_{AL}) \subseteq For(\mathcal{L}_{AL})$
 2. $\{\neg X \mid X \in For(\mathcal{L}_{AL})\} \subseteq For(\mathcal{L}_{AL})$
 3. $\{(X \odot Y) \mid X, Y \in For(\mathcal{L}_{AL})\} \subseteq For(\mathcal{L}_{AL})$ für jeden binären Junktor \odot
 4. für alle Mengen \mathbf{P} , die die Bedingungen 1–3 erfüllen, gilt:
 $For(\mathcal{L}_{AL}) \subseteq \mathbf{P}$

- Aufgabe (ohne Nummer).

Zeigen Sie, dass es eine eindeutig bestimmte "kleinste Menge", die die in Definition 2.2.2 geforderten Eigenschaften besitzt, gibt und dass sie in allen Mengen, die die Eigenschaften besitzen, enthalten ist. (Tipp: Bilden Sie den Durchschnitt aller Mengen, die obige Eigenschaften haben und zeigen Sie, dass diese Menge auch die Bedingungen erfüllt.)

Zum Selbststudium

Beim Nacharbeiten
der Vorlesung

Aufgabe 2-1

- Geben Sie eine eindeutige kontextfreie Grammatik für die Menge der aussagenlogischen Formeln an, wobei mit jedem binären Junktor auch zugehörige Klammern generiert werden sollen.

Aufgabe 2-2

- Geben Sie eine kontextfreie Grammatik für die Menge der aussagenlogischen Formeln an, die einerseits eindeutig ist (für jede Formel gibt es nur einen Weg, sie zu erzeugen) und andererseits Klammerersparnis gegenüber der Grammatik aus Aufgabe 2-1 möglich ist.
- Auch die Übergenerierung von Klammern ((A)) soll durch diese Grammatik zulässig sein.
- Wie erreichen Sie es, die unterschiedliche Bindungskraft von Junktoren in der Grammatik abzubilden?

- Zeigen Sie jeweils, dass Ihre Grammatik tatsächlich eindeutig ist.

Aufgabe (ohne Nummer)

1. Werden die folgenden Zeichenketten von Ihrer Grammatik generiert? $\neg(P_1)$, $P_3 \wedge P_1 \wedge P_2$ und $P_3 \wedge P_1 \vee P_2$.
2. Welche Teilformeln (s. übernächste Folie) haben die Formeln ((A)), $P_3 \wedge P_1 \wedge P_2$ und $P_3 \wedge P_1 \vee P_2$?

Formel-Repräsentation in den Prolog-Programmen

Konkrete Syntax

Folien: $\neg ((P_1 \supset P_2) \vee ((P_3 \wedge P_2) \supset \top))$

Prolog: $\sim((p1 > p2) \vee ((p3 \& p2) > top))$

Eigentlich ist egal,

- welche Zeichen zur Repräsentation der Junktoren etc. verwendet werden
- welche Klammerkonventionen gelten

Abstrakte Syntax

- Behandelt Formelmengen und Junktoren als algebraische Strukturen

Teilformeln

Definition 2.2.6: Unmittelbare Teilformeln einer Formel

1. Atomare Formeln besitzen keine unmittelbaren Teilformeln.
2. Die einzige unmittelbare Teilformel von $\neg X$ ist X .
3. Für $(X \odot Y)$ – wobei \odot ein binärer Junktor ist – sind X und Y die beiden unmittelbaren Teilformeln.

Definition 2.2.7

Die Menge S der Teilformeln einer Formel X ist die kleinste Menge, die X enthält, und die für alle ihre Elemente auch deren unmittelbare Teilformeln enthält.

[X ist eine unechte oder uneigentliche Teilformel von sich selbst.]

Vgl. Ben-Ari (2001) Def. 2.17

Prinzip der Strukturellen Induktion

Theorem 2.2.3

Eine Eigenschaft Q gilt für alle Formeln der Aussagenlogik, wenn

Induktionsbasis

- [V1] Jede atomare Formel besitzt die Eigenschaft Q .

Induktionsschritt

- [V2] Falls die Formel X die Eigenschaft Q hat, dann hat auch $\neg X$ die Eigenschaft Q .
- [V3] Falls die Formeln X und Y die Eigenschaft Q haben und \odot ein binärer Junktor ist, dann hat auch die Formel $(X \odot Y)$ die Eigenschaft Q .

➤ Dieses Theorem hilft uns, Beweise intern zu strukturieren.

- Ben-Ari (2001) Th. 2.5
- Das Prinzip der strukturellen Induktion erlaubt zu beweisen, dass eine Eigenschaft für alle (unendlich vielen) Formeln gilt. Die Beweise sind, wenn man das Prinzip durchschaut hat, vergleichsweise einfach. Die Teilbeweise entsprechen den Bildungsregeln für die Formeln und können sich jeweils auf den Beitrag der einzelnen Bildungsregeln konzentrieren. Allerdings gibt es durchaus Eigenschaften von Formeln, für die dieses Beweisschema nicht geeignet ist, da die Eigenschaft gar nichts mit dem Formelaufbau zu tun hat (z.B. sollte man ‚Alle Tautologien sind äquivalent‘ nicht nach diesem Prinzip beweisen wollen.)
- Es gibt aber durchaus verschiedene Formen, in denen Induktionsprinzipien formuliert werden können. (Später verwenden wir noch eines, das auf der sog. uniformen Notation beruht). Die Form auf dieser Folie ist diejenige, die sich am einfachsten beweisen lässt.

Beweis: Prinzip der Strukturellen Induktion



Voraussetzungen

- [V1] Jede atomare Formel besitzt die Eigenschaft Q .
- [V2] Falls die Formel X die Eigenschaft Q hat, dann hat auch $\neg X$ die Eigenschaft Q .
- [V3] Falls die Formeln X und Y die Eigenschaft Q haben und \odot ein binärer Junktor ist, dann hat auch die Formel $(X \odot Y)$ die Eigenschaft Q .
- [V4] Es sei M die Menge aller Formeln, die die Eigenschaft Q haben.

Behauptung: $For(\mathcal{L}_{AL}) \subseteq M$

Beweis

Die Voraussetzungen besagen, dass M alle in Definition 2.2.2 angeführten Bedingungen erfüllt. $For(\mathcal{L}_{AL})$ ist die kleinste Menge, die das tut, damit gilt: $For(\mathcal{L}_{AL}) \subseteq M$

Struktur eines Beweises nach dem Prinzip der Strukturellen Induktion

Voraussetzungen: z.B. Definition der Eigenschaft Q

Behauptung

- Alle Formeln der Aussagenlogik haben die Eigenschaft Q .

Induktionsbasis

⇒ Teilbeweis für: Jede atomare Formel hat die Eigenschaft Q .

Induktionsannahme

- Es seien X und Y Formeln, die die Eigenschaft Q haben.
 - kein Beweis erforderlich, keine Einschränkung erlaubt

Induktionsschritt

⇒ Teilbeweis für: $\neg X$ hat die Eigenschaft Q .

⇒ Teilbeweis für: Für jeden binären Junktor \circ gilt:

$(X \circ Y)$ hat die Eigenschaft Q .

- Induktionsschritt greift auf die Induktionsannahme zurück.

Resümee

- Nach dem Prinzip der Strukturellen Induktion hat also jede Formel der Aussagenlogik die Eigenschaft Q .

Prinzip der Strukturellen Rekursion (Fitting)

Theorem 2.2.4

Es existiert genau eine Funktion f definiert über der Menge der Formeln der Aussagenlogik ($\text{For}(\mathcal{L}_{AL})$), so dass gilt:

- **Rekursionsbasis**

Der Wert von f ist auf den atomaren Formeln explizit spezifiziert.

- **Rekursionsschritt**

Der Wert $f(\neg X)$ ist spezifiziert über den Wert von f bzgl. X .

Der Wert $f((X \odot Y))$ ist spezifiziert über den Werten von f bzgl. X und Y , wobei \odot ein binärer Junktor ist.

- Das Prinzip der strukturellen Rekursion ist das grundlegende Definitionsprinzip, um Funktionen über der Menge aller Formeln der Aussagenlogik zu definieren, die von der Wertefestlegung bzgl. der atomaren Formeln ausgehen, und dann für die nicht-atomaren Formeln eine systematische Fortsetzung vornehmen.

Aufgabe 2-3.

Erläutern Sie, warum es wichtig ist, dass die Grammatiken für die Menge der Formeln eindeutig sind?

Prinzip der Strukturellen Rekursion (reformuliert, allgemeine Form)

- Theorem 2.2.4*

Es sei \mathbf{D} eine Menge, und Funktionen seien wie folgt gegeben:

$$f_{At}: \text{At}(\mathcal{L}_{AL}) \rightarrow \mathbf{D}$$

$$f_{\neg}: \text{For}(\mathcal{L}_{AL}) \times \mathbf{D} \rightarrow \mathbf{D}$$

für jeden binären Junktor \odot eine Funktion

$$f_{\odot}: \text{For}(\mathcal{L}_{AL}) \times \text{For}(\mathcal{L}_{AL}) \times \mathbf{D} \times \mathbf{D} \rightarrow \mathbf{D}$$

Dann existiert genau eine Funktion $f: \text{For}(\mathcal{L}_{AL}) \rightarrow \mathbf{D}$, so dass gilt:

- **Rekursionsbasis**

Für jede atomare Formel P ist $f(P) = f_{At}(P)$

- **Rekursionsschritt**

Für jede Formel X gilt: $f(\neg X) = f_{\neg}(X, f(X))$

Für alle Formeln X und Y und jeden binären Junktor \odot gilt:

$$f((X \odot Y)) = f_{\odot}(X, Y, f(X), f(Y))$$

Aufgabe (ohne Nummer)

Beweisen Sie das Prinzip der Strukturellen Rekursion für Formeln der Aussagenlogik.

Prinzip der Strukturellen Rekursion (Spezialisierung)

Theorem 2.2.4**

Es sei \mathbf{D} eine Menge, und Funktionen seien wie folgt gegeben:

$$f_{As}: \mathcal{AS}(\mathcal{L}_{AL}) \rightarrow \mathbf{D}$$

$$f_{op}(\perp), f_{op}(\top) \in \mathbf{D}, f_{op}(\neg): \mathbf{D} \rightarrow \mathbf{D}$$

$$f_{op}(\odot): \mathbf{D} \times \mathbf{D} \rightarrow \mathbf{D}, \text{ für jeden binären Junktor } \odot$$

Dann existiert genau eine Funktion $f: \text{For}(\mathcal{L}_{AL}) \rightarrow \mathbf{D}$, so dass gilt:

- **Rekursionsbasis**

Für jedes Aussagensymbol P ist $f(P) = f_{As}(P)$

Für jede logische Konstante P ist $f(P) = f_{op}(P)$

- **Rekursionsschritt**

Für jede Formel X gilt: $f(\neg X) = f_{op}(\neg)(f(X))$

Für alle Formeln X und Y und jeden binären Junktor \odot gilt:

$$f((X \odot Y)) = f_{op}(\odot)(f(X), f(Y))$$

- Hier haben wir genau zwei Funktionen vorgegeben. Eine, die den Aussagesymbolen Werte zuordnet f_{As} , und eine, die den logischen Symbolen Werte zuordnet f_{op} .
- Der Negation wird als Wert von f_{op} eine einstellige Funktion zugeordnet.
- Den binären Junktoren werden als Wert zweistellige Funktionen zugeordnet.
- Daraus ergibt sich dann wieder eindeutig eine Funktion, die den Formeln Werte zuordnet.
- Beispiel:
Die *Interpretation* für aussagenlogische Formel ist dem Prinzip der strukturellen Rekursion folgend definiert. Die Basis ist in der *Belegung* der atomaren Formeln gegeben, hierauf aufbauend wird der Wert von nicht-atomaren Formeln berechnet.
- Über Bedeutungsfunktionen, die in dieser Form definiert sind, sagt man auch, dass sie dem ‚Kompositionalitätsprinzip‘ genügen: Die Bedeutung des zusammengesetzten Ausdrucks ergibt sich genau aus der Bedeutung der Bestandteile. Die Aussagenlogik genügt in diesem Sinne dem Kompositionalitätsprinzip. Prädikatenlogik und Modallogik sind komplizierter.

Beispiel: Rekursive Definition vom Grad einer Formel

Definition: (**Grad** einer Formel)

$d: \text{For}(\mathcal{L}_{AL}) \rightarrow \mathbb{N}_0$

- **Rekursionsbasis**

Ist P eine atomare Formel, so ist $d(P) = 0$

- **Rekursionsschritt**

Sind X und Y Formeln, so ist:

- $d(\neg X) = d(X) + 1$
- $d((X \odot Y)) = d(X) + d(Y) + 1$

\Rightarrow Jede Formel besitzt einen eindeutig bestimmten Grad.

$d_{As}: \mathcal{As}(\mathcal{L}_{AL}) \rightarrow \mathbb{N}_0, P \mapsto 0$

$d_{op}(\perp) = d_{op}(\top) = 0,$

$d_{op}(\neg): \mathbb{N}_0 \rightarrow \mathbb{N}_0, n \mapsto n + 1$

$d_{op}(\odot): \mathbb{N}_0 \times \mathbb{N}_0 \rightarrow \mathbb{N}_0, (n, m) \mapsto n + m + 1$



Beispiel: Rekursive Definition: Teilformeln

Rekursive Definition (Teilformeln)

- Die Funktion TF , die jede Formel auf die Menge ihrer Teilformeln abbildet, wird rekursiv wie folgt definiert:
 - $TF: For(\mathcal{L}_{AL}) \rightarrow \mathcal{P}(For(\mathcal{L}_{AL}))$
 - **Rekursionsbasis**
Ist P eine atomare Formel, so ist $TF(P) = \{P\}$
 - **Rekursionsschritt**
Sind X und Y Formeln, so ist:
 - $TF(\neg X) = \{\neg X\} \cup TF(X)$
 - $TF((X \odot Y)) = \{(X \odot Y)\} \cup TF(X) \cup TF(Y)$
- \Rightarrow Zu jeder Formel gibt es eine eindeutig bestimmte Menge von Teilformeln.

- Ganz formell lässt sich das wie folgt schreiben:

$$TF_{At}: At(\mathcal{L}_{AL}) \rightarrow \mathcal{P}(For(\mathcal{L}_{AL})), P \mapsto \{P\}$$

$$TF_{\neg}: For(\mathcal{L}_{AL}) \times \mathcal{P}(For(\mathcal{L}_{AL})) \rightarrow \mathcal{P}(For(\mathcal{L}_{AL})), (X, M) \mapsto \{\neg X\} \cup M$$

für jeden binären Junktor \odot

$$TF_{\odot}: For(\mathcal{L}_{AL}) \times For(\mathcal{L}_{AL}) \times \mathcal{P}(For(\mathcal{L}_{AL})) \times \mathcal{P}(For(\mathcal{L}_{AL})) \rightarrow \mathcal{P}(For(\mathcal{L}_{AL})),$$

$$(X, Y, M_X, M_Y) \mapsto \{(X \odot Y)\} \cup M_X \cup M_Y$$

- Beachten Sie, dass hier die allgemeine und nicht die spezielle Form des rekursiven Schemas verwendet wird.

Mögliche Programmierübung

Eine einfache rekursive Funktion über den Formelaufbau

- Besorgen Sie das Programm-Paket zu dieser Vorlesung
- lesen Sie sich die Definitionen der Prädikate durch
- probieren Sie die Prädikate aus und überzeugen Sie sich, dass sie wissen, warum was passiert
 - Schrittweise Abarbeitung z.B. mit Tracer.
- ergänzen sie ein einfaches rekursives Prädikat, z.B.
 - zählen der Junktoren der Formel
 - Liste aller Teilformeln einer Formel
- ergänzen Sie das Vokabular, tragen Sie neue Beispiele ein.

- Lernziel dieser Übung ist, mit Prolog und der Programm-Umgebung (wieder) vertraut zu werden.

```
[wsv7:~/TestProgs/L05progs/aussagenlogik] ce% swipl
Welcome to SWI-Prolog (Multi-threaded, Version 5.4.6)
Copyright (c) 1990-2003 University of Amsterdam.
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software,
and you are welcome to redistribute it under certain conditions.
Please visit http://www.swi-prolog.org for details.

For help, use ?- help(Topic), or ?- apropos(Word).

?- emacs.

Yes
?- [main].
% examples compiled into examples 0.00 sec, 6.168 bytes
% formulae compiled into main 0.01 sec, 2.916 bytes
% modelCheckerAL compiled into main 0.00 sec, 3.768 bytes
% main compiled into main 0.01 sec, 17.696 bytes

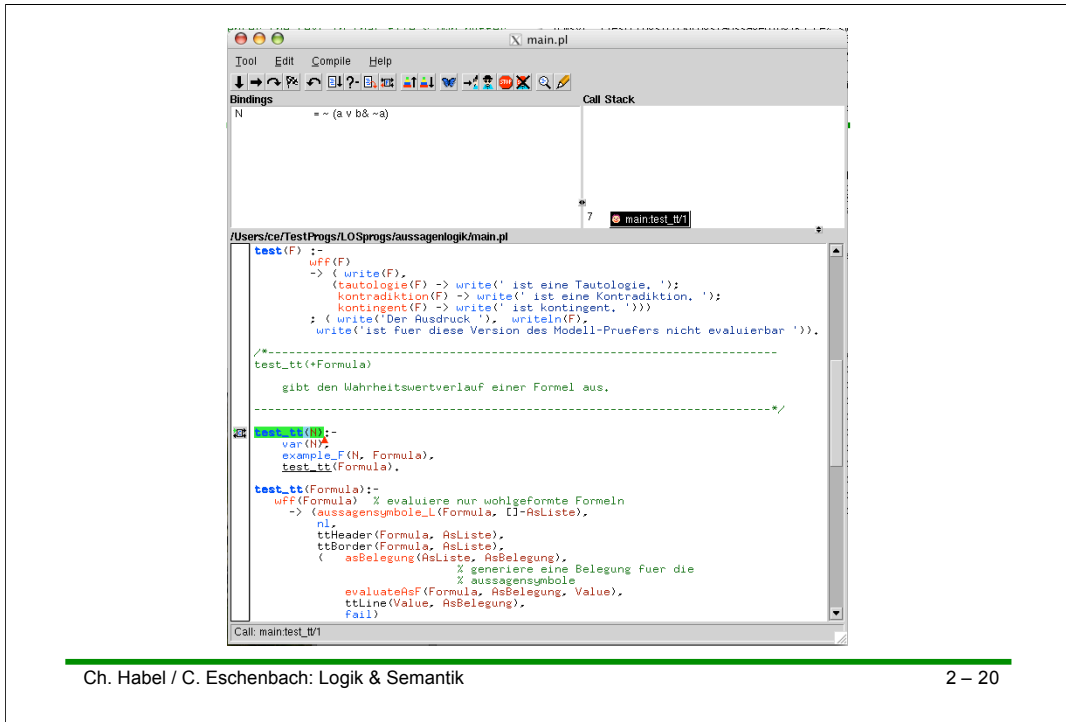
Yes
?- guitracen.
% The graphical front-end will be used for subsequent tracing

Yes
?- trace.

Yes
[trace] ?-
| test_tt("(a v (b & "a))).
```

Ch. Habel / C. Eschenbach: Logik & Semantik 2 - 19

- Die Prolog-Umgebung verlässt man durch 'halt.'



Definition eines logischen Systems: Generelles Schema

Spezifikation

- einer formalen Sprache (zur Repräsentation)
- von Evaluations- / Interpretationsprinzipien
- semantischer Kategorisierungen und Beziehungen
- Ableitungs-, Beweisverfahren

Wahrheitswerte – Belegungen – Interpretationen

Wahrheitswerte

- Der Semantik liegt ein System von Wahrheitswerten zugrunde:
Zweiwertige Logik : $\mathbf{Tr} = \{ \mathbf{t}, \mathbf{f} \}$

Belegungen (Wahrheitswertzuweisungen)

- sind Funktionen $\mathcal{A}_S(\mathcal{L}_{AL}) \rightarrow \mathbf{Tr}$, die den Aussagensymbolen Wahrheitswerte zuordnen (freie Interpretationen).

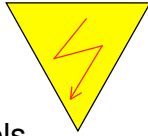
Interpretationen (Boolesche Auswertungen, *valuations*)

- sind Funktionen $\mathcal{F}or(\mathcal{L}_{AL}) \rightarrow \mathbf{Tr}$,
 - dem Prinzip der strukturellen Rekursion entsprechend definiert
 - basierend auf einer Belegung
 - und einer festen Interpretation \mathcal{V}_{op} von logische Konstanten und Junktoren

Wahrheitswertfunktionale Interpretationen von Junktoren

Boolesche Auswertungen: Interpretation von Junktoren

- Junktoren der Stelligkeit n werden durch n -stellige Wahrheitswertfunktionen interpretiert.
 - ⇒ Wahrheitswertfunktional
 - $\nu_{op}(\perp), \nu_{op}(\top) \in \text{Tr}$
 - $\nu_{op}(\neg): \text{Tr} \rightarrow \text{Tr}$
 - $\nu_{op}(\odot): \text{Tr} \times \text{Tr} \rightarrow \text{Tr}$ falls \odot ein binärer Junktor.
- Festzulegen sind also die Interpretationen der logischen Symbole.
 - Es gibt 4 einstellige Wahrheitswertfunktionen, z.B.:
 $neg: \text{Tr} \rightarrow \text{Tr}, t \mapsto f, f \mapsto t$
 - Es gibt 16 zweistellige Wahrheitswertfunktionen:
 $\text{Tr} \times \text{Tr} \rightarrow \text{Tr}$

- Hier – und in den kommenden Folien – ist eine **wichtige Abweichung** gegenüber der Darstellung bei Fitting. 
 - Für ein richtiges Verständnis des Zusammenspiels von Syntax und Semantik ist es wichtig, den Unterschied zwischen den beiden Ebenen zu kennen.
 - Syntaktische Ebene: **Konnektive (Junktoren)**
Dies sind Teile des Vokabulars der Sprache
 - Semantische Ebene: **Wahrheitswertfunktionen**
Diese werden verwendet, um Ausdrücke der logischen Sprache zu interpretieren.
- Syntaktische und semantische Ebene dürfen nicht durcheinander gebracht werden!
- Wenn jemand – wirklich – weiß, wie Logik funktioniert (etwa M. Fitting), dann kann er/sie es sich leisten, die beiden Ebenen nicht zu trennen. Fitting (p. 15). „to make the relationship between symbol and operation more memorable, we will overload our notation. From now on, \wedge will be a binary operation *symbol* of our formal language and will also denote an *operation* on Tr ,....this double usage should cause no difficulty, as context will make clear which is meant.
- Gegebenenfalls wird in der Prüfung geklärt, ob der Unterschied wirklich klar ist.

Tafel der zweistelligen Funktionen: $\text{Tr} \times \text{Tr} \rightarrow \text{Tr}$

		f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}	f_{16}
t	t	f	t	f	t	f	t	f	t	f	t	f	t	f	t	f	t
t	f	f	f	t	t	f	f	t	t	f	f	t	t	f	f	t	t
f	t	f	f	f	f	t	t	t	t	f	f	f	f	t	t	t	t
f	f	f	f	f	f	f	f	f	f	t	t	t	t	t	t	t	t

Dieses System von Wahrheitswertfunktionen (WWF) ist vollständig.

Primäre und sekundäre Wahrheitswertfunktionen (WWF)

		f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}	f_{16}
t	t	f	t	f	t	f	t	f	t	f	t	f	t	f	t	f	t
t	f	f	f	t	t	f	f	t	t	f	f	t	t	f	f	t	t
f	t	f	f	f	f	t	t	t	t	f	f	f	f	t	t	t	t
f	f	f	f	f	f	f	f	f	f	t	t	t	t	t	t	t	t

triviale und fast triviale WWF (6 Stück)	Sind nicht in relevanter Weise von der Kombination der zwei Wahrheitswerte abhängig
Primäre WWF (8 Stück)	Haben eine 3 : 1 Verteilung
Sekundäre WWF (2 Stück)	Haben eine 2 : 2 Verteilung

Ch. Habel / C. Eschenbach: Logik & Semantik

2 – 25

- Triviale und nicht-triviale WWF sind für die Interpretation der binären Junktoren nicht interessant, da sie besser durch Konstanten oder unäre Junktoren vertreten werden.
- Der Unterschied zwischen primären und sekundären WWFen wird später – u.a. bei der Uniformen Notation – eine Rolle spielen.

Interpretation der Junktoren

Definition

Für die **Boolesche Operatoreninterpretation** v_{op} gilt:

$$v_{op}(\perp) = \mathbf{f}, v_{op}(\top) = \mathbf{t}, v_{op}(\neg) = \mathit{neg}$$

und für die binären Junktoren:

\odot		\wedge	\supset	$\not\supset$	$\not\equiv$	\vee	\downarrow	\equiv		\subset		\supset	\uparrow				
$v_{op}(\odot)$	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}	f_{16}	
t	t	f	t	f	t	f	t	f	t	f	t	f	t	f	t	f	t
t	f	f	f	t	t	f	f	t	t	f	f	t	t	f	f	t	t
f	t	f	f	f	f	t	t	t	t	f	f	f	f	t	t	t	t
f	f	f	f	f	f	f	f	f	f	t	t	t	t	t	t	t	t

Ch. Habel / C. Eschenbach: Logik & Semantik

2 – 26

- Die Funktion f_{14} wird auch als ‚materiale Implikation‘ bezeichnet. Viele Logiker sind der Meinung, dass diese Funktion das natürliche ‚wenn ... dann...‘ nicht adäquat modelliert. Viele alternative Logiken (Modallogik, Relevanzlogik, Intuitionistische Logik) beschäftigen sich entsprechend mit der Frage, wie das natürliche ‚wenn ... dann...‘ besser zu modellieren ist.

Boolesche Auswertung – Interpretation (1)

Definition 2.4.1*

Gegeben sei eine Belegung ν_{AS} . Die nach dem Rekursionsprinzip auf der Basis von ν_{AS} und ν_{op} eindeutig definierte Funktion $\nu: For(\mathcal{L}_{AL}) \rightarrow Tr$ nennen wir **Boolesche Auswertung (Interpretation)**.

Der Einfachheit halber schreiben wir im weiteren auch einfach \odot oder $\nu(\odot)$ für $\nu_{op}(\odot)$ und ν anstelle von ν_{AS} . Außerdem benutzen wir bei \odot auch die Infix-Notation.

- Wichtig ist hier, dass alle Interpretationen die logischen Operatoren (logische Konstanten, Junktoren) gleich bewerten. Unterschiede zwischen den Interpretationen beruhen stets auf Unterschieden in der Belegung der Aussagesymbole.
- Operatoren mit fester Interpretation nennen wir auch ‚logisch‘, da die Logik ihnen ihre Bedeutung zuordnet.
- Dieses ist die von Fitting verwendete Schreibweise, der Junktoren, d.h. syntaktische Objekte und Wahrheitswertfunktionen gleich bezeichnet. Vgl. das Zitat in den Anmerkungen zu Folie 16.
- Wir werden im weiteren meist der Fittingschen Schreibweise folgen, durch den Farbcode die Unterscheidung aber aufrechterhalten.

Boolesche Auswertung – Interpretation (2)

Definition 2.4.1

Eine **Boolesche Auswertung** (Interpretation) ist eine Abbildung

$v: \text{For}(\mathcal{L}_{AL}) \rightarrow \mathbf{Tr}$, die die folgenden Bedingungen erfüllt:

1. $v(\top) = t$ $v(\perp) = f$
2. Falls $X \in \text{For}(\mathcal{L}_{AL})$, dann $v(\neg X) = \text{neg}(v(X))$ $\neg = \text{neg}$
3. Falls \odot ein binärer Junktor, dann gilt:
$$v((X \odot Y)) = \odot(v(X), v(Y))$$
$$= v(X) \odot v(Y)$$

Dualität von Operatoren

Definition 2.4.6

Seien \odot und \otimes zwei binäre Operatoren über Tr .

\otimes ist genau dann *dual* zu \odot , wenn für alle Wahrheitswerte $x, y \in \text{Tr}$ gilt:

$$\text{neg}(\odot(x, y)) = \otimes(\text{neg}(x), \text{neg}(y))$$

Wenn \otimes dual zu \odot ist, dann werden auch die korrespondierenden Junktoren \boxtimes und \boxdot als dual bezeichnet.

Anm.: Dualität ist symmetrisch.

Beispiel: Konjunktion (\wedge) und Disjunktion (\vee) sind dual.

- Aufgabe (ohne Nummer).
 - a. Finden Sie weitere Paare dualer Wahrheitsfunktionen.
 - b. Beweisen Sie die Symmetrie von *Dualität*.
- Dualität ist auch für unäre Junktoren interessant (im Allgemeinen aber nicht so gut über Wahrheitswerte definierbar):
Seien \prec und \succ zwei unäre Junktoren.
 \prec ist *dual* zu \succ , falls für alle Formel X gilt:
 $\neg\prec X$ und $\succ\neg X$ sind äquivalent.
- All- und Existenzquantor in der Prädikatenlogik sind dual.
- Möglichkeit und Notwendigkeit in der Modallogik sind dual.

Variationen bei Wahrheitswerten

Satz 2.4.3

Es sei $As(X)$ die Menge der Teilformeln einer aussagenlogischen Formel X , die Aussagensymbole sind. Wenn zwei Interpretationen v_1 und v_2 für alle Elemente von $As(X)$ übereinstimmen, dann gilt auch $v_1(X) = v_2(X)$.

⇒ Alle Variationen beruhen auf unterschiedlichen **Belegungen**.

⇒ Nur vorkommende Aussagensymbole beeinflussen den Wahrheitswert.

⇒ Wahrheitstabellen zeigen genau die Variationen in den Belegungen der betroffenen Aussagensymbole.

- Ben-Ari (2001) Th. 2.10
- Aufgabe (ohne Nummer)

Geben Sie eine rekursive Definition für As an.

Wahrheitstafel

Zelle: Wert einer Formel unter einer Belegung

Zeile

- (Klasse von) Belegungen / Interpretationen

Spalten

- **Wahrheitswertverlauf** einer Formel

	P_1	P_2	P_3	X	Y	Z
v_1	$v_1(P_1)$	$v_1(P_2)$	$v_1(P_3)$	$v_1(X)$	$v_1(Y)$	$v_1(Z)$
v_2	$v_2(P_1)$	$v_2(P_2)$	$v_2(P_3)$	$v_2(X)$	$v_2(Y)$	$v_2(Z)$
v_3	$v_3(P_1)$	$v_3(P_2)$	$v_3(P_3)$	$v_3(X)$	$v_3(Y)$	$v_3(Z)$
...						
v_8	$v_8(P_1)$	$v_8(P_2)$	$v_8(P_3)$	$v_8(X)$	$v_8(Y)$	$v_8(Z)$

Ch. Habel / C. Eschenbach: Logik & Semantik

2 – 31

- Für mathematisch versierte: Wahrheitswertverläufe können als Funktionen, die Belegungen / Interpretationen auf Wahrheitswerte abbilden, angesehen werden. Wie in der Wahrheitstafel ersichtlich, gibt es zu jeder Formel einen systematisch zugeordneten Wahrheitswertverlauf.
- Als ‚Bedeutung einer aussagenlogischen Formel‘ sollte man eher den zugehörigen Wahrheitswertverlauf als ihren Wahrheitswert unter einer Belegung auffassen.

Definition eines logischen Systems: Generelles Schema

Spezifikation

- einer formalen Sprache (zur Repräsentation)
- von Evaluations- / Interpretationsprinzipien
- **semantischer Kategorisierungen und Beziehungen**
- Ableitungs-, Beweisverfahren

Grundkonzepte der Semantik

Definitionen (2.4.4, 2.4.5, ...)

Eine Formel X heißt genau dann

Tautologie (*allgemeingültig*), wenn für jede Interpretation v gilt:

$v(X) = \mathbf{t}$. \Rightarrow Wahrheitswertverlauf konstant \mathbf{t} .

Kontradiktion (*unerfüllbar*), wenn für jede Interpretation v gilt:

$v(X) = \mathbf{f}$. \Rightarrow Wahrheitswertverlauf konstant \mathbf{f} .

erfüllbar, wenn für eine Interpretation v gilt: $v(X) = \mathbf{t}$.

falsifizierbar, wenn für eine Interpretation v gilt: $v(X) = \mathbf{f}$.

kontingent, wenn sie erfüllbar und falsifizierbar ist.

\Rightarrow Wahrheitswertverlauf nicht konstant.



- Ben-Ari (2001) Def. 2.24
- Englische Begriffe:
 - *valid* : allgemeingültig
 - *satisfiable* : erfüllbar
 - *unsatisfiable, contradictory* : unerfüllbar
 - *not-valid, falsifiable* : falsifizierbar
- Diese und die folgenden Definitionen bilden einen wohletablierten Standard in der klassischen, zweiwertigen Aussagenlogik.
- Die Begriffe werden auf andere Logiken (z.B. Prädikatenlogik) direkt übertragen. Es ist dabei jeweils zu klären, wie die Interpretation von Formeln erfolgt.
- Im Zusammenhang mit den mehrwertigen Logiken zeigt sich aber, dass die Übertragung der Begrifflichkeit nicht immer einfach ist.
- Der Begriff *falsifizierbar* gehört (leider) nicht zum wohletablierten Standard (s. aber: Ben-Ari, 2001).

Tautologien – Kontingente Formeln – Kontradiktionen

- Tautologien und Kontradiktionen sind **logisch interessant**,
 - da sie sich unter allen Belegungen / Interpretationen gleich verhalten
- Kontingente Formeln sind **epistemisch interessant**,
 - da sie Aussagen über die Welt machen

Erfüllbare Formeln		Kontradiktionen (unerfüllbar)
Tautologien (nicht falsifizierbar)	Kontingente Formeln	
Falsifizierbare Formeln		
Wahrheitswertverlauf konstant t	Wahrheitswertverlauf nicht konstant	Wahrheitswertverlauf konstant f

Äquivalenz und Folgerung

Definitionen

Zwei Formeln X und Y sind genau dann *äquivalent*, wenn für jede Interpretation v gilt: $v(X) = v(Y)$.

⇒ Identische Wahrheitswertverläufe

Formel Y *folgt* genau dann *aus* Formel X , wenn jede Interpretation v , die $v(X) = \mathbf{t}$ liefert, auch $v(Y) = \mathbf{t}$ liefert. In Zeichen: $X \models Y$

Eine Menge S von Formeln ist genau dann *erfüllbar*, wenn es eine Interpretation v gibt, die für alle $X \in S$ $v(X) = \mathbf{t}$ ergibt.

Eine Formel Y *folgt* genau dann *aus* einer Menge von Formeln S , wenn jede Interpretation v , die für alle Formeln X aus S $v(X) = \mathbf{t}$ liefert, auch $v(Y) = \mathbf{t}$ liefert. In Zeichen: $S \models Y$

- Vgl. Ben-Ari (2001), 2.13, 2.30, 2.36
- Englische Begriffe:
 - *logical consequence* : Folgerung
 - *follows from* : folgt aus
- Beachten Sie, dass die Menge S von Formeln leer sein kann aber auch unendlich viele Formeln enthalten kann.
- Die leere Menge ist erfüllbar.

Theorien

Definition (Ben-Ari (2001), 2.41*)

Eine Formelmengens T ist genau dann *unter Folgerung abgeschlossen*, wenn alle aus ihr folgerbaren Formeln in ihr enthalten sind ($F \in T$ gdw. $T \models F$). Eine Formelmengens ist genau dann eine *Theorie*, wenn sie unter Folgerung abgeschlossen ist.

Es sei U eine (endliche) Formelmengens. $T(U) = \{F \mid U \models F\}$ ist die *Theorie von U* . Die Formeln aus U sind die *Axiome* und $T(U)$ ist (*endlich*) *axiomatisierbar*.

Aufgabe

- Gibt es eine kleinste bzw. größte Theorie (bezogen auf die Ordnung der Mengeninklusion)?
- Wenn ja, wie lassen sie sich spezifizieren?
- Wieviele inkonsistente Theorien gibt es?

Aufgabe

- Zeigen Sie, dass $T(U)$ für beliebige Formelmengens U unter Folgerung abgeschlossen ist.

Zu beachten ist, dass für den hier eingeführten Begriff von Axiomatisierbarkeit gilt, dass die Menge der Tautologien durch die leere Menge axiomatisiert wird. Das bedeutet, dass der Begriff nicht auf die Logik selbst anwendbar ist, sondern nur auf Theorien anderer Gegenstandsbereiche (Arithmetik, Geometrie, ...), die mit Hilfe der Logik beschrieben werden.

Programmierübung

Semantische Auswertungen

- für aussagenlogische Formeln gibt es endlich viele verschiedene Belegungen, die sich konstruktiv bestimmen lassen
 - Beispiele für Berechnung semantischer Eigenschaften (Tautologie, erfüllbar, ...) von aussagenlogischen Formeln sind gegeben
 - Ergänzen Sie Prädikate zur Bestimmung von Äquivalenz von Formeln und Folgerbarkeit aus Formeln
 - Ergänzen Sie Prädikate zur Bestimmung der Erfüllbarkeit von Formelmengen/-listen und zur Folgerung aus Formelmengen/-listen

Interaktionen zwischen semantischen Begriffen

Theoreme

- Alle **Tautologien** sind **äquivalent**.
- Alle **Kontradiktionen** sind **äquivalent**.
- (Kontingente Formeln müssen nicht äquivalent sein.)

- Jede **Tautologie** **folgt** aus jeder Formelmenge (auch der leeren).
- Aus **Tautologien** **folgen** nur **Tautologien**.

- Aus jeder **unerfüllbaren** Formelmenge **folgt** jede Formel.
- **Kontradiktionen** **folgen** nur aus **unerfüllbaren** Formelmengen.

- Zwei Formeln sind genau dann **äquivalent**, wenn jede aus der anderen **folgt**.

- Im folgenden schreibe ich kurz $\models X$ für ‚ X ist eine Tautologie‘ und $X \models$ für ‚ X ist eine Kontradiktion‘.
- Aufgabe (ohne Nummer)
Geben Sie Beweise (oder Beweisideen) für die Theoreme an.
Überlegen Sie sich aufgrund der Beweise, welche Merkmale der klassischen Aussagenlogik für die Gültigkeit der Aussagen entscheidend sind.

Entscheidungsprozeduren

Definition (Ben-Ari (2001), 2.26)

Es sei \mathbf{V} eine Menge von Formeln der Logiksprache \mathcal{L} ($\mathbf{V} \subseteq \text{For}(\mathcal{L})$). Ein Algorithmus P ist genau dann eine **Entscheidungsprozedur für \mathbf{V}** , wenn P bei Eingabe einer beliebigen Formel $F \in \text{For}(\mathcal{L})$ terminiert und die Antwort ‚ja‘ für $F \in \mathbf{V}$ und die Antwort ‚nein‘ für $F \notin \mathbf{V}$ ausgibt.

Aussagenlogik

Für viele semantisch charakterisierte Formelmengen lassen sich Entscheidungsprozeduren auf der Basis von Wahrheitstafeln spezifizieren.

Andere Verfahren beruhen auf syntaktischen Ersetzungsverfahren.

Folgende Theoreme erlauben die einfache Reduktion von solchen Entscheidungsfragen aufeinander.

Interaktion der Junktoren mit den semantischen Begriffen

Theoreme

- Formel X ist genau dann eine **Tautologie**,
 - wenn $\neg X$ eine **Kontradiktion** (unerfüllbar) ist.
- Zwei Formeln X und Y sind genau dann **äquivalent**,
 - wenn $(X \equiv Y)$ eine **Tautologie** ist.
- Formel X **folgt** genau dann aus Formel Y ,
 - wenn $(Y \supset X)$ eine **Tautologie** ist.
 - wenn $(Y \wedge \neg X)$ eine **Kontradiktion** ist.
- Formel X **folgt** genau dann aus $\{Y_1, Y_2, \dots, Y_{n-1}, Y_n\}$,
 - wenn X aus $\{Y_1, Y_2, \dots, (Y_{n-1} \wedge Y_n)\}$ **folgt**.
 - wenn X aus $(Y_1 \wedge (Y_2 \wedge \dots (Y_{n-1} \wedge Y_n) \dots))$ **folgt**.
 - wenn $((Y_1 \wedge (Y_2 \wedge \dots (Y_{n-1} \wedge Y_n) \dots)) \supset X)$ eine **Tautologie** ist.
 - wenn $((Y_1 \wedge (Y_2 \wedge \dots (Y_{n-1} \wedge Y_n) \dots)) \wedge \neg X)$ eine **Kontradiktion** ist.

- Diese Theoreme gelten in der klassischen Aussagenlogik und Prädikatenlogik. Es gibt aber auch Logiken, in denen sie nicht gelten. Ihre Gültigkeit ist also keine Selbstverständlichkeit sondern das Resultat bestimmter Entscheidungen bei dem Design der klassischen Logik.
- Wir schreiben im Folgenden auch kurz $\models (X \equiv Y)$ für X und Y sind *äquivalent*
- Aufgabe (ohne Nummer)
Geben Sie Beweise (oder Beweisideen) für die Theoreme an. Überlegen Sie sich aufgrund der Beweise, welche Merkmale der klassischen Aussagenlogik für die Gültigkeit der Aussagen entscheidend sind.

Ersetzungstheorem

Definition (Ersetzung)

Es sei P ein Aussagensymbol, F und X seien Formeln.

$F\{P/X\}$ bezeichnet die Formel, die aus F entsteht, wenn alle Vorkommnisse von P in F durch die Formel X ersetzt werden.

Theorem 2.5.1* (Ersetzungstheorem Version 1)

Es seien P ein Aussagensymbol, F , X und Y aussagenlogische Formeln und v eine Interpretation.

Falls $v(X) = v(Y)$, dann gilt: $v(F\{P/X\}) = v(F\{P/Y\})$



- Die rekursive Definition sieht wie folgt aus:
 - $\{P/X\}: \text{For}(\mathcal{L}_{AL}) \rightarrow \text{For}(\mathcal{L}_{AL})$
 - $P\{P/X\} = X$
 - $Q\{P/X\} = Q$, für alle atomaren Formeln $Q \neq P$
 - Sind Y und Z Formeln, so ist:
 - $(\neg Y)\{P/X\} = \neg(Y\{P/X\})$
 - $(Y \odot Z)\{P/X\} = ((Y\{P/X\}) \odot (Z\{P/X\}))$
- Fitting führt eine andere Schreibweise für Ersetzungen ein, die es aber nicht erlauben, entsprechende Funktionen rekursiv zu definieren:
- **Ersetzung**
Ist P ein Aussagensymbol in der Formel F , so schreiben wir $F(P)$, wenn wir auf das Vorkommen von P in F hinweisen wollen.
 $F(X)$ bezeichnet die Formel, die aus $F(P)$ entsteht, wenn alle Vorkommnisse von P in F durch die Formel X ersetzt werden.
- Ersetzungstheoreme sind insbesondere deswegen von Bedeutung, da sie die Rechtfertigung für „semantisch gerechtfertigte“ Umformungen darstellen.

Vgl. auch Ben-Ari (2001), Def. 2.19

Ersetzungstheorem: Beweis

Voraussetzung: P ist ein Aussagensymbol, X, Y sind Formeln und v eine Interpretation mit $v(X) = v(Y)$

Arbeitsdefinition:

Eine Formel F heißt **gut**, falls $v(F\{P/X\}) = v(F\{P/Y\})$.

Behauptung: Alle Formeln sind gut.

Induktionsbasis: Zu zeigen: jede atomare Formel ist gut.

Sei F eine atomare Formel. Dann gibt es zwei Fälle:

- (1) $F = P$: dann: $F\{P/X\} = X$ und $F\{P/Y\} = Y$
 F ist gut, denn $v(X) = v(Y)$ (nach Voraussetzung)
- (2) $F \neq P$: dann: $F\{P/X\} = F = F\{P/Y\}$
also ist F gut

- Die Eigenschaft **gut** für Formeln, ist genau die, die es für das Ersetzungstheorem zu beweisen gilt.
- ➔ Der Beweis folgt dem Prinzip der strukturellen Induktion im Hinblick auf die Eigenschaft **gut**.
- Der Fall (2): Generell gilt: Wenn P nicht in F vorkommt, dann wird die Ersetzung von P in F nichts „bewirken“, also: $F\{P/X\} = F = F\{P/Y\}$

Ersetzungstheorem: Beweis (Fortsetzung)

Induktionsannahme: Es seien F und G gute Formeln.

Induktionsschritt: Es sei \odot ein binärer Junktor.

Zu zeigen: Die Formel $(F \odot G)$ ist gut.

$v((F \odot G)\{P/X\})$	$= v((F\{P/X\} \odot G\{P/X\}))$	Def. Ersetzung
	$= v(F\{P/X\}) \odot v(G\{P/X\})$	Boolesche Auswertung
	$= v(F\{P/Y\}) \odot v(G\{P/Y\})$	Induktionsannahme
	$= v((F\{P/Y\} \odot G\{P/Y\}))$	Boolesche Auswertung
	$= v((F \odot G)\{P/Y\})$	Def. Ersetzung

Zum Üben: Teilbeweis für $\neg F$

Resümee: Nach dem Prinzip der Strukturellen Induktion ist also jede Formel der Aussagenlogik *gut*.

- **Aufgabe 2-4.**
- a) Führen Sie für den Beweis des Theorems (2.5.1) [Ersetzungstheorem] den Induktionsschritt für Formeln mit dem unären Junktor “ \neg ” durch.
- b) Überprüfen Sie den Beweis daraufhin, an welchen Stellen es wichtig ist, dass es sich bei der auf die Formeln angewendete Funktion um eine Interpretation handelt. Auf welche Funktionen können Sie den Satz verallgemeinern?

Ein weiteres Ersetzungstheorem

Theorem 2.5.2* (Ersetzungstheorem Version 2)

Falls X und Y äquivalent sind,
dann sind auch $F\{P/X\}$ und $F\{P/Y\}$ äquivalent
($\equiv F\{P/X\} \equiv F\{P/Y\}$)

Beweis

Sei v eine beliebige Boolesche Auswertung. Da X und Y äquivalent sind, ist $v(X) = v(Y)$ und nach Theorem 2.5.1 gilt dann auch $v(F\{P/X\}) = v(F\{P/Y\})$. Da die Wahl der Belegung unbeschränkt war, haben wir damit gezeigt, dass $F\{P/X\}$ und $F\{P/Y\}$ äquivalent sind.

Diese Form des Ersetzungstheorems rechtfertigt „Umformungen“ – etwa die Erstellung gewisser Normalformen – als Mittel für semantische Untersuchungen.

Vgl. Ben-Ari (2001) Th. 2.21

Aufgabe 2-5.

- Beweisen Sie folgendes Ersetzungstheorem: Wenn F eine Tautologie ist, P ein Aussagensymbol und X eine beliebige Formel, dann ist $F\{P/X\}$ eine Tautologie.
- Zeigen Sie (durch ein Gegenbeispiel), dass folgende Aussage nicht gilt: Wenn F erfüllbar ist, P ein Aussagensymbol und X eine beliebige Formel, dann ist $F\{P/X\}$ erfüllbar.

Ein weiteres Ersetzungstheorem

NEU 2007

Theorem (Ersetzungstheorem Version 3)

Es sei \mathbf{M} eine Formelmenge, so dass $\mathbf{M} \models (X \equiv Y)$. Dann gilt auch

- $\mathbf{M} \models (F\{P/X\} \equiv F\{P/Y\})$ und
- aus \mathbf{M} folgt $F\{P/X\}$ gdw. aus \mathbf{M} $F\{P/Y\}$ folgt.

Diese Form des Ersetzungstheorems rechtfertigt „Umformungen“ auf Basis weiterer Annahmen als Mittel für semantische Untersuchungen.

- Beweis

Ist ν eine Boolesche Auswertung, die alle Formeln aus \mathbf{M} wahr macht, dann macht (wegen $\mathbf{M} \models (X \equiv Y)$) ν auch $(X \equiv Y)$ wahr und damit ist $\nu(X) = \nu(Y)$. Nach Theorem 2.5.1 gilt dann auch $\nu(F\{P/X\}) = \nu(F\{P/Y\})$ und damit $\nu((F\{P/X\} \equiv F\{P/Y\})) = \mathbf{t}$. Ist nun für alle solche Auswertungen auch noch $\nu(F\{P/X\}) = \mathbf{t}$, dann ist natürlich auch $\nu(F\{P/Y\}) = \mathbf{t}$ (und umgekehrt.)