FGI-1

Formale Grundlagen der Informatik

Vorlesung
Michael Köhler-Bußmeier, Christopher Habel

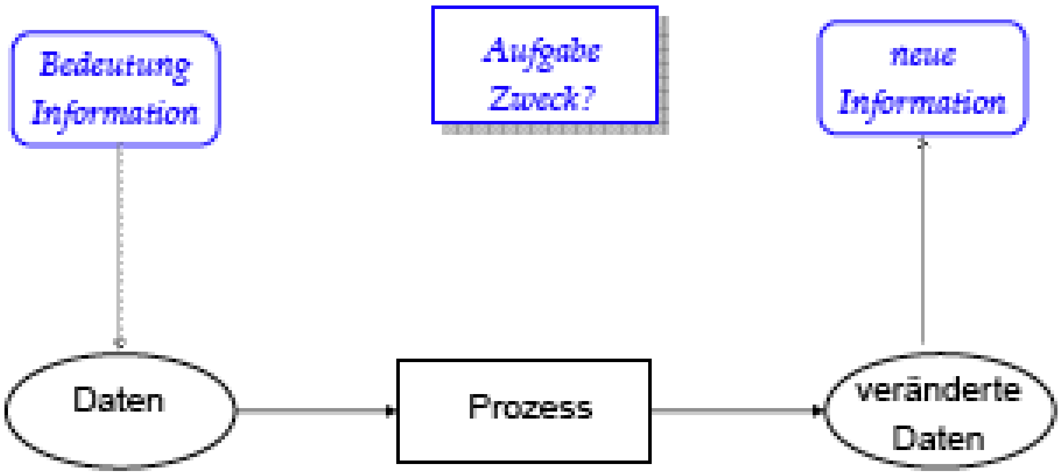
Materialien zur Logik

Christopher Habel und Carola Eschenbach
nach
Uwe Schöning (2000): *Logik für Informatiker*. Heidelberg: Spektrum.
Marcus Spies (2003): *Einführung in die Logik*. Heidelberg: Spektrum.

Sommersemester 2013
Universität Hamburg

Beschreibungen & Verarbeitungen: Die SE-1 Übersicht

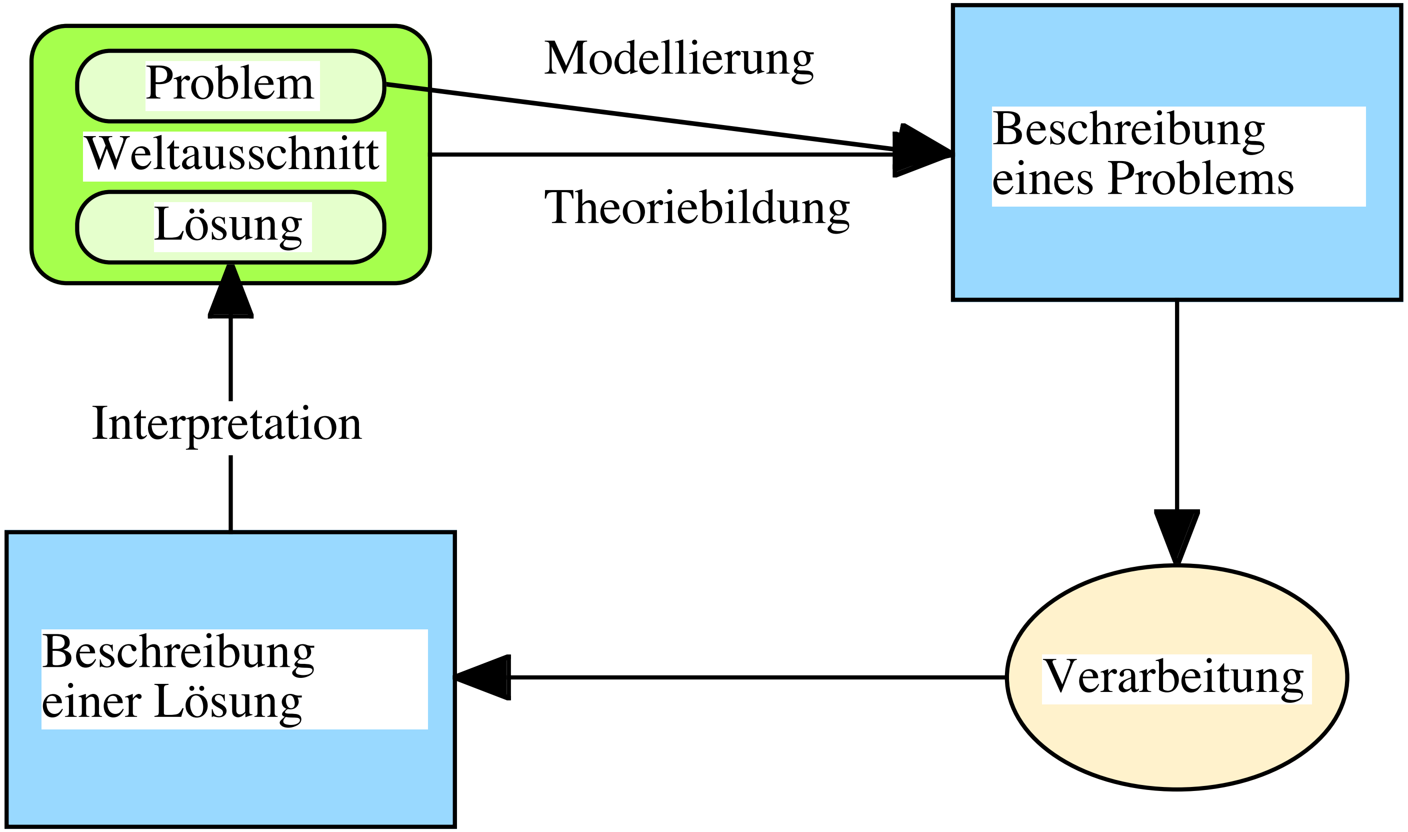
Datenverarbeitung, Software, Algorithmus



- **Datenverarbeitungsprozesse** können als **Software-Programme** formuliert und auf dem Computer ausgeführt werden.
- Software-Programme realisieren **maschinelle Verfahren**, wie z.B. die Auswertungsvorschriften für einen Wert.
- Diese Verfahren bezeichnen wir als **Algorithmen**

SE1 - Intro 6

Beschreibungen und Verarbeitungen allgemein



Das Informationsverarbeitungsparadigma

Grundannahmen der Informatik

- Problemlösen basiert auf Informationsverarbeitung
- „Berechnung ist Informationsverarbeitung – Informationsverarbeitung ist Berechnung"
- Informationsverarbeitung ist Manipulation von Symbolen eines Symbolsystems

Symbolsysteme

- Inventar von Basis-Symbolen (Lexikon, Alphabet)
- Regeln der Wohlgeformtheit von Symbolkombinationen (Syntax)
- Systematische Beziehung von Symbolen und Symbolkombinationen einerseits und Bedeutungen andererseits (Semantik)

➔ **Formale Sprachen + Abstrakte Mechanismen (Automaten & Maschinen) zur Symbolverarbeitung**

Logik: Gegenstand und Zielsetzung

Was untersucht die Logik?

- Schlüsse und Argumente
  - insbesondere ihre formale Rechtfertigung
- Beweise und ihre Struktur (oft als ‚Form‘ bezeichnet)
  - Kommunikation von Begründungszusammenhängen
- Sprachen, in denen die Welt beschrieben werden kann, so dass
  - mit diesen Beschreibungen formal gerechtfertigt geschlossen werden kann
  - über diese Beschreibungen rational argumentiert werden kann
- Theorien und ihre Konsequenzen
  - Vorhersagekraft einer Theorie
  - Bewertung der Theorie über die Bewertung ihrer Konsequenzen

➔ **Eine spezifische Familie formaler Sprachen, bei der die Bedeutung der Zeichenketten systematisch berechnet wird, und die deswegen, für das Lösen von Problemen semantisch gerechtfertigt eingesetzt werden können.**

Lernziele dieser Vorlesung

Fachterminologie der Theoretischen Informatik, d.h. der Logik, der formalen Sprachen, der Automaten und der Komplexitätstheorie

- Ausdrücke mit spezifisch festgelegter Bedeutung, mit denen Symbolsysteme, die Struktur und Bedeutung von Zeichenketten und Berechnungsprozesse über Symbolsystemen gesprochen wird.

Aussagenlogik und Prädikatenlogik

- Grundprinzipien und Eigenschaften jeweils eines ausgewählten Dialektes.
- Verfahren zum Umgang mit Formeln

Formale Grammatiken und Automaten, Komplexität

- Grundprinzipien und Eigenschaften der wichtigsten Typen von Grammatiken und Automaten.
- Zusammenhänge zwischen der Komplexität von Berechnungsprozessen und den strukturellen Eigenschaften von Problemklassen

Beweisen

- als wissenschaftliche Technik

Zum Selbststudium

Folien mit diesem Titel werden in der Vorlesung nicht aufgelegt

- Sie können dennoch Informationen enthalten, die für die Bearbeitung der Übungsaufgaben oder das Gesamtverständnis nützlich sind.
- Auch vor der Klausur ist es sinnvoll, diese Seiten anzusehen.
- Wenn auf Seiten zum Selbststudium komplexere Beweise stehen, dann kann man davon ausgehen, dass bei der Klausur solch ein Beweis nicht ‚abgefragt' wird. Wer aber eine Begründung sucht, warum die bewiesene Behauptung gilt, wird hier fündig.

Aussagenlogik: Grundbegriffe, Syntax

Aussagen der natürlichen Sprache

- sind Sätze, die **wahr** oder **falsch** sein können,
  (auch wenn wir nicht wissen, ob sie **wahr** oder **falsch** sind)

Beispiele

• Hamburg ist die deutsche Stadt mit der zweitgrößten Einwohnerzahl.	**wahr**
• Hamburg liegt südlich von München.	**falsch**
• Am 6. April 1299 fielen in Hamburg 17,5 mm Niederschlag.	??
• Am 27. April 2011 scheint in Hamburg die Sonne länger als 4 Stunden.	??
• Am 6. April 2299 wird der Pegel der Elbe die 7m-Marke übersteigen.	??

keine Aussagen sind z.B.

• *Fragen:*	Liegt Hamburg nördlich von München?
• *Aufforderungen, Befehle:*	Fahr nach Kiel!
• *Inhärent widersprüchliche Sätze:*	Dieser Satz ist falsch.

Aussagen, Fragen & Befehle

- Die Untersuchung von Fragen kann systematisch auf die Untersuchung von Aussagen zurückgeführt werden.
  - Liegt Hamburg nördlich von München?
    besitzt zwei korrespondierende Antworten:
    - ✶ Ja! ≈ Hamburg liegt nördlich von München.
    - ✶ Nein! ≈ Hamburg liegt nicht nördlich von München.
- Die Untersuchung von Befehlen kann auf die Untersuchung von Aussagen zurückgeführt werden.
  - Begib Dich nach Kiel!
    ≈ Verändere Deinen Aufenthaltsort derart, dass Du in Kiel bist.

  ➔ Voraussetzung der Befehlsausführung:

  „Du bist nicht in Kiel" ist **wahr**.

  ➔ Der Befehl ist erfolgreich ausgeführt:

  „Du bist in Kiel" ist **wahr**.

Inhärent widersprüchliche Sätze

- sind Sätze, denen man überhaupt keinen Wahrheitswert zuordnen kann, ohne in Probleme zu kommen.

(1) Dieser Satz ist falsch.

  - Die Annahme, dass (1) wahr ist, führt automatisch dazu, dass er auch falsch ist.
  - Die Annahme, dass (1) falsch ist, führt automatisch dazu, dass er auch wahr ist.

- Eine Wahrheitswertzuordnung macht also keinen Sinn.

Kontradiktionen

- sind Sätze, die auf jeden Fall falsch sind.

(2) Es regnet und es regnet nicht.

  - (2) ist ganz einfach falsch und aus der Annahme, dass (2) falsch ist, ergibt sich kein weiteres Problem.
  - Der durch (2) ausgedrückte Widerspruch ist einer, mit dem die Logik umgehen kann. Die Logik ist gewissermaßen dafür da, solche Widersprüche aufzudecken.

- Mit dem durch (1) ausgedrückten Widerspruch kann die Logik nicht umgehen, und deshalb werden solche Sätze von der Betrachtung in der Logik ausgeschlossen.

Die symbolische Logik

Formeln der symbolischen Logik sind Zeichenketten,

- die aus den Symbolen eines speziellen Alphabets zusammengesetzt sind,
- und bestimmte Bedingungen erfüllen (Wohlgeformtheitsbedingungen).

Objektsprache

- die Menge der Zeichenketten, über die wir sprechen
- auf Folien und in pdf-Dateien in dieser Schrift dargestellt
- Die Zeichen F, G, H, … 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

Form-Bezogenheit der Logik

Formeln der symbolischen Logik

- machen ‚logische Muster' in der Sprache explizit

Logische Muster, die die Aussagenlogik behandelt

- Wiederholung einer (Teil-)Aussage
- Gewisse Verwendungen

von	symbolisiert durch
‚nicht'	$\neg$
‚und'	$\wedge$
‚oder'	$\vee$
‚wenn …, dann…'	$\Rightarrow$
‚genau dann …, wenn …'	$\Leftrightarrow$

Logische Muster, die die Prädikatenlogik behandelt

- Wiederholung von Namen, Nomen, Verben, Adjektiven
- Gewisse Verwendungen von ‚ein', ‚einige', ‚jeder', ‚alle', ‚kein' („Quantoren")

Repräsentation von Aussagen durch (aussagenlogische) Formeln (1)

Übersetzungsschlüssel

- A : 734 ist durch 3 teilbar
- B : Die Quersumme von 74 ist durch 3 teilbar.

Formel	Aussage	
$\neg A$	734 ist nicht durch 3 teilbar.	**wahr**
$\neg B$	Die Quersumme von 74 ist nicht durch 3 teilbar.	**wahr**
$(A \land B)$	734 ist durch 3 teilbar und die Quersumme von 74 ist durch 3 teilbar.	**falsch**
$(A \lor B)$	734 ist durch 3 teilbar oder die Quersumme von 74 ist durch 3 teilbar.	**falsch**
$(A \Rightarrow B)$	Wenn 734 durch 3 teilbar ist, dann ist die Quersumme von 74 durch 3 teilbar.	**wahr**
$(A \Leftrightarrow B)$	734 ist genau dann durch 3 teilbar, wenn die Quersumme von 74 durch 3 teilbar ist.	**wahr**
$(B \land A)$	Die Quersumme von 74 ist durch 3 teilbar und 734 ist durch 3 teilbar.	**falsch**
$(A \land A)$	734 ist durch 3 teilbar und 734 ist durch 3 teilbar.	**falsch**
$(A \lor \neg A)$	734 ist durch 3 teilbar oder 734 ist nicht durch 3 teilbar.	**wahr**

Repräsentation von Aussagen durch Formeln (2)

Übersetzungsschlüssel

- A : 44 ist durch 11 teilbar
- B : Die Quersumme von 44 ist durch 11 teilbar.

Formel	**Aussage**	
$\neg A$	44 ist nicht durch 11 teilbar.	**falsch**
$\neg B$	Die Quersumme von 44 ist nicht durch 11 teilbar.	**wahr**
$(A \wedge B)$	44 ist durch 11 teilbar und die Quersumme von 44 ist durch 11 teilbar.	**falsch**
$(A \vee B)$	44 ist durch 11 teilbar oder die Quersumme von 44 ist durch 11 teilbar.	**wahr**
$(A \Rightarrow B)$	Wenn 44 durch 11 teilbar ist, dann ist die Quersumme von 44 durch 11 teilbar.	**falsch**
$(A \Leftrightarrow B)$	44 ist genau dann durch 11 teilbar, wenn die Quersumme von 44 durch 11 teilbar ist.	**falsch**
$(B \wedge A)$	Die Quersumme von 44 ist durch 11 teilbar und 44 ist durch 11 teilbar.	**falsch**
$(A \wedge A)$	44 ist durch 11 teilbar und 44 ist durch 11 teilbar.	**wahr**
$(A \vee \neg A)$	44 ist durch 11 teilbar oder 44 ist nicht durch 11 teilbar.	**wahr**

Repräsentation von Aussagen durch Formeln (3)

Übersetzungsschlüssel

- A : Abianer sagen immer die Wahrheit.
- B : Bebianer lügen immer.

Formel	Aussage	
$\neg A$	Abianer sagen nicht immer die Wahrheit.	
$\neg B$	Bebianer lügen nicht immer.	
$(A \wedge B)$	Abianer sagen immer die Wahrheit und Bebianer lügen immer.	
$(A \vee B)$	Abianer sagen immer die Wahrheit oder Bebianer lügen immer.	
$(A \Rightarrow B)$	Wenn Abianer immer die Wahrheit sagen, dann lügen Bebianer immer.	
$(A \Leftrightarrow B)$	Abianer sagen genau dann immer die Wahrheit, wenn Bebianer immer lügen.	
$(B \wedge A)$	Bebianer lügen immer und Abianer sagen immer die Wahrheit.	
$(A \wedge A)$	Abianer sagen immer die Wahrheit und Abianer sagen immer die Wahrheit.	
$(A \vee \neg A)$	Abianer sagen immer die Wahrheit oder Abianer sagen nicht immer die Wahrheit.	wahr

Verschiedene Objektsprachen

- $\mathcal{L}_{\mathrm{AL}}$: Die Objektsprache der Aussagenlogik
  - Beispiele: A, $\neg A$, $(A \vee C)$, $(A \Rightarrow B)$, $(A \Leftrightarrow (A \vee C))$
  - Die kleinsten Einheiten (atomaren Formeln) sind Aussagensymbole (A, B, C)
  - Aus ihnen werden mit Junktoren ($\vee$, $\wedge$, $\neg$, $\Rightarrow$, $\Leftrightarrow$) komplexe Formeln gebildet.
- $\mathcal{L}_{\mathrm{PL}}$: Die Objektsprache der Prädikatenlogik
  - Anreicherung der Aussagenlogik
  - Atomare Formeln haben eine interne Struktur: $P(a)$, $P(x)$, $R(a, b)$
    Sie werden aus Prädikatssymbolen (P, R) und Termen (a, b, x) gebildet
  - Quantoren bilden zusätzliche Formeln: $\forall x\ (R(x, a) \Rightarrow P(x))$, $\exists x\ (P(x) \wedge Q(x))$

Gemeinsamkeiten in der Metasprache

- Viele Fachbegriffe der Logik werden an Hand der Aussagenlogik eingeführt.
- Sie werden dann an die reichere Struktur der Prädikatenlogik angepasst in entsprechender Art angewendet.

Definition 2.1

Das *Alphabet* der (symbolischen) Aussagenlogik besteht aus

- einer abzählbaren Menge von *Aussagensymbolen* (‘Elementaraussagen’, ‘atomaren Aussagen’): A, B, C, D, A', B', C', D', A", B", C", D", …
  Die Menge der Aussagensymbole bezeichnen wir mit $\mathcal{A}s_{\mathrm{AL}}$
- den Junktoren: $\neg$, $\wedge$, $\vee$, $\Rightarrow$, $\Leftrightarrow$
- und Klammern:), (

Die Junktoren und Klammern gehören alle nicht zu den Aussagensymbolen.

Die Aussagensymbole stehen stellvertretend für einfache Aussagen, z.B.

- 734 ist durch 3 teilbar.
- Abianer sagen immer die Wahrheit.

Die Junktoren	Symbol	offizieller Name	deutsche Übersetzung
einstellig	$\neg$	Negation	*nicht …*
zweistellig	$\wedge$	Konjunktion	… *und* …
zweistellig	$\vee$	Disjunktion	… *oder* …
zweistellig	$\Rightarrow$	Implikation	*wenn …, dann …*
zweistellig	$\Leftrightarrow$	Biimplikation	… *genau dann, wenn* …

Zum Selbststudium

Aussagensymbole

- Für die Menge der Aussagensymbole ist eigentlich nur folgendes wichtig:
  - Es sind höchstens abzählbar unendlich viele, d.h. nicht 'mehr' als natürliche Zahlen
  - Sie sind klar unterscheidbar
  - Sie sind auch von den Junktoren und Klammern klar unterscheidbar.
- Ob diese Symbole eine interne Struktur haben, oder nicht, interessiert die Aussagenlogik nicht, oder anders ausgedrückt: die interne Struktur der atomaren Aussagen wird in der Aussagenlogik nicht berücksichtigt. Sollten sie aber eine interne Struktur haben, dann sollten die Junktoren nicht in ihnen vorkommen.
  - Wir könnten also auch folgende Zeichen als Aussagensymbole verwenden: ☸, ♈, ♉, ♊, ♋, ♌, ♍, ♎, ♏, ♐, 🕒, 🕔, 🕛, ⓪, ⑦, ☐, ⇩, ⌑, ❼, 🖫, 🕯, ❍, ⮝, 🡒, ⌘, 🖭, ⬋, 🡇, 🡖, ✩, ♒, ❍, 🡑, …
  - oder 'Wörter' wie: BenIstSohnVonHans, HansIstVaterVonPeter, AbianerSagenDieWahrheit, …
  - A, B, C, D, A', B', C', D', A'', B'', C'', D'', … haben den Vorteil, dass sie nicht viel Platz brauchen und gut auszusprechen sind.

Formeln der symbolischen Aussagenlogik

- $F, G, H, \ldots, F_1, F_2, \ldots$ verwenden wir als Variablen, die Formeln als Wert haben.
- $A_1, A_2, \ldots, A_i, A_j, \ldots$ sind Variablen, die Aussagensymbole als Wert haben.

Definition 2.2

Es sei $\mathcal{As}_{AL}$ eine endliche oder abzählbar unendliche Menge von Aussagesymbolen, so dass $\{\neg, \wedge, \vee, \Rightarrow, \Leftrightarrow,), (\} \cap \mathcal{As}_{AL} = \emptyset$.

Die *wohlgeformten Ausdrücke der Aussagenlogik (Formeln)* sind induktiv definiert:

1. Alle Aussagensymbole aus $\mathcal{As}_{AL}$ sind (atomare) Formeln. Beispiele: $A, B, C, \ldots$
2. Falls F und G Formeln sind, so sind $(F \wedge G)$, $(F \vee G)$, $(F \Rightarrow G)$ und $(F \Leftrightarrow G)$ (*komplexe*) Formeln.
   Beispiele: $(A \wedge A)$, $(A \vee C)$, $(A \Rightarrow B)$, $(A \Leftrightarrow (A \vee C))$, $((A \Rightarrow B) \vee C)$, …
3. Falls F eine Formel ist, so ist auch $\neg F$ eine (*komplexe*) Formel.
   Beispiele: $\neg A$, $\neg(A \wedge A)$, $\neg((A \Rightarrow B) \vee C)$, $\neg\neg A$, $\neg\neg\neg A$, $(\neg A \Rightarrow B)$
4. Es gibt keine anderen Formeln, als die, die durch endliche Anwendung der Schritte 1–3 erzeugt werden.

- Die Menge aller aussagenlogischen Formeln bezeichnen wir als $\mathcal{L}_{AL}$.
- Gleichheit von Formeln verstehen wir immer als buchstäbliche Übereinstimmung
  - $(A \Rightarrow B) = (A \Rightarrow B)$, aber $(A \wedge A) \neq A$

Zur Sprache der Aussagenlogik

Das Alphabet (vgl. Def. 2.1)

$\Sigma_{\text{AL}} = \mathcal{A}s_{\text{AL}} \cup \{\neg, \wedge, \vee, \Rightarrow, \Leftrightarrow,), (\ \}$

Die Sprache der Aussagenlogik

- Die induktive Definition der wohlgeformten Formeln (Def. 2.2) stellt eine Methode dar, eine Sprache $\mathcal{L}_{\text{AL}} \subseteq \Sigma_{\text{AL}}{}^{*}$ zu spezifizieren.
- Eine andere Methode für die Spezifikation von $\mathcal{L}_{\text{AL}}$ besteht in der Verwendung von formalen Grammatiken.

Grammatik für Formeln der Aussagenlogik

Vokabular der terminalen Symbole

(Symbole, die in der Formel vorkommen):

$\Sigma_{AL} = \mathcal{A}s_{AL} \cup \{\neg, \wedge, \vee, \Rightarrow, \Leftrightarrow,), (\,\}$

nicht-terminales Symbol

(Symbole, die für die Erzeugung der Formeln benötigt werden, aber nicht in der Formel vorkommen):

$N = \{S\}$, wobei S $\mathcal{A}s_{AL}$

Startsymbol: S

Anmerkung:

Diese Grammatik verwendet im Gegensatz zu einer KGF, ein abzählbares Alphabet der atomaren Formeln und eine abzählbare Menge von Regeln des Typs $S \rightarrow A_i$ ($A_i \in \mathcal{A}s_{AL}$). Ist $\mathcal{A}s_{AL}$ endlich, dann ist dies tatsächlich eine KFG.

Regeln (Produktionen):

$P = \{\, S \rightarrow A_i \mid A_i \in \mathcal{A}s_{AL}\}$

$\cup \{\, S \rightarrow (S \circ S) \mid \circ \in \{\wedge, \vee, \Rightarrow, \Leftrightarrow\}\,\}$

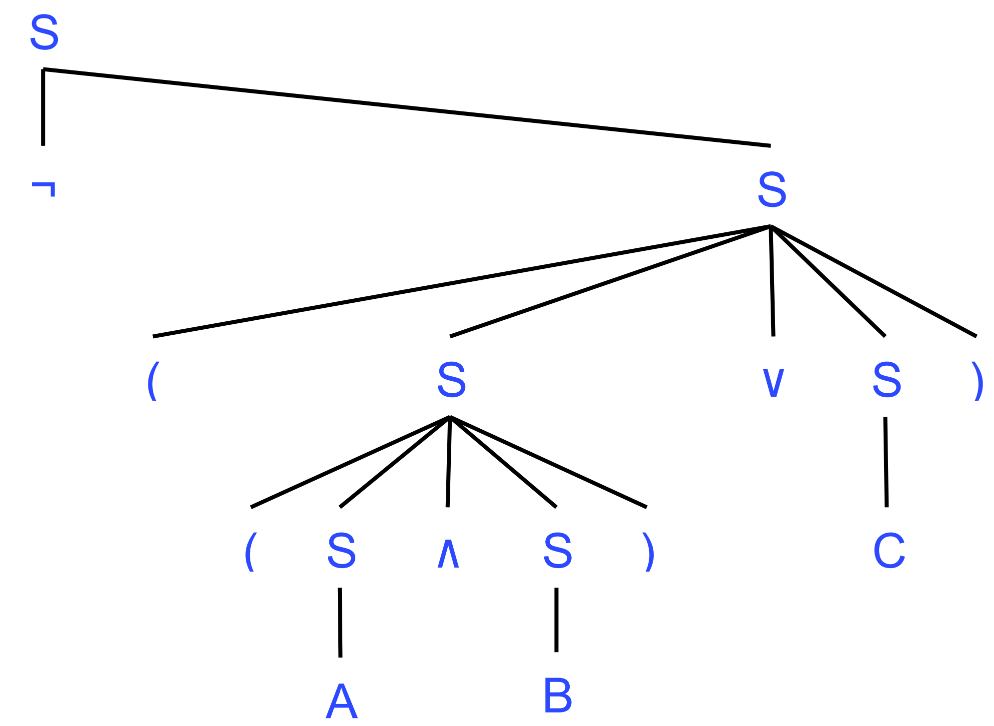
$\cup \{\, S \rightarrow \neg S \,\}$

Ableitung: $S \rightarrow \neg S \rightarrow \neg(S \vee S) \rightarrow \neg((S \wedge S) \vee S) \rightarrow \neg((A \wedge S) \vee S)$

$\rightarrow \neg((A \wedge B) \vee S) \rightarrow \neg((A \wedge B) \vee C)$

Beispiel: Ableitung der Formel $\neg((A \wedge B) \vee C)$

$S \rightarrow \neg S \rightarrow \neg(S \vee S) \rightarrow \neg((S \wedge S) \vee S) \rightarrow \neg((A \wedge S) \vee S) \rightarrow \neg((A \wedge B) \vee S) \rightarrow \neg((A \wedge B) \vee C)$



Ableitungsbaum

Regeln (Produktionen):

$P = \{\ S \rightarrow A$,

$S \rightarrow B$,

$S \rightarrow C$,

$S \rightarrow D$,

$\ldots$,

$S \rightarrow (S \wedge S)$,

$S \rightarrow (S \vee S)$,

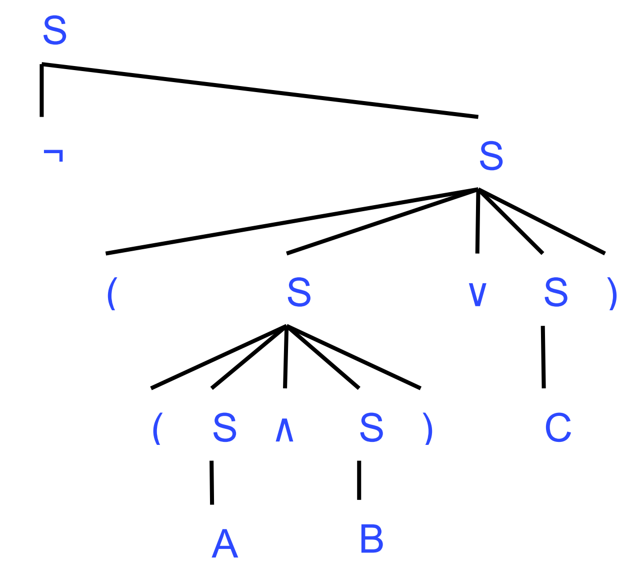
$S \rightarrow (S \Rightarrow S)$,

$S \rightarrow (S \Leftrightarrow S)$,

$S \rightarrow \neg S\ \}$

Zum Selbststudium: Ableitungsbäume

Den Ableitungsweg einer Formel können wir auch in einer (hierarchischen) Baumstruktur (mit Wurzel) darstellen. (vgl. Biggs, Kapitel 8.5 und 9.1)

Terminalsymbole	Blatt-Knoten des Baumes Aussagesymbole, Junktoren, Klammern	$A, \neg, \wedge, \ldots$
Nichtterminalsymbole	innere Knoten des Baums	S
(komplexe) Formeln $\neg((A \wedge B) \vee C)$	Sequenz der Blätter eines Baums ● Teilformeln entsprechen Teilbäumen. ● Die Reihenfolge der Teilformeln wird beibehalten.	

- Die Bäume werden mit der Wurzel oben und den Blättern unten gezeichnet.
- Kommt eine Teilformel mehrfach in der Formel vor, dann kommt der entsprechende Teilbaum auch mehrfach (als Kopie) vor.

Die rigide Sprachdefinition für die Aussagenlogik

z.B. Punkt 4 der Sprachdefinition

- Es gibt keine anderen Formeln, als die, die durch endliche Anwendung der Schritte 1–3 erzeugt werden.

ist erforderlich

- wenn man Aussagen über alle Formeln der Aussagenlogik machen und beweisen will.

Aufgrund der gegebenen Sprachdefinition wissen wir,

- dass jede Formel tatsächlich durch die Aufbauregeln erzeugt wurde,
- dass Texte, die nicht diesen Aufbauregeln genügen, keine Formeln sind,
- dass auch Knut keine Formel war,
- dass es für jede Formel genau einen Ableitungsbaum gibt, der sie erzeugt (Eindeutigkeit der Grammatik).

Die formale Sprachdefinition unterstützt

- induktive Beweise (‚strukturelle Induktion', an den Aufbauregeln orientiert),
- rekursive Funktionsdefinitionen (davon im Verlauf der Vorlesung mehr)

Zum Selbststudium: Dialekte der Logik

Die symbolische Darstellung der Logik weist viele Varianten auf Variationen entstehen durch

- Klammerkonventionen
- Menge der Aussagensymbole ($\mathcal{As}_{\mathrm{AL}}$)
- Junktorenbasis (also die Menge der Junktoren)
  - z.B. könnte man darauf verzichten, die Implikation als Junktor einzuführen, oder man könnte weitere Junktoren (z.B. exklusives oder) einführen
- Symbole für die einzelnen Junktoren
  - z.B. wählt Schöning zur Darstellung der Junktoren an Stelle von $\Rightarrow$ das Symbol $\rightarrow$ und an Stelle von $\Leftrightarrow$ das Symbol $\leftrightarrow$. [Diese Junktoren sind bei Schöning nicht Teil des Basisvokabulars, sie sind als „Abkürzungen“ anzusehen.
  - z.B. wählt Salmon zur Darstellung der Junktoren an Stelle von $\Rightarrow$ das Symbol $\supset$ und an Stelle von $\Leftrightarrow$ das Symbol $\equiv$. Für die Relation, die wir später durch $\equiv$ symbolisieren, führt Salmon dagegen kein Symbol ein.
  - Entsprechend trifft man manchmal die Balkennotation $\overline{\mathsf{A}}$ anstelle der Negation $\neg\mathsf{A}$ an. Auch dieses Symbol werden wir später mit einem etwas anderen Sinn verwenden.

- Um zu zeigen, dass und in welchem Sinn die verschiedenen Logik-Dialekte gleichwertig sind, benötigt man das formale Instrumentarium, das wir in dieser Vorlesung vorstellen werden.
- Die Beschränkung auf einen einheitlichen Logik-Dialekt erleichtert die Vorstellung dieser Prinzipien, die dann auf weitere Dialekte übertragbar sind.
- Verwenden Sie bitte bei der Bearbeitung der Übungsaufgaben den Logik-Dialekt der Vorlesung. Dieser Dialekt stimmt (weitgehend) mit dem Dialekt in den Büchern von Schöning und Spies überein.
- Wenn sie weitere Logikbücher lesen, dann achten Sie auf die Dialektvariationen.

Zu abzählbar unendlichen Mengen (von Aussagensymbolen, Formeln)

- vgl. Biggs Kapitel 2.5
- Dass wir eine abzählbar unendliche Menge von Aussagesymbolen zulassen,
  - wird nie wirklich stören, da in endlichen Formelmengen sowieso nur endlich viele Aussagensymbole verwendet werden, und
  - hat nur einen einzigen Grund: der unbeschränkte Zeichenvorrat ist sehr nützlich, wenn man zeigen will, dass man gewisse Aspekte der Prädikatenlogik auch in Aussagenlogik behandeln kann. Bis wir an dieser Stelle der Vorlesung sind, kann ignoriert werden, dass die Menge der Aussagensymbole unendlich sein kann, wobei man aber immer davon ausgehen kann, dass noch genügend unverbrauchte Symbole da sind, wenn man mal eins benötigt.

Formeln und andere Zeichenketten

Formeln	**Zeichenketten, die keine Formeln sind**
A	$\neg$
$\neg B$	$) A \wedge B ($
$(A \wedge B)$	$(\wedge A B)$
$((A \wedge B) \vee (A \wedge \neg C))$	$(A \wedge B \vee A \wedge \neg C)$
$((A \wedge B) \Rightarrow C)$	$\neg(A)$

Eine Vereinfachung: Das äußerste Klammerpaar kann weggelassen werden

Formeln	**Zeichenketten, die keine Formeln sind**
$A \wedge B$ (steht für $(A \wedge B)$)	$(A \wedge B \vee A \wedge \neg C)$
$A \Leftrightarrow (C \vee B)$ (steht für $(A \Leftrightarrow (C \vee B))$)	$((A \wedge B))$

Zeichenketten, die keine Formeln sind, gehören nicht zu $\mathcal{L}_{AL}$.

Aufgabe zum Selbststudium: Beweisen Sie für die beiden "grau unterlegten Zeichenketten", dass die in der Tabelle vorgenommene Zuordnung korrekt ist.

Zum Selbststudium

Voraussetzung: Definitionen 2.1, 2.2

Behauptung: $((A \wedge B) \Rightarrow C)$ ist eine Formel.

Beweis

Man muss zeigen, dass diese Zeichenkette durch die Schritte 1–3 der Definition von ‚Formel' konstruierbar ist. In aller Ausführlichkeit sieht das dann so aus:

A	(atomare) Formel nach 1, da Aussagensymbol.
B	(atomare) Formel nach 1, da Aussagensymbol.
$(A \wedge B)$	Formel nach 2, wobei wir $F := A$ und $G := B$ setzen. Dass A und B Formeln sind, haben wir schon gezeigt.
C	(atomare) Formel nach 1, da Aussagensymbol.
$((A \wedge B) \Rightarrow C)$	Formel nach 2, wobei wir $F := (A \wedge B)$ und $G := C$ setzen. Dass $(A \wedge B)$ und C Formeln sind, haben wir schon gezeigt.

Zum Selbststudium

Voraussetzung: Definitionen 2.1, 2.2

Behauptung: $H := (A \wedge B \vee A \wedge \neg C)$ ist keine Formel.

Beweis

Wir nehmen an, dass H eine Formel ist, und führen dies zum Widerspruch.
Wenn H eine Formel ist, dann muss der letzte Konstruktionsschritt Nr. 2 gewesen sein, denn das erste Zeichen von H ist eine Klammer (.

Es gibt drei Möglichkeiten, H an einem Junktor $\circ$ zu zerlegen mit $H = (F \circ G)$:

F	G	$\circ$	
A	$B \vee A \wedge \neg C$	$\wedge$	G ist keine Formel, denn G ist kein Aussagensymbol und beginnt weder mit (noch mit $\neg$. Also kann auch G nicht durch 1–3 konstruiert sein.
$A \wedge B$	$A \wedge \neg C$	$\vee$	F und G sind keine Formeln. (Begründung wie vorher.)
$A \wedge B \vee A$	$\neg C$	$\wedge$	F ist keine Formel. (Begründung wie vorher.)

Wichtig: Die Klammerersparnisregel dürfen wir hier nicht verwenden, da sie nur auf das äußerste Klammerpaar von H anwendbar ist.
Es gibt also keine Zerlegung von H, die den Aufbauregeln für Formeln gehorcht.

Teilformeln und Hauptoperatoren

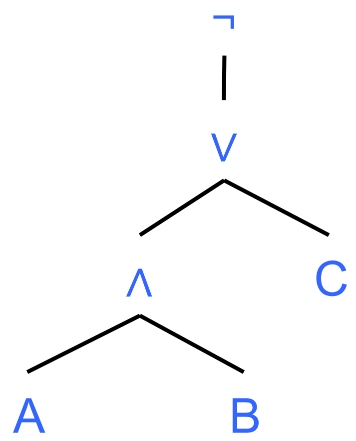
Definitionen 2.3

- Eine Formel, die beim Aufbau einer Formel F verwendet wird, heißt *Teilformel* von F. Außerdem werden wir auch F als (uneigentliche) *Teilformel* von F bezeichnen.
  - **Beispiel:** $F := \neg((A \land B) \lor \neg C)$
  - Teilformeln: $\neg((A \land B) \lor \neg C)$, $((A \land B) \lor \neg C)$, $(A \land B)$, A, B, $\neg C$, C
  - keine Teilformeln: D, $B) \lor \neg C)$, $(C \Rightarrow (A \land B))$, $(B \land A)$
- Der Junktor, der im letzten Konstruktionsschritt einer komplexen Formel F verwendet wurde, heißt *Hauptoperator* von F.
- Komplexe Formeln benennen wir auch nach ihrem Hauptoperator

Formel	Hauptoperator	offizieller Name
$\neg C$	$\neg$	Negation
$(A \land \neg C)$	$\land$	Konjunktion
$((A \land B) \lor (A \land \neg C))$	$\lor$	Disjunktion
$((A \land B) \Rightarrow (A \lor C))$	$\Rightarrow$	Implikation
$(A \Leftrightarrow B)$	$\Leftrightarrow$	Biimplikation
C	< kein Hauptoperator >	Aussagensymbol

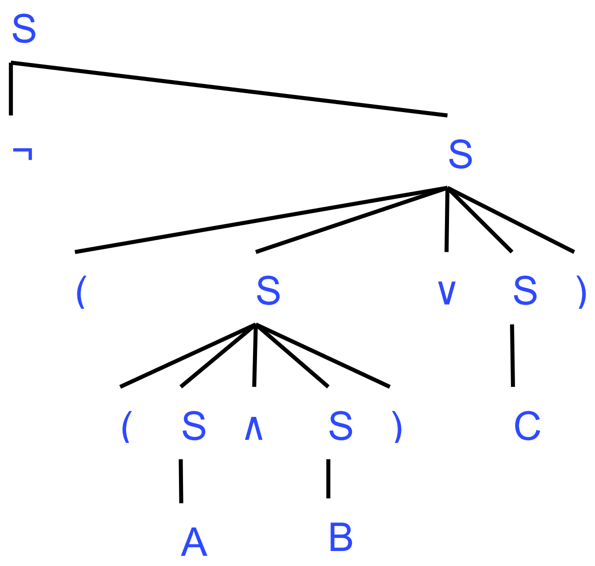
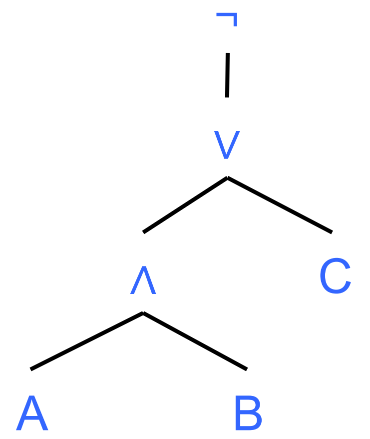
Strukturbäume

Obwohl die Formeln der Logik (lineare) Zeichenketten sind, können wir sie auch in einer (hierarchischen) Baumstruktur (mit Wurzel) darstellen. (vgl. Biggs, Kapitel 8.5 und 9.1)

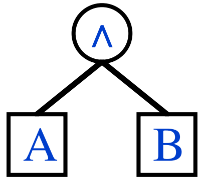
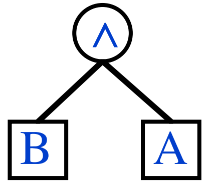
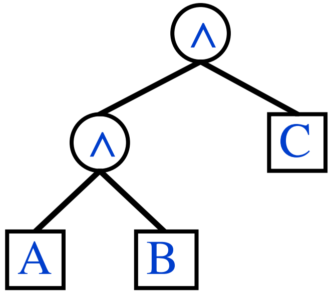
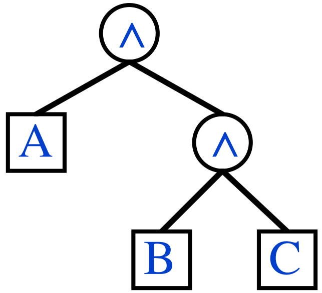
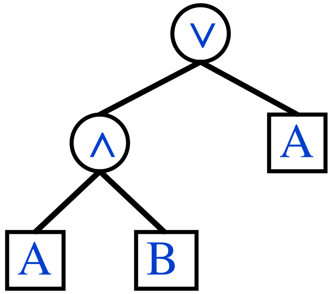
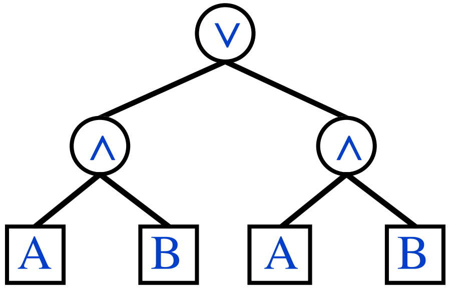
Aussagesymbole	Blatt-Knoten des Baumes	A, B, ...
Junktoren	innere Knoten des Baums	$\neg$, $\wedge$, $\vee$, ...
komplexe Formeln $\neg((A \wedge B) \vee C)$	Bäume ● Hauptoperator markiert die Wurzel. ● Teilformeln entsprechen Teilbäumen. ● Die Reihenfolge der Teilformeln wird beibehalten.	

- Die Bäume werden mit der Wurzel oben und den Blättern unten gezeichnet.
- $\neg$ hat einen Nachfolger, die anderen Junktor-Knoten haben zwei Nachfolger.
- Klammern kommen in den Bäumen nicht vor.
- Kommt eine Teilformel mehrfach in der Formel vor, dann kommt der entsprechende Teilbaum auch mehrfach (als Kopie) vor.

Ableitungsbaum vs. Strukturbaum

<table>
<tr><th>Ableitungsbaum</th><th>Strukturbaum</th></tr>
<tr><td>
</td><td>
</td></tr>
<tr><td>• syntaktische Struktur
• grammatische Struktur</td><td>• logische Struktur
• semantische Struktur</td></tr>
<tr><td>• Alle Terminalsymbole werden gleich behandelt (Blattknoten)</td><td>• Junktoren, und Aussagensymbole werden unterschiedlich behandelt, Hilfssymbole verschwinden</td></tr>
</table>

Strukturbäume: Beispiele

Reihenfolge	$(A \wedge B)$	$(B \wedge A)$
		
Klammerung	$((A \wedge B) \wedge C)$	$(A \wedge (B \wedge C))$
		
doppelte Teilformeln	$((A \wedge B) \vee A)$	$((A \wedge B) \vee (A \wedge B))$
		

Prinzip der strukturellen Induktion

Der Aufbau von komplexen Formeln aus einfacheren Formeln dient als Grundlage, um Eigenschaften von Formeln nachzuweisen:

Um zu beweisen, dass eine Behauptung $\mathcal{B}(F)$ für jede Formel $F \in \mathcal{L}_{AL}$ gilt, genügt es, folgende Schritte durchzuführen:

Induktionsanfang *(induction basis)*: Man zeigt, dass $\mathcal{B}(F)$ für jede atomare Formel F gilt, also für die Aussagensymbole ($\mathcal{As}_{AL}$) A, B, C, D,…

Induktionsannahme *(induction hypothesis)*: Man nimmt an, dass F und G Formeln sind, für die $\mathcal{B}(F)$ und $\mathcal{B}(G)$ gelten.

Induktionsschritt: Man zeigt, dass dann auch $\mathcal{B}(\neg F)$, $\mathcal{B}((F \wedge G))$, $\mathcal{B}((F \vee G))$, $\mathcal{B}((F \Rightarrow G))$ und $\mathcal{B}((F \Leftrightarrow G))$ gelten.

➔ Die Bedingung 4 der Definition von Formeln legitimiert die strukturelle Induktion.
4. Es gibt keine anderen Formeln, als die, die durch endliche Anwendung der Schritte 1–3 erzeugt werden.

➔ Die strukturelle Induktion ist eine (beweisbare) Verallgemeinerung der vollständigen Induktion (vgl. Biggs, Kapitel 1.4)

Beispiel für einen induktiven Beweis

Satz (ohne Nummer) Voraussetzung: Definitionen 2.1, 2.2, Satz 2.10

Behauptung: Jede aussagenlogische Formel hat endlich viele Aussagensymbole als Teilformeln.

Beweis

Induktionsanfang

Nach Def. 2.2 hat jede atomare Formel genau ein Aussagensymbol als Teilformel, nämlich sich selbst.

Induktionsannahme

Es seien F und G Formeln mit endlich vielen Aussagensymbolen. Die Anzahl der Aussagensymbole von F sei n, die Anzahl der Aussagensymbole von G sei m.

Induktionsschritt

Da nach Def. 2.1 $\neg$ kein Aussagensymbol ist, hat die Formel $\neg F$ genauso viele Aussagensymbole wie F, nach Induktionsannahme also n.

Da nach Def. 2.1 $\wedge$, $\vee$, $\Rightarrow$, $\Leftrightarrow$, (,) keine Aussagensymbole sind, haben die Formeln $(F \wedge G)$, $(F \vee G)$, $(F \Rightarrow G)$ und $(F \Leftrightarrow G)$ höchstens so viele Aussagensymbole wie F und G zusammen, nach Induktionsannahme also höchstens $m + n$ Aussagensymbole.

Resümee: Gemäß dem Prinzip der strukturellen Induktion hat jede Formel der Aussagenlogik endlich viele Aussagensymbole als Teilformeln.

Struktur von Beweisen nach dem Prinzip der strukturellen Induktion

Voraussetzung: Def. 2.1, 2.2, Satz 2.10, z.B. Definition der Eigenschaft $\mathcal{B}$

Behauptung: Jede Formel der Aussagenlogik hat die Eigenschaft $\mathcal{B}$.

Beweis

Induktionsanfang

➔ Teilbeweis für: Jede atomare Formel hat die Eigenschaft $\mathcal{B}$.
Dieser Teilbeweis greift auf die Voraussetzungen zurück.

Induktionsannahme

Es seien F und G Formel, die die Eigenschaft $\mathcal{B}$ haben.

➔ Kein Beweis erforderlich, keine Einschränkung erlaubt.

➔ Ergänzende Definitionen sind hier möglich.

Induktionsschritt

➔ Teilbeweise für: Die Formeln $\neg F$, $(F \wedge G)$, $(F \vee G)$, $(F \Rightarrow G)$ und $(F \Leftrightarrow G)$ haben die Eigenschaft $\mathcal{B}$.
Diese Teilbeweise greifen auf Voraussetzungen und Induktionsannahme zurück.

Resümee

Gemäß dem Prinzip der strukturellen Induktion hat jede Formel der Aussagenlogik die Eigenschaft $\mathcal{B}$.

Prinzip der strukturellen Rekursion

Der Aufbau von komplexen Formeln aus einfacheren Formeln dient auch als Grundlage, um Funktionen über die Formelmenge zu definieren.

Es sei **D** eine (beliebige) Menge. Um eine Funktion f, die Formeln in **D** abbildet, zu definieren, genügt es, folgende (einfache) Funktionen festzulegen:

1. eine Abbildung $f_{\mathcal{As}}$ der Aussagensymbole auf Elemente von **D**:

   $f_{\mathcal{As}}: \mathcal{As}_{\mathrm{AL}} \rightarrow \mathbf{D}$
2. eine Abbildung $f_\neg$ von **D** nach **D**: $f_\neg: \mathbf{D} \rightarrow \mathbf{D}$
3. vier Abbildungen von Paaren von Elementen von **D** auf Elemente von **D**, die den Junktoren zugeordnet werden: $f_\wedge, f_\vee, f_\Rightarrow, f_\Leftrightarrow: \mathbf{D} \times \mathbf{D} \rightarrow \mathbf{D}$

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

Rekursionsbasis: Für jedes Aussagensymbol $\mathsf{A} \in \mathcal{As}_{\mathrm{AL}}$ ist $f(\mathsf{A}) = f_{\mathcal{As}}(\mathsf{A})$.

Rekursionsschritt: Für alle Formeln $\mathsf{F}, \mathsf{G} \in \mathcal{L}_{\mathrm{AL}}$ gilt:

$f(\neg\mathsf{F}) = f_\neg(f(\mathsf{F}))$, $\quad f((\mathsf{F} \wedge \mathsf{G})) = f_\wedge(f(\mathsf{F}), f(\mathsf{G}))$,

$f((\mathsf{F} \vee \mathsf{G})) = f_\vee(f(\mathsf{F}), f(\mathsf{G}))$, $\quad f((\mathsf{F} \Rightarrow \mathsf{G})) = f_\Rightarrow(f(\mathsf{F}), f(\mathsf{G}))$

$f((\mathsf{F} \Leftrightarrow \mathsf{G})) = f_\Leftrightarrow(f(\mathsf{F}), f(\mathsf{G}))$.

Zum Selbststudium: Funktionen

Vgl. Biggs Kapitel 2

Funktionen

- ordnen Objekten eines *Definitionsbereiches* (*Domäne*) (M_D)
- Objekte eines *Wertebereichs* (M_W) zu.
- Definitionsbereich und Wertebereich einer Funktion sind Mengen.
- Symbolisch stellen wir das wie folgt dar f: $M_D \rightarrow M_W$
- Die Art und Weise wie diese Zuordnung erfolgt, kann durch eine *Abbildungsvorschrift* beschrieben werden.
- Abbildungsvorschriften werden üblicherweise wie folgt notiert:
  $f(x) =$ … und hier kommt eine Spezifikation des Wertes …
  x ist hier eine Variable, die alle Elemente des Definitionsbereichs M_D als Wert annehmen kann, $f(x)$ liegt aber in M_W, dem Wertebereich der Funktion.

Beispiele

- Der Ausdruck ‚genetischer Vater' steht für eine Funktion, die allen Menschen einen Menschen männlichen Geschlechts zuordnet.
- ‚Gewicht in Gramm' steht für eine Funktion, die allen materiellen Objekten eine Zahl zuordnet.

Beispiel: Rekursive Definition vom Grad einer Formel

Definition 2.11 (Grad einer Formel)

Es seien folgende Funktionen gegeben:

$grad_{\mathcal{As}}: \mathcal{As}_{\mathrm{AL}} \to \mathbb{N}_0$, $grad_{\mathcal{As}}(A) = 0$ für alle $A \in \mathcal{As}_{\mathrm{AL}}$

$grad_{\neg}: \mathbb{N}_0 \to \mathbb{N}_0$, $grad_{\neg}(n) = n + 1$ für alle $n \in \mathbb{N}_0$

$grad_{\wedge}: \mathbb{N}_0 \times \mathbb{N}_0 \to \mathbb{N}_0$, $grad_{\wedge}(n, m) = n + m + 1$ für alle $n, m \in \mathbb{N}_0$

$grad_{\vee} = grad_{\Rightarrow} = grad_{\Leftrightarrow} = grad_{\wedge}$

Dann existiert genau eine Funktion $grad: \mathcal{L}_{\mathrm{AL}} \to \mathbb{N}_0$, so dass gilt:

Rekursionsbasis: Für $A \in \mathcal{As}_{\mathrm{AL}}$ ist $grad(A) = grad_{\mathcal{As}}(A) = 0$.

Rekursionsschritt: Für alle Formeln $F, G \in \mathcal{L}_{\mathrm{AL}}$ gilt:

$$grad(\neg F) = grad_{\neg}(grad(F)) = grad(F) + 1$$
$$grad((F \wedge G)) = grad_{\wedge}(grad(F), grad(G)) = grad(F) + grad(G) + 1$$
$$grad((F \vee G)) = grad_{\vee}(grad(F), grad(G)) = grad(F) + grad(G) + 1$$
$$grad((F \Rightarrow G)) = grad_{\Rightarrow}(grad(F), grad(G)) = grad(F) + grad(G) + 1$$
$$grad((F \Leftrightarrow G)) = grad_{\Leftrightarrow}(grad(F), grad(G)) = grad(F) + grad(G) + 1$$

Jede Formel hat einen eindeutig bestimmten Grad.

Beispiel: $\mathit{grad}(\neg((A \wedge B) \vee C))$

$\mathit{grad}(\neg((A \wedge B) \vee C))$

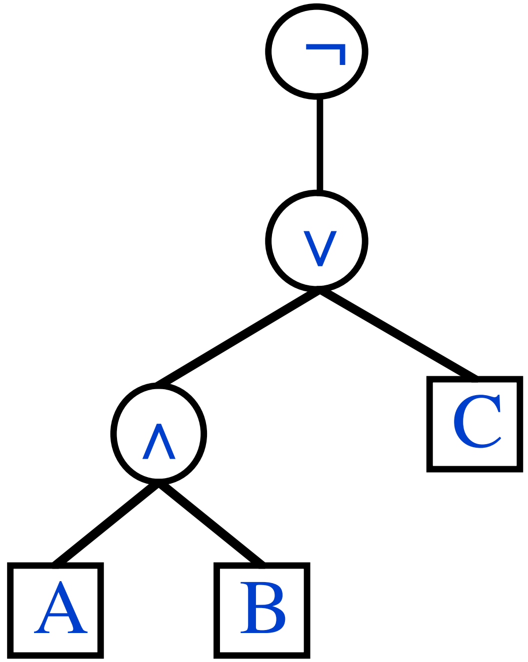
$= \mathit{grad}(((A \wedge B) \vee C)) + 1$

$= (\mathit{grad}((A \wedge B)) + \mathit{grad}(C) + 1) + 1$

$= ((\mathit{grad}(A) + \mathit{grad}(B) + 1) + \mathit{grad}(C) + 1) + 1$

$= ((0 + 0 + 1) + 0 + 1) + 1$

$= 3$



Beispiel: Rekursive Definition von der Tiefe einer Formel

Definition 2.12 (Tiefe einer Formel)

Es seien folgende Funktionen gegeben:

$tiefe_{\mathcal{As}}: \mathcal{As}_{AL} \to \mathbb{N}_0$, $\quad tiefe_{\mathcal{As}}(A) = 0$ $\quad$ für alle $A \in \mathcal{As}_{AL}$

$tiefe_{\neg}: \mathbb{N}_0 \to \mathbb{N}_0$, $\quad tiefe_{\neg}(n) = n + 1$ $\quad$ für alle $n \in \mathbb{N}_0$

$tiefe_{\wedge}: \mathbb{N}_0 \times \mathbb{N}_0 \to \mathbb{N}_0$, $\quad tiefe_{\wedge}(n, m) = max(n, m) + 1$ $\quad$ für alle $n, m \in \mathbb{N}_0$

$tiefe_{\vee} = tiefe_{\Rightarrow} = tiefe_{\Leftrightarrow} = tiefe_{\wedge}$

Dann existiert genau eine Funktion $tiefe: \mathcal{L}_{AL} \to \mathbb{N}_0$, so dass gilt:

Rekursionsbasis: Für $A \in \mathcal{As}_{AL}$ ist $tiefe(A) = tiefe_{\mathcal{As}}(A) = 0$.

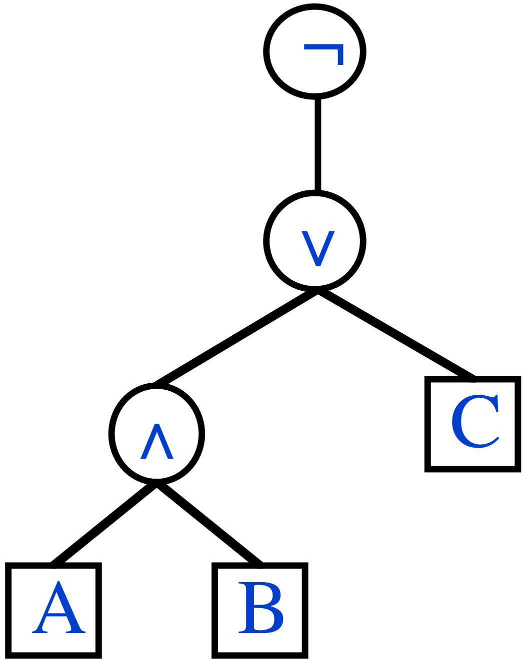
Rekursionsschritt: Für alle Formeln $F, G \in \mathcal{L}_{AL}$ gilt:

$$
\begin{aligned}
tiefe(\neg F) &= tiefe_{\neg}(tiefe(F)) &&= tiefe(F) + 1\\
tiefe((F \wedge G)) &= tiefe_{\wedge}(tiefe(F), tiefe(G)) &&= max(tiefe(F), tiefe(G)) + 1\\
tiefe((F \vee G)) &= tiefe_{\vee}(tiefe(F), tiefe(G)) &&= max(tiefe(F), tiefe(G)) + 1\\
tiefe((F \Rightarrow G)) &= tiefe_{\Rightarrow}(tiefe(F), tiefe(G)) &&= max(tiefe(F), tiefe(G)) + 1\\
tiefe((F \Leftrightarrow G)) &= tiefe_{\Leftrightarrow}(tiefe(F), tiefe(G)) &&= max(tiefe(F), tiefe(G)) + 1
\end{aligned}
$$

Jede Formel hat eine eindeutig bestimmte Tiefe.

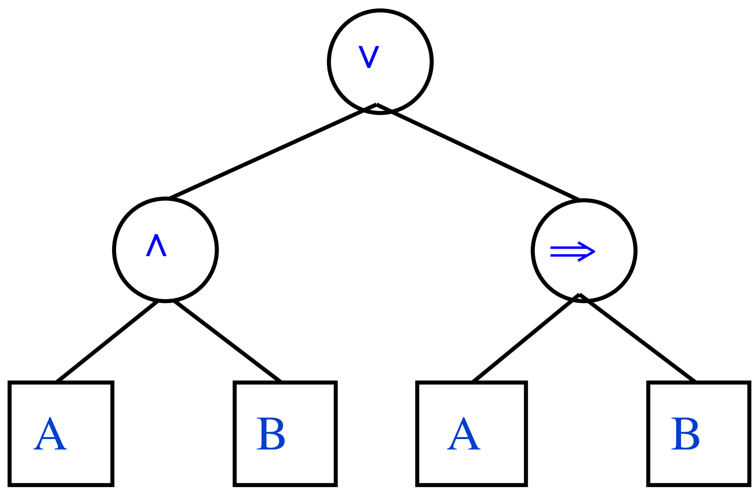
Zum Selbststudium: Beispiel: $\mathit{tiefe}(\neg((A \wedge B) \vee C))$

$$
\begin{aligned}
&\mathit{tiefe}(\neg((A \wedge B) \vee C)) \\
&\quad = \mathit{tiefe}(((A \wedge B) \vee C)) + 1 \\
&\quad = (\mathit{max}(\mathit{tiefe}((A \wedge B)), \mathit{tiefe}(C)) + 1) + 1 \\
&\quad = (\mathit{max}(\mathit{max}(\mathit{tiefe}(A), \mathit{tiefe}(B)) + 1, \mathit{tiefe}(C)) + 1) + 1 \\
&\quad = (\mathit{max}(\mathit{max}(0, 0) + 1, 0) + 1) + 1 \\
&\quad = (\mathit{max}(1, 0) + 1) + 1 \\
&\quad = (1 + 1) + 1 \\
&\quad = 3
\end{aligned}
$$



Zum Selbststudium: Unterschied von *grad* und *tiefe*

Vergleichen Sie die Funktionen *grad* und *tiefe* an Hand der Berechnungen für die Formel $((A \wedge B) \vee (A \Rightarrow B))$ mit dem Strukturbaum



Zum Selbststudium: Unterschied zwischen den Folien und Schöning

Schöning führt die Ausdrücke $(F \Rightarrow G)$ und $(F \Leftrightarrow G)$ als Abkürzungen ein.
Damit sind $\Rightarrow$ und $\Leftrightarrow$ eigentlich keine Junktoren.

(Der Grund von Schöning ist wahrscheinlich, dass damit die strukturelle Induktion etwas einfacher wird.)

Wir verzichten hier auf die Einführung von Abkürzungen, da die Abkürzungen in früheren Semestern zu Unklarheiten geführt haben. Stattdessen benutzen wir ein reicheres Junktoreninventar und machen uns bei den Beweisen etwas mehr Arbeit.

Grundsätzlich gilt: Die Form des Induktionsanfangs und des Induktionsschrittes hängt mit der Definition der Syntax der gewählten Sprache zusammen.

FAQ: Zur Definition von Formeln

Ist es ein Fehler, wenn wir in Übungsaufgaben zu viele oder zu wenig Klammern schreiben?

- Es kommt drauf an.
  - Ja, wenn es in der Übungsaufgabe gezielt um die Klammerung geht.
  - Ja, wenn dadurch Mehrdeutigkeiten entstehen (→ zu wenig Klammern).
  - Nein, bei überflüssigen Klammern, wenn es nicht um die Klammerung geht.
- Es werden auch noch weitere Klammerersparnisregeln eingeführt.

Wozu dient die Definition der Formeln?

- Sie dient im Wesentlichen dazu, festzulegen, welche Dinge wir zu berücksichtigen haben, wenn wir Behauptungen über Formeln aufstellen und diese beweisen wollen.
- Die Prinzipien der Strukturellen Induktion und der Strukturellen Rekursion greifen auf diese Definition zurück.
- Sie ist die Basis eines Formel-Parsers, also eines Programms, das Formeln einliest und ihre syntaktische Struktur analysiert.
- http://logik.phl.univie.ac.at/~chris/gateway/formular-zentral.html (8.4.07)
  Verarbeitungsmodus: "Ausdrucksbaum als Graphik"

Zum Selbststudium

In früheren Semestern

- wurde darum gebeten, den Beweis für das Prinzip der strukturellen Induktion vorzuführen.
- Der Beweis folgt auf den nächsten Folien. Ob sie präsentiert werden, hängt von den Wünschen der HörerInnen ab.
- Der Beweis wird nicht Teil der Prüfung sein, das Prinzip der strukturellen Induktion und seine Anwendung kann aber sehr wohl in der Klausur vorkommen!
- Der Beweis ist ein einfaches Beispiel dafür, wie über Abschlussbedingungen argumentiert werden kann. Diese Beweisform trifft man in der Informatik häufiger an.

Definition 2.4

Wir bilden folgende Mengen von Zeichenketten über dem Alphabet der Aussagenlogik (Def. 2.1):

- $\mathcal{F}_0 := \mathcal{As}_{AL}$

für alle $i \in \mathbb{N}_0$ sei

- $\mathcal{F}_{i+1} := \mathcal{F}_i \cup \{ \neg F \mid F \in \mathcal{F}_i\}$
$\cup \{ (F \wedge G) \mid F, G \in \mathcal{F}_i\}$
$\cup \{ (F \vee G) \mid F, G \in \mathcal{F}_i\}$
$\cup \{ (F \Rightarrow G) \mid F, G \in \mathcal{F}_i\}$
$\cup \{ (F \Leftrightarrow G) \mid F, G \in \mathcal{F}_i\}$

All diese Mengen fassen wir zusammen:

- $\mathcal{F} := \bigcup_{n \in \mathbb{N}_0} \mathcal{F}_n = \{ F \mid$ es gibt ein $n \in \mathbb{N}_0$, so dass $F \in \mathcal{F}_n\}$

Beobachtung 2.5

$\mathcal{F} = \mathcal{L}_{AL}$

Begründung: Die Mengenkonstruktion bildet die (informelle) Beschreibung in Def. 2.2 formal ab.

Zum Selbststudium: Abschlussbedingung

Definition 2.6

Eine Menge **M** von Zeichenketten heißt genau dann abgeschlossen bzgl. der Regeln zur Formelbildung, wenn gilt:

[A1] $\mathcal{A}s_{AL} \subseteq \mathbf{M}$ (die Aussagensymbole gehören zu **M**)

und für alle $F, G \in \mathbf{M}$ gilt

[A2] $\neg F, (F \wedge G), (F \vee G), (F \Rightarrow G), (F \Leftrightarrow G) \in \mathbf{M}$

Es sei $\mathcal{K} := \{ \mathbf{M} \mid \mathbf{M}$ ist abgeschlossen bzgl. der Regeln zur Formelbildung$\}$.

Hilfssatz (Lemma) 2.7

$\mathcal{L}_{AL}$ ist abgeschlossen bzgl. der Regeln zur Formelbildung. ($\mathcal{L}_{AL} \in \mathcal{K}$)

Voraussetzung: Def. 2.2, 2.6

Beweis

Nach Def. 2.2.1 ist $\mathcal{A}s_{AL} \subseteq \mathcal{L}_{AL}$, also ist Bedingung [A1] von Def. 2.6 erfüllt

Nach Def. 2.2.2 und 2.2.3 gilt für alle $F, G \in \mathcal{L}_{AL}$:

$\neg F, (F \wedge G), (F \vee G), (F \Rightarrow G), (F \Leftrightarrow G) \in \mathcal{L}_{AL}$,

also ist auch Bedingung [A2] von Def. 2.6 erfüllt.

Nach Def. 2.6 ist also $\mathcal{L}_{AL}$ ist abgeschlossen bzgl. der Regeln zur Formelbildung.

Zum Selbststudium: Zusammenhang zwischen $\mathcal{K}$ und $\mathcal{F}_n$

Hilfssatz (Lemma) 2.8

Jeder Menge, die abgeschlossen bzgl. der Regeln zur Formelbildung ist, umfasst alle in Definition 2.4 definierten Mengen $\mathcal{F}_n$. (Für jedes $\mathbf{M} \in \mathcal{K}$ und jedes $n \in \mathbb{N}_0$ gilt: $\mathcal{F}_n \subseteq \mathbf{M}$.)

Voraussetzung: Def. 2.4, 2.6

Beweis

Es sei $\mathbf{M} \in \mathcal{K}$ beliebig gewählt.

Der Beweis erfolgt durch vollständige Induktion über den Index.

Induktionsanfang: $\mathcal{F}_0 = \mathcal{As}_{AL}$ (Def. 2.4) und

$\mathcal{As}_{AL} \subseteq \mathbf{M}$ (Def. 2.6.[A1]) damit dann auch $\mathcal{F}_0 \subseteq \mathbf{M}$.

Induktionsannahme: Es sei $i \in \mathbb{N}_0$, so dass $\mathcal{F}_i \subseteq \mathbf{M}$.

Induktionsschritt

$$\mathcal{F}_{i+1} = \mathcal{F}_i \cup \{ \neg F \mid F \in \mathcal{F}_i\} \cup \{ (F \wedge G) \mid F, G \in \mathcal{F}_i\} \cup \{ (F \vee G) \mid F, G \in \mathcal{F}_i\}$$
$$\cup \{ (F \Rightarrow G) \mid F, G \in \mathcal{F}_i\} \cup \{ (F \Leftrightarrow G) \mid F, G \in \mathcal{F}_i\} \subseteq \mathbf{M},$$

(Def. 2.4), (Annahmen $\mathcal{F}_i \subseteq \mathbf{M} \in \mathcal{K}$), (Def. 2.6.[A2])

Also gilt nach dem Prinzip der vollständigen Induktion für alle $n \in \mathbb{N}_0$: $\mathcal{F}_n \subseteq \mathbf{M}$.

Da die Wahl von $\mathbf{M} \in \mathcal{K}$ nicht eingeschränkt war, gilt die Behauptung für alle $\mathbf{M} \in \mathcal{K}$.

Zum Selbststudium: Zusammenhang zwischen $\mathcal{K}$ und $\mathcal{L}_{\mathbf{AL}}$

Konsequenz (Korollar) 2.9 (von 2.4, 2.5, 2.7 und 2.8)

$\mathcal{L}_{\mathrm{AL}}$ ist die kleinste Menge, die abgeschlossen bzgl. der Regeln zur Formelbildung ist.
(Für alle $\mathbf{M} \in \mathcal{K}$ gilt: $\mathcal{L}_{\mathrm{AL}} \subseteq \mathbf{M}$.)

Erläuterung

- Wie in der formelleren Variante deutlich wird, ist ‘die kleinste Menge’ hier bezogen auf Mengeninklusion gemeint. Man beachte, dass $\mathcal{L}_{\mathrm{AL}}$ nicht endlich ist.
- Das Korollar ergibt sich daraus, dass jedes Element von $\mathcal{L}_{\mathrm{AL}}$ in irgendeinem $\mathcal{F}_{\mathrm{n}}$ enthalten sein muss und die $\mathcal{F}_{\mathrm{n}}$ wie eben gezeigt alle in $\mathbf{M}$ enthalten sind.

Zum Selbststudium: Prinzip der strukturellen Induktion

Satz (Theorem) 2.10

Es sei $\mathcal{B}$ eine Eigenschaft, die Zeichenketten haben können (oder auch nicht).

Wenn

[V1] jede atomare Formel F die Eigenschaft $\mathcal{B}$ hat (*Induktionsanfang*) und

[V2] für alle Formeln F und G, die die Eigenschaft $\mathcal{B}$ haben, auch gilt, dass die hieraus gebildeten Formeln $\neg F$, $(F \wedge G)$, $(F \vee G)$, $(F \Rightarrow G)$ und $(F \Leftrightarrow G)$ die Eigenschaft $\mathcal{B}$ haben (*Induktionsannahme und -schritt*),

dann hat jede Formel aus $\mathcal{L}_{AL}$ die Eigenschaft $\mathcal{B}$.

Voraussetzung: Def. 2.2, 2.4, 2.6, Kor. 2.9

Beweis

- Es sei $\mathcal{B}$ eine Eigenschaft, die [V1] und [V2] erfüllt.
- Es sei $\mathbf{M}_{\mathcal{B}}$ die Menge aller Zeichenketten, die die Eigenschaft $\mathcal{B}$ haben.
  ($\mathbf{M}_{\mathcal{B}}$ wird auch als die Extension der Eigenschaft $\mathcal{B}$ bezeichnet.)
- [V1] und [V2] sagen (entsprechend Def. 2.6) aus, dass $\mathbf{M}_{\mathcal{B}}$ abgeschlossen bzgl. der Regeln zur Formelbildung ist und damit, dass $\mathbf{M}_{\mathcal{B}} \in \mathcal{K}$.
- Mit Korollar 2.9 gilt dann auch $\mathcal{L}_{AL} \subseteq \mathbf{M}_{\mathcal{B}}$,
  Also: jede Formel gehört zu $\mathbf{M}_{\mathcal{B}}$, also hat auch jede Formel die Eigenschaft $\mathcal{B}$.

Ergänzung: Logik-Fragmente

Spezielle Formelmengen

- Als ‚Logik-Fragment' bezeichnet man Teilmengen der Logik-Sprache, die interessante Eigenschaften haben und durch grammatische Beschränkungen definiert sind.

Beispiele

- die Formeln in konjunktiver Normalform (KNF)
- die Formeln in disjunktiver Normalform (DNF)
- die Formeln in Negations-Normalform (NNF)
- die Hornformeln (zwei Varianten: KNF vs. Implikationsschreibweise)

Nutzen: unterschiedlich

- KNF, DNF, NNF: keine Beschränkung der Ausdrucksmächtigkeit, angenehme Eigenschaften für die Verarbeitung, wenig stilistische Variation
- Hornformeln: Beschränkung der Ausdrucksmächtigkeit, Verarbeitung wird richtig effizient, aber man kann auch nicht alles bearbeiten.

Ergänzung: Definition von Logikfragmenten

Typische Beschränkungen

- Junktorenauswahl (z.B. nur $\neg$, $\wedge$, $\vee$)
- Hierarchische Beschränkung in der Junktoren-Anordnung (z.B. Negation nur an Aussagensymbolen)

Nicht-Terminalsymbole

- oft werden mehr Nicht-Terminalsymbole in den Grammatiken von Logik-Fragmenten benötigt als in der Grammatik der vollen Aussagenlogik, da hierarchische Beschränkungen über Nicht-Terminalsymbole gesteuert werden.

Die Regelmengen

- werden entsprechend auch komplizierter.

Beispiele

- Auf den folgenden Folien werden einige Grammatiken für Logikfragmente gegeben, auf die wir später noch einmal zurückkommen oder in den Aufgaben verweisen.

Ergänzung: Grammatik für Konjunktive Normalformen

Vokabular der terminalen Symbole (Symbole, die in der Formel vorkommen): $\Sigma_{KNF} = \mathcal{A}s_{AL} \cup \{\neg, \wedge, \vee,), (\}$ **nicht-terminales Symbol** (Symbole, die für die Erzeugung der Formeln benötigt werden, aber nicht in der Formel vorkommen): $N_{KNF} = \{KNF, Kl, L, As\}$ **Startsymbol:** KNF KNF: Konjunktive Normalform Kl: Klausel L: Literal As: Aussagesymbol	**Regeln (Produktionen):** $P = \{$ KNF $\rightarrow$ (KNF $\wedge$ KNF) , KNF $\rightarrow$ Kl , Kl $\rightarrow$ (Kl $\vee$ Kl) , Kl $\rightarrow$ L , L $\rightarrow$ $\neg$As , L $\rightarrow$ As , As $\rightarrow$ A , As $\rightarrow$ B , As $\rightarrow$ C , As $\rightarrow$ D , … $\}$

Ergänzung: Grammatik für Disjunktive Normalformen

Vokabular der terminalen Symbole	**Regeln (Produktionen):**
(Symbole, die in der Formel vorkommen): $\Sigma_{DNF} = \mathcal{A}s_{AL} \cup \{\neg, \wedge, \vee,), (\}$	$P = \{$ DNF $\rightarrow$ (DNF $\vee$ DNF) ,
nicht-terminales Symbol	DNF $\rightarrow$ DKl ,
(Symbole, die für die Erzeugung der Formeln benötigt werden, aber nicht in der Formel vorkommen):	DKl $\rightarrow$ (DKl $\wedge$ DKl) ,
$\mathcal{N}_{DNF} = \{$DNF, DKl, L, As$\}$	DKl $\rightarrow$ L ,
Startsymbol: DNF	L $\rightarrow$ $\neg$As ,
	L $\rightarrow$ As ,
Anmerkung:	As $\rightarrow$ A ,
DNF: Disjunktive Normalform	As $\rightarrow$ B ,
DKl: Duale Klausel	As $\rightarrow$ C ,
L: Literal	As $\rightarrow$ D ,
As: Aussagesymbol	… $\}$

Ergänzung: Grammatik für Negations-Normalformen

Vokabular der terminalen Symbole	Regeln (Produktionen):
(Symbole, die in der Formel vorkommen): $\Sigma_{NNF} = \mathcal{A}s_{AL} \cup \{\neg, \wedge, \vee,), (\ \}$	$P = \{$ NNF $\rightarrow$ (NNF $\wedge$ NNF) ,
nicht-terminales Symbol	NNF $\rightarrow$ (NNF $\vee$ NNF) ,
(Symbole, die für die Erzeugung der Formeln benötigt werden, aber nicht in der Formel vorkommen):	NNF $\rightarrow$ L ,
$N_{NNF} = \{$NNF, L, As $\}$	L $\rightarrow$ ¬As ,
Startsymbol: NNF	L $\rightarrow$ As ,
	As $\rightarrow$ A ,
Anmerkung:	As $\rightarrow$ B ,
NNF: Konjunktive Normalform	As $\rightarrow$ C ,
L: Literal	As $\rightarrow$ D ,
As: Aussagesymbol	… $\}$

Ergänzung: Grammatik für Hornformel (KNF-Spezialfall)

Vokabular der terminalen Symbole	Regeln (Produktionen):
Vokabular der terminalen Symbole (Symbole, die in der Formel vorkommen): $\Sigma_{HFK} = \mathcal{A}s_{AL} \cup \{\neg, \wedge, \vee,), (\}$ **nicht-terminales Symbol** (Symbole, die für die Erzeugung der Formeln benötigt werden, aber nicht in der Formel vorkommen): $N_{HFK} = \{HFK, HKl, L, As\}$ **Startsymbol:** HFK Anmerkung: HFK: Hornformel KNF-Schreibweise HKl: Horn-Klausel L: Literal NL: negatives Literal As: Aussagesymbol (positives Literal)	**Regeln (Produktionen):** P = { HFK → (HFK ∧ HFK) , HFK → HKl , HKl → (NL ∨ HKl) , HKl → (HKl ∨ NL) , HKl → L , NL → ¬As , L → NL , L → As , As → A , As → B , As → C , As → D , … }

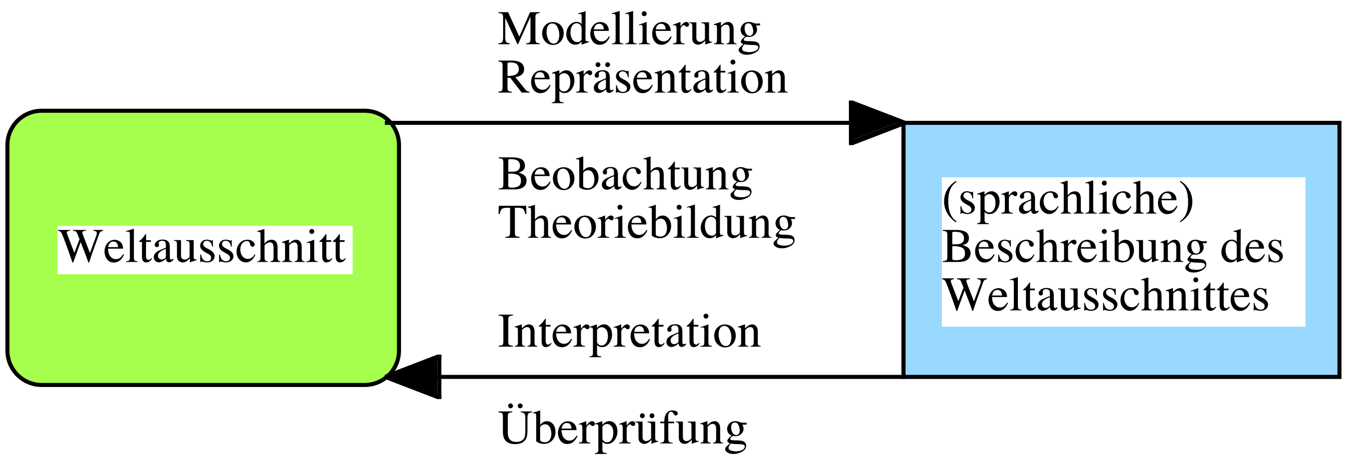
Ergänzung: Grammatik für Hornformel (Implikationsschreibweise)

Vokabular der terminalen Symbole (Symbole, die in der Formel vorkommen): $\Sigma_{HFI} = \mathcal{As}_{AL} \cup \{\wedge, \Rightarrow, \bot, \top,), (\}$ **nicht-terminales Symbol** (Symbole, die für die Erzeugung der Formeln benötigt werden, aber nicht in der Formel vorkommen): $N_{HFI} = \{HFI, HI, L, As\}$ **Startsymbol:** HF Anmerkung: HFI: Hornformel Implikationsschreibweise HI: Horn-Implikation KAt: Konjunktion von Atomen As: Aussagesymbol (positives Literal)	**Regeln (Produktionen):** $P = \{$ HFI $\rightarrow$ HI $\wedge$ HFI , HFI $\rightarrow$ HI , HI $\rightarrow$ (KAt $\Rightarrow$ As) , HI $\rightarrow$ (KAt $\Rightarrow$ $\bot$) , HI $\rightarrow$ ($\top$ $\Rightarrow$ As) , KAt $\rightarrow$ As $\wedge$ KAt , KAt $\rightarrow$ As , As $\rightarrow$ A , As $\rightarrow$ B , As $\rightarrow$ C , As $\rightarrow$ D , … $\}$

Wichtige Konzepte in diesem Foliensatz

- Aussagensymbol, Junktor, Formel
- Teilformel, Hauptoperator
- Ableitungsbaum, Strukturbaum
- Strukturelle Induktion
- Rekursive Definition
- Grad einer Formel, Tiefe einer Formel

Aussagenlogik: Grundbegriffe, Semantik



Beschreibungen / Formeln

- können einen Weltausschnitt (unter einer Interpretation)
- korrekt oder inkorrekt beschreiben

Mathematische Abstraktion (Aussagenlogik)

- eine Interpretation (Belegung) ordnet Formeln Wahrheitswerte zu.

Ein Beispiel

Unsere Aussagen zur Beschreibung des Weltausschnittes

$\mathcal{As}_{\mathrm{AL}} = \{$ HP7_e, HP7_d, K23_e, K23_d$\}$

- HP7_e ≈ Harry Potter 7 ist auf Englisch erschienen.
- HP7_d ≈ Harry Potter 7 ist auf Deutsch erschienen.
- K23_e ≈ Kunde 23 kauft englische Bücher.
- K23_d ≈ Kunde 23 kauft deutsche Bücher.

Formelmengen stehen für Zustände des Weltausschnittes,

- in denen alle gegebenen Formeln wahr sind.
- {HP7_e, HP7_d, K23_d , ¬K23_e} : Harry Potter 7 ist in Deutsch und in Englisch erschienen, K23 kauft deutsche aber keine englischen Bücher.
- {HP7_e, HP7_d, K23_d} : Harry Potter 7 ist in Deutsch und in Englisch erschienen, K23 kauft deutsche Bücher (ob er englische Bücher kauft, ist nicht bekannt).

Zum Selbststudium: Datenbanken

Datenbanken ≈ Sammlung von atomarer Aussagen

- In Datenbanken werden atomare Aussagen über die Welt gespeichert, z.B. solche, wie im Beispiel der vorigen Folie, die in einer Datenbank eines Internet-Buchhändlers auftauchen:
  - Die HP7-Aussagen betreffen die Verfügbarkeit von Produkten (Büchern), sind also relevant für die Lieferbarkeit und für Ankündigungen.
  - Die K23-Aussagen betreffen das Kaufverhalten eines Kunden, sind also interessant für die on-line Information des Kunden.
- Da in der Datenbank keine Negation dargestellt wird, ist es üblich, zu vereinbaren, dass eine nicht enthaltene Aussage als falsch gilt (das ist die ‚closed world assumption').
- Für Anfragen an die Datenbank gibt es eine komplexe Sprache (SQL), die aufgrund der Datenbankinhalte ausgewertet wird.
- Das hier verwendete Darstellungsformat der Aussagenlogik, d.h. eine Aussage wird durch ein Aussagensymbol repräsentiert, ist etwas vereinfachend.
  - Die Prädikatenlogik (wird in späteren Kapiteln behandelt) macht es möglich, die interne Struktur von Aussagen zu berücksichtigen.
  - In Datenbankanfragesprachen werden meist spezielle Subsysteme/Subsprachen der Prädikatenlogik verwendet; so ist etwa SQL eine Einschränkung der Prädikatenlogik.

Wahrheitswerte

- sind keine Formeln und können nicht in Formeln auftreten
- Die Menge der *Wahrheitswerte* enthält genau zwei Elemente.
- Wir verwenden die Menge {**0**, **1**}, wobei wir **0** auch ‚falsch' aussprechen und **1** als ‚wahr' aussprechen.
- Häufig werden auch die Mengen {**falsch**, **wahr**}, {**f**, **w**}, {**f**, **t**} verwendet.

Aussagen / Formeln

- können bezüglich einer Interpretation / Belegung wahr oder falsch sein

Die aussagenlogische Semantik

- regelt, wie komplexe Formeln zu ihren Wahrheitswerten kommen
- modelliert die Bedeutung von Formeln über die Gesamtheit aller Möglichkeiten der Wahrheitswertzuordnung (Wahrheitswertverläufe)
- stellt Fachtermini für die Beschreibung von Wahrheitswertverläufen und den Vergleich von Wahrheitswertverläufen bereit.

Belegung von Aussagensymbolen: Zuordnung von Wahrheitswerten

Definition 3.1 (wird fortgesetzt)

Eine *Belegung* (oder *Wahrheitswertzuordnung*) für $\mathcal{A}s_{\text{AL}}$ ist eine Funktion $\mathcal{A}_{\mathcal{A}s}$: $\mathcal{A}s_{\text{AL}} \rightarrow \{\mathbf{0}, \mathbf{1}\}$, die die Aussagensymbole auf Wahrheitswerte abbildet.

Beispiel

16 verschiedene Belegungen (Weltzustände)

	HP7_e	HP7_d	K23_e	K23_d
$\mathcal{A}_1$	0	0	0	0
$\mathcal{A}_2$	1	0	0	0
$\mathcal{A}_3$	0	1	0	0
$\mathcal{A}_4$	1	1	0	0
$\mathcal{A}_5$	0	0	1	0
$\mathcal{A}_6$	1	0	1	0
$\mathcal{A}_7$	0	1	1	0
$\mathcal{A}_8$	1	1	1	0

	HP7_e	HP7_d	K23_e	K23_d
$\mathcal{A}_9$	0	0	0	1
$\mathcal{A}_{10}$	1	0	0	1
$\mathcal{A}_{11}$	0	1	0	1
$\mathcal{A}_{12}$	1	1	0	1
$\mathcal{A}_{13}$	0	0	1	1
$\mathcal{A}_{14}$	1	0	1	1
$\mathcal{A}_{15}$	0	1	1	1
$\mathcal{A}_{16}$	1	1	1	1

Definition 3.1

Eine ***Belegung*** (oder *Wahrheitswertzuordnung*) ist eine Funktion $\mathcal{A}_{\mathcal{A}s}\colon \mathcal{A}s_{\mathrm{AL}} \to \{\mathbf{0}, \mathbf{1}\}$, die die Aussagensymbole auf Wahrheitswerte abbildet.

Für jede Belegung $\mathcal{A}_{\mathcal{A}s}$ wird rekursiv eine Funktion $\mathcal{A}\colon \mathcal{L}_{\mathrm{AL}} \to \{\mathbf{0}, \mathbf{1}\}$ definiert, die alle Formeln bewertet.

Für jedes Aussagensymbol $\mathsf{A} \in \mathcal{A}s_{\mathrm{AL}}$ sei $\mathcal{A}(\mathsf{A}) = \mathcal{A}_{\mathcal{A}s}(\mathsf{A})$.

Für alle Formeln $\mathsf{F}, \mathsf{G} \in \mathcal{L}_{\mathrm{AL}}$ sei

$$\mathcal{A}(\neg \mathsf{F}) = \begin{cases} \mathbf{1}, \text{ falls } \mathcal{A}(\mathsf{F}) = \mathbf{0} \\ \mathbf{0}, \text{ sonst} \end{cases}$$

$$\mathcal{A}((\mathsf{F} \wedge \mathsf{G})) = \begin{cases} \mathbf{1}, \text{ falls } \mathcal{A}(\mathsf{F}) = \mathbf{1} \text{ und } \mathcal{A}(\mathsf{G}) = \mathbf{1} \\ \mathbf{0}, \text{ sonst} \end{cases}$$

$$\mathcal{A}((\mathsf{F} \vee \mathsf{G})) = \begin{cases} \mathbf{1}, \text{ falls } \mathcal{A}(\mathsf{F}) = \mathbf{1} \text{ oder } \mathcal{A}(\mathsf{G}) = \mathbf{1} \\ \mathbf{0}, \text{ sonst} \end{cases}$$

$$\mathcal{A}((\mathsf{F} \Rightarrow \mathsf{G})) = \begin{cases} \mathbf{1}, \text{ falls } \mathcal{A}(\mathsf{F}) = \mathbf{0} \text{ oder } \mathcal{A}(\mathsf{G}) = \mathbf{1} \\ \mathbf{0}, \text{ sonst} \end{cases}$$

$$\mathcal{A}((\mathsf{F} \Leftrightarrow \mathsf{G})) = \begin{cases} \mathbf{1}, \text{ falls } \mathcal{A}(\mathsf{F}) = \mathcal{A}(\mathsf{G}) \\ \mathbf{0}, \text{ sonst} \end{cases}$$

Verknüpfungstafeln für Junktoren

- stellen die Bedingungen der Definition 3.1 übersichtlicher dar.

	F	G	$(F \wedge G)$	$(F \vee G)$	$(F \Rightarrow G)$	$(F \Leftrightarrow G)$
$\mathcal{A}_1$	0	0	0	0	1	1
$\mathcal{A}_2$	0	1	0	1	1	0
$\mathcal{A}_3$	1	0	0	1	0	0
$\mathcal{A}_4$	1	1	1	1	1	1

	F	$\neg F$
$\mathcal{A}'_1$	0	1
$\mathcal{A}'_2$	1	0

- In jeder Zeile steht eine Belegung.
- Alle aufgeführten Belegungen unterscheiden sich in der interessierenden Domäne

Zum Selbststudium: Das allgemeine Schema für Verknüpfungstafeln

	Teilformeln		komplexe, zusammengesetzte Formeln			
	F	G	$(F \wedge G)$	$(F \vee G)$	$(F \Rightarrow G)$	$(F \Leftrightarrow G)$
$\mathcal{A}_1$	$\mathcal{A}_1(F)$	$\mathcal{A}_1(G)$	$\mathcal{A}_1((F \wedge G))$	$\mathcal{A}_1((F \vee G))$	$\mathcal{A}_1((F \Rightarrow G))$	$\mathcal{A}_1((F \Leftrightarrow G))$
$\mathcal{A}_2$	$\mathcal{A}_2(F)$	$\mathcal{A}_2(G)$	$\mathcal{A}_2((F \wedge G))$	$\mathcal{A}_2((F \vee G))$	$\mathcal{A}_2((F \Rightarrow G))$	$\mathcal{A}_2((F \Leftrightarrow G))$
$\mathcal{A}_3$	$\mathcal{A}_3(F)$	$\mathcal{A}_3(G)$	$\mathcal{A}_3((F \wedge G))$	$\mathcal{A}_3((F \vee G))$	$\mathcal{A}_3((F \Rightarrow G))$	$\mathcal{A}_3((F \Leftrightarrow G))$
$\mathcal{A}_4$	$\mathcal{A}_4(F)$	$\mathcal{A}_4(G)$	$\mathcal{A}_4((F \wedge G))$	$\mathcal{A}_4((F \vee G))$	$\mathcal{A}_4((F \Rightarrow G))$	$\mathcal{A}_4((F \Leftrightarrow G))$

	F	$\neg F$
$\mathcal{A}'_1$	$\mathcal{A}'_1(F)$	$\mathcal{A}'_1(\neg F)$
$\mathcal{A}'_2$	$\mathcal{A}'_2(F)$	$\mathcal{A}'_2(\neg F)$

- Jede Kombination von Wahrheitswerten für die Teilformeln (F und G) muss in einer der Belegungen $\mathcal{A}_i$ vorkommen.

➔ Die Verknüpfungstafel für einen n-stelligen Junktor enthält 2^n Zeilen.

Beispiel: $\mathcal{A}_1(\neg((A \wedge B) \vee C))$

Es sei $\mathcal{A}_{\mathcal{A}s1}$ eine Belegung, für die gilt: $\mathcal{A}_{\mathcal{A}s1}(A) = \mathbf{0}$, $\mathcal{A}_{\mathcal{A}s1}(B) = \mathbf{1}$, $\mathcal{A}_{\mathcal{A}s1}(C) = \mathbf{0}$

Dann gilt für die Funktion $\mathcal{A}_1$ (als rekursive Fortsetzung von $\mathcal{A}_{\mathcal{A}s1}$)

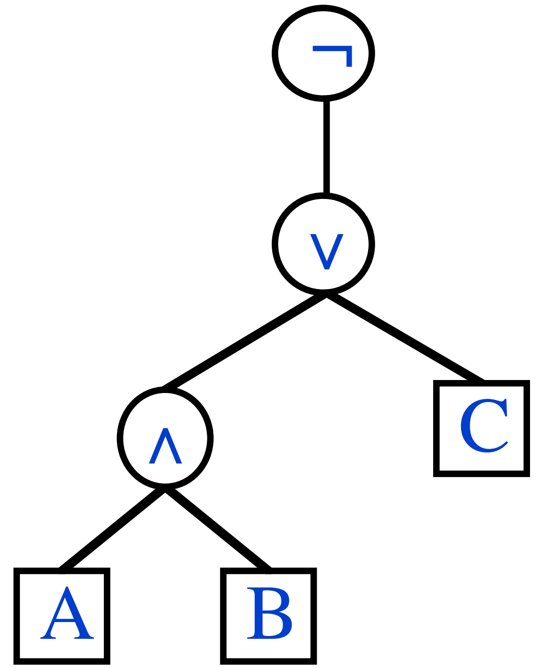
F	G	(F ∧ G)	(F ∨ G)	¬F
0	0	0	0	1
0	1	0	1	1
1	0	0	1	0
1	1	1	1	0

$\mathcal{A}_1(A) = \mathbf{0}$, $\mathcal{A}_1(B) = \mathbf{1}$, $\mathcal{A}_1(C) = \mathbf{0}$

$\mathcal{A}_1((A \wedge B)) = \mathbf{0}$

$\mathcal{A}_1(((A \wedge B) \vee C)) = \mathbf{0}$

$\mathcal{A}_1(\neg((A \wedge B) \vee C)) = \mathbf{1}$



Beispiel: $\mathcal{A}_2(\neg((A \wedge B) \vee C))$

Es sei $\mathcal{A}_{\mathcal{A}s2}$ eine Belegung, für die gilt: $\mathcal{A}_{\mathcal{A}s2}(A) = \mathbf{1}$, $\mathcal{A}_{\mathcal{A}s2}(B) = \mathbf{0}$, $\mathcal{A}_{\mathcal{A}s2}(C) = \mathbf{1}$

Dann gilt für die Funktion $\mathcal{A}_2$ (als rekursive Fortsetzung von $\mathcal{A}_{\mathcal{A}s2}$)

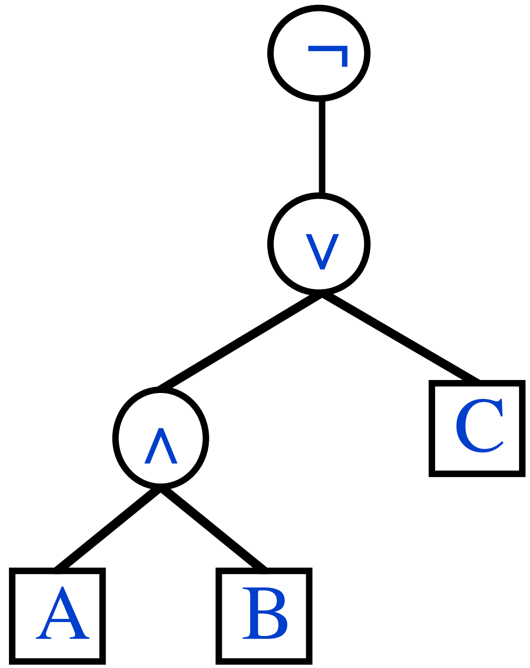
F	G	(F ∧ G)	(F ∨ G)	¬F
0	**0**	**0**	**0**	**1**
0	**1**	**0**	**1**	**1**
1	**0**	**0**	**1**	**0**
1	**1**	**1**	**1**	**0**

$\mathcal{A}_2(A) = \mathbf{1}$, $\mathcal{A}_2(B) = \mathbf{0}$, $\mathcal{A}_2(C) = \mathbf{1}$

$\mathcal{A}_2((A \wedge B)) = \mathbf{0}$

$\mathcal{A}_2(((A \wedge B) \vee C)) = \mathbf{1}$

$\mathcal{A}_2(\neg((A \wedge B) \vee C)) = \mathbf{0}$



Beobachtungen zu Definition 3.1

Beobachtung 3.1.1

Es sei $\mathcal{A}_{\mathcal{A}s}$ eine Belegung.

- Dann weist $\mathcal{A}$ der Formel F einen eindeutig bestimmten Wahrheitswert zu.
  - ➔ Die Funktionen $\mathcal{A}$ und $\mathcal{A}_{\mathcal{A}s}$ stimmen auf den Aussagensymbolen überein.
  - ➔ $\mathcal{A}$ ist eine eindeutige Erweiterung von $\mathcal{A}_{\mathcal{A}s}$. (Prinzip der strukturellen Rekursion)
  - ➔ Im weiteren keine typographische Unterscheidung zwischen $\mathcal{A}$ und $\mathcal{A}_{\mathcal{A}s}$.

Beobachtung 3.1.2

Es seien $\mathcal{A}$ und $\mathcal{A}'$ zwei Belegungen.

- Wenn für alle Aussagensymbole A_1 von F: $\mathcal{A}(A_1) = \mathcal{A}'(A_1)$, dann auch $\mathcal{A}(F) = \mathcal{A}'(F)$
  - ➔ Für den Wahrheitswert einer Formel F unter einer Belegung spielt die Zuordnung von Wahrheitswerten zu Aussagensymbolen, die nicht in F vorkommen, keine Rolle.
  - ➔ Wer mag, sollte ruhig einmal versuchen, diese Beobachtung zu beweisen. Mit dem Prinzip der strukturellen Rekursion geht das ganz einfach.

Zum Selbststudium

Bei Schöning

- werden Belegungen als partielle Funktionen definiert, die also nicht allen Aussagesymbolen einen Wert zuordnen müssen.
- Das macht Beweise viel komplizierter als erforderlich, ohne je richtig zu nützen.
- Deshalb nutzen wir nur Belegungen, die wirklich jedes Aussagensymbol bewerten.
- Daher ist zu bedenken, dass Belegungen, die in Beispielen auftauchen (z.B. auf der nächsten Seite), in der Regel nur unvollständig spezifiziert sind.

Zum Selbststudium: Mögliche Welten und Zustandsbeschreibungen (Spies)

Mögliche Welt (Spies Abs. 1.5 nennt Belegungen ‚Mögliche Welten')

Eine aussagenlogische mögliche Welt zu einer Liste atomarer Aussagen gibt zu jeder Aussage der Liste genau einen Wahrheitswert an.

	A	B	C	D	E	F	G	H	…
	0	0	0	0	0	0	0	0	…
	1	0	0	0	0	0	0	0	…
	0	1	0	0	0	0	0	0	…
	1	1	0	0	0	0	0	0	…
	…	…	…	…	…	…	…	…	…

Anmerkungen

Jede Zeile beschreibt eine Konstellation von Wahrheitswertzuordnungen (Belegungen), d.h. einen (prinzipiell) möglichen Zustand der Welt („Mögliche Welt").

Werden n atomare Aussagen betrachtet, so existieren 2^n verschiedene Wahrheitswertzuordnungen zu diesen atomaren Aussagen, d.h. 2^n mögliche Welten.

Zum Selbststudium: Natürliche Sprache und Junktoren

Die Verknüpfungstafeln der Junktoren

- legen die Bedeutung der aussagenlogischen Junktoren fest
- bestimmen, welche natürlich-sprachlichen Elemente durch die Junktoren symbolisiert werden können

Beispiel: ‚und'

- Der Satz ‚Peter trägt eine Kiste und Paul trägt eine Kiste' ist falsch, wenn ‚Peter trägt eine Kiste' falsch ist oder wenn ‚Paul trägt eine Kiste' falsch ist. Sonst ist der Satz wahr.
  ➔ Diese Verwendung von ‚und' kann durch $\wedge$ symbolisiert werden.
- Der Satz ‚Peter und Paul tragen eine Kiste' beschreiben eine Gemeinschaftsaktion.
  ➔ Diese Verwendung von ‚und' kann nicht einfach durch $\wedge$ symbolisiert werden.
- Der Satz ‚Peter ist gefallen und hat sich verletzt' ist falsch, wenn ‚Peter ist gefallen' falsch ist oder wenn ‚Peter hat sich verletzt' falsch ist. Sonst ist der Satz wahr.
  ➔ Diese Verwendung von ‚und' kann durch $\wedge$ symbolisiert werden.
  Mit $\wedge$ wird aber kein zeitlicher oder kausaler Zusammenhang ausgedrückt.

Die zweistelligen Junktoren

- Eine Formel mit n Aussagensymbolen hat 2^n bzgl. der Aussagensymbole unterscheidbare Belegungen.
- Es gibt 2^{2^n} Möglichkeiten, n Formeln mit n-stelligen (wahrheitsfunktionalen) Junktoren zu verknüpfen. ➔ Es gibt 16 zweistellige Junktoren.
- *Wahrheitsfunktional*: Wahrheitswert der Formel ergibt sich eindeutig aus den Wahrheitswerten der Teilformeln.

	F	G	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)	(13)	(14)	(15)	(16)
$\mathcal{A}_1$	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
$\mathcal{A}_2$	0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
$\mathcal{A}_3$	1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
$\mathcal{A}_4$	1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
				$\wedge$						$\vee$		$\Leftrightarrow$				$\Rightarrow$		
					$\not\Rightarrow$		$\not\Leftarrow$		$\not\Leftrightarrow$		$\downarrow$			$\Leftarrow$			$\uparrow$	

$\downarrow$ Peirce-Pfeil (NOR) $\uparrow$ Sheffer-Strich (NAND)

Zum Selbststudium

Triviale und fast-triviale Wahrheitswertfunktionen

- Trivial: Die Wahrheitswertverläufe in den Spalten (1) und (16) sind konstant, also ignorieren die Eingabewerte völlig. Entsprechend macht es keinen Sinn, einen Junktor zu definieren, der eine dieser Funktionen als semantischen Wert hat.
- Fast Trivial: Die Wahrheitswertverläufe in den Spalten (4) und (6) stimmen genau mit einem der Eingabewerte überein, der zweite Eingabewert wird ignoriert. Entsprechend macht es keinen Sinn, einen zweistelligen Junktor zu definieren, der eine dieser Funktionen als semantischen Wert hat.
- Fast Trivial: Die Wahrheitswertverläufe in den Spalten (11) und (13) stimmen genau mit der Negation von einem der Eingabewerte überein, der zweite Eingabewert wird wieder ignoriert. Auch hier macht es keinen Sinn, einen zweistelligen Junktor zu definieren, der eine dieser Funktionen als semantischen Wert hat. Dafür haben wir schließlich die (einstellige) Negation $\neg$.
- Bei den anderen Wahrheitswertverläufen kann man anfangen, darüber zu diskutieren, welchen Nutzen sie haben. Entsprechend gibt es Standard-Symbole und Namen, wie in obiger Tabelle aufgelistet. Wir verwenden die 4 Symbole in der ersten Zeile, könnten aber durchaus auch die anderen in unsere Sprache aufnehmen.

Wahrheitstafeln: Verallgemeinerung der Verknüpfungstafeln

Das Prinzip: Gegeben sei eine Formel F mit n (verschiedenen) Aussagensymbolen.
Tafel mit 2^n Zeilen, die ersten n Spalten mit den Aussagensymbolen beschriftet.
Für jede komplexe Teilformel von F eine weitere Spalte.
In jeder Zeile steht eine Belegung. Belegungen in verschiedenen Zeilen unterscheiden sich bei mindestens einem der interessierenden Aussagensymbole.
Die Berechnung erfolgt von den Teilformeln zu den komplexeren Formeln.

Beispiel: $F := (A \wedge B) \Rightarrow (A \vee \neg C)$

	A	B	C	$(A \wedge B)$	$\neg C$	$(A \vee \neg C)$	$(A \wedge B) \Rightarrow (A \vee \neg C)$
$\mathcal{A}_1$	0	0	0				
$\mathcal{A}_2$	0	0	1				
$\mathcal{A}_3$	0	1	0				
$\mathcal{A}_4$	0	1	1				
$\mathcal{A}_5$	1	0	0				
$\mathcal{A}_6$	1	0	1				
$\mathcal{A}_7$	1	1	0				
$\mathcal{A}_8$	1	1	1				

Wahrheitstafeln zur Berechnung von Wahrheitswertverläufen

für komplexe Formeln

- Mehrere komplexe Formeln (und ihre Teilformeln) können in eine Tafel eingetragen werden.

Beispiel

	A	B	¬A	(¬A ∨ B)	(A ⇒ B)
$\mathcal{A}_1$	0	0	1	1	1
$\mathcal{A}_2$	0	1	1	1	1
$\mathcal{A}_3$	1	0	0	0	0
$\mathcal{A}_4$	1	1	0	1	1

Definition 3.2

Eine Spalte der Wahrheitstafel bezeichnen wir auch als den *Wahrheitswertverlauf* der zugehörigen Formel.

➔ Die Formeln (¬A ∨ B) und (A ⇒ B) haben denselben Wahrheitswertverlauf.

Zum Selbststudium: Formeln vs. Wahrheitswerte

- Formeln sind Zeichenketten.
- Wahrheitswerte sind keine Zeichenketten.
- Aus diesem Grund können eine Formel und ein Wahrheitswert auch nicht identisch sein. Gemäß unserer Farbkodierung gilt: blaue Objekte und grüne Objekte können nicht gleichgesetzt werden.
- Trotzdem finden sich Gleichsetzungen der Art $\mathsf{A} = \mathbf{1}$, die ausdrücken sollen, dass A unter einer (ungenannten) Belegung wahr ist, also als Kurzform für $\mathcal{A}(\mathsf{A}) = \mathbf{1}$. Wahrheitswertverläufe, an denen wir hier primär interessiert sind, werden durch die Kurzschreibweise nicht gut erfasst. Gewöhnen Sie sich deshalb lieber Schreibweisen an, die die jeweils gemeinte Belegung explizit machen.
- Manche Autoren verwenden auch $\mathsf{0}$ und $\mathsf{1}$ als logische Konstanten, die Kontradiktionen oder Tautologien sind. Für diese Konstanten gilt dann also für jede Belegung $\mathcal{A}(\mathsf{0}) = \mathbf{0}$ bzw. $\mathcal{A}(\mathsf{1}) = \mathbf{1}$. Diese Konstanten sind dann aber keine Wahrheitswerte sondern Formeln, wir haben gemäß unserer Farbdarstellung also blaue und grüne 0-en und 1-en. Wir werden in Kürze die Zeichen $\perp$ (bottom) und $\top$ (top) als entsprechende logische Konstanten einsetzen.

Modelle

Definition 3.3

- Eine Belegung $\mathcal{A}$ heißt genau dann *Modell* für eine Formel F, wenn $\mathcal{A}(F) = \mathbf{1}$
  ➔ Um festzustellen ob $\mathcal{A}$ Modell von F ist, berücksichtigt man genau die Zelle der Wahrheitstafel, die durch $\mathcal{A}$ und F festgelegt wird.
- Weitere Sprechweisen / Schreibweisen für „$\mathcal{A}$ ist Modell für F"
  - $\mathcal{A}$ macht F wahr, $\mathcal{A}$ erfüllt F
  - F gilt unter der Belegung $\mathcal{A}$
  - $\mathcal{A} \models F$
- Falls $\mathcal{A}$ kein Modell für F ist ($\mathcal{A}(F) = \mathbf{0}$), dann sagen wir auch
  - $\mathcal{A}$ macht F falsch, $\mathcal{A}$ falsifiziert F
  - F gilt nicht unter der Belegung $\mathcal{A}$
  - $\mathcal{A} \not\models F$
- Wenn **M** eine Formelmenge ist, dann heißt eine Belegung $\mathcal{A}$, die Modell für jede Formel aus **M** ist, *Modell* für **M**.

Beispiel: Modelle von Formelmengen

	HP7_e	HP7_d	K23_e	K23_d
$\mathcal{A}_1$	0	0	0	0
$\mathcal{A}_2$	1	0	0	0
$\mathcal{A}_3$	0	1	0	0
$\mathcal{A}_4$	1	1	0	0
$\mathcal{A}_5$	0	0	1	0
$\mathcal{A}_6$	1	0	1	0
$\mathcal{A}_7$	0	1	1	0
$\mathcal{A}_8$	1	1	1	0

	HP7_e	HP7_d	K23_e	K23_d
$\mathcal{A}_9$	0	0	0	1
$\mathcal{A}_{10}$	1	0	0	1
$\mathcal{A}_{11}$	0	1	0	1
$\mathcal{A}_{12}$	1	1	0	1
$\mathcal{A}_{13}$	0	0	1	1
$\mathcal{A}_{14}$	1	0	1	1
$\mathcal{A}_{15}$	0	1	1	1
$\mathcal{A}_{16}$	1	1	1	1

Die Formelmenge {HP7_e, HP7_d, K23_d , ¬K23_e} hat hier ein Modell: $\mathcal{A}_{12}$

Die Formelmenge {HP7_e, HP7_d, K23_d} hat hier zwei Modelle: $\mathcal{A}_{12}$, $\mathcal{A}_{16}$

{(HP7_e ∨ HP7_d), (K23_d ∧ ¬K23_e)} hat hier drei Modelle: $\mathcal{A}_{10}$, $\mathcal{A}_{11}$, $\mathcal{A}_{12}$

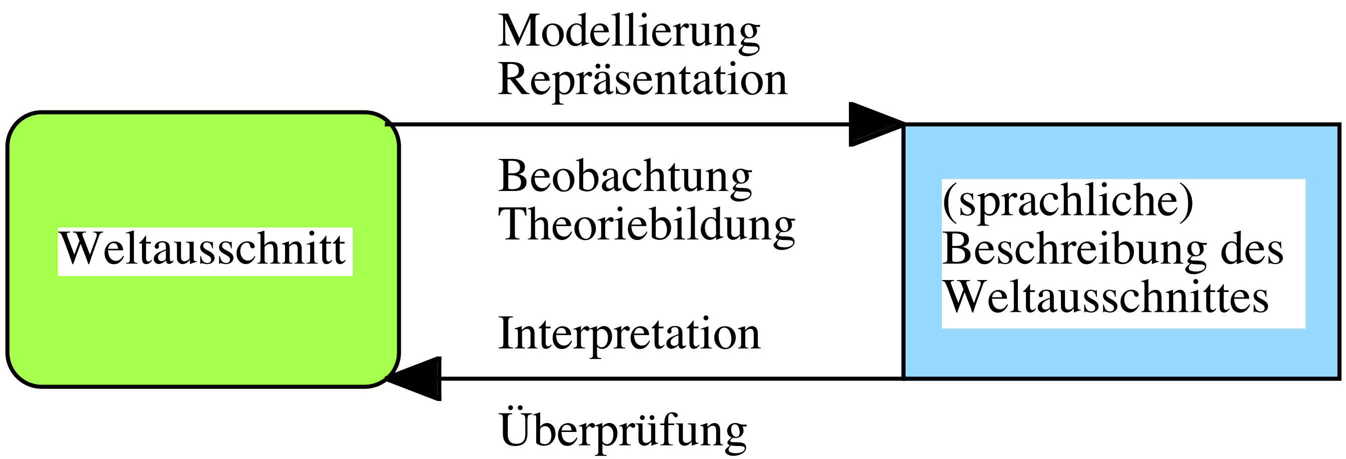
Begriffe der Metasprache: Semantik (1)

Belegung

- Wahrheitswertzuordnung zu Formeln ➔ eine Zeile der Wahrheitstafel

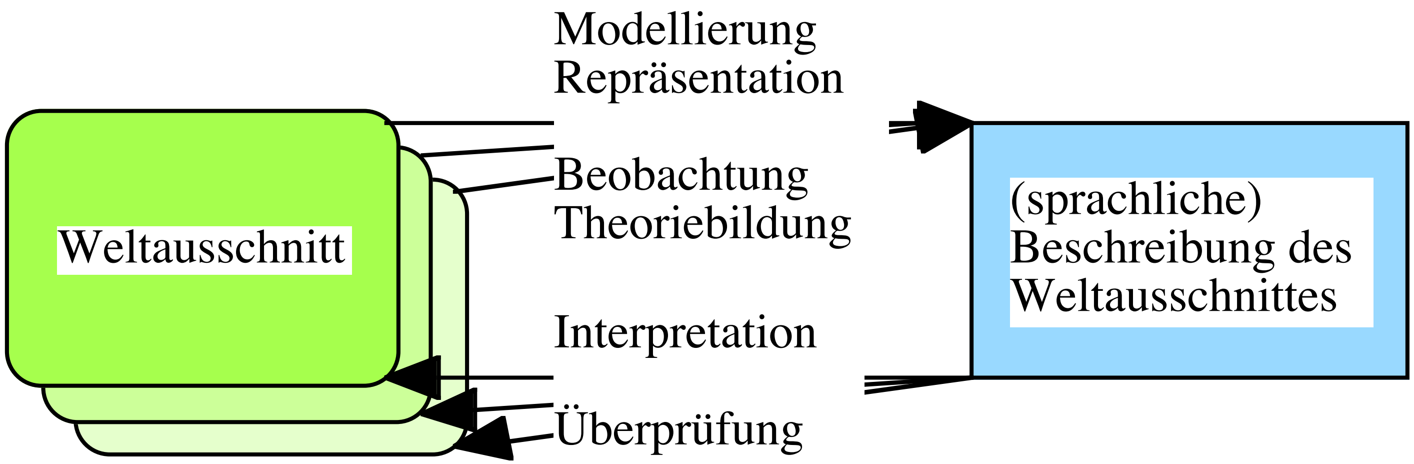
Modell einer Formel

- Belegung, die die Formel wahr macht ➔ eine Zelle der Wahrheitstafel



Wahrheitswertverlauf einer Formel

- Die Wahrheitswerte der Belegungen → eine Spalte der Wahrheitstafel



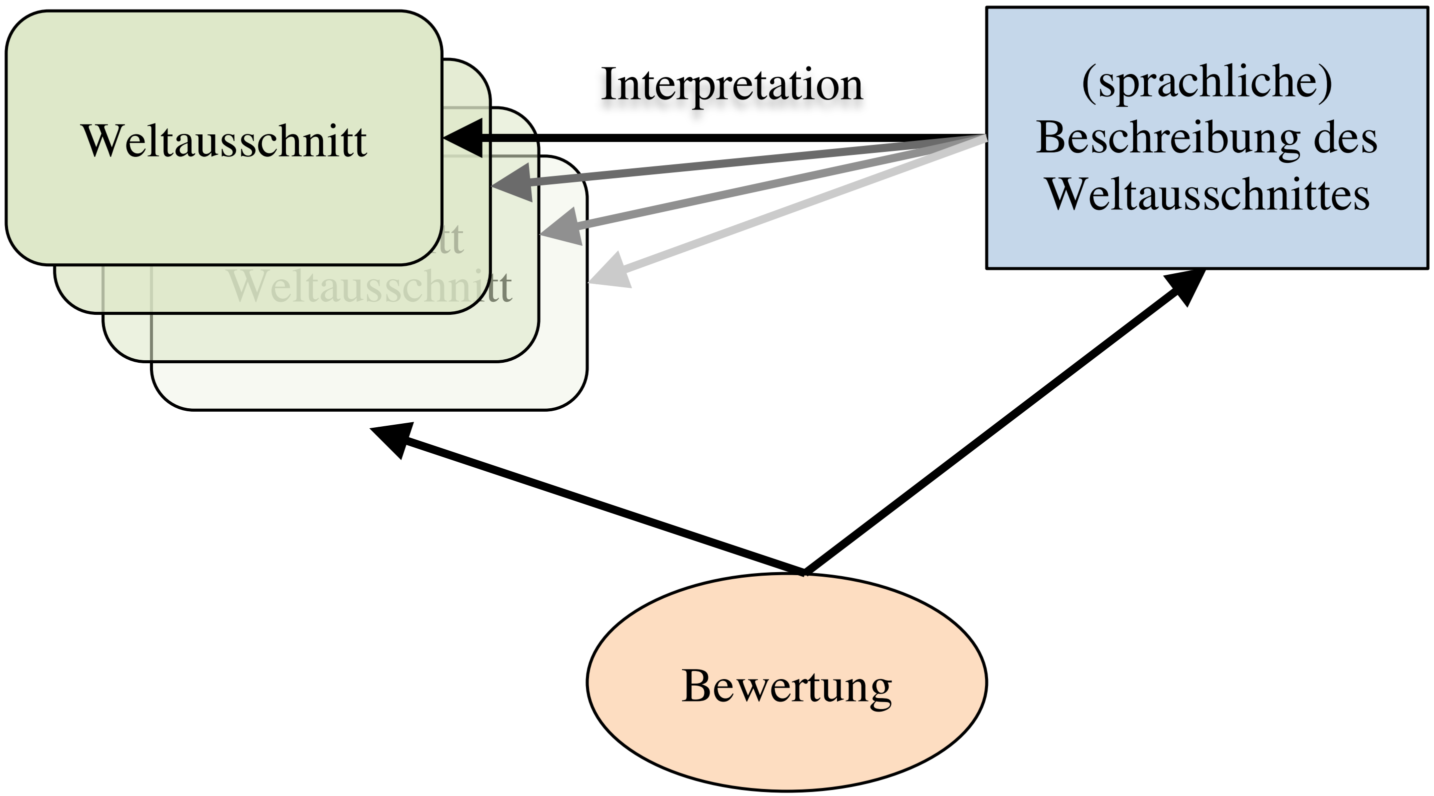
- Formeln und Formelmengen können verschiedene Modelle haben.

- Mathematische Abstraktion: Interpretation in einem Weltausschnitt als Belegung von Aussagensymbolen.

Begriffe der Metasprache: Semantik (2)

Erfüllbarkeit – Unerfüllbarkeit – Gültigkeit – Kontingenz

- Eigenschaften von Formeln, die über ihre Wahrheitswertverlauf definiert sind.
- Klassifikation einer Formel gemäß der zugehörigen Spalte in der Wahrheitstafel.



Verhalten von Formeln (Formelmengen) unter beliebigen Belegungen

Definition 3.4

- Eine Formel F heißt genau dann *erfüllbar*, falls ein Modell für F existiert, anderenfalls heißt F *unerfüllbar* (*Kontradiktion*)
- Eine Formel F heißt genau dann *falsifizierbar*, falls eine falsifizierende Belegung für F existiert.
- Eine Formelmenge **M** heißt genau dann *erfüllbar*, falls ein Modell für **M** existiert, andernfalls heißt sie *unerfüllbar*.

Beispiele 3.4.1

- Jedes Aussagensymbol ist erfüllbar und falsifizierbar. z.B. $\mathcal{A}_1(A) = \mathbf{1}$, $\mathcal{A}_2(A) = \mathbf{0}$
- $(A \wedge B) \Rightarrow (A \vee \neg C)$ ist erfüllbar. z.B. $\mathcal{A}_1(A) = \mathbf{1}$, $\mathcal{A}_1(B) = \mathbf{0}$, $\mathcal{A}_1(C) = \mathbf{0}$
- $A \wedge \neg A$ ist unerfüllbar.
- $\{A \vee B, A \wedge B\}$ ist erfüllbar. z.B. $\mathcal{A}_3(A) = \mathbf{1}$, $\mathcal{A}_3(B) = \mathbf{1}$
- $\{A, \neg A\}$ ist unerfüllbar, obwohl die Formeln A und $\neg A$ erfüllbar sind.

Zum Selbststudium

Die leere (Formel-)Menge

- Es hat sich eingebürgert, die Definitionen, die über Formelmengen reden, wie folgt auf die leere (Formel-)Menge anzuwenden:
  - Jede Belegung ist Modell der leeren Menge.
  - Die leere Menge ist *erfüllbar*.

Unerfüllbare und allgemeingültige Formel(menge)n

- Eine unerfüllbare Formel(menge) kann keinen Weltausschnitt korrekt beschreiben.
- Eine allgemeingültige Formel beschreibt jeden Weltausschnitt unter jeder Interpretation korrekt.
  - ➔ Unerfüllbare und allgemeingültige Formeln können nicht dazu dienen einen Weltausschnitt oder eine Interpretation von dem/der anderen zu unterscheiden.

Allgemeingültigkeit – Kontingenz

Definitionen 3.5

- Eine Formel F heißt genau dann *allgemeingültig*, falls jede Belegung $\mathcal{A}$ ein Modell für F ist.
- Weitere Sprechweisen / Schreibweisen für „F ist allgemeingültig“:
  - F ist *gültig*
  - F ist eine *Tautologie*
  - $\models$ F
- Eine Formel F heißt genau dann *kontingent*, wenn sie erfüllbar und falsifizierbar ist (d.h. falls F ein Modell $\mathcal{A}$ hat, aber es auch Belegung $\mathcal{A}'$ gibt, die kein Modell für F ist).

Beispiele

- Aussagensymbole sind kontingente Formeln
- $A \lor \neg B$ ist kontingent $\mathcal{A}_3(A) = \mathbf{1}$, $\mathcal{A}_3(B) = \mathbf{1}$, dann $\mathcal{A}_3(A \lor \neg B) = \mathbf{1}$

  $\mathcal{A}_2(A) = \mathbf{0}$, $\mathcal{A}_2(B) = \mathbf{1}$, dann $\mathcal{A}_2(A \lor \neg B) = \mathbf{0}$
- $A \lor \neg A$ ist allgemeingültig

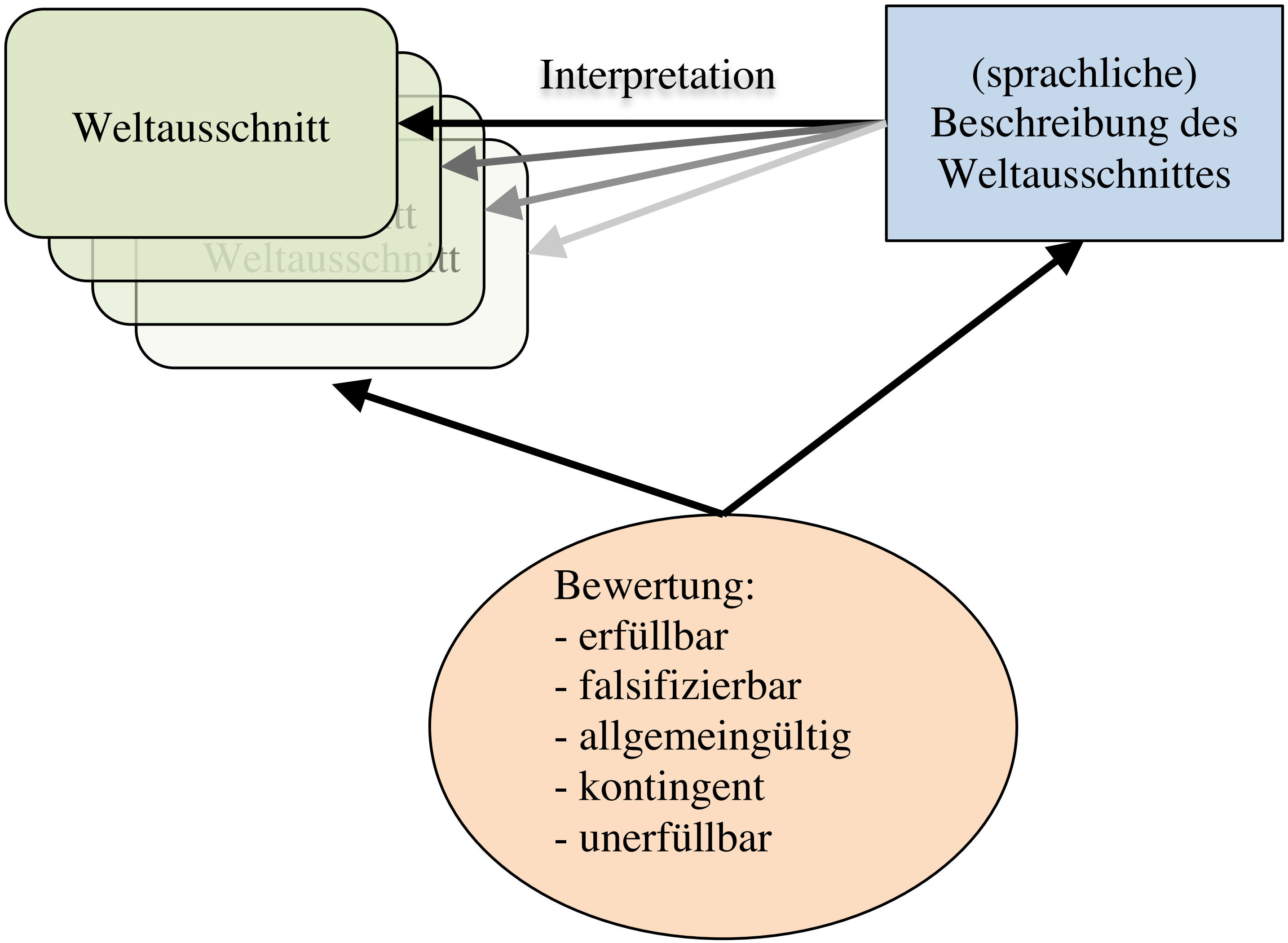
Allgemeingültigkeit – Kontingenz – Unerfüllbarkeit

(Aussagen-)Logische Formeln

<table>
<tr><td colspan="2">erfüllbare Formeln</td><td rowspan="2">unerfüllbare Formeln
Kontradiktionen</td></tr>
<tr><td rowspan="2">(allgemein-) gültige Formeln
Tautologien</td><td>kontingente Formeln</td></tr>
<tr><td colspan="2">falsifizierbare Formeln</td></tr>
<tr><td>wahr unter allen Belegungen
Wahrheitswertverlauf konstant 1
jede Belegung ist ein Modell</td><td>wahr / falsch unter manchen Belegungen
Wahrheitswertverlauf nicht konstant
es gibt ein Modell und eine falsifizierende Belegung</td><td>falsch unter allen Belegungen
Wahrheitswertverlauf konstant 0
es existiert kein Modell</td></tr>
</table>

- Die allgemeingültigen und die unerfüllbaren Formeln sind *logisch interessant*, da sie sich unter jeder beliebigen Belegung gleich verhalten.
- Die kontingenten Formeln sind *epistemisch interessant*, da sie Aussagen über die Welt machen.

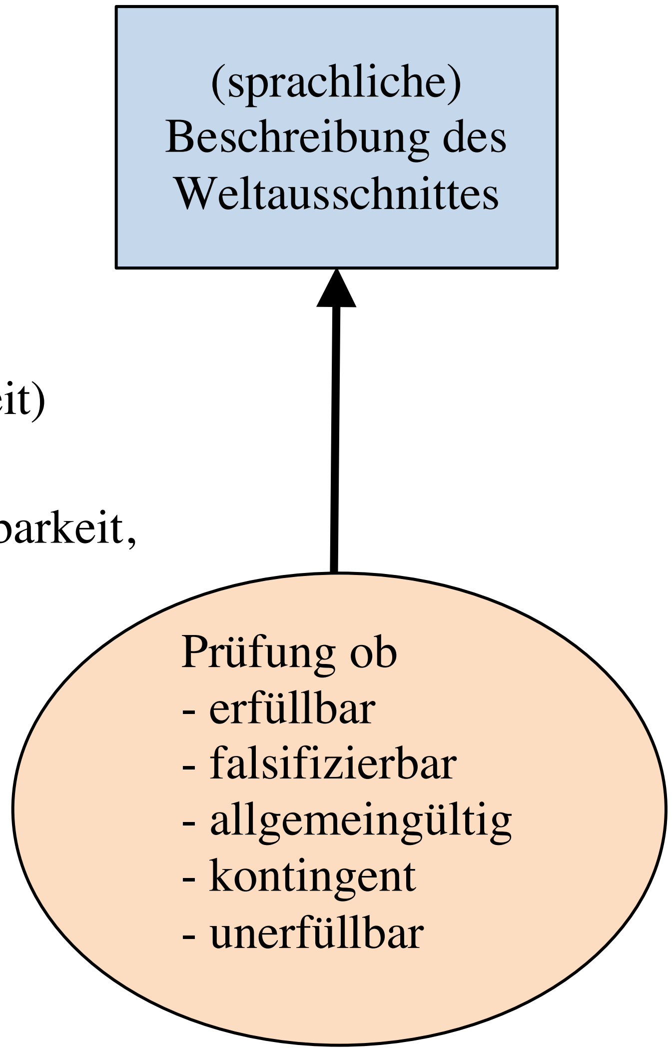
Allgemeingültigkeit – Kontingenz – Unerfüllbarkeit



Verfahren zur Klassifikation von Formeln

Wie kann man beliebige Formeln oder Fomelmengen automatisch semantisch bewerten (klassifizieren)

- Wie laufen die Verfahren?
- Können wir uns auf das Ergebnis verlassen? (Korrektheit)
- Wie viele Verfahren benötigen wir? (Reduzierbarkeit)
- Wie aufwendig ist die Prüfung mit den Verfahren?
- Wie aufwendig ist das beste Verfahren? (Entscheidbarkeit, Komplexität)



Prüfung auf Allgemeingültigkeit, Erfüllbarkeit, Unerfüllbarkeit

- Für eine Formel F mit n Aussagensymbolen kann eine komplette Wahrheitstafel berechnet werden.
  - F ist *erfüllbar*, falls mindestens eine Belegung den Wahrheitswert **1** für F ergibt.
  - F ist *allgemeingültig*, falls alle Belegungen den Wahrheitswert **1** für F ergeben.
  - F ist *falsifizierbar*, falls mindestens eine Belegung den Wahrheitswert **0** für F ergibt.
  - F ist *unerfüllbar*, falls keine Belegung den Wahrheitswert **1** für F ergibt.

Ein einfaches Verfahren

- zur Prüfung dieser Eigenschaften basiert also auf der Erstellung und Auswertung von Wahrheitstafeln
- Die Tafeln können aber sehr groß werden.

Verallgemeinerbarkeit ist aber beschränkt

- nicht direkt anwendbar für
  - (Un)erfüllbarkeitsprüfung von (unendlichen) Formelmengen
  - kompliziertere Logiken (Prädikatenlogik)

Zur Erinnerung: Ein erster Algorithmus-Begriff SE1 - Level 1 Folie 76

- Wir verstehen unter einem Algorithmus einen **endlichen Text**, in dem ein für einen Prozessor (Interpreter) eindeutiges allgemeines und schrittweises **Problemlösungsverfahren** aus **Aktionen**, die auf **sprachlichen Einheiten** arbeiten, beschrieben ist.
- Ein Algorithmus terminiert, wenn er nicht nur in einer endlichen Vorschrift beschrieben ist, sondern auch nach endlich vielen Schritten seine Bearbeitung beendet.

[Formulierung und Hervorhebung: SE1 - Level 1 Folie 76]

Zwei ergänzende Anmerkungen zur Termination:

- Es ist wichtig, dass ein Algorithmus für jede Eingabe (für die der Algorithmus konzipiert ist) nach endlich vielen Verarbeitungsschritten stoppt.
- Die Eigenschaft eines Berechnungsverfahrens zu terminieren, sollte bewiesen werden.

Informelle Beschreibung des Begriffs "Algorithmus"

(1) Ein Algorithmus wird durch eine endliche Menge von Instruktionen endlicher Größe gegeben.

(2) Es existiert ein berechnender Agent (Mensch oder Maschine), der auf der Grundlage Instruktionen die Berechnungen durchführen kann.

(3) Es existieren Einrichtungen (Komponenten) für die Durchführung, Speicherung und den Wiederabruf von Berechnungsschritten.

(4) Sei *ALG* ein Algorithmus entsprechend (1) und *AG* ein berechnender Agent im Sinne von (2). Dann führt *AG* die Berechnungen auf der Grundlage von *ALG* so durch, dass für jede gegebene Eingabe (Aufgabenstellung) die Berechnung in diskreten Berechnungsschritten, d.h. ohne die Methode von kontinuierlichen Methoden oder analogen Verfahren, erfolgt.

(5) *AG* führt die Berechnungen auf der Grundlage von *ALG* deterministisch durch, d.h. ohne Verwendung von Zufallsmethoden oder –hilfsmitteln (etwa einem Würfel, etc.).

[Charakterisierung folgt: Rogers 1987]

Entscheidungsprozeduren

Definition

Sei $\mathcal{P}$ eine Menge von Formeln, d.h. $\mathcal{P} \subseteq \mathcal{L}_{\mathrm{AL}}$. Ein Algorithmus A_{EP} ist eine Entscheidungsprozedur für $\mathcal{P}$, falls für jede Formel $\mathsf{F} \in \mathcal{L}_{\mathrm{AL}}$ gilt:

- der Algorithmus A_{EP} terminiert,
- A_{EP} liefert die Antwort 'JA' nur wenn $\mathsf{F} \in \mathcal{P}$ (Korrektheit),
- A_{EP} liefert die Antwort 'JA' immer wenn $\mathsf{F} \in \mathcal{P}$ (Vollständigkeit)
- A_{EP} liefert die Antwort 'NEIN' sonst, also wenn $\mathsf{F} \notin \mathcal{P}$.

Anmerkungen:

- der Algorithmus A_{EP} liefert für jede Formel eine Antwort, er "entscheidet" zwischen den $\mathcal{P}$-Fällen und den Nicht-$\mathcal{P}$-Fällen.
- Entsprechend zur obigen Definition kann auch für andere Wortmengen, d.h. andere Sprachen, die Frage der Entscheidbarkeit gestellt werden.
- Das Thema "Entscheidbarkeit" betrifft insbesondere die Frage, welche Probleme *entscheidbar* und welche *unentscheidbar* sind. [Unentscheidbare Probleme sind solche, für die es keine Entscheidungsprozeduren gibt.]

Zum Selbststudiumg: Literaturhinweise zu *Algorithmen & Entscheidbarkeit*

- Auf die folgenden Bücher, die in Veranstaltungen des Master-Studiums für Sie interessant werden können, wurde in den letzten Folien Bezug genommen:
  - Rogers, Hartley (1987). *Theory of recursive functions and effective computability*. Cambridge, MA: MIT-Press.
  - Ben-Ari, Mordechai (2001). *Mathematical logic for computer science*. London: Springer-Verlag. (second edition).

- Das Thema "Entscheidbarkeit" wird im weiteren Verlauf von FGI-1 noch vertieft behandelt werden.

Allgemeingültigkeit – Unerfüllbarkeit

Satz 3.1: Eine Formel F ist genau dann eine Tautologie, wenn ¬F unerfüllbar ist.

Voraussetzungen: Definitionen 3.1, 3.3, 3.4, 3.5, F sei eine Formel.

Bew.: F ist eine Tautologie

GDW. jede Belegung $\mathcal{A}$ Modell für F ist, d.h. $\mathcal{A}(F) = 1$ [Def. Tautologie]

GDW. jede Belegung $\mathcal{A}$ ¬F falsch macht, d.h. $\mathcal{A}(\neg F) = 0$ [Def. 3.1]

GDW. ¬F besitzt kein Modell [Def. Modell]

GDW. ¬F ist unerfüllbar [Def. unerfüllbar]

Satz 3.2: Eine Formel F ist genau dann unerfüllbar, wenn ¬F eine Tautologie ist.

Voraussetzungen und Bew.: Analog zum vorherigen.

➔ Zu jeder allgemeingültigen Formel existiert eine korrespondierende unerfüllbare Formel und umgekehrt.

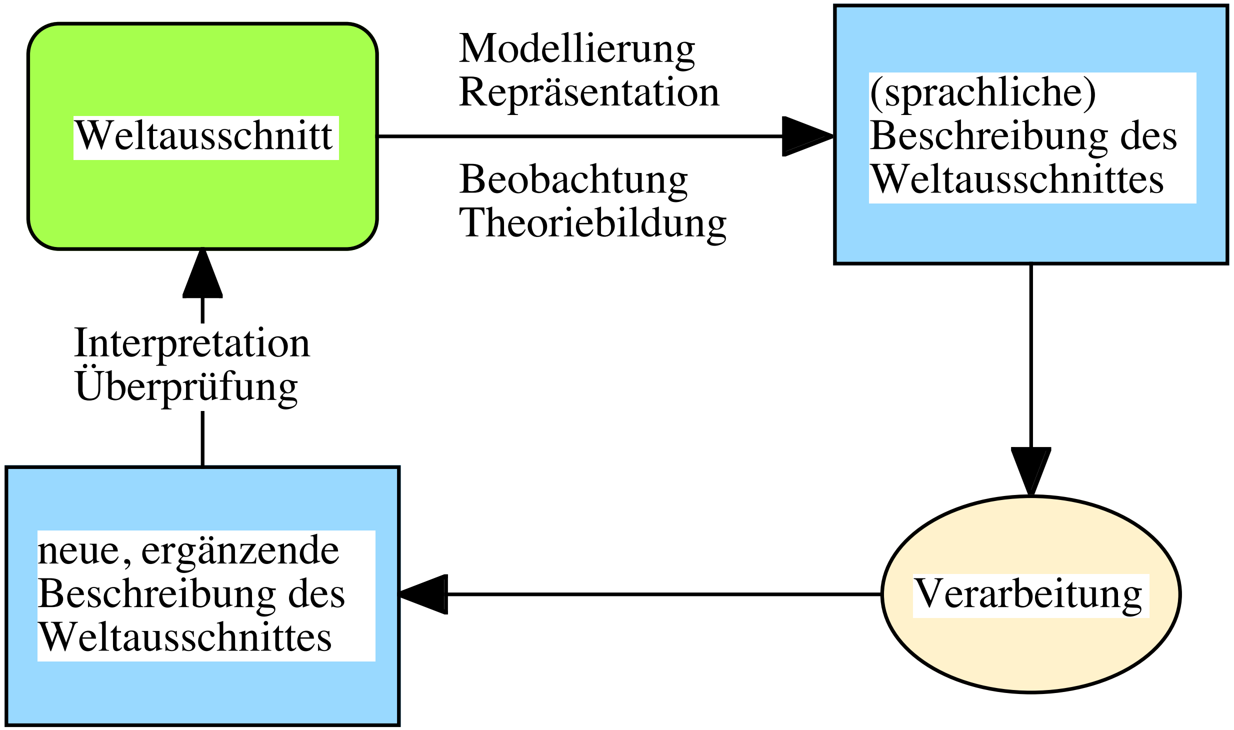
allgemeingültig		**unerfüllbar**
$A \vee \neg A$	$\rightarrow$	$\neg(A \vee \neg A)$
$\neg(A \wedge \neg A)$	$\leftarrow$	$A \wedge \neg A$

➔ Jedes Verfahren, das Allgemeingültigkeit einer Formel prüfen kann, kann ich mit geringem Zusatzaufwand verwenden, um Unerfüllbarkeit einer Formel zu prüfen, und umgekehrt.

So geht es weiter: Begriffe der Metasprache: Semantik (3)

Äquivalenz – Folgerung

- Beziehungen zwischen Formeln, die über ihre Wahrheitswertverläufe definiert sind
- Vergleich zweier Spalten in der Wahrheitstafel.



Erfüllbarkeit – Unerfüllbarkeit – Folgerung

- Auch für unendliche Formelmengen definiert.

Wichtige Konzepte in diesem Foliensatz

- Wahrheitswert, Belegung, Wahrheitstafel
- Wahrheitswertverlauf
- Modell
- Erfüllbarkeit, Falsifizierbarkeit, Kontingenz
- Unerfüllbarkeit / Kontradiktion, Allgemeingültigkeit / Tautologie
- Entscheidungsverfahren

Aussagenlogik: Äquivalenz – Normalformen – Formelklassen

Algorithmische Verfahren zur Prüfung von

Allgemeingültigkeit, Erfüllbarkeit, Unerfüllbarkeit

- existieren,
- können aber sehr aufwendig sein.

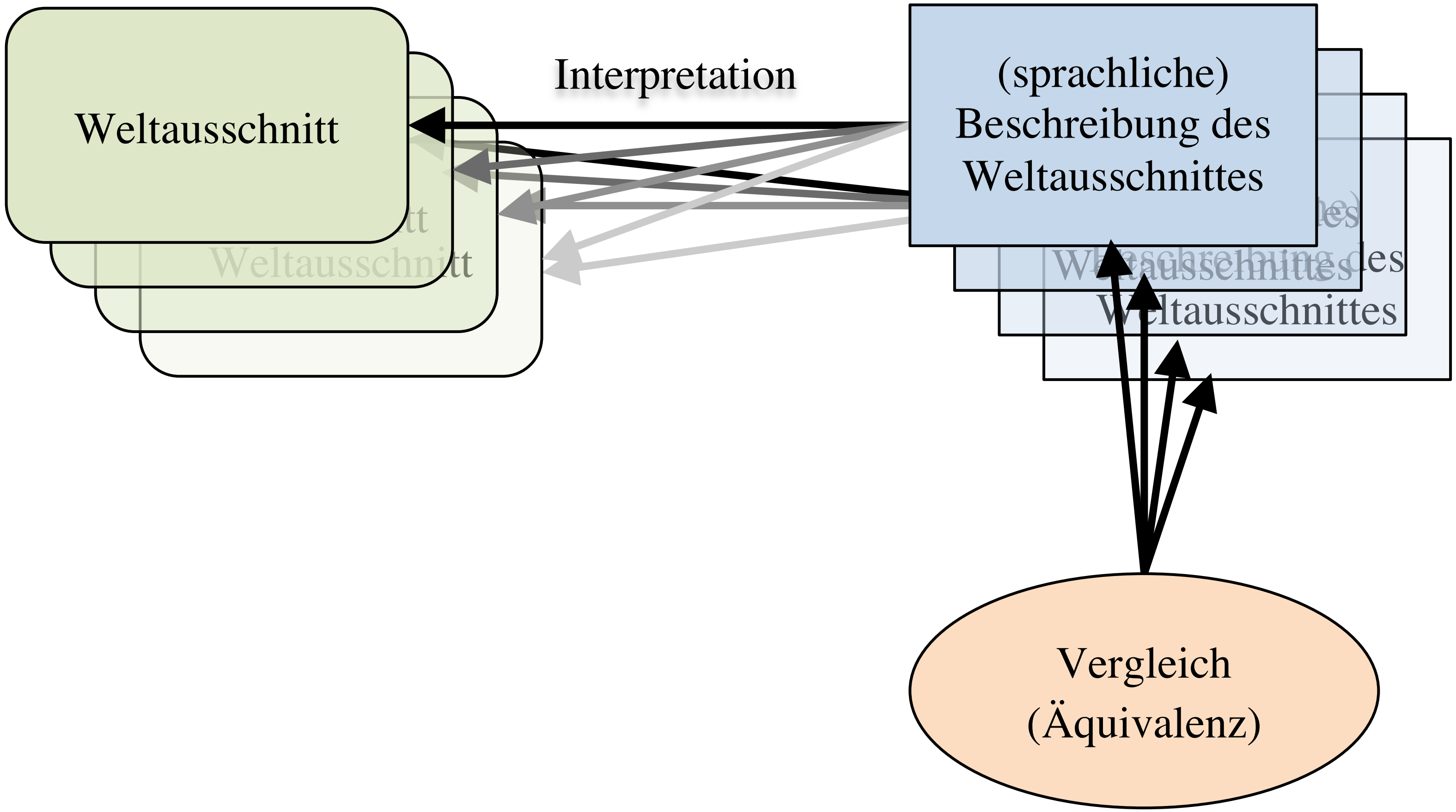
➔ Die Lösung des Erfüllbarkeitsproblems (‚Ist diese Formelmenge erfüllbar?‘) erfordert exponentielle Rechenzeit

➔ Komplexitätstheoretische Untersuchungen ➔ spätere Kapitel von FGI-1

Formeln spezifischer Form: Normalformen

- *Normalformen*
  - sind Formeln mit bestimmter Form, also Elemente einer systematisch festgelegten Teilmenge von $\mathcal{L}_{AL}$
  - ermöglichen Verfahren, die einfach(er) zu entwerfen und programmieren sind.
- *Ersetzungsregeln* erzeugen zu gegebenen Formeln äquivalente Formeln in Normalform.

Vergleich alternativer Beschreibungen



Logische Äquivalenz

Definition 4.1

Zwei Formeln F und G heißen genau dann *(logisch / semantisch)* *äquivalent*, falls jede Belegung $\mathcal{A}$ beiden Formeln denselben Wahrheitswert zuordnet ($\mathcal{A}$(F) = $\mathcal{A}$(G))
Dies wird als F ≡ G geschrieben.

- ➔ Formeln sind genau dann äquivalent, wenn sie denselben Wahrheitswertverlauf haben.
- ➔ Zwei Formeln F und G sind genau dann äquivalent, wenn sie dieselben Modelle besitzen (also für alle Belegungen gilt $\mathcal{A}$(F) = **1** GDW. $\mathcal{A}$(G) = **1**).
- ➔ Beispiel: Die Formeln (¬A ∨ B) und (A ⇒ B) sind äquivalent. (s. S. 3-[18])

Satz 4.2: Äquivalenz mit Tautologien

Ist eine Formel F äquivalent mit einer Tautologie G, dann ist F auch eine Tautologie.

Voraussetzungen: Def. 3.1, 3.5, 4.1

Bew.: Es sei G eine Tautologie und F eine zu G äquivalente Formel.
Da G eine Tautologie ist, sind alle Belegungen Modelle von G.
Da F und G äquivalent sind, sind dann auch alle Belegungen Modelle von F.
Also ist F eine Tautologie.

Satz 4.3: Äquivalenz mit Kontradiktionen

Ist eine Formel F äquivalent mit einer Kontradiktion G, dann ist F auch eine Kontradiktion.

Bew.: Analog zum Beweis von Satz 4.2 unter Bezug auf Def. 3.4.

Satz 4.4: Äquivalenz von Tautologien

Alle Tautologien sind äquivalent. ($\models F$ und $\models G$, dann auch $F \equiv G$)

Voraussetzungen: Def. 3.1, 3.5, 4.1

Bew.: Es seien F und G zwei Tautologien
Damit ist jede Belegung Modell beider Formeln.
Also haben F und G dieselben Modelle und sind damit äquivalent.

Satz 4.5: Äquivalenz von Kontradiktionen

Alle Kontradiktionen sind äquivalent.

Bew.: Analog zum Beweis von Satz 4.4 unter Bezug auf Def. 3.4.

Bemerkung

Kontingente Formeln können äquivalent miteinander sein, müssen aber nicht.

Äquivalenz und Biimplikation

Das Symbol für Äquivalenz $\equiv$ ist kein Junktor, aber eng mit der Biimplikation verwandt.

Satz 4.6: F und G sind genau dann äquivalent, wenn $F \Leftrightarrow G$ allgemeingültig ist

$F \equiv G$ GDW. $\models F \Leftrightarrow G$.

Voraussetzungen: Def. 3.1, 3.5, 4.1

Bew.: Es seien F und G zwei Formeln.

F und G sind äquivalent ($F \equiv G$)

GDW. Für alle Belegungen $\mathcal{A}$ gilt: $\mathcal{A}(F) = \mathcal{A}(G)$

GDW. (s. Verknüpfungstafel für $\Leftrightarrow$)

alle Belegungen $\mathcal{A}$ Modelle von $(F \Leftrightarrow G)$ sind,

d.h. $\mathcal{A}((F \Leftrightarrow G)) = \mathbf{1}$

GDW. $F \Leftrightarrow G$ ist allgemeingültig ($\models F \Leftrightarrow G$)

Zum Selbststudium

Wozu nützen uns die obigen Sätze?

- Die Definitionen von Allgemeingültigkeit, (Un-)Erfüllbarkeit und Äquivalenz suggerieren drei verschiedene Verfahren, um zu überprüfen, ob eine Formel F allgemeingültig oder unerfüllbar ist, bzw. ob zwei Formeln F und G äquivalent sind.
- Die Sätze 3.2 und 4.6 zeigen aber, dass es ausreichend ist, ein Verfahren zu haben, mit dem man Allgemeingültigkeit einer Formel H prüft.
  - Will man wissen, ob F (un-)erfüllbar ist,
    dann wendet man das Verfahren einfach auf $H = \neg F$ an.
  - Will man wissen, ob F und G äquivalent ist,
    dann wendet man das Verfahren einfach auf $H = F \Leftrightarrow G$ an
- Die Sätze 3.1 und 4.6 reduzieren entsprechend die Frage nach Allgemeingültigkeit und Äquivalenz auf die Frage der Unerfüllbarkeit.
- Die Sätze 4.2 und 4.3 zeigen, dass es ausreichend ist, eine Tautologie und eine Kontradiktion zu kennen und ein Verfahren zu haben, mit dem man Äquivalenz von zwei Formeln prüft.
- Die Unerfüllbarkeit von **unendlichen** Formelmengen ist mit Verfahren für Tautologie- und Äquivalenzprüfung aber nicht erfassbar.

Wichtige Äquivalenzen (Satz 4.7)

Es seien F, G und H beliebige Formeln:

Elimination von $\Leftrightarrow$	$(F \Leftrightarrow G) \equiv (F \Rightarrow G) \wedge (G \Rightarrow F)$ $(F \Leftrightarrow G) \equiv (F \wedge G) \vee (\neg G \wedge \neg F)$	
Elimination von $\Rightarrow$	$(F \Rightarrow G) \equiv (\neg F \vee G)$	
doppelte Negation	$\neg\neg F \equiv F$	
Assoziativität	$((F \wedge G) \wedge H) \equiv (F \wedge (G \wedge H))$ $((F \vee G) \vee H) \equiv (F \vee (G \vee H))$	
Distributivität	$(F \wedge (G \vee H)) \equiv ((F \wedge G) \vee (F \wedge H))$ $(F \vee (G \wedge H)) \equiv ((F \vee G) \wedge (F \vee H))$	
Kommutativität	$(F \wedge G) \equiv (G \wedge F)$	$(F \vee G) \equiv (G \vee F)$
de Morgansche Regeln	$\neg(F \wedge G) \equiv (\neg F \vee \neg G)$	$\neg(F \vee G) \equiv (\neg F \wedge \neg G)$
Absorption	$(F \wedge (F \vee G)) \equiv F$	$(F \vee (F \wedge G)) \equiv F$
Idempotenz	$(F \wedge F) \equiv F$	$(F \vee F) \equiv F$
Tautologieregeln wenn F eine Tautologie	$(F \wedge G) \equiv G$	$(F \vee G) \equiv F$
Unerfüllbarkeitsregeln wenn F unerfüllbar	$(F \wedge G) \equiv F$	$(F \vee G) \equiv G$

Anwendung der Wahrheitstafelmethode

- Distributivität: $(F \lor (G \land H)) \equiv ((F \lor G) \land (F \lor H))$

	F	G	H	$G \land H$	$F \lor (G \land H)$	$F \lor G$	$F \lor H$	$(F \lor G) \land (F \lor H)$
$\mathcal{A}_1$	0	0	0	0	0	0	0	0
$\mathcal{A}_2$	0	0	1	0	0	0	1	0
$\mathcal{A}_3$	0	1	0	0	0	1	0	0
$\mathcal{A}_4$	0	1	1	1	1	1	1	1
$\mathcal{A}_5$	1	0	0	0	1	1	1	1
$\mathcal{A}_6$	1	0	1	0	1	1	1	1
$\mathcal{A}_7$	1	1	0	0	1	1	1	1
$\mathcal{A}_8$	1	1	1	1	1	1	1	1

- Tautologieregeln: Falls F eine Tautologie, dann $(F \land G) \equiv G$ und $(F \lor G) \equiv F$

	F	G	$F \land G$	$F \lor G$
$\mathcal{A}'_1$	1	0	0	1
$\mathcal{A}'_2$	1	1	1	1

Äquivalenzregeln dienen der syntaktischen Vereinfachung

„Überflüssige" Teilformeln

- Mehrfach auftretende Teilformeln können in bestimmten Fällen reduziert werden

➔ Idempotenz: $(F \wedge F) \equiv F$, $(F \vee F) \equiv F$

➔ Absorption: $(F \wedge (F \vee G)) \equiv F$, $(F \vee (F \wedge G)) \equiv F$

- Allgemeingültige oder unerfüllbare Teilformeln machen bestimmte Teilformeln überflüssig:

➔ Tautologieregel: Falls F eine Tautologie: $(F \vee G) \equiv F$, $(F \wedge G) \equiv G$

➔ Unerfüllbarkeitsregel: Falls F unerfüllbar: $(F \vee G) \equiv G$, $(F \wedge G) \equiv F$

Weitere Klammerersparnisregel: Nutzung der Assoziativität

- Die interne Klammerung bei mehrfache Konjunktion und mehrfache Disjunktion drückt keine Wahrheitswert-relevanten Unterschied aus (➔ Assoziativität)

  $(F \wedge G \wedge H)$ können wir verwenden, wenn $((F \wedge G) \wedge H)$ und $(F \wedge (G \wedge H))$ gleichermaßen gemeint sind.

  Aber: $(F \Rightarrow G \Rightarrow H)$ und $(F \Leftrightarrow G \Leftrightarrow H)$ sind weiterhin keine Formeln.

Der nächste Schritt

- Wir zeigen, dass die Äquivalenzen auch für Teilformeln relevant sind.

Ersetzung von Formeln

Definition 4.8 (Ersetzung)

Seien F und G zwei Formeln und sei H eine Formel mit der Teilformel F und H' eine Formel mit Teilformel G. Wenn der einzige Unterschied zwischen H und H' darin besteht, dass an einer oder mehreren Positionen, an denen in H die Teilformel F steht in H' die Teilformel G steht, dann heißt H' *durch Ersetzung von* F *durch* G *aus* H *hervorgegangen*.

- Beispiel:

  Ersetzungen von F := $\neg A$ durch G := $(B \vee C)$

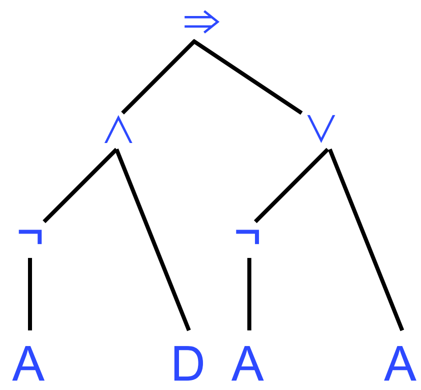
  (0) $(\neg A \wedge D) \Rightarrow (\neg A \vee A)$ = H

  (1) $((B \vee C) \wedge D) \Rightarrow (\neg A \vee A)$ = H'1

  (2) $(\neg A \wedge D) \Rightarrow ((B \vee C) \vee A)$ = H'2

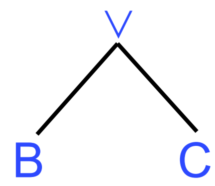
  (3) $((B \vee C) \wedge D) \Rightarrow ((B \vee C) \vee A)$ = H'3

  ! Ersetzung von zwei Vorkommen von $\neg A$!



- Keine Ersetzung von F durch G:

  (4) $(\neg A \wedge D) \Rightarrow (\neg A \vee \neg(B \vee C))$



Ersetzbarkeitstheorem

Satz 4.9

- Es seien F und G äquivalente Formeln.
  Wenn H eine Formel mit Teilformel F ist und H' eine Formel ist, die durch Ersetzung von F durch G aus H hervorgeht, dann sind H und H' äquivalent.

Voraussetzungen: Def. 2.2, 2.3, 3.1, 4.1, 4.8; Satz 2.10; F und G sind äquivalente Formeln

Bew.: (folgt dem Prinzip der strukturellen Induktion über den Aufbau von H)

Induktionsanfang

Ist H eine atomare Formel und H' eine Formel, die durch Ersetzung von F durch G aus H hervorgeht, dann muss F = H und H' = G sein (Def. 2.3, 4.8).

Es gilt $H \equiv H'$, wegen $H = F \equiv G = H'$.

Induktionsannahme

Wir nehmen an, dass H1 und H2 Formeln sind, für die gilt:

Für jede Formel H1' bzw. H2', die aus H1 bzw. H2 durch Ersetzung von F durch G hervorgegangen ist, gilt: $H1 \equiv H1'$ bzw. $H2 \equiv H2'$.

Ersetzbarkeitstheorem – Der Induktionsschritt (1)

Induktionsschritt

Zu zeigen: Wenn H eine der Formeln ¬H1, (H1 ∨ H2), (H1 ∧ H2), (H1 ⇒ H2) oder (H1 ⇔ H2) ist und H' eine Formel ist, die durch Ersetzung von F durch G aus H hervorgeht, dann sind H und H' äquivalent.

Wenn F = H, dann gilt, wie beim Induktionsanfang, H = F ≡ G = H'.

Für alles weitere betrachten wir den Fall, dass F ist eine echte Teilformel von H ist.

Ist H = ¬H1,

- dann ist F eine Teilformel von H1 und gibt es eine Formel H1', die aus H1 durch Ersetzung von F durch G hervorgegangen ist, so dass H' = ¬H1' (Def. 4.8)
- Nach Induktionsannahme gilt H1 ≡ H1'
  - Sei $\mathcal{A}$ eine beliebige Belegung. Dann gilt:
    $\mathcal{A}$(H1') = $\mathcal{A}$(H1), und wegen Def. 3.1:
    $\mathcal{A}$(¬H1') = $\mathcal{A}$(¬H1)
    also auch $\mathcal{A}$(H') = $\mathcal{A}$(¬H1') = $\mathcal{A}$(¬H1) = $\mathcal{A}$(H)
- und damit H' ≡ H. (Def. 4.1)

Ersetzbarkeitstheorem – Der Induktionsschritt (2)

Ist $H = (H1 \vee H2)$,

- dann ist F eine Teilformel von $H1$ oder von $H2$ (oder beiden) und es gibt Formeln $H1'$ und $H2'$, die durch Ersetzung von F durch G aus $H1$ bzw. $H2$ hervorgegangen oder mit $H1$ bzw. $H2$ identisch sind, so dass $H' = (H1' \vee H2')$ (Def. 4.8)
- Nach Induktionsannahme gilt $H1 \equiv H1'$ und $H2 \equiv H2'$
  - Sei $\mathcal{A}$ eine beliebige Belegung. Dann gilt:
    $\mathcal{A}(H1') = \mathcal{A}(H1)$ und $\mathcal{A}(H2') = \mathcal{A}(H2)$, und wegen Def. 3.1 gilt:
    $\mathcal{A}(H') = \mathcal{A}((H1' \vee H2')) = \mathcal{A}((H1 \vee H2)) = \mathcal{A}(H)$
- also gilt insgesamt $H' \equiv H$. (Def. 4.1)

Die Beweise für Formeln mit den Hauptoperatoren $\wedge$, $\Rightarrow$, $\Leftrightarrow$ laufen entsprechend.

Resümee: Nach dem Prinzip der strukturellen Induktion gibt es also: Wenn F und G äquivalente Formeln sind, H eine Formel ist und H' eine Formel ist, die aus H durch Ersetzung von F durch G hervorgeht, dann sind H und H' äquivalent.

Elimination von $\Leftrightarrow$

Satz 4.10

Zu jeder Formel gibt es eine äquivalente Formel, in der der Junktor $\Leftrightarrow$ nicht vorkommt.

Vor.: Def. 2.1, 2.2, Satz 2.10, 4.7, 4.9

Bew.: (folgt dem Prinzip der strukturellen Induktion)

Induktionsanfang

In atomaren Formeln kommt $\Leftrightarrow$ nicht vor. Also erfüllen atomare Formeln die Bedingung automatisch selbst.

Induktionsannahme

F und G sind Formeln, zu denen es äquivalente Formeln F' bzw. G' ohne $\Leftrightarrow$ gibt.

Induktionsschritt

Gemäß Ersetzbarkeitstheorem (Satz 4.9) gelten die folgenden Äquivalenzen:

$\neg F \equiv \neg F'$, $(F \vee G) \equiv (F' \vee G')$, $(F \wedge G) \equiv (F' \wedge G')$, $(F \Rightarrow G) \equiv (F' \Rightarrow G')$

und die jeweils rechts stehende Formel enthält den Junktor $\Leftrightarrow$ nicht.

Nach Satz 4.7 gilt weiterhin: $(F \Leftrightarrow G) \equiv ((F \Rightarrow G) \wedge (G \Rightarrow F))$, mit dem Ersetzbarkeitstheorem gilt $(F \Leftrightarrow G) \equiv ((F' \Rightarrow G') \wedge (G' \Rightarrow F'))$ und in der letzten Formel kommt $\Leftrightarrow$ nicht vor.

Resümee: Nach dem Prinzip der strukturellen Induktion gibt es also zu jeder Formel eine äquivalente Formel, in der $\Leftrightarrow$ nicht vorkommt.

Zum Selbststudium

Äquivalenz bedeutet semantische Gleichwertigkeit

- d.h. äquivalente Formeln verhalten sich immer gleich, wenn es um Fragen geht, die allein durch die Betrachtung der Wahrheitswertverläufe beantwortet werden können.
- Andere Eigenschaften müssen äquivalente Formeln nicht teilen.

Satz 4.10: Elimination von $\Leftrightarrow$

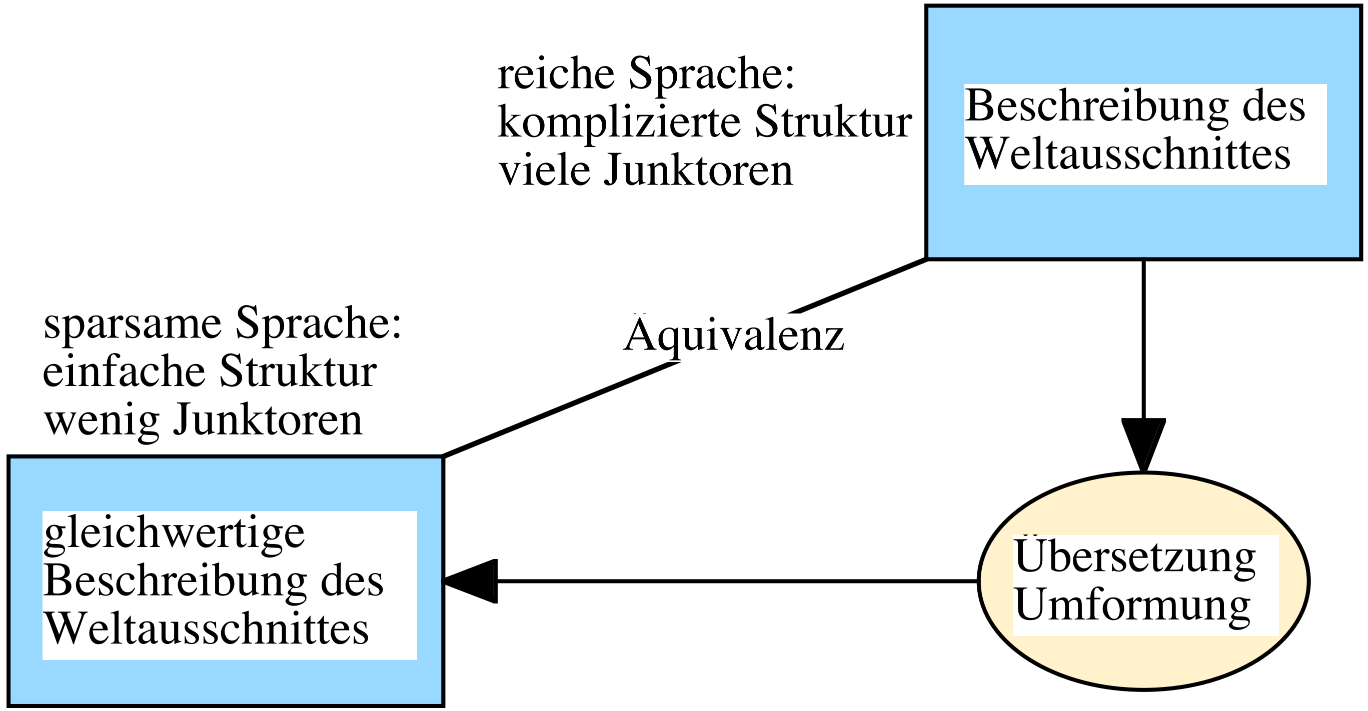
- ist ein Beispiel dafür, dass man die logische Sprache einschränken kann, ohne dadurch (logische / semantische) Ausdruckskraft zu verlieren. Was man verliert sind eher stilistische Variationsmöglichkeiten.
- Mögliche Einschränkungen betreffen die Wahl der Junktorenbasis, oder auch die Struktur der Formeln, wie z.B. bei den konjunktiven und disjunktiven Normalformen.

Satz 4.10.1

Zu jeder Formel gibt es eine äquivalente Formel, in der die Junktoren $\Leftrightarrow$ und $\Rightarrow$ nicht vorkommen.

Bew.: s. übernächste Seite, Einbettung in einen induktiven Beweis zur Übung

Übersetzung / Umformung von Formeln



Minimale Inventare für zweistellige Junktoren

$(F \Leftrightarrow G) \equiv (F \Rightarrow G) \wedge (G \Rightarrow F)$

- Der Junktor $\Leftrightarrow$ ist *durch* die Junktoren $\Rightarrow$ und $\wedge$ *ausdrückbar*.

Entsprechend kann z.B. auch der Junktor $\Rightarrow$ eliminiert werden:

$(F \Rightarrow G) \equiv (\neg F \vee G)$

- Der Junktor $\Rightarrow$ ist *durch* die Junktoren $\neg$ und $\vee$ *ausdrückbar*.

Welche Junktoren können durch welche Junktorenbasis ausgedrückt werden?

- Jeder zweistellige Junktor ist durch $\{\neg, \vee\}$ ausdrückbar.
- Jeder zweistellige Junktor ist durch $\{\neg, \wedge\}$ ausdrückbar.
- Jeder zweistellige Junktor und auch $\neg$ ist durch $\{\uparrow\}$ (NAND) ausdrückbar.
- Jeder zweistellige Junktor und auch $\neg$ ist durch $\{\downarrow\}$ (NOR) ausdrückbar.

➔ Es gilt sogar: Jeder Junktor mit endlich vielen Stellen ist durch jede dieser vier Mengen ausdrückbar.

➔ Unsere Junktorenbasis beschränkt die Ausdrückbarkeit nicht, wir könnten sogar noch einige Junktoren streichen.

➔ Normalformbildung durch Beschränkung der Junktorenbasis

Strukturbäume von konjunktiven und disjunktiven Normalformen

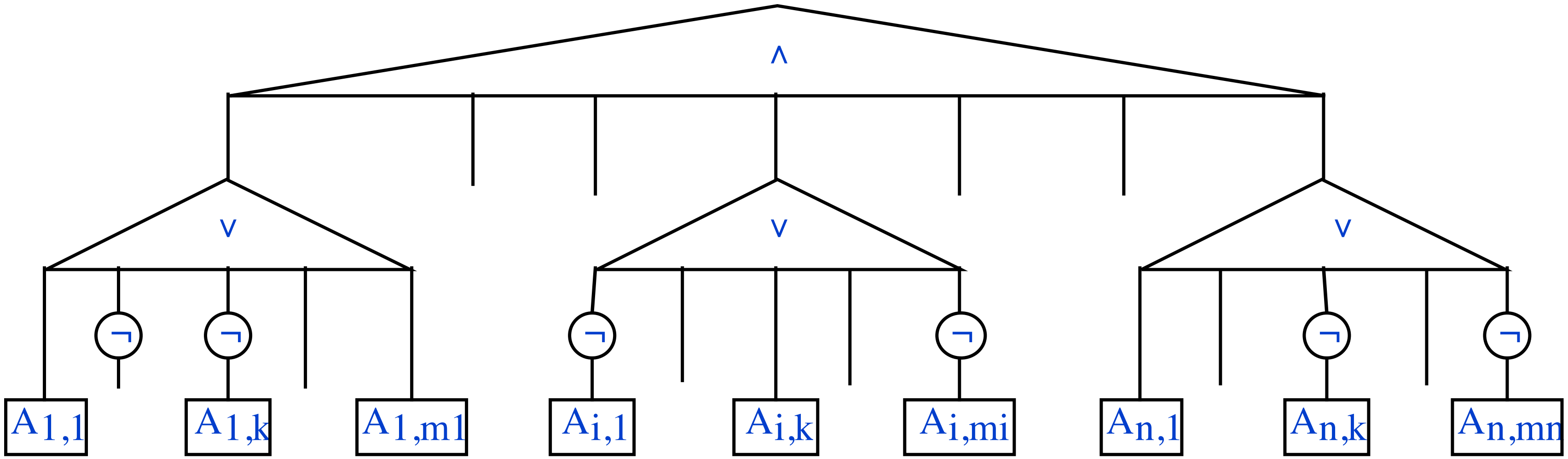
Einschränkung der Junktorenbasis

1. Es kommen als Junktoren nur $\wedge$, $\vee$, $\neg$ vor.

Einschränkung der Bildungsregeln (Grammatik), Strukturbäume

2. Kein Negationsknoten steht über einem Junktorknoten.
3. KNF: Kein Disjunktionsknoten steht über einem Konjunktionsknoten.
4. DNF: Kein Konjunktionsknoten steht über einem Disjunktionsknoten.

Beispiel: KNF



Literale

Definition 4.11 (Literal)

Atomare Formeln und ihre Negationen werden als *Literale* bezeichnet.

- Atomare Formeln heißen *positive Literale*,
  negierte atomare Formeln *negative Literale*.
- Die Symbole L, L_1, L_2, ..., L_i, L_k, ... verwenden wir als Variablen, die Literale als Wert haben können.
- Beispiele: Literale: A, $\neg B$
  keine Literale: $\neg\neg A$, $(A \vee B)$

Zwei Literale heißen genau dann *komplementär*, wenn sie positives und negatives Literal bezüglich der gleichen atomaren Formel sind;

- z.B. A und $\neg A$ sind komplementäre Literale.
- Wir sagen auch $\neg A$ ist das komplementäre Literal zu A und
  A ist das komplementäre Literal zu $\neg A$
- Der Balken über einem Literal zeigt, dass das komplementäre Literal gemeint ist:
  $\overline{A} = \neg A$ und $\overline{\neg A} = A$

Abkürzende Schreibweisen: Konjunktionen und Disjunktionen

Es sei $n \in \mathbb{N}$ und es seien $F_1, F_2 \ldots F_n$ Variablen, die Formeln als Wert haben.

Kurzschreibweisen für die Konjunktion und die Disjunktion der F_i

Konjunktionen		Disjunktionen	
$(\bigwedge_{i=1}^{n} F_i)$	$((\ldots (F_1 \wedge F_2) \wedge \ldots) \wedge F_n)$	$(\bigvee_{i=1}^{n} F_i)$	$((\ldots (F_1 \vee F_2) \vee \ldots) \vee F_n)$
$(\bigwedge_{i=1}^{3} F_i)$	$((F_1 \wedge F_2) \wedge F_3)$	$(\bigvee_{i=1}^{3} F_i)$	$((F_1 \vee F_2) \vee F_3)$
$(\bigwedge_{i=1}^{2} F_i)$	$(F_1 \wedge F_2)$	$(\bigvee_{i=1}^{2} F_i)$	$(F_1 \vee F_2)$
$(\bigwedge_{i=1}^{1} F_i)$	F_1	$(\bigvee_{i=1}^{1} F_i)$	F_1

Aufgrund der Assoziativgesetze ist die interne Klammerung eigentlich unerheblich.

Definition 4.12 (Klausel)

Literale und Disjunktionen von Literalen werden als (nicht leere) *Klauseln* bezeichnet.

- Allgemeine Form einer Klausel: $(\bigvee_{i=1}^{n} L_i) = (L_1 \vee \ldots \vee L_n)$
- Zu beachten: Ist n = 1, dann ist $(\bigvee_{i=1}^{1} L_i) = L_1$
- Die Symbole K, K_1, ..., K_i, ... verwenden wir als Variablen, die Klauseln als Wert haben können.

Satz 4.13: Jede (nicht-leere) Klausel K ist erfüllbar.

Beweis: Jedes Literal ist erfüllbar (Bsp. 3.4.1) und Disjunktionen von erfüllbaren Formeln sind (Def. 3.1) auch erfüllbar.

Satz 4.14: Eine (nicht-leere) Klausel ist genau dann allgemeingültig, wenn in ihr mindestens ein Paar von komplementären Literalen vorkommt. (Beweis zur Übung)

➔ Tautologieprüfung von Klauseln ist als Prüfung der Existenz komplementärer Literale realisierbar.

➔ Ein rein syntaktisches Verfahren ohne Aufbau der Wahrheitstafel.

Duale Klauseln

Definition 4.15 (Duale Klausel)

Literale und Konjunktionen von Literalen werden als (nicht leere) *Duale Klauseln* bezeichnet.

- Allgemeine Form einer dualen Klausel: $(\bigwedge_{i=1}^{n} L_i) = (L_1 \wedge \ldots \wedge L_n)$
- Zu beachten: Ist n = 1, dann ist $(\bigwedge_{i=1}^{1} L_i) = L_1$

Satz 4.16: Jede (nicht-leere) duale Klausel ist falsifizierbar. (Keine (nicht-leere) duale Klausel ist gültig.)
Beweis: Jedes Literal ist falsifizierbar (Bsp. 3.4.1) und Konjunktionen von falsifizierbaren Formeln sind auch falsifizierbar (Def. 3.1).

Satz 4.17: Eine (nicht-leere) duale Klausel ist genau dann unerfüllbar, wenn in ihr mindestens ein Paar von komplementären Literalen vorkommt. (Beweis zur Übung)

➔ Unerfüllbarkeitsprüfung von dualen Klauseln ist als Prüfung der Existenz komplementärer Literale realisierbar.

Definition 4.18

Eine Formel F ist genau dann in konjunktiver Normalform (KNF), wenn

- Keine Junktoren außer $\wedge$, $\vee$ und $\neg$ in F vorkommen,
- jede Teilformel von F mit Hauptoperator $\neg$ als (echte) Teilformel genau eine atomare Formel hat,
- keine Teilformel von F mit Hauptoperator $\vee$ ein $\wedge$ enthält.

➔ Eine Formel F ist in KNF, wenn sie folgende Form hat:

$$F = \left(\bigwedge_{i=1}^{n} \left(\bigvee_{k=1}^{mi} L_{i,k}\right)\right)$$

wobei $n, m_1, \ldots, m_n \in \mathbb{N}$ und $L_{1,1}, \ldots, L_{n,mi}$ Literale sind.

- Vereinfachend (aber nicht immer ganz korrekt) sagen wir dann auch:
  F ist eine Konjunktion von Klauseln.
  F ist eine Konjunktion von Disjunktionen von Literalen.

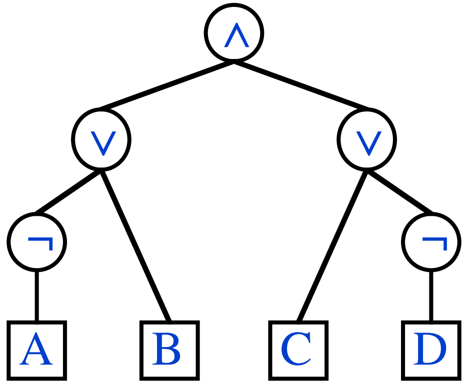
Beispiele KNF

$(\neg A \vee B) \wedge (C \vee \neg D)$

$n = 2, m_1 = m_2 = 2$

$L_{1,1} = \neg A, L_{1,2} = B$

$L_{2,1} = C, L_{2,2} = \neg D$



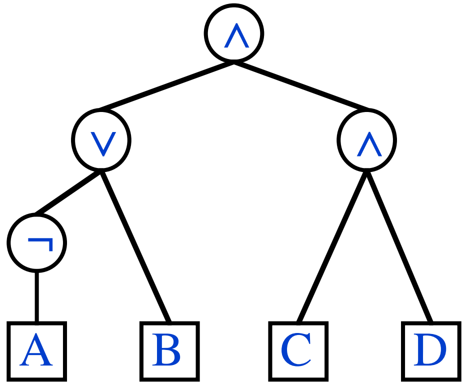
$(\neg A \vee B) \wedge C \wedge D$

$n = 3, m_1 = 2, m_2 = 1, m_3 = 1$

$L_{1,1} = \neg A, L_{1,2} = B$

$L_{2,1} = C$

$L_{3,1} = D$



C

$n = 1, m_1 = 1$

$L_{1,1} = C$

Definition 4.19

Eine Formel F ist genau dann in *disjunktiver Normalform* (DNF), wenn

- Keine Junktoren außer $\wedge$, $\vee$ und $\neg$ in F vorkommen,
- jede Teilformel von F mit Hauptoperator $\neg$ als (echte) Teilformel genau eine atomare Formel hat,
- keine Teilformel von F mit Hauptoperator $\wedge$ ein $\vee$ enthält.

➔ Eine Formel F ist in DNF, wenn sie folgende Form hat:

$$F = \left(\bigvee_{i=1}^{n} \left(\bigwedge_{k=1}^{mi} L_{i,k}\right)\right)$$

- Vereinfachend (aber nicht immer ganz korrekt):
  F ist eine Disjunktion von dualen Klauseln.
  F ist eine Disjunktion von Konjunktionen von Literalen.

Ergänzung: Grammatik für Konjunktive Normalformen

Vokabular der terminalen Symbole (Symbole, die in der Formel vorkommen): $\Sigma_{KNF} = \mathcal{As}_{AL} \cup \{\neg, \wedge, \vee, \ ), \ (\ \}$ **nicht-terminales Symbol** (Symbole, die für die Erzeugung der Formeln benötigt werden, aber nicht in der Formel vorkommen): $N_{KNF} = \{KNF, Kl, L, As\}$ **Startsymbol:** KNF KNF: Konjunktive Normalform Kl: Klausel L: Literal As: Aussagesymbol	**Regeln (Produktionen):** $P = \{ \ KNF \rightarrow (KNF \wedge Kl)$, $KNF \rightarrow Kl$, $Kl \rightarrow (Kl \vee L)$, $Kl \rightarrow L$, $L \rightarrow \neg As$, $L \rightarrow As$, $As \rightarrow A$, $As \rightarrow B$, $As \rightarrow C$, $As \rightarrow D$, … $\}$

Nutzen von Normalformen: Beispiel

Beobachtung 4.20

- Eine KNF ist genau dann gültig, wenn alle ihre Klauseln gültig sind.
- Eine KNF ist genau dann gültig, wenn in allen ihren Klauseln mindestens ein Paar von komplementären Literalen vorkommt.

➔ Tautologieprüfung von KNF ist als Prüfung der Existenz komplementärer Literale realisierbar.

➔ Resolution (Logische Basis von PROLOG / SE-3) ist ein Verfahren, das Unerfüllbarkeit auf der Basis von KNF überprüft. [Wird ausführlich in Kap 8. (Aussagenlogik) und Kap. 12 Prädikatenlogik behandelt.]

Beobachtung 4.21

- Eine DNF ist genau dann unerfüllbar, wenn alle ihre dualen Klauseln unerfüllbar sind.
- Eine DNF ist genau dann unerfüllbar, wenn in allen ihren dualen Klauseln mindestens ein Paar von komplementären Literalen vorkommt.

➔ Unerfüllbarkeitsprüfung von DNF ist als Prüfung der Existenz komplementärer Literale realisierbar.

➔ Grundlage des Tableau-Verfahrens (vertieft im Masterstudium: FGI-3)

Normalformtheorem

Definition 4.22

- Eine Formel F' ist genau dann eine *konjunktive Normalform zu einer Formel* F, wenn F' in konjunktiver Normalform und äquivalent zu F ist.
- Eine Formel G' ist genau dann eine *disjunktive Normalform zu einer Formel* G, wenn G' in disjunktiver Normalform und äquivalent zu G ist.

Satz 4.23

Zu jeder Formel F gibt es (mindestens) eine konjunktive Normalform und (mindestens) eine disjunktive Normalform

So geht es weiter

- Beweis der Existenz von Normalformen
- Algorithmus für die Erstellung von Normalformen durch Umformung
- Erstellung von Normalformen auf der Basis der Wahrheitswertverläufe

Beweis des Normalformtheorems

Vor.: Def. 2.2, 4.11, 4.18, 4.19, 4.22; Satz 2.10, 4.7, 4.10, 4.10.1

Beweis (von Satz 4.23)

- Gemäß der Eliminationsregeln für $\Leftrightarrow$ (Satz 4.10) und $\Rightarrow$ (Satz 4.10.1) gibt es zu jeder Formel eine äquivalente Formel, in der die Junktoren $\Leftrightarrow$ und $\Rightarrow$ nicht vorkommen. Entsprechend reicht es zu zeigen, dass es zu jeder Formel, in der nur die Junktoren $\{\neg, \vee, \wedge\}$ vorkommen, eine KNF und eine DNF gibt.
- Die Beweise für KNF und DNF verlaufen weitgehend analog und miteinander verzahnt nach dem Prinzip der strukturellen Induktion.
- *Induktionsanfang*: Jede atomare Formel ist ein positives Literal und damit in konjunktiver und in disjunktiver Normalform.
- *Induktionsannahme*: Es seien G und H Formeln, zu denen es konjunktive Normalformen G_K bzw. H_K und disjunktive Normalformen G_D bzw. H_D gibt.
- *Induktionsschritt*: Wir haben zu zeigen, dass es zu den Formeln $\neg G$, $(G \vee H)$ und $(G \wedge H)$ konjunktive und disjunktive Normalformen gibt.

Beweis NF-Theorem – Induktionsschritt für $\neg G$ [DNF]

Zu zeigen: Es gibt eine DNF zu $\neg G$.

- Nach Induktionsannahme ist G_K eine KNF von G:

$$G \equiv G_K = (\bigwedge_{i=1}^{n} (\bigvee_{k=1}^{mi} L_{i,k})) \qquad \text{[Def. KNF]}$$

also: $\neg G \equiv \neg G_K = \neg(\bigwedge_{i=1}^{n} (\bigvee_{k=1}^{mi} L_{i,k}))$ [Ersetzungstheorem]

$$\equiv (\bigvee_{i=1}^{n} \neg (\bigvee_{k=1}^{mi} L_{i,k})) \qquad \text{[de Morgansche Regel bzgl. } \wedge \text{ und Ersetzungstheorem]}$$

$$\equiv (\bigvee_{i=1}^{n} (\bigwedge_{k=1}^{mi} \neg L_{i,k})) \qquad \text{[de Morgansche Regel bzgl. } \vee \text{ und Ersetzungstheorem]}$$

$$\equiv (\bigvee_{i=1}^{n} (\bigwedge_{k=1}^{mi} \overline{L_{i,k}})) \qquad \text{[Doppelte Negation und Ersetzungstheorem]}$$

mit $\overline{L_{i,k}} = \begin{cases} A_j, \text{ falls } L_{i,k} = \neg A_j \\ \neg A_j, \text{ falls } L_{i,k} = A_j \end{cases}$

➔ dies ist eine DNF [Def. DNF]

- entsprechend kann aus G_D eine KNF zu $\neg G$ erstellt werden.

Beweis NF-Theorem – Induktionsschritt für $(G \wedge H)$ [KNF]

Zu zeigen: Es gibt eine KNF zu $(G \wedge H)$.

- Nach Induktionsannahme ist G_K eine KNF von G und H_K eine KNF von H:

$$G \equiv G_K = (\bigwedge_{i=1}^{n'} K'_i) \qquad H \equiv H_K = (\bigwedge_{j=1}^{n''} K''_j)$$, wobei K'_i und K''_j Klauseln sind

also: $(G \wedge H) \equiv (G_K \wedge H_K) = ((\bigwedge_{i=1}^{n'} K'_i) \wedge (\bigwedge_{j=1}^{n''} K''_j))$ [Ersetzungstheorem]

➔ Linksklammerung ist gemäß Assoziativgesetz möglich.

- Die gewünschte Form erkennen wir dann mit der folgenden Zuordnung:

$$(G \wedge H) \equiv F_K := (\bigwedge_{i=1}^{n} K_i) \qquad \text{mit } n := n' + n'' \quad \text{und} \quad K_i := \begin{cases} K'_i, \text{ für } 1 \leq i \leq n' \\ K''_{i-n'}, \text{ für } n' < i \leq n \end{cases}$$

- entsprechend kann aus G_D und H_D eine DNF zu $(G \vee H)$ erstellt werden.

Beweis NF-Theorem – Induktionsschritt für $(G \vee H)$ [KNF]

Zu zeigen: Es gibt eine KNF zu $(G \vee H)$.

- Nach Induktionsannahme ist G_K eine KNF von G und $H_{K'}$ eine KNF von H:

$$G \equiv G_K = (\bigwedge_{i=1}^{n'} K'_i) \qquad H \equiv H_K = (\bigwedge_{j=1}^{n''} K''_j)$$, wobei K'_i und K''_j Klauseln sind

also: $(G \vee H) \equiv (G_K \vee H_K) = ((\bigwedge_{i=1}^{n'} K'_i) \vee (\bigwedge_{j=1}^{n''} K''_j))$ [Ersetzungstheorem]

$$\equiv (\bigwedge_{i=1}^{n'} (\bigwedge_{j=1}^{n''} (K'_i \vee K''_j)))$$ [Distributivitätsgesetz / Assoziativitätsgesetz]

- Die gewünschte Form erkennen wir dann mit der folgenden Zuordnung:

$(G \vee H) \equiv F_K := (\bigwedge_{i=1}^{n} K_i)$ mit $n := n' * n''$ und $K_{n''*(i-1)+j} := (K'_i \vee K''_j)$

➔ Linksklammerung in den neuen Klauseln ist gemäß Assoziativitätsgesetz möglich.

- entsprechend kann aus G_D und H_D eine DNF zu $(G \wedge H)$ erstellt werden.

Resümee: Nach dem Prinzip der strukturellen Induktion gibt es also zu jeder Formel eine äquivalente Formel in KNF und eine äquivalente Formel in DNF.

Verfahren für die Erstellung von Normalformen

Gegeben: eine Formel F

0) Ersetze alle Teilformeln der Form

$(G \Leftrightarrow H)$	durch	$(\neg G \vee H) \wedge (\neg H \vee G)$	[Elimination $\Leftrightarrow$]
$(G \Rightarrow H)$	durch	$(\neg G \vee H)$	[Elimination $\Rightarrow$]

1) Ersetze alle Teilformeln der Form

$\neg\neg G$	durch	G	[Doppelte Negation]
$\neg(G \wedge H)$	durch	$(\neg G \vee \neg H)$	[de Morgan]
$\neg(G \vee H)$	durch	$(\neg G \wedge \neg H)$	[de Morgan]

[„Treibe Negationen nach innen, zu den atomaren Formeln.“]

2a) Um die KNF zu bilden: Ersetze alle Teilformeln der Form

$(F \vee (G \wedge H))$	durch	$((F \vee G) \wedge (F \vee H))$	[Distributivität]
$((F \wedge G) \vee H)$	durch	$((F \vee H) \wedge (G \vee H))$	[Distributivität]

[„Treibe Disjunktionen nach innen und Konjunktionen nach außen.“]

2b) Um die DNF zu bilden: Ersetze alle Teilformeln der Form

$(F \wedge (G \vee H))$	durch	$((F \wedge G) \vee (F \wedge H))$	[Distributivität]
$((F \vee G) \wedge H)$	durch	$((F \wedge H) \vee (G \wedge H))$	[Distributivität]

[„Treibe Konjunktionen nach innen und Disjunktionen nach außen.“]

Erstellung der konjunktiven Normalform – Ein Beispiel

	$(A \land (B \Rightarrow C)) \Rightarrow D$	[Elimination von $\Rightarrow$]
$\equiv$	$(A \land (\neg B \lor C)) \Rightarrow D$	[Elimination von $\Rightarrow$]
$\equiv$	$\neg(A \land (\neg B \lor C)) \lor D$	[de Morgan]
$\equiv$	$(\neg A \lor \neg(\neg B \lor C)) \lor D$	[de Morgan]
$\equiv$	$(\neg A \lor (\neg \neg B \land \neg C)) \lor D$	[Doppelte Negation]
$\equiv$	$(\neg A \lor (B \land \neg C)) \lor D$	[Distributivität]
$\equiv$	$((\neg A \lor B) \land (\neg A \lor \neg C)) \lor D$	[Distributivität]
$\equiv$	$((\neg A \lor B) \lor D) \land ((\neg A \lor \neg C) \lor D)$	

Verfahren für die Erstellung von Normalformen – Fortsetzung

- Das Verfahren führt zu Formeln, die äquivalent zur Ausgangsformel sind.
- Das Verfahren terminiert.
- Das Verfahren liefert eine Formel in KNF (2a) bzw. in DNF (2b).

Begründungen

- Da alle Ersetzungen auf Äquivalenzen beruhen (Satz 4.7), sind alle (Zwischen- und Endprodukte) äquivalent zur ursprünglichen Formel (Satz 4.9).
- Die Elimination von $\Leftrightarrow$ terminiert, da eine (zwischenzeitliche) Vermehrung von Biimplikationen immer nur auf unteren Ebenen des Strukturbaums erfolgt.
- Alle weiteren Ersetzungen können zwar die Formellänge vergrößern, aber nicht die Tiefe des Strukturbaums.
- Ist die Elimination von $\Leftrightarrow$ (bzw. $\Rightarrow$) abgeschlossen, dann wird durch keinen Schritt $\Leftrightarrow$ (bzw. $\Rightarrow$) wieder eingeführt.
- Der Prozess endet, wenn
  - (i) Negationen nur noch bei atomaren Formeln auftreten (negative Literale) und wenn
  - (ii) keine Teilformel, deren Hauptoperator die Disjunktion ist, eine Konjunktion enthält (2a) bzw. keine Teilformel, deren Hauptoperator die Konjunktion ist, eine Disjunktion enthält (2b)

Erstellung von Normalformen aus Wahrheitstafeln – I (DNF)

Die Idee

- Komplette Wahrheitstafeln (Angabe aller unterschiedlichen Belegungen $\mathcal{A}_i$ zu einer Menge von Aussagensymbolen) geben einen Wahrheitswertverlauf an.
- Hieraus soll eine Formel in DNF oder KNF erzeugt werden, die genau diesen Wahrheitswertverlauf hat.

Verfahren zur Konstruktion einer DNF–Formel

Gegeben: ein kompletter Wahrheitswertverlauf

- Konstruiere zu jeder Zeile der Wahrheitstafel, die den Wert **1** aufweist, eine duale Klausel wie folgt:
  - Falls der Wert für ein Aussagensymbol in der Zeile **1** ist, so nimm das positive Literal in die Konjunktion auf.
  - Falls der Wert **0** ist, so nimm das negative Literal in die Konjunktion auf.
- Bilde die Disjunktion aller Konjunktionen, die nach diesem Verfahren konstruiert wurden. (Ignoriere alle Zeilen, die den Wert **0** aufweisen.)
- Wenn der Wahrheitswertverlauf keine **1** enthält, wähle ein beliebiges Aussagensymbol A und bilde die Formel $(A \wedge \neg A)$

Die Wahrheitstafel

	A	B	C	F	
$\mathcal{A}_1$	0	0	0	1	$(\neg A \wedge \neg B \wedge \neg C)$
$\mathcal{A}_2$	0	0	1	0	
$\mathcal{A}_3$	0	1	0	0	
$\mathcal{A}_4$	0	1	1	0	
$\mathcal{A}_5$	1	0	0	1	$(A \wedge \neg B \wedge \neg C)$
$\mathcal{A}_6$	1	0	1	1	$(A \wedge \neg B \wedge C)$
$\mathcal{A}_7$	1	1	0	0	
$\mathcal{A}_8$	1	1	1	0	

- DNF: $(\neg A \wedge \neg B \wedge \neg C) \vee (A \wedge \neg B \wedge \neg C) \vee (A \wedge \neg B \wedge C)$

Normalformen aus Wahrheitstafel – III – Die Idee (DNF)

- Eine Disjunktion ist genau dann wahr, wenn eine ihrer direkten Teilformeln wahr ist.
- Wir verwenden daher Formeln, die unter genau einer der Belegungen wahr sind.
- F = F1 ∨ F5 ∨ F6

	A	B	C	F	F1	F2	F3	F4	F5	F6	F7	F8
$\mathcal{A}_1$	0	0	0	1	1	0	0	0	0	0	0	0
$\mathcal{A}_2$	0	0	1	0	0	1	0	0	0	0	0	0
$\mathcal{A}_3$	0	1	0	0	0	0	1	0	0	0	0	0
$\mathcal{A}_4$	0	1	1	0	0	0	0	1	0	0	0	0
$\mathcal{A}_5$	1	0	0	1	0	0	0	0	1	0	0	0
$\mathcal{A}_6$	1	0	1	1	0	0	0	0	0	1	0	0
$\mathcal{A}_7$	1	1	0	0	0	0	0	0	0	0	1	0
$\mathcal{A}_8$	1	1	1	0	0	0	0	0	0	0	0	1

- Konjunktionen aus Literalen, in denen jedes Aussagensymbol genau einmal auftritt, werden unter genau einer der Belegungen wahr.
- z.B. F2 = ¬A ∧ ¬B ∧ C, F5 = A ∧ ¬B ∧ ¬C

Normalformen aus Wahrheitstafel – VI (Der KNF-Fall)

- Eine Konjunktion ist genau dann falsch, wenn eine ihrer direkten Teilformeln falsch ist. Wir betrachten daher Formeln, die unter genau einer der Belegungen falsch sind.
- $F = G2 \wedge G3 \wedge G4 \wedge G7 \wedge G8$

	A	B	C	F	G1	G2	G3	G4	G5	G6	G7	G8
$\mathcal{A}_1$	0	0	0	1	0	1	1	1	1	1	1	1
$\mathcal{A}_2$	0	0	1	0	1	0	1	1	1	1	1	1
$\mathcal{A}_3$	0	1	0	0	1	1	0	1	1	1	1	1
$\mathcal{A}_4$	0	1	1	0	1	1	1	0	1	1	1	1
$\mathcal{A}_5$	1	0	0	1	1	1	1	1	0	1	1	1
$\mathcal{A}_6$	1	0	1	1	1	1	1	1	1	0	1	1
$\mathcal{A}_7$	1	1	0	0	1	1	1	1	1	1	0	1
$\mathcal{A}_8$	1	1	1	0	1	1	1	1	1	1	1	0

- Disjunktionen aus Literalen, in denen jedes Aussagensymbol genau einmal auftritt, werden unter genau einer der Belegungen falsch.
- z.B. $G2 = A \vee B \vee \neg C$

Normalformen aus Wahrheitstafel – V (KNF)

- Konstruiere zu jeder Zeile der Wahrheitstafel mit dem Wert **0**, eine Klausel wie folgt:
  - Falls der Wert für ein Aussagensymbol in der Zeile **0** ist, so nimm das positive Literal in die Disjunktion auf, sonst nimm das negative Literal auf.
- Bilde die Konjunktion aller so gewonnenen Disjunktionen. (Ignoriere alle Zeilen mit **1**.)
- Wenn der Wahrheitswertverlauf keine **0** enthält, wähle ein beliebiges Aussagensymbol A und bilde die Formel $(A \vee \neg A)$

	A	B	C	F	
$\mathcal{A}_1$	0	0	0	1	
$\mathcal{A}_2$	0	0	1	0	$(A \vee B \vee \neg C)$
$\mathcal{A}_3$	0	1	0	0	$(A \vee \neg B \vee C)$
$\mathcal{A}_4$	0	1	1	0	$(A \vee \neg B \vee \neg C)$
$\mathcal{A}_5$	1	0	0	1	
$\mathcal{A}_6$	1	0	1	1	
$\mathcal{A}_7$	1	1	0	0	$(\neg A \vee \neg B \vee C)$
$\mathcal{A}_8$	1	1	1	0	$(\neg A \vee \neg B \vee \neg C)$

KNF:
$(A \vee B \vee \neg C) \wedge (A \vee \neg B \vee C) \wedge (A \vee \neg B \vee \neg C) \wedge (\neg A \vee \neg B \vee C) \wedge (\neg A \vee \neg B \vee \neg C)$

Zusammenfassung: Äquivalenz und Normalformen

Äquivalenz

➔ Formeln sind äquivalent, wenn sie unter allen Belegungen denselben Wahrheitswert liefern.

➔ semantische Gleichwertigkeit: Immer wenn es um semantische Fragen (Wahrheitswertverläufe) geht, ist es egal welche der äquivalenten Formeln man betrachtet

➔ syntaktische Unterschiede können aber bestehen, Formelgleichheit / Identität bezieht diese Unterschiede mit ein.

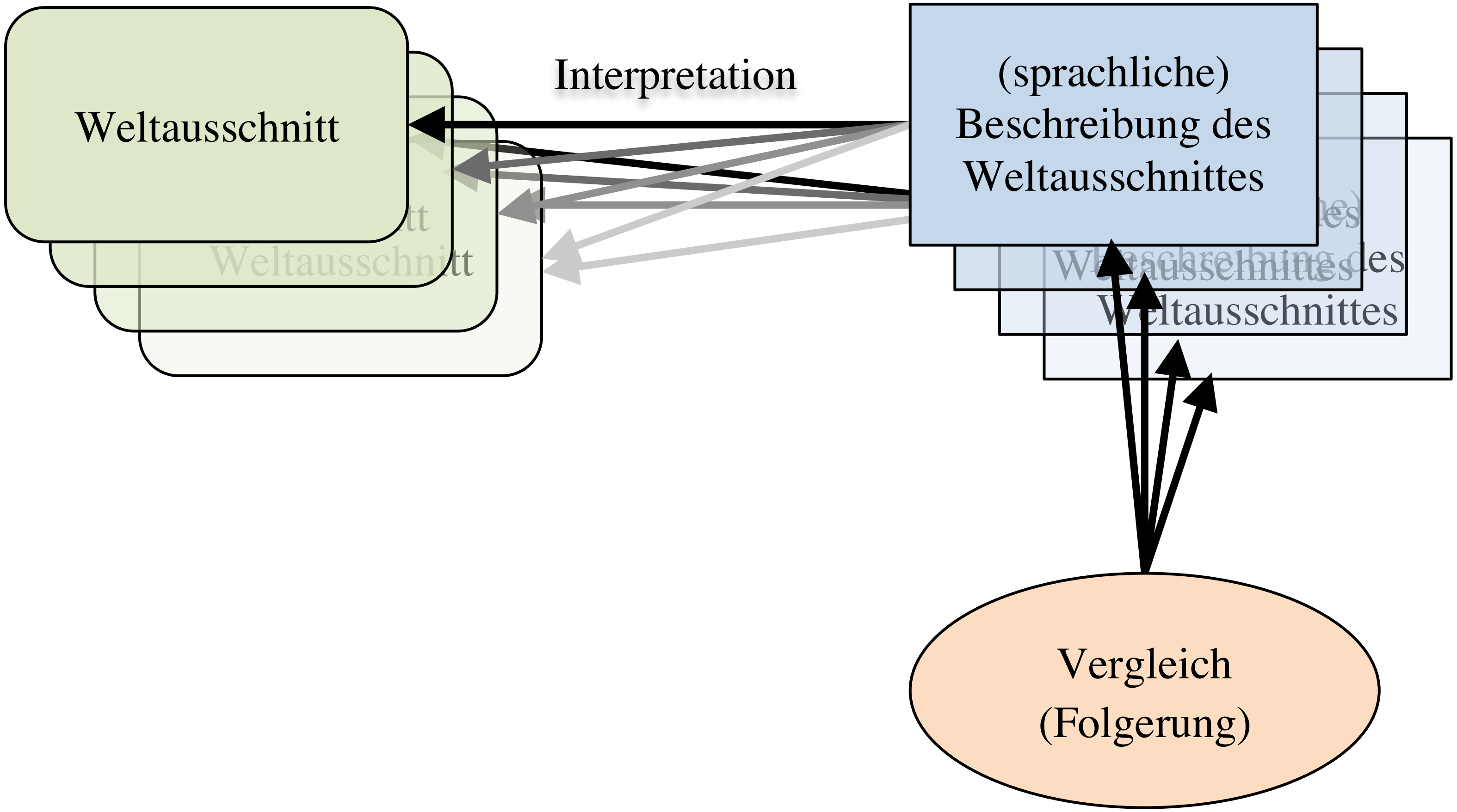
Normalformen: insb. KNF, DNF

- Restriktion der Syntax der Sprache (Junktoreninventar, Bildungsregeln) ohne Einschränkung des Ausdrucksvermögens
- Normalform zu einer Formel: Äquivalente Formel, die zur eingeschränkten Sprache gehört.
- Nutzen: der geringere Formenreichtum kann die weitere Verarbeitung erleichtern
- Eindeutigkeit der Normalform muss aber nicht gewährleistet sein.

Wichtige Konzepte in diesem Foliensatz

- (logische) Äquivalenz
- Konjunktive Normalform, Disjunktive Normalform, Klausel, duale Klausel, Literal
- Ersetzung, Ersetzbarkeitstheorem
- Junktorenelimination
- KNF-/DNF-Erzeugung durch Umformung
- KNF-/DNF-Erzeugung aus Wahrheitstafel

Folgerung – Folgerbarkeit



Definition: Folgerung

Definition 5.1

- Eine Formel G *folgt* genau dann *(logisch)* aus einer Formelmenge **M**, falls für jede Belegung $\mathcal{A}$ gilt:
  Wenn $\mathcal{A}$ ein Modell für **M** ist, dann ist $\mathcal{A}$ auch ein Modell für G.
- Weitere Sprechweisen / Schreibweisen für „G folgt aus **M**"
  - G ist folgerbar aus **M**
  - G ist eine Folgerung der Formeln aus **M**
  - **M** $\vDash$ G
  - Statt {F} $\vDash$ G wird F $\vDash$ G geschrieben. „G folgt aus der Formel F."

➔ WICHTIG: $\vDash$ ist kein Junktor, sondern eine Relation zwischen Formelmengen und Formeln.

{$\mathcal{A}$ | $\mathcal{A}$ ist Belegung}

{$\mathcal{A}$ | $\mathcal{A}$ ist Modell für G}

{$\mathcal{A}$ | $\mathcal{A}$ ist Modell für **M**}

Beispiel: Folgerung

$(A \wedge B) \models (A \vee B)$

	A	B	$(A \wedge B)$	$(A \vee B)$
$\mathcal{A}_1$	0	0	0	0
$\mathcal{A}_2$	0	1	0	1
$\mathcal{A}_3$	1	0	0	1
$\mathcal{A}_4$	1	1	1	1

Alle Belegungen, die Modell von $(A \wedge B)$ sind, sind Modell von $(A \vee B)$

$\{(A \vee B), \neg B\} \models A$

	A	B	$\neg B$	$(A \vee B)$
$\mathcal{A}_1$	0	0	1	0
$\mathcal{A}_2$	0	1	0	1
$\mathcal{A}_3$	1	0	1	1
$\mathcal{A}_4$	1	1	0	1

Alle Belegungen, die Modell von $(A \vee B)$ und von $\neg B$ sind, sind Modell von A

Das sprachliche Inventar einer Daten- / Wissensbasis (DB/WB)

- HP7_e ≈ Harry Potter 7 ist auf englisch erschienen.
- HP7_d ≈ Harry Potter 7 ist auf deutsch erschienen.

DB / WB

- **WB_3** = { HP7_d ⇒ HP7_e, HP7_d }

Belegungen über dem Inventar { HP7_e, HP7_d }

	HP7_e	HP7_d	HP7_d ⇒ HP7_e
$\mathcal{A}_1$	0	0	1
$\mathcal{A}_2$	1	0	1
$\mathcal{A}_3$	0	1	0
$\mathcal{A}_4$	1	1	1

$\mathcal{A}_4$ ist ein Modell für HP7_e, d.h. alle Modelle von **WB_3** sind auch Modelle von HP7_e.

➔ **WB_3** $\models$ HP7_e

Folgerung: indirekte Kodierung von Fakten

Modellierung mit logischen Formeln

- Sammeln von Fakten, die (in dem zu modellierendem Weltausschnitt) wahr sind.
  - Darunter können sein

    ⇒ atomare Aussagen („Harry Potter 7 ist auf deutsch erschienen.“)

    ⇒ komplexe Aussagen („Wenn Harry Potter 7 auf deutsch erschienen ist, dann ist Harry Potter 7 (auch schon) auf englisch erschienen.“)

    ⇒ Tautologien (die sind aber wenig nützlich)

    ⇒ keine Kontradiktionen
- Bestimmung eines Übersetzungsschlüssels
- Kodierung der Fakten mit dem Übersetzungsschlüssel (direkte Kodierung).

  ⇒ { HP7_d, HP7_d ⇒ HP7_e }
- Die Faktenmenge
  - ist erfüllbar (es gibt ein Modell, das dem modellierten Weltausschnitt entspricht).
  - beschreibt den modellierten Weltausschnitt in der Regel unvollständig.
  - kodiert indirekt weitere Aussagen, die in allen Modellen der Faktenbasis wahr sind, und damit auch im modellierten Weltausschnitt (HP7_e, (HP7_e ∧ HP7_d)).

➔ Folgerungen sind indirekt kodierte Aussagen.

Das Symbol $\models$

hat verschiedene Verwendungen

$\mathcal{A} \models \mathsf{F}$	$\mathcal{A}$ ist Modell für F.	
$\models \mathsf{F}$	F ist allgemeingültig.	Alle Belegungen sind Modelle für F.
$\mathbf{M} \models \mathsf{F}$	F folgt aus $\mathbf{M}$.	Alle Modelle von $\mathbf{M}$ sind Modelle von F.
$\mathsf{G} \models \mathsf{F}$	F folgt aus G.	Alle Modelle von G sind Modelle von F.

- Woran können Sie jeweils erkennen, welche Verwendung Sie vor sich haben?
- Überlegen Sie, welche Motive die Logiker gehabt haben könnten, gerade für diese Bedingungen das einheitliche Symbol $\models$ einzusetzen.

Anderenorts findet man auch noch folgende Verwendungen

$\mathbf{M} \models$	$\mathbf{M}$ ist unerfüllbar.	Es gibt keine Modelle für $\mathbf{M}$.
$\mathsf{G} \models$	G ist unerfüllbar.	Es gibt keine Modelle für G.

- Erkennen Sie auch die Motivation für diese Verwendung?

Folgerbarkeit – Allgemeingültigkeit

Satz 5.2: Folgerbarkeit von allgemeingültigen Formeln

Jede allgemeingültige Formel F ist aus jeder Formelmenge **M** folgerbar.

$\models F$ dann auch $\mathbf{M} \models F$

Vor.: Def. 3.3, 3.5, 5.1

Bew.: Ist F eine Tautologie, dann macht jede Belegung F wahr, also auch die Modelle von **M.** D.h. F folgt aus **M**.

Satz 5.3: Folgerbarkeit aus allgemeingültigen Formeln

Wenn F aus einer Menge von Tautologien **M** folgt, dann ist F eine Tautologie.

Vor.: Def. 3.1, 3.3, 3.5, 5.1

Bew.: Es sei **M** eine Menge von Tautologien und F folge aus **M**.

Es sei $\mathcal{A}$ eine beliebige Belegung.

Da alle Elemente von **M** allgemeingültig sind, ist $\mathcal{A}$ ein Modell von **M**.

Da F aus **M** folgt, ist $\mathcal{A}$ auch ein Modell von F.

Damit zeigt sich, dass jede Belegung ein Modell von F ist.

➔ Wenn Folgerungen bestimmbar sind, dann brauchen wir Tautologien nicht in die Beschreibung eines Weltausschnittes aufzunehmen.

Satz 5.4: Folgerbarkeit aus der leeren Menge

Wenn F aus der leeren Menge folgt, dann ist F eine Tautologie.

Vor.: Def. 3.1, 3.3, 3.5, 5.1

Bew.: Es sei F eine Formel, die aus der leeren Menge folgt.

Es sei $\mathcal{A}$ eine beliebige Belegung.

$\mathcal{A}$ ist Modell der leeren Menge (*kein Gegenbeispiel*, nützliche Konvention).

Da F aus der leeren Menge folgt, ist $\mathcal{A}$ auch ein Modell von F.

Da die Wahl von $\mathcal{A}$ beliebig war, ist F eine Tautologie.

- Satz 5.4 zeigt, dass es für die Prüfung von Allgemeingültigkeit ausreichend ist, ein Verfahren zu haben, mit dem man Folgerung von Formeln aus Formelmengen prüfen kann.
- Nach den oben vorgestellten Ergebnissen kann man dann entsprechend Unerfüllbarkeit einer Formel und Äquivalenz prüfen.
- Die Umkehrung ist nicht immer so einfach.

Zum Selbststudium: Folgerbarkeit – Unerfüllbarkeit

Satz 5.5: Folgerbarkeit aus unerfüllbaren Formelmengen

Aus jeder unerfüllbaren Formelmenge **M** ist jede beliebige Formel F folgerbar.

Vor.: Def. 3.1, 3.3, 3.4, 5.1

Bew.: Es sei **M** eine unerfüllbare Formelmenge und F eine Formel.
Keine Belegung ist Modell für **M**.
Also macht jede Belegung, die Modell für **M** ist, F wahr.
(Es ist *kein Gegenbeispiel* konstruierbar.)
Das heißt, dass F aus **M** folgt.

Satz 5.6: Folgerbarkeit von Kontradiktionen

Folgt eine Kontradiktion F aus einer Formelmenge **M**, dann ist **M** unerfüllbar.

Vor.: Def. 3.1, 3.3, 3.4, 5.1

Bew.: Es sei F eine Kontradiktion, die aus der Formelmenge **M** folgt.
Es sei $\mathcal{A}$ eine beliebige Belegung.
Da F unerfüllbar ist macht $\mathcal{A}$ F falsch
Da F aus **M** folgt, kann $\mathcal{A}$ kein Modell von **M** sein.
Also hat **M** kein Modell und ist unerfüllbar.

➔ Folgerbarkeit ist besonders interessant für kontingente Formeln, d.h. sie ist eine *logisch* interessante Beziehung zwischen *epistemisch* interessanten Aussagen.

Logische Äquivalenz und Folgerung

Satz 5.7: $F \equiv G$ GDW. $F \models G$ und $G \models F$.

Vor.: Def. 3.1, 3.3, 4.1, 5.1

Bew.: $F \equiv G$

GDW. für jede Belegung $\mathcal{A}$ gilt: $\mathcal{A}(F) = \mathcal{A}(G)$

GDW. jedes Modell für F, auch Modell für G ist,

und jedes Modell für G auch Modell für F ist,

GDW. $F \models G$ und $G \models F$

$\{\mathcal{A} \mid \mathcal{A}$ ist eine Belegung$\}$

$\boxed{\{\mathcal{A} \mid \mathcal{A} \text{ ist Modell für } G\} = \{\mathcal{A} \mid \mathcal{A} \text{ ist Modell für } F\}}$

Noch ein wichtiger Zusammenhang

- Wenn F_1 und F_2 äquivalent sind, sowie G_1 und G_2 äquivalent sind, dann folgt G_1 genau dann aus F_1, wenn G_2 aus F_2 folgt.

  (Wenn $F_1 \equiv F_2$ und $G_1 \equiv G_2$, dann $F_1 \models G_1$ GDW. $F_2 \models G_2$)

Folgerung und Implikation

Das Folgerungssymbol $\vDash$ ist kein Junktor, aber dennoch eng mit der Implikation verwandt.

Satz 5.8: $(F \Rightarrow G)$ ist allgemeingültig GDW. G aus F folgt.

($\vDash(F \Rightarrow G)$ GDW. $F \vDash G$).

Vor.: Def. 3.1, 3.3, 3.5, 5.1

Bew.: $(F \Rightarrow G)$ ist allgemeingültig, d.h. $\vDash (F \Rightarrow G)$

GDW. alle Belegungen $\mathcal{A}$ Modelle von $(F \Rightarrow G)$ sind, [Def. 3.5]

d.h. $\mathcal{A}((F \Rightarrow G)) = \mathbf{1}$ [Def. 3.3]

GDW. Für alle Belegungen $\mathcal{A}$ gilt: $\mathcal{A}(F) = \mathbf{0}$ oder $\mathcal{A}(G) = \mathbf{1}$ [Def. 3.1]

GDW. Für alle Belegungen $\mathcal{A}$ gilt: Wenn $\mathcal{A}$ Modell für F ist, dann ist $\mathcal{A}$ auch Modell für G (d.h. Wenn $\mathcal{A}(F) = \mathbf{1}$, dann $\mathcal{A}(G) = \mathbf{1}$) [Def. 3.3]

GDW. G folgt aus F, d.h. $F \vDash G$ [Def. 5.1]

- Satz 5.8 reduziert die Frage der Folgerung *zwischen Formeln* auf die Frage der Allgemeingültigkeit.

Zum Selbststudium: Folgerung und Implikation

Dieser Satz ist eine Verallgemeinerung des vorherigen.

Satz 5.8x: G folgt aus $M \cup \{F\}$ GDW. $(F \Rightarrow G)$ aus M folgt.

$(M \cup \{F\} \models G$ GDW. $M \models (F \Rightarrow G)$).

Vor.: Def. 3.1, 3.3, 5.1

Bew.: Es sei **M** eine Formelmenge und F und G Formeln.

$(F \Rightarrow G)$ folgt aus **M**, d.h. $\mathbf{M} \models (F \Rightarrow G)$

GDW. alle Belegungen $\mathcal{A}$, die Modelle von **M** sind,
Modelle von $(F \Rightarrow G)$ sind, [Def. 5.1]

GDW. für alle Modelle $\mathcal{A}$ von **M** gilt: $\mathcal{A}(F) = \mathbf{0}$ oder $\mathcal{A}(G) = \mathbf{1}$ [Def. 3.1, 3.3]

GDW. alle Belegungen $\mathcal{A}$, die Modelle von $\mathbf{M} \cup \{F\}$ sind,
Modelle für G sind [Def. 3.3]

GDW. G folgt aus $\mathbf{M} \cup \{F\}$, d.h. $\mathbf{M} \cup \{F\} \models G$ [Def. 5.1]

Folgerung und Unerfüllbarkeit

Satz 5.9: $F \models G$ GDW. $(F \wedge \neg G)$ unerfüllbar ist.

Vor.: Def. 3.1, 3.3, 3.4, 5.1

Bew.1: $F \models G$ GDW. $\models (F \Rightarrow G)$ [Satz 5.8]

GDW. $\neg(F \Rightarrow G)$ unerfüllbar ist [Satz 3.1]

GDW. $\neg(\neg F \vee G)$ unerfüllbar ist [Satz 4.7 Elimination $\Rightarrow$, Ersetzungstheorem]

GDW. $(\neg\neg F \wedge \neg G)$ unerfüllbar ist [Satz 4.7 de Morgan]

GDW. $(F \wedge \neg G)$ unerfüllbar ist [Satz 4.7 doppelte Negation, Ersetzungstheorem]

Bew.2: $F \models G$ GDW. Alle Modelle $\mathcal{A}$ von F auch Modelle von G sind

(Wenn $\mathcal{A}(F) = \mathbf{1}$, dann $\mathcal{A}(G) = \mathbf{1}$) [Def. 5.1]

GDW. Alle Modelle $\mathcal{A}$ von F keine Modelle von $\neg G$ sind

(Wenn $\mathcal{A}(F) = \mathbf{1}$, dann $\mathcal{A}(\neg G) = \mathbf{0}$) [Def. 3.1, 3.3]

GDW. Alle Belegungen $\mathcal{A}$ $(F \wedge \neg G)$ falsch machen ($\mathcal{A}((F \wedge \neg G)) = \mathbf{0}$)[Def. 3.1]

GDW. $(F \wedge \neg G)$ unerfüllbar ist [Def. 3.4]

Satz 5.10: $M \models G$ GDW. $M \cup \{\neg G\}$ unerfüllbar ist.

Bew.: Ganz entsprechend.

Allgemeingültigkeit, Erfüllbarkeit, Unerfüllbarkeit, Äquivalenz, Folgerbarkeit

Prüfung auf Allgemeingültigkeit, Erfüllbarkeit, Unerfüllbarkeit

- Für eine Formel F mit n Aussagensymbolen kann eine komplette Wahrheitstafel mit 2^n unterschiedlichen Belegungen berechnet werden.
  - F ist *erfüllbar*, falls mindestens eine Belegung den Wahrheitswert **1** für F ergibt.
  - F ist *allgemeingültig*, falls alle Belegungen den Wahrheitswert **1** für F ergeben.
  - F ist *unerfüllbar*, falls keine Belegung den Wahrheitswert **1** für F ergibt.

Prüfung auf Äquivalenz und Folgerbarkeit

- Für eine Formel F und eine Formel G mit insgesamt n Aussagensymbolen kann eine komplette Wahrheitstafel mit 2^n unterschiedlichen Belegungen berechnet werden.
  - F und G *sind äquivalent*, falls F und G denselben Wahrheitswertverlauf haben.
  - G *folgt aus* F, falls alle Belegungen, die für F den Wahrheitswert **1** liefern, auch für G den Wahrheitswert **1** ergeben.

➔ algorithmische Verfahren durch Aufstellung der kompletten Wahrheitstafel erfordern exponentieller Aufwand: im ungünstigen Fall sehr aufwendig.

Formelmengen: Erfüllbarkeit, Unerfüllbarkeit, Folgerbarkeit

Formelmenge M

- **M** ist *erfüllbar*, falls mindestens eine Belegung den Wahrheitswert **1** für alle Elemente von **M** ergibt.
- **M** ist *unerfüllbar*, falls jede Belegung den Wahrheitswert **0** für mindestens ein Element von **M** ergibt.
- eine Formel F *folgt aus* **M**, falls alle Belegungen, die für die Formeln aus **M** den Wahrheitswert **1** liefern, auch für F den Wahrheitswert **1** ergeben.

➔ Was ist los, wenn **M** unendlich viele Formeln (mit unendlich vielen Aussagensymbolen) enthält?

Zur Erinnerung: Die Menge der Aussagensymbole, $\mathcal{As}_{\mathrm{AL}}$, und die Menge der wohlgeformten Formeln, $\mathcal{L}_{\mathrm{AL}}$, sind abzählbar.

➔ Die Wahrheitstafelmethode ist dann nicht mehr anwendbar.

➔ Unendliche Formelmengen zur Weltbeschreibung können mit endlichen Mitteln erzeugt werden. (➔ Prädikatenlogik)

Folgerung aus Formelmengen – Folgerung aus Konjunktionen

- Zur Erinnerung: Es sei **M** eine Formelmenge und G eine Formel.
  M $\models$ G GDW. jedes Modell von **M** auch Modell von G ist.

Satz 5.11: $\{F1, F2\} \cup \mathbf{M} \models G$ GDW. $\{(F1 \land F2)\} \cup \mathbf{M} \models G$

Vor.: Def. 3.1, 3.3, 5.1

Bew.: Entsprechend der Wahrheitstafel von $\land$ sind die Modelle von $(F1 \land F2)$ mit den Modellen von $\{F1, F2\}$ identisch.
Damit gilt auch: die Modelle von $\{F1, F2\} \cup \mathbf{M}$ und die Modelle von $\{(F1 \land F2)\} \cup \mathbf{M}$ sind dieselben.

Satz 5.12: $\{F1, F2\} \cup \mathbf{M}$ ist unerfüllbar GDW. $\{(F1 \land F2)\} \cup \mathbf{M}$ unerfüllbar ist.

Bew.: Analog zum vorherigen.

➔ Die Bildung einer Formelmenge entspricht hier einer impliziten Konjunktion.
➔ Formeln sind aber stets endlich, Formelmengen können unendlich sein.

Folgerung aus endlichen Formelmengen – Folgerung aus Formeln

Abkürzende Schreibweisen (vgl. 4-20)

Konjunktionen		Disjunktionen	
$(\bigwedge_{i=1}^{n} F_i)$	$((\ldots (F_1 \wedge F_2) \wedge \ldots) \wedge F_n)$	$(\bigvee_{i=1}^{n} F_i)$	$((\ldots (F_1 \vee F_2) \vee \ldots) \vee F_n)$
$(\bigwedge_{i=1}^{3} F_i)$	$((F_1 \wedge F_2) \wedge F_3)$	$(\bigvee_{i=1}^{3} F_i)$	$((F_1 \vee F_2) \vee F_3)$
$(\bigwedge_{i=1}^{2} F_i)$	$(F_1 \wedge F_2)$	$(\bigvee_{i=1}^{2} F_i)$	$(F_1 \vee F_2)$
$(\bigwedge_{i=1}^{1} F_i)$	F_1	$(\bigvee_{i=1}^{1} F_i)$	F_1

Satz 5.13: $\{F_1, F_2, \ldots, F_n\} \models G$ GDW. $(\bigwedge_{i=1}^{n} F_i) \models G$

Bew.: Vollständige Induktion über n und Verwendung des Satzes 5.11.

WICHTIG: Dieser Satz ist nur auf endliche Formelmengen anwendbar !

Formelmengen: Modelle, Erfüllbarkeit, Folgerung

Beobachtung 5.14

Es seien **M** und **M'** Formelmengen und **M** $\subseteq$ **M'**.

- Alle Modelle von **M'** sind auch Modelle von **M**. (Def. 3.3)
- Wenn **M** unerfüllbar ist, dann ist auch **M'** unerfüllbar. (Def. 3.4)
- (Wenn **M'** erfüllbar ist, dann ist auch **M** erfüllbar.)
- Wenn eine Formel F aus **M** folgt, dann folgt F auch aus **M'** (Def. 5.1). (Monotonie der Folgerung.)

➔ Die Umkehrungen müssen nicht gelten.

Beobachtung 5.15

Es sei **M** eine Formelmenge, G eine Formel, die aus **M** folgt (**M** $\models$ G).

- Die Modelle von **M** $\cup$ {G} sind genau die Modelle von **M**. (Def. 3.3, 5.1)
- **M** ist genau dann unerfüllbar, wenn **M** $\cup$ {G} unerfüllbar ist. (Def. 3.4, 5.1)
- (**M** ist genau dann erfüllbar, wenn **M** $\cup$ {G} erfüllbar ist.)
- Eine Formel F folgt genau dann aus **M**, wenn F aus **M** $\cup$ {G} folgt. (Def. 5.1)

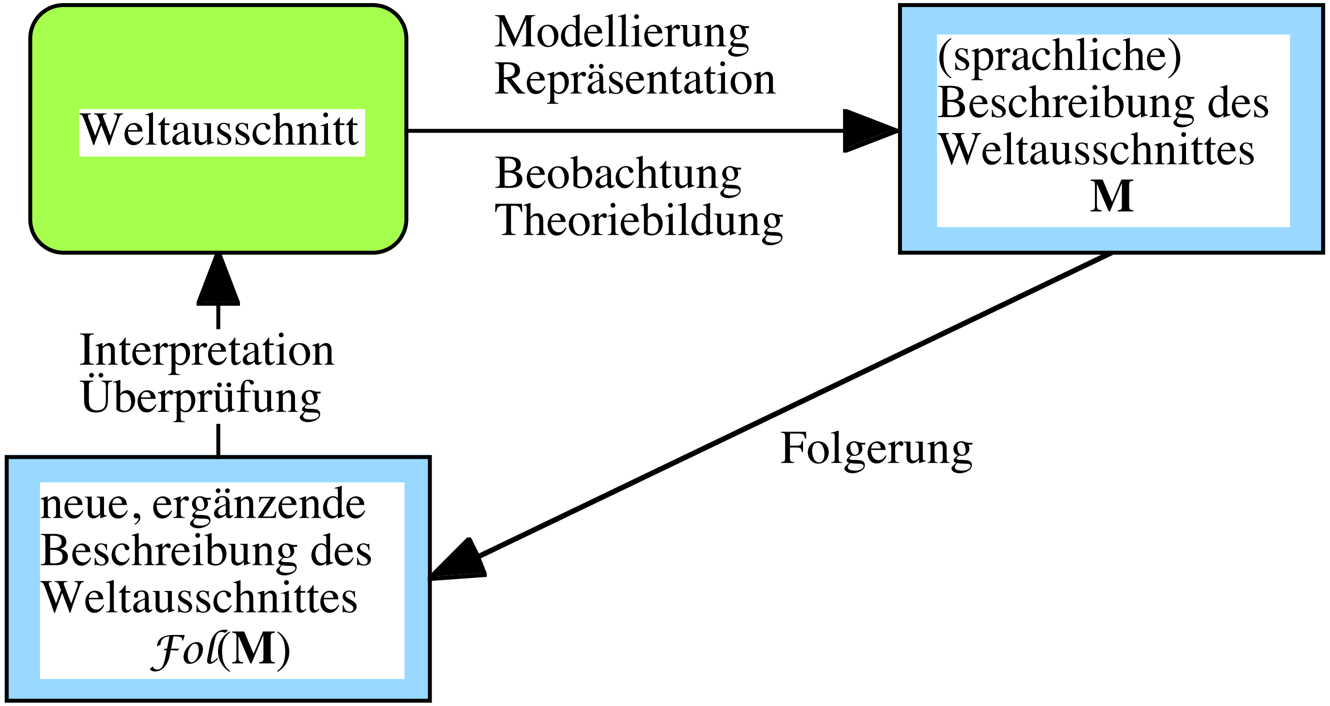
➔ Die Umkehrungen gelten alle.

➔ Die Ergänzung oder Reduktion um folgerbare Formeln verändert die semantischen Eigenschaften einer Formelmenge nicht.

Definition 5.16

Für jede (aussagenlogische) Formelmenge **M** sei $\mathcal{Fol}(\mathbf{M})$ die Menge aller aus **M** folgerbaren Formeln ($\mathcal{Fol}(\mathbf{M}) := \{\ F \in \mathcal{L}_{AL} \mid \mathbf{M} \models F\ \}$) .

$\mathcal{Taut}_{AL}$ sei die Menge aller aussagenlogischen Tautologien ($\mathcal{Taut}_{AL} = \{\ F \in \mathcal{L}_{AL} \mid\ \models F\ \}$).



Eigenschaften der Folgerung

Beobachtungen 5.17

- Für zwei Formelmengen $\mathbf{M} \subseteq \mathbf{M'}$ gilt:
  - $\mathcal{Fol}(\mathbf{M}) \subseteq \mathcal{Fol}(\mathbf{M'})$

- $\mathcal{Taut}_{\mathrm{AL}} = \mathcal{Fol}(\ \{\ \}\ ) = \mathcal{Fol}(\mathcal{Taut}_{\mathrm{AL}})$
  - Die Menge der Tautologien ist unter Folgerung abgeschlossen.
  - Die Menge der Tautologien ist die kleinste unter Folgerung abgeschlossene Formelmenge.

- Für jede Formelmenge $\mathbf{M}$ gilt:
  - $\mathbf{M} \subseteq \mathcal{Fol}(\mathbf{M})$
  - $\mathcal{Taut}_{\mathrm{AL}} \subseteq \mathcal{Fol}(\mathbf{M})$
  - $\mathcal{Fol}(\mathbf{M}) = \mathcal{Fol}(\mathcal{Fol}(\mathbf{M}))$
  - $\mathcal{Fol}(\mathbf{M}) = \mathcal{Fol}(\mathbf{M} \cup \mathcal{Taut}_{\mathrm{AL}})$
  - $\mathbf{M}$ ist genau dann unerfüllbar, wenn gilt: $\mathcal{Fol}(\mathbf{M}) = \mathcal{L}_{\mathrm{AL}}$

Endlichkeitssatz

Satz 5.18 (Endlichkeitssatz, Kompaktheitstheorem):

Eine Formelmenge **M** ist genau dann erfüllbar,
wenn jede endliche Teilmenge von **M** erfüllbar ist.

Bedeutung des Endlichkeitssatzes in der Informatik

- Aus Eigenschaften der endlichen Teilmengen kann auf eine unendliche Formelmenge geschlossen werden.

➔ Wenn **M** unerfüllbar ist,
dann existiert auch eine endliche Teilmenge **M*** von **M**, die unerfüllbar ist.

  - Von einer unendliche Gesamtmenge kann auf die Existenz einer spezifischen endlichen Teilmenge geschlossen werden.

- Der Endlichkeitssatz sagt, dass bei dem Übergang von endlichen zu unendlichen Formelmengen eigentlich nichts Neues passiert.

➔ Wiedersprüche manifestieren sich im Endlichen.

➔ Jeder Widerspruch hat eine endliche Basis.

- Bei Behandlung der Prädikatenlogik wird der Endlichkeitssatz eingesetzt.

Beweis (5.18)

- Es sei **M** eine Formelmenge.

Teil 1: Wenn **M** erfüllbar ist, dann ist jede endliche Teilmenge von **M** erfüllbar.

- Ergibt sich aus Beobachtung 5.14

Teil 2: Wenn jede endliche Teilmenge von **M** erfüllbar ist, dann ist **M** erfüllbar.
Überblick über den Beweis

- Wir bilden eine Folge von Teilmengen $\mathbf{M_i}$ von **M**, so dass jede Formel aus **M** in unendlich vielen $\mathbf{M_i}$ vorkommt.
- Wir zeigen: Zu jedem $\mathbf{M_i}$ gibt es eine endliche Teilmenge $\mathbf{M'_i}$, so dass $\mathbf{M_i}$ und $\mathbf{M'_i}$ genau dieselben Modelle haben.
- Damit finden wir eine Folge von Belegungen $\mathcal{A}_i$, die jeweils Modelle für die $\mathbf{M_i}$ sind, sich aber bei der Bewertung der Aussagensymbole widersprechen können.
- Wir bilden daraus eine neue Folge von Belegungen $\mathcal{A}'_i$, die jeweils Modelle für die $\mathbf{M_i}$ sind, und bei der Bewertung der Aussagensymbole keine Unterschiede aufweisen.
- Wir definierten daraus eine Belegung $\mathcal{A}$ und zeigen, dass $\mathcal{A}$ ein Modell für **M** ist.

Zum Selbststudium: Endlichkeitssatz – Beweis (2)

Es sei **M** eine Formelmenge, so dass jede endliche Teilmenge von **M** erfüllbar ist.

- Es sei $A_1, \dots A_i, \dots$ eine Folge aller Aussagensymbole.

  [Abzählbarkeit der Aussagensymbole, Def. 2.1]
- Für jedes $n \geq 1$ wird nun folgende Formelmenge $\mathbf{M_n}$ gebildet:

  $\mathbf{M_n} := \{F \in \mathbf{M} \mid F$ enthält kein Aussagensymbol außer $A_1, \dots A_n\}$
  - Jede Formel $F \in \mathbf{M}$ kommt in unendlich vielen $\mathbf{M_n}$ vor.
  - Es gilt: $\mathbf{M_1} \subseteq \mathbf{M_2} \subseteq \dots \subseteq \mathbf{M_n} \subseteq \dots \subseteq \mathbf{M}$
  - Zu $A_1, \dots A_n$ gibt es 2^n unterscheidbare Belegungen.
    Es gibt in $\mathbf{M_n}$ höchstens 2^{2^n} Formeln mit verschiedenen Wahrheitswertverläufen.
- Wir wählen eine Menge $\mathbf{M'_n} \subseteq \mathbf{M_n}$,

  so dass es für jede Formel $F \in \mathbf{M_n}$ eine Formel $G \in \mathbf{M'_n}$ mit $F \equiv G$ gibt und

  so dass keine zwei Formeln in $\mathbf{M'_n}$ äquivalent sind.
  - Jedes Modell für $\mathbf{M'_n}$ ist ein Modell für $\mathbf{M_n}$. (Def. 3.3)
  - $\mathbf{M'_n}$ hat höchstens 2^{2^n} Elemente (und ist endlich!).
- Nach Voraussetzung besitzt $\mathbf{M'_n} \subseteq \mathbf{M}$ damit ein Modell, wir nennen es $\mathcal{A}_n$.
  - $\mathcal{A}_n$ ist auch ein Modell für $\mathbf{M_n}$ und für alle $\mathbf{M_i}$ mit $i < n$.

Zum Selbststudium: Endlichkeitssatz–Definition eines Modells (1)

- Wenn es nur endliche viele Aussagensymbole in **M** gibt –sagen wir $A_1, \ldots, A_m$– dann ist $\mathbf{M} = \mathbf{M_m}$ und wir sind mit $\mathcal{A} := \mathcal{A}_m$ als Modell von **M** fertig.
- Wenn nicht, dann haben wir eine unendliche Folge $\mathcal{A}_i$ von Modellen für die Mengen $\mathbf{M_i}$
  - Die $\mathcal{A}_i$ können einzelnen Aussagensymbolen verschiedene Wahrheitswerte zuordnen.
- Der nächste Schritt: Die Definition einer Folge von Modellen $\mathcal{A}'_0, \mathcal{A}'_1, \ldots, \mathcal{A}'_n, \ldots$, die in der Wahrheitswertzuordnung weitgehend übereinstimmen.
  - Die Entscheidung über die Zuordnung richtet sich nach der Mehrheit der Modelle.
  - Die Folge von Indexmengen $\mathbf{I_n}$ protokolliert, welche Belegungen der Folge $\mathcal{A}_i$ für alle behandelten Aussagensymbolen mit der Mehrheit übereinstimmen.

Stufe 0: $\mathcal{A}'_0(A_i) = \mathbf{1}$, für alle i $\qquad \mathbf{I_0} := \{1, 2, \ldots\}$

Stufe n > 0: $\mathcal{A}'_n(A_k) = \mathcal{A}'_{n-1}(A_k)$, für $k < n$;

$$\mathcal{A}'_n(A_n) = \begin{cases} \mathbf{1}, \text{ falls es unendlich viele Indizes } i \in \mathbf{I_{n-1}} \text{ mit } \mathcal{A}_i(A_n) = \mathbf{1} \text{ gibt} \\ \mathbf{0}, \text{ sonst} \end{cases}$$

$\mathcal{A}'_n(A_i) = \mathbf{1}$, für $i > n$ $\qquad \mathbf{I_n} := \{i \in \mathbf{I_{n-1}} \mid \mathcal{A}_i(A_n) = \mathcal{A}'_n(A_n)\}$

- Die $\mathcal{A}'_1, \mathcal{A}'_2, \ldots, \mathcal{A}'_n, \ldots$ sind Modelle für $\mathbf{M_1} \subseteq \mathbf{M_2} \subseteq \ldots \subseteq \mathbf{M_n} \subseteq \ldots \subseteq \mathbf{M}$
- Die $\mathbf{I_n}$ enthalten unendlich viele Elemente, insbesondere für jedes n noch Indizes $i > n$.

Zum Selbststudium: Endlichkeitssatz–Definition eines Modells (2)

- Der letzte Schritt: Definition eines Modells $\mathcal{A}$ für **M** aus der Folge $\mathcal{A}'_i$

$$\mathcal{A}: \{A_1, \ldots, A_n, \ldots\} \rightarrow \{\mathbf{0}, \mathbf{1}\}$$

$$\mathcal{A}(A_n) = \mathcal{A}'_n(A_n), \text{ für alle n}$$

Behauptung: $\mathcal{A}$ ist ein Modell für **M**.

Wir müssen also zeigen, dass $\mathcal{A}$ ein Modell für jede Formel F aus **M** ist.

- Sei F eine beliebige Formel aus **M**.
  - F kann nur endlich viele Aussagensymbole haben.
    Den größten Index eines Aussagensymbols von F nennen wir k.
  - Dann ist $F \in \mathbf{M_k} \subseteq \mathbf{M_{k+1}} \subseteq \ldots$
    und $\mathcal{A}'_k, \mathcal{A}'_{k+1}, \ldots$ sind Modelle für $\mathbf{M_k}$ und damit auch für F.
  - $\mathcal{A}$ stimmt mit $\mathcal{A}'_k$ für alle Aussagensymbole $A_1, \ldots, A_k$ überein.

    $$\mathcal{A}(A_1) = \mathcal{A}'_k(A_1), \ldots, \mathcal{A}(A_k) = \mathcal{A}'_k(A_k)$$

    Damit ist $\mathcal{A}$ auch ein Modell für F.

Zum Selbststudium: Endlichkeitssatz: Anmerkungen

zur Definition eines Modells für M

- Der Beweis des Endlichkeitssatzes ist nicht konstruktiv.
- Die falls-Bedingung der Konstruktionsvorschrift ist nicht algorithmisch effektiv, denn es wäre eine unendliche Menge von Indizes zu prüfen, und auf der Basis dieser Prüfung eine Berechnung durchzuführen.
- Diese Berechnung kann – gegebenenfalls – selbst wieder eine nicht-endliche Anzahl von Berechnungsschritten beinhalten.

➔ gedankliche Konstruktion
➔ Existenzbeweis in mathematischer Tradition
➔ nicht programmierbares Konstruktionsverfahren

Zum Selbststudium: Exkurs: Nicht-konstruktive Beweise

Theorem:

Es gibt Lösungen der Gleichung $x^y = z$,
mit z rational, und x und y irrational, d.h. $z \in \mathbb{Q}$ und $x, y \in \mathbb{R} \setminus \mathbb{Q}$.

Beweis:

$\sqrt{2}$ ist irrational und $\sqrt{2}^{\sqrt{2}}$ ist rational oder irrational.

- Fall 1. $\sqrt{2}^{\sqrt{2}}$ ist rational.
  Sei $x = \sqrt{2}$ und $y = \sqrt{2}$, dann ist $z = x^y = \sqrt{2}^{\sqrt{2}}$ rational nach Voraussetzung (Fall 1).
- Fall 2. $\sqrt{2}^{\sqrt{2}}$ ist irrational.
  Sei $x = \sqrt{2}^{\sqrt{2}}$ und $y = \sqrt{2}$, dann ist $z = x^y = (\sqrt{2}^{\sqrt{2}})^{\sqrt{2}} = (\sqrt{2})^{\sqrt{2} \cdot \sqrt{2}} = (\sqrt{2})^2 = 2 \in \mathbb{Q}$.

Beobachtungen / Anmerkungen

1. Der Beweis liefert die Existenz einer Lösung der Gleichung $x^y = z$ (unter den geforderten Bedingungen), ohne dass eine Lösung angegeben (konstruiert) würde. Der Beweis zeigt, dass eine der beiden Lösungen die Bedingungen erfüllt, lässt aber offen, welche.
2. Intuitionistische / Konstruktivistische Mathematik sieht konstruktive Beweisverfahren als notwendig an. [Zur Vertiefung (Master-Studiengang Informatik oder Mathematik):
   M. Dummett (1977). *Elements of intuitionism*. Oxford: Clarendon Press.]

Eine Anwendung des Endlichkeitssatzes

Satz 5.19

- Es seien $A_1, \ldots, A_n, \ldots$ verschiedene Aussagensymbole und

  $\mathbf{M} := \{A_1 \vee A_2, \neg A_1 \vee \neg A_2 \vee \neg A_3, A_2 \vee A_3, \neg A_2 \vee \neg A_3 \vee \neg A_4, \ldots\}$

  $= \{A_i \vee A_{i+1} \mid i \in \mathbb{N}\} \cup \{\neg A_i \vee \neg A_{i+1} \vee \neg A_{i+2} \mid i \in \mathbb{N}\}$

  Dann ist **M** erfüllbar.

Beweis

- Aufgrund des Endlichkeitssatzes reicht es zu zeigen, dass jede endliche Teilmenge von **M** erfüllbar ist.
- Der Beweis erfolgt über vollständige Induktion über den größten Index eines Aussagensymbols in der Teilmenge.
- *Induktionsanfang*: Ist der größte Index eines Aussagensymbols in $\mathbf{M'} \subseteq \mathbf{M}$ kleiner als 3, dann ist **M'** erfüllbar, denn die leere Menge und $\{A_1 \vee A_2\}$sind erfüllbar.
- *Induktionsannahme*: Alle Teilmengen von **M**, bei denen der größte Index eines Aussagensymbols $< k$ ist, sind erfüllbar.

Induktionsschritt (5.19)

- *Induktionsschritt*: Sei **M'** eine endliche Teilmenge von **M**, so dass k der größte Index eines Aussagensymbols in **M'** ist.
  - Sei **M"** := {$F \in$ **M'** | A_k ist keine Teilformel von F}
  - Gemäß Induktionsannahme hat **M"** hat ein Modell $\mathcal{A}''$.
  - Wir definieren die Belegung $\mathcal{A}'$ wie folgt:

i < k–2 $\quad \mathcal{A}'(A_i) = \mathcal{A}''(A_i)$

$$\mathcal{A}'(A_{k-2}) = \begin{cases} \mathcal{A}''(A_{k-2}), \text{ falls } A_{k-2} \text{ in } \mathbf{M''} \text{ vorkommt} \\ \mathbf{0}, \text{ sonst} \end{cases}$$

$$\mathcal{A}'(A_{k-1}) = \begin{cases} \mathcal{A}''(A_{k-1}), \text{ falls } A_{k-1} \text{ in } \mathbf{M''} \text{ vorkommt} \\ \mathbf{0}, \text{ sonst} \end{cases}$$

$$\mathcal{A}'(A_k) = \begin{cases} \mathbf{0}, \text{ falls } \mathcal{A}''(A_{k-1}) = \mathbf{1} \\ \mathbf{1}, \text{ sonst} \end{cases}$$

i > k (dann kommt A_i nicht in **M'** vor) $\quad \mathcal{A}'(A_i) = \mathbf{0}$

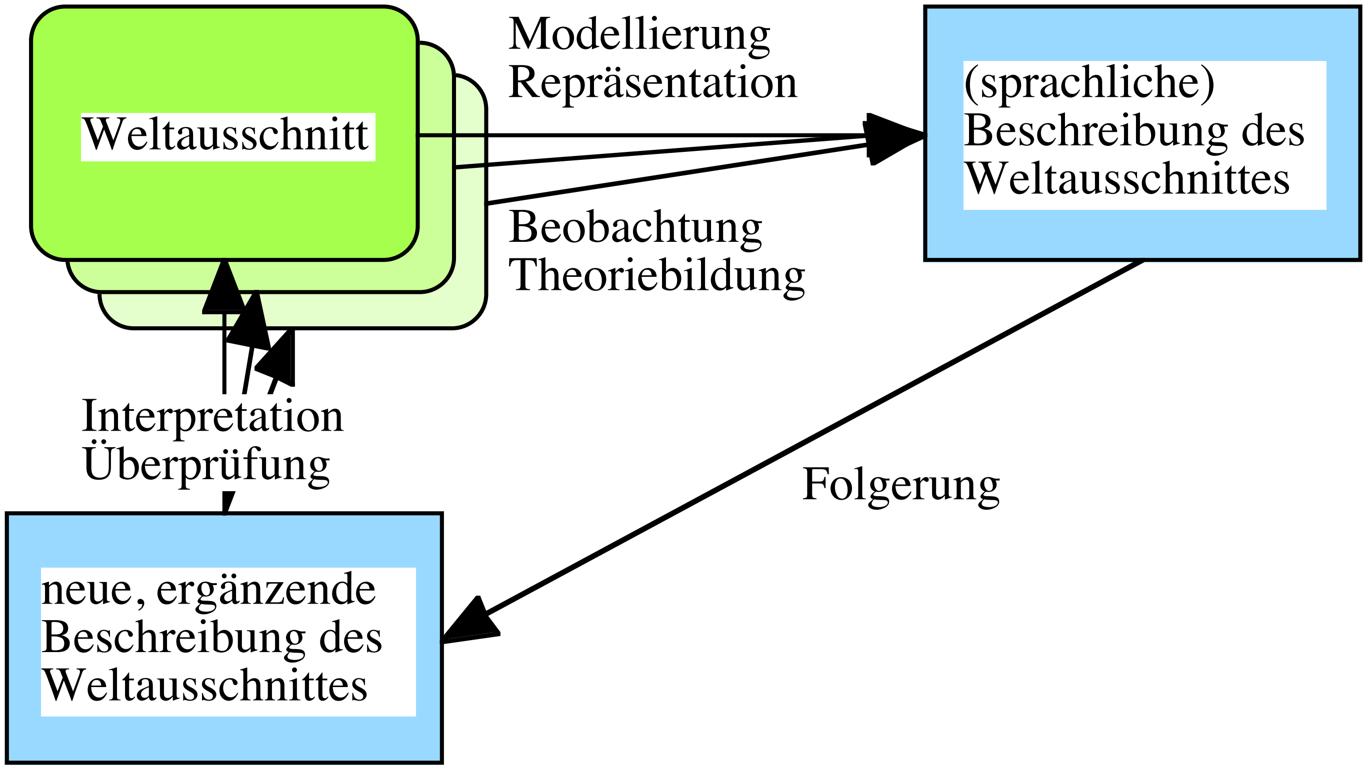
  - Gemäß dieser Definition ist $\mathcal{A}'$ ein Modell von **M"** und von $\{A_{k-1} \vee A_k, \neg A_{k-2} \vee \neg A_{k-1} \vee \neg A_k\}$. Damit ist $\mathcal{A}'$ auch ein Modell von **M'**.

Resümee: Also hat jede endliche Teilmenge von **M** ein Modell.

Wichtige Konzepte in diesem Foliensatz

- (logische) Folgerung: zwischen Formeln, zwischen Formelmengen und Formeln
- Beziehung von Folgerung zu Allgemeingültigkeit, Unerfüllbarkeit, Äquivalenz
- Endlichkeitssatz / Kompaktheit (Wiedersprüche manifestieren sich im Endlichen)

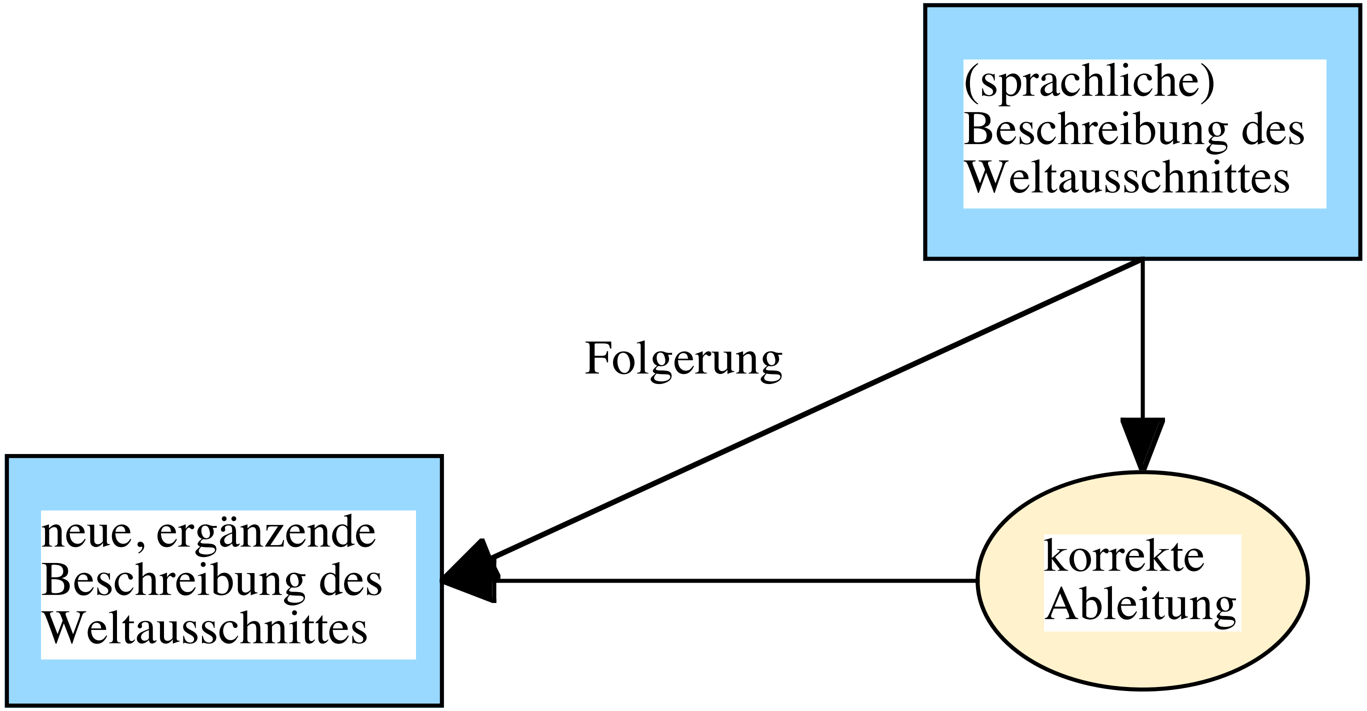
Aussagenlogik: Beweisbarkeit
Ableitung und Widerlegung



Ableitung und Widerlegung (deduction & refutation)

- Grundidee:

  Auf der Basis der syntaktischen Eigenschaften von Formeln und Formelmengen auf ihre semantischen Eigenschaften schließen, statt die Belegungen / Wahrheitswertverläufe der Formeln direkt zu prüfen.



Zum Selbststudium

Wenn korrekte Ableitung nichts anderes ist als Folgerung, warum brauchen wir es dann?

- Folgerung ist eine Relation zwischen Formeln, die über die Wahrheitswertverläufe definiert ist. Zur Überprüfung liegt es damit nahe, die Wahrheitstafeln zu verwenden. Dieses Verfahren ist für die Aussagenlogik auch weitgehend praktikabel.
- Für komplexere Logiken ist ein entsprechendes Vorgehen nicht mehr möglich. Z.B. ist es für prädikatenlogische Formeln im allgemeinen nicht möglich, (endliche) Tabellen aufzuschreiben, die alle möglichen Interpretationen aufführen.
- Der Ableitungsbegriff, den wir einführen werden, legt andere Berechnungsverfahren für die logischen Eigenschaften nahe. Vollständigkeit und Korrektheit zeigen, dass diese Verfahren unbesorgt anwendbar sind.
- Ableitungen kann man auch in anderen Logiken definieren und damit dann Folgerungsbeziehungen zwischen Formeln ausrechnen. Die Einführung von Ableitungsverfahren in der Aussagenlogik dient vor allem dazu, das Zusammenspiel der verschiedenen Aspekte an einem einfachen Beispiel vorzustellen. Dieses soll dann die Übertragung auf die Prädikatenlogik erleichtern.

Semantische Eigenschaften und syntaktische Muster: Beispiele

- Unabhängig von der Bedeutung von F gilt:
  - $F \wedge \neg F$ ist unerfüllbar
  - $F \vee \neg F$ ist allgemeingültig

  ➔ Es ist unnötig, die Belegungen zu prüfen,
  wenn eines dieser syntaktischen Muster vorliegt.
- Entsprechend: Syntaktische Tautologieprüfung bei Klauseln:
  Prüfung auf Existenz von komplementären Literalen

  ➔ Nutzung von syntaktischen Mustern mit bekannten semantischen Eigenschaften.

  ➔ Was sind syntaktische Muster (und wie erkennen wir sie)?

Definition 6.1 ((Formel-) Substitution)

Eine *(Uniforme Formel-) Substitution* ist eine rekursiv definierte Funktion $\mathrm{sub}: \mathcal{L}_{AL} \rightarrow \mathcal{L}_{AL}$.

(s. Prinzip der strukturellen Rekursion)

- Aussagensymbole A_i werden auf Formeln $\mathrm{sub}(A_i)$ abgebildet.
- Für Formeln $F, G \in \mathcal{L}_{AL}$ gilt:

  $\mathrm{sub}(\neg F) = \neg \mathrm{sub}(F)$

  $\mathrm{sub}((F \wedge G)) = (\mathrm{sub}(F) \wedge \mathrm{sub}(G))$ $\qquad$ $\mathrm{sub}((F \vee G)) = (\mathrm{sub}(F) \vee \mathrm{sub}(G))$

  $\mathrm{sub}((F \Rightarrow G)) = (\mathrm{sub}(F) \Rightarrow \mathrm{sub}(G))$ $\qquad$ $\mathrm{sub}((F \Leftrightarrow G)) = (\mathrm{sub}(F) \Leftrightarrow \mathrm{sub}(G))$
- Ist **M** eine Formelmenge, dann ergibt die Substitution die Formelmenge

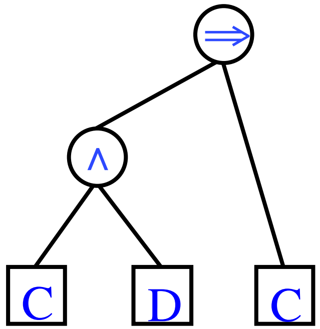
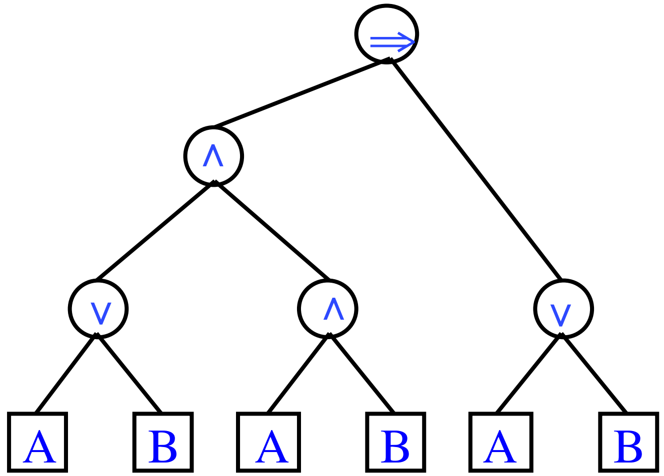
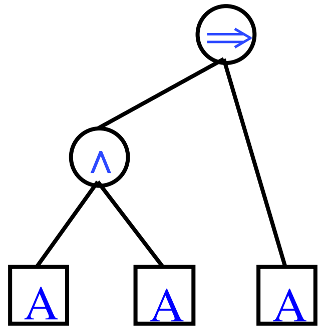
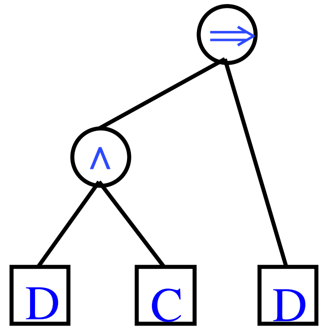
  $\mathbf{sub}(\mathbf{M}) = \{\mathrm{sub}(F) \mid F \in \mathbf{M}\}$

  ➔ Durch eine (Formel-)Substitution wird jede Formel F auf eine Formel $\mathrm{sub}(F)$ abgebildet, wobei *jedes* Vorkommen eines Aussagensymbols A_i in F durch die entsprechende Formel $\mathrm{sub}(A_i)$ ersetzt wird.

  ➔ *Uniformität:* Jedes Aussagensymbol wird immer wieder durch die gleiche Formel ersetzt.

  ➔ Die Substitution ist durch die Zuordnung von Formeln zu den Aussagensymbolen eindeutig bestimmt.

Beispiele für Substitutionen

	$sub_1(C) = (A \vee B)$ $sub_1(D) = (A \wedge B)$	$sub_2(C) = A$ $sub_2(D) = A$	$sub_3(C) = D$ $sub_3(D) = C$
$F =$ $(C \wedge D) \Rightarrow C$	$sub_1(F) =$ $((A \vee B) \wedge (A \wedge B)) \Rightarrow (A \vee B)$	$sub_2(F) =$ $(A \wedge A) \Rightarrow A$	$sub_3(F) =$ $(D \wedge C) \Rightarrow D$
			
M = $\{(C \wedge D) \Rightarrow C,$ $(C \wedge D) \Rightarrow D\}$	**sub_1(M) =** $\{((A \vee B) \wedge (A \wedge B)) \Rightarrow (A \vee B),$ $((A \vee B) \wedge (A \wedge B)) \Rightarrow (A \wedge B)\}$	**sub_2(M) =** $\{(A \wedge A) \Rightarrow A\}$	**sub_3(M) =** $\{(D \wedge C) \Rightarrow D,$ $(D \wedge C) \Rightarrow C\}$

Zum Selbststudium: Zusammenhang zwischen Ersetzung und Substitution

Definition 4.8 (Ersetzung) (*Umformuliert*)

Seien G' und G" zwei Formeln und sei H' eine Formel mit der Teilformel G' und H" eine Formel mit Teilformel G". Wenn der einzige Unterschied zwischen H' und H" darin besteht, dass an einer oder mehreren Positionen, an denen in H' die Teilformel G steht in H" die Teilformel G" steht, dann heißt H" *durch Ersetzung von* G' *durch* G" *aus* H' *hervorgegangen*.

Definition 6.1 ((Formel-) Substitution) (*Umformuliert*)

Eine *(uniforme Formel-) Substitution* ist eine rekursiv definierte Funktion sub: $\mathcal{L}_{AL} \rightarrow \mathcal{L}_{AL}$.
(s. Prinzip der strukturellen Rekursion)

- Aussagensymbole A_i werden auf Formeln sub(A_i) abgebildet.
- Für Formeln F, F' $\in \mathcal{L}_{AL}$ gilt:

  sub(¬F) = ¬sub(F)

  sub((F ∧ F')) = (sub(F) ∧ sub(F')) sub((F ∨ F')) = (sub(F) ∨ sub(F'))

  sub((F ⇒ F')) = (sub(F) ⇒ sub(F')) sub((F ⇔ F')) = (sub(F) ⇔ sub(F'))

Beobachtung

Ist H" durch Ersetzung von G' durch G" aus H' hervorgegangen, dann gibt es eine Formel H, ein Aussagensymbol A, das nicht in H' oder H" vorkommt, und zwei Substitutionen sub' und sub", so dass gilt: sub'(A) = G', sub"(A) = G", für alle Aussagesymbole B ≠ A gilt: sub'(B) = sub'(B) = B und sub'(H) = H', sub"(H) = H".

Substitutionen und Allgemeingültigkeit

Satz 6.2

Wenn eine Formel F allgemeingültig und sub eine Substitution ist, dann ist auch die Formel $G := sub(F)$ allgemeingültig.

Beweis

- Es sei $\mathcal{A}$ eine Belegung,
  $A_1, \ldots, A_n$ seien die in F vorkommenden Aussagensymbole.
- Wir betrachten nun eine Belegung $\mathcal{A}'$, für die gilt
  $\mathcal{A}'(A_i) = \mathcal{A}(sub(A_i))$.
  - So eine Belegung gibt es, denn die Aussagensymbole sind kontingente Formeln.
- Es gilt: $\mathcal{A}'(F) = \mathcal{A}(G)$.
  [zu zeigen durch strukturelle Induktion über F als Übung]
- Da F allgemeingültig ist, ist $\mathcal{A}'(F) = \mathbf{1}$, also auch $\mathcal{A}(G) = \mathbf{1}$

Vorsicht: die Umkehrung gilt nicht

- Gegenbeispiel: $(A \vee B)$ ist nicht allgemeingültig, aber $(A \vee \neg A)$ ist allgemeingültig.

Zum Selbststudium: Substitutionen — Unerfüllbarkeit etc.

Sätze 6.3

Es sei sub eine Substitution.

- Wenn eine Formel F unerfüllbar ist, dann ist auch die Formel sub(F) unerfüllbar.
- Wenn die Formeln F und G äquivalent sind, dann sind auch sub(F) und sub(G) äquivalent.
- Wenn eine Formelmenge **M** unerfüllbar ist, dann ist auch die Formelmenge **sub**(**M**) unerfüllbar.
- Wenn eine Formel F aus einer Formelmenge **M** folgt, dann folgt auch sub(F) aus **sub**(**M**).

Beweise: ganz analog zu 6.2 und zur Übung.

Aber zu beachten ist:

- Die Formeln F und sub(F) sind im Allgemeinen nicht äquivalent !
- Vergleichen Sie Substitutionen mit den spezifischen Voraussetzungen des Ersetzbarkeitstheorems.

Definition 6.4 (Darstellung von Inferenzregeln)

- Es seien $F_1,\ldots,F_n$ und G Formeln.
- $R = \dfrac{F_1,\ldots,F_n}{G}$

  stellt eine *Inferenzregel* dar,
  die F_i werden als *Prämissen*, G wird als *Konklusion* der Regel bezeichnet.
- Eine alternative Schreibweise: $R = F_1,\ldots,F_n \therefore G$

Inferenzregeln

- sind keine Formeln der Aussagenlogik.
- spezifizieren Muster in Formelmengen
- sind als Ableitungs- bzw. Schlussfiguren aufzufassen.

Modus Ponens

Modus Ponens ist die Inferenzregel $MP = \dfrac{A, A \Rightarrow B}{B}$

Anwendbarkeit einer Inferenzregel

- Inferenzregeln spezifizieren Muster in Formelmengen
- Inferenzregeln sind als Ableitungs- bzw. Schlussfiguren aufzufassen.

Definition 6.5

- Es seien $F_1, \ldots, F_n$ und G Formeln und R = $\dfrac{F_1, \ldots, F_n}{G}$ eine Inferenzregel.
- *Das Muster* $F_1, \ldots, F_n$ *liegt in einer Formelmenge* **M** genau dann *vor*, wenn es eine Substitution sub gibt, so dass $\mathbf{sub}(\{F_1, \ldots, F_n\}) \subseteq \mathbf{M}$
  - Andere Sprechweise: Die Regel R kann auf die Formelmenge **M** *angewendet* werden.
- In diesem Fall kann die Formel $sub(G)$ aus **M** mit R *in einem Schritt abgeleitet* werden.
  - Andere Schreib- / Sprechweisen:
    $sub(G)$ ist mit R *direkt ableitbar* aus **M**
    $sub(G)$ ist mit R *direkt ableitbar* aus $sub(F_1), \ldots, sub(F_n)$
    $\mathbf{M} \vdash_R sub(G)$ bzw. $\{sub(F_1), \ldots, sub(F_n)\} \vdash_R sub(G)$

Modus ponens

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

- $\mathbf{M_1} = \{A, A \Rightarrow B, A \Rightarrow C\}$

  $sub_1(A) = A$, $sub_1(B) = B$

  $\mathbf{sub_1}(\{A, A \Rightarrow B\}) = \{A, A \Rightarrow B\} \subseteq \mathbf{M_1}$

  $\mathbf{M_1} \vdash_{MP} B$

- $\mathbf{M_1} = \{A, A \Rightarrow B, A \Rightarrow C\}$

  $sub_2(A) = A$, $sub_2(B) = C$

  $\mathbf{sub_2}(\{A, A \Rightarrow B\}) = \{A, A \Rightarrow C\} \subseteq \mathbf{M_1}$

  $\mathbf{M_1} \vdash_{MP} C$

- $\mathbf{M_3} = \{A \wedge B, (A \wedge B) \Rightarrow \neg(C \vee D)\}$

  $sub_3(A) = A \wedge B$, $sub_3(B) = \neg(C \vee D)$

  $\mathbf{sub_3}(\{A, A \Rightarrow B\}) = \{A \wedge B, (A \wedge B) \Rightarrow \neg(C \vee D)\} \subseteq \mathbf{M_3}$

  $\mathbf{M_3} \vdash_{MP} \neg(C \vee D)$

Zum Selbststudium: Regeln und Ableitungen

Wie man hier sieht, gehen den Formalisten schnell die Wörter aus …

- Regeln gibt es in kontextfreien Grammatiken, mit ihnen werden Wörter aus einem Startsymbol 'abgeleitet', man kann auch sagen 'generiert'.
- (Inferenz-)Regeln gibt es aber auch in logischen Kalkülen, mit ihnen werden Formeln aus Formelmengen 'abgeleitet', man kann auch sagen 'erschlossen'. Beweise sind spezielle Fälle solcher Ableitungen.
- Man sieht hier, wie unterschiedliche Typen von 'Regeln' zu unterschiedlichen Sequenzen von Regelanwendungen führen, die als 'Ableitungen' bezeichnet werden.
- Entsprechende Mehrdeutigkeiten gibt es aber auch in der natürlichen Sprache, man muss einfach akzeptieren, dass das so ist.

Übrigens

- Regeln in kontextfreien Grammatiken haben immer eine 'Prämisse' (ein Non-Terminalsymbol)
- Inferenz-Regeln können mehr oder weniger Prämissen haben. (Inferenzregeln ohne Prämissen sind Axiome). In jedem logischen Kalkül gibt es aber mindestens eine Regel, die zwei Prämissen hat (sehr oft ist das der Modus Ponens).

Korrektheit des Modus Ponens

- Wahrheitsverhalten bei Anwendung des Modus Ponens: $\dfrac{A, A \Rightarrow B}{B}$

	F	G	(F ⇒ G)
$\mathcal{A}_1$	**0**	**0**	**1**
$\mathcal{A}_2$	**0**	**1**	**1**
$\mathcal{A}_3$	**1**	**0**	**0**
$\mathcal{A}_4$	**1**	**1**	**1**

Für alle Belegungen mit $\mathcal{A}(F) = \mathbf{1}$ und $\mathcal{A}(F \Rightarrow G) = \mathbf{1}$ gilt auch $\mathcal{A}(G) = \mathbf{1}$.

➔ Modus Ponens erhält Wahrheit.

➔ Wenn die Prämissen zu MP wahr sind,
dann ist die Wahrheit der Konklusion garantiert.

- Falls **M** eine Formelmenge und $\mathbf{M} \vdash_{MP} G$, dann $\mathbf{M} \models G$

➔ Was durch Modus Ponens aus einer Menge **M** abgeleitet werden kann,
ist auch aus **M** folgerbar.

Definition 6.6 (Korrektheit einer Inferenzregel)

Eine Inferenzregel R = $\dfrac{F_1,\ldots,F_n}{G}$ heißt genau dann *korrekt* (sound), falls für alle Formelmengen **M** und alle Formeln H gilt: wenn $\mathbf{M} \vdash_R H$, dann $\mathbf{M} \vDash H$.

➔ Was mit einer korrekten Regel aus einer Formelmenge **M** abgeleitet werden kann, ist auch aus **M** folgerbar.

Sätze 6.7

- Wenn $F_1, \ldots, F_n$ allgemeingültig, R eine korrekte Regel und $\{ F_1, \ldots, F_n \} \vdash_R G$, dann ist G allgemeingültig.
- Eine Inferenzregel R = $\dfrac{F_1,\ldots,F_n}{G}$ ist genau dann korrekt, wenn $\{F_1,\ldots, F_n\} \vDash G$ gilt.

Beweise zur Übung

Korrekte Inferenzregeln: Beispiele

- Modus tollens (MT): $\dfrac{\neg B,\ A \Rightarrow B}{\neg A}$
- Hypothetischer Syllogismus (HS): $\dfrac{A \Rightarrow B,\ B \Rightarrow C}{A \Rightarrow C}$
- Konjunktionseinführung (KE): $\dfrac{A,\ B}{A \wedge B}$
- Konjunktionslöschung (KL): $\dfrac{A \wedge B}{A}$ $\qquad$ $\dfrac{A \wedge B}{B}$
- Disjunktionseinführung (DE): $\dfrac{A}{A \vee B}$ $\qquad$ $\dfrac{B}{A \vee B}$
- Disjunktiver Syllogismus (DS): $\dfrac{\neg B,\ A \vee B}{A}$ $\qquad$ $\dfrac{\neg A,\ A \vee B}{B}$
- Konstruktives Dilemma (KD): $\dfrac{A \Rightarrow B,\ C \Rightarrow D,\ A \vee C}{B \vee D}$

Definition 6.8 ($Abl_{\mathcal{R}}(\mathbf{M})$: Menge der mit $\mathcal{R}$ aus **M** ableitbaren Formeln)

- Es sei **M** eine Formelmenge und $\mathcal{R}$ eine Menge von Inferenzregeln.
- $Abl_{\mathcal{R}}^{0}(\mathbf{M}) := \mathbf{M}$
- $Abl_{\mathcal{R}}^{1}(\mathbf{M}) := \mathbf{M} \cup \{F \in \mathcal{L}_{AL} \mid R \in \mathcal{R} \text{ und } \mathbf{M} \vdash_R F\}$
- $Abl_{\mathcal{R}}^{n}(\mathbf{M}) := Abl_{\mathcal{R}}^{n-1}(\mathbf{M}) \cup \{F \in \mathcal{L}_{AL} \mid R \in \mathcal{R} \text{ und } Abl_{\mathcal{R}}^{n-1}(\mathbf{M}) \vdash_R F\}$
- $Abl_{\mathcal{R}}(\mathbf{M}) := \bigcup_{n \geq 0} Abl_{\mathcal{R}}^{n}(\mathbf{M})$
- Ist F Element von $Abl_{\mathcal{R}}^{n}(\mathbf{M})$ dann ist F mit $\mathcal{R}$ (in n Schritten) aus **M** ableitbar.

Beobachtungen zu 6.8

- Bei mehrschrittigen Ableitungen kann jeder Schritt auf beliebige zuvor eingeführte oder abgeleitete Formeln zugreifen.
- Sind alle Inferenzregeln in $\mathcal{R}$ korrekt, dann ist $Abl_{\mathcal{R}}(\mathbf{M}) \subseteq \mathcal{Fol}(\mathbf{M})$.

Definition 6.9 (Kalkül)

Ein Kalkül C der Aussagenlogik wird gebildet durch:

- Eine Logik-Sprache $\mathcal{L}_{\mathrm{AL}}$,
  spezifiziert durch ein Alphabet und Regeln zur Bildung von wohlgeformten Formeln,
- Eine ausgezeichnete Teilmenge von $\mathcal{L}_{\mathrm{AL}}$, genannt die *Axiome* von C.
- Eine endliche Menge von Inferenzregeln für $\mathcal{L}_{\mathrm{AL}}$.

➔ C kann als Tripel $C = (\mathcal{L}_{\mathrm{AL}}, \mathcal{A}x, \mathcal{R})$ dargestellt werden:

(1) Sprache $\mathcal{L}_{\mathrm{AL}}$,

(2) Menge von Axiomen $\mathcal{A}x$,

(3) Menge von Inferenzregeln $\mathcal{R}$

Ableitung in einem Kalkül

Definition 6.10

Sei $\mathcal{C}$ ein Kalkül der Aussagenlogik, **M** eine Formelmenge und G eine Formel aus $\mathcal{L}_{AL}$.
Eine (nicht-leere) endliche Folge von Formeln $F_1, \ldots, F_n$ heißt genau dann eine aussagen-logische *Ableitung* bzgl. $\mathcal{C}$ von G aus **M**, wenn $G = F_n$ und für jedes k ($1 \leq k \leq n$) wenigstens eine der folgenden Bedingungen gilt:

1. $F_k \in \mathbf{M}$,
2. $F_k = sub(H)$, wobei H ein Axiom aus $\mathcal{C}$ und sub eine Substitution ist,
3. $\{F_1, \ldots, F_{k-1}\} \vdash_R F_k$, wobei R eine Inferenzregel aus $\mathcal{C}$ ist.

Gibt es eine Ableitung von G bzgl. $\mathcal{C}$ aus **M**, dann sagen / schreiben wir auch:

- G ist in $\mathcal{C}$ *ableitbar* aus **M**; $\quad \mathbf{M} \vdash_{\mathcal{C}} G$
- $Abl_{\mathcal{C}}(\mathbf{M}) := \{F \in \mathcal{L}_{AL} \mid \mathbf{M} \vdash_{\mathcal{C}} G\}$ ist die Menge aller aus **M** mit $\mathcal{C}$ ableitbaren Formeln.

Ableitung: Beispiel

M:= $\{A \vee B, (A \vee B) \Rightarrow C, C \Rightarrow D, D \Rightarrow E\}$

Kalkül: $\mathcal{C}_0 = (\mathcal{L}_{AL}, \{\}, \{MP\})$, Regel: MP: $\dfrac{A, A \Rightarrow B}{B}$

Nr		Formel	Begründung für Aufnahme	
(1)	$F_1 :=$	$A \vee B$	aus **M**	
(2)	$F_2 :=$	$(A \vee B) \Rightarrow C$	aus **M**	
(3)	$F_3 :=$	C	(1), (2) MP	$sub_3(A) = (A \vee B)$, $sub_3(B) = C$
(4)	$F_4 :=$	$C \Rightarrow D$	aus **M**	
(5)	$F_5 :=$	D	(3), (4) MP	$sub_5(A) = C$, $sub_5(B) = D$
(6)	$F_6 :=$	$D \Rightarrow E$	aus **M**	
(7)	$F_7 :=$	E	(5), (6) MP	$sub_7(A) = D$, $sub_7(B) = E$

- **M** $\vdash_{\mathcal{C}_0} E$, E ist aus der Formelmenge **M** ableitbar.
- Bei dieser Ableitung wurde nicht auf Axiome zurückgegriffen.

Beobachtung 6.11

1. Wenn G ein Axiom von C ist oder $G \in \mathbf{M}$, so gilt: $\mathbf{M} \vdash_C G$
2. Wenn $\mathbf{M_1}$ und $\mathbf{M_2}$ Formelmengen sind und G eine Formel ist, so gilt für jeden Kalkül C:

   Falls $\mathbf{M_1} \subseteq \mathbf{M_2}$ und $\mathbf{M_1} \vdash_C G$, dann $\mathbf{M_2} \vdash_C G$. [Monotonie]
3. Wenn $\mathbf{M_1}$ eine Formelmenge und G eine Formel ist, so gilt:

   $\mathbf{M_1} \vdash_C G$ gdw.

   eine endliche Teilmenge $\mathbf{M_2} \subseteq \mathbf{M_1}$ existiert mit $\mathbf{M_2} \vdash_C G$. [Kompaktheit]

Beweisideen

- zu 1. Länge der Ableitung ist 1, denn schon für $F_1 = G$ kann eine der Bedingungen 1. oder 2. in der Definition von Ableitung erreicht werden.
- zu 2. Wenn eine Ableitung aus $\mathbf{M_1}$ möglich ist, kann die gleiche Ableitung auch gebildet werden, wenn zusätzliche Formeln (aus $\mathbf{M_2}$) zur Verfügung stehen.
- zu 3. Da Ableitungen endlich sind, können auch nur endlich viele Elemente aus $\mathbf{M_1}$ (und auch nur endlich viele Axiome) in der Ableitung von G berücksichtigt werden. Also muss ein endliches $\mathbf{M_2} \subseteq \mathbf{M_1}$ ausreichen, um die Ableitung für G aufzubauen.

Axiom

αξιομα Ein Satz, der keines Beweises bedarf.
Ein logischer Satz, den kein Vernünftiger bezweifelt. (Euklid)

Axiome für die Aussagenlogik

- Die Idee: Formeln der Aussagenlogik, „die keines Beweises bedürfen".
- Eine Basismenge von Tautologien um weitere Tautologien abzuleiten.

Beweisbarkeit: Ableitbarkeit aus der leeren Menge (Definition 6.12)

Eine Formel G, für die gilt $\emptyset \vdash_C$ G, heißt ***beweisbar*** in C. (in Zeichen: $\vdash_C$ G)

- Beweisbare Formeln werden auch als *Theoreme* von C bezeichnet.
- Eine Ableitung, die nur Axiome und Regeln von C verwendet, heißt auch *formaler Beweis in C*.
  - Beweise sind nicht „Ableitungen aus dem Nichts".
  - ➔ „ableitbar ausschließlich aus den Axiomen, ohne Bezug auf weitere Formeln"

Hilbert-Systeme

- Klasse aussagenlogischer Kalküle, die ausschließlich den Modus Ponens als Inferenzregel verwenden.
  Unterschiede existieren in der Festlegung der Axiome.
- Modus Ponens: $\dfrac{A, A \Rightarrow B}{B}$
- Axiome für das System $\mathcal{H}$
  - $H_1 :=$ $(A \Rightarrow (B \Rightarrow A))$
  - $H_2 :=$ $((A \Rightarrow B) \Rightarrow ((A \Rightarrow (B \Rightarrow C)) \Rightarrow (A \Rightarrow C)))$
  - $H_3 :=$ $(A \Rightarrow (B \Rightarrow (A \wedge B)))$
  - $H_{4a} :=$ $(A \Rightarrow (A \vee B))$
  - $H_{4b} :=$ $(B \Rightarrow (A \vee B))$
  - $H_5 :=$ $((A \Rightarrow B) \Rightarrow ((A \Rightarrow \neg B) \Rightarrow \neg A))$
  - $H_{6a} :=$ $((A \wedge B) \Rightarrow A)$
  - $H_{6b} :=$ $((A \wedge B) \Rightarrow B)$
  - $H_7 :=$ $((A \Rightarrow C) \Rightarrow ((B \Rightarrow C) \Rightarrow ((A \vee B) \Rightarrow C)))$
  - $H_8 :=$ $(\neg\neg A \Rightarrow A)$

Die Axiome von $\mathcal{H}$

Beispiel

- $(A \Rightarrow (B \Rightarrow A))$ stellt durch Substitutionen u.a. die folgenden Formeln bereit:
  $(A \Rightarrow (B \Rightarrow A))$,
  $(A \Rightarrow (A \Rightarrow A))$,
  $(B \Rightarrow (A \Rightarrow B))$,
  $(A \Rightarrow (C \Rightarrow A))$
  $((A \vee B) \Rightarrow (C \Rightarrow (A \vee B)))$,
  $((A \wedge B) \Rightarrow (C \Rightarrow (A \wedge B)))$, …

- Unterschiedliche Axiomenmengen des Hilbert-Stils sind u.a. durch unterschiedliche Basissätze von Junktoren (‘Junktorenbasen’) bedingt.

Beweisbeispiel in $\mathcal{H}$

- $H_1 \quad (A \Rightarrow (B \Rightarrow A))$
- $H_2 \quad ((A \Rightarrow B) \Rightarrow ((A \Rightarrow (B \Rightarrow C)) \Rightarrow (A \Rightarrow C)))$

Satz 6.13: $\vdash_{\mathcal{H}} \mathbf{D} \Rightarrow \mathbf{D}$

Beweis

Nr.	Formaler Beweis	Begründung für die Aufnahme	
(1) $F_1 :=$	$(D \Rightarrow (D \Rightarrow D)) \Rightarrow ((D \Rightarrow ((D \Rightarrow D) \Rightarrow D)) \Rightarrow (D \Rightarrow D))$	Axiom H_2	$sub_1(A) = D$ $sub_1(B) = (D \Rightarrow D)$ $sub_1(C) = D$
(2) $F_2 :=$	$(D \Rightarrow (D \Rightarrow D))$	Axiom H_1	$sub_2(A) = D$, $sub_2(B) = D$
(3) $F_3 :=$	$((D \Rightarrow ((D \Rightarrow D) \Rightarrow D)) \Rightarrow (D \Rightarrow D))$	(1), (2) MP	$sub_3(A) = (D \Rightarrow (D \Rightarrow D))$ $sub_3(B) = ((D \Rightarrow ((D \Rightarrow D) \Rightarrow D)) \Rightarrow (D \Rightarrow D))$
(4) $F_4 :=$	$(D \Rightarrow ((D \Rightarrow D) \Rightarrow D))$	Axiom H_1	$sub_4(A) = D$ $sub_4(B) = (D \Rightarrow D)$
(5) $F_5 :=$	$(D \Rightarrow D)$	(3), (4) MP	$sub_5(A) = (D \Rightarrow ((D \Rightarrow D) \Rightarrow D))$ $sub_5(B) = (D \Rightarrow D)$

Formale Beweise vs. ‚normale' Beweise

Formaler Beweis	**‚normale' Beweise**
$(D \Rightarrow (D \Rightarrow D)) \Rightarrow$ $((D \Rightarrow ((D \Rightarrow D) \Rightarrow D)) \Rightarrow (D \Rightarrow D))$ $(D \Rightarrow (D \Rightarrow D))$ $((D \Rightarrow ((D \Rightarrow D) \Rightarrow D)) \Rightarrow (D \Rightarrow D))$ $(D \Rightarrow ((D \Rightarrow D) \Rightarrow D))$ $(D \Rightarrow D)$	● machen größere Schritte ● benutzt andere Sätze und Hilfssätze ● benutzen verschiedene Inferenzregeln ● sind korrekt, wenn sich die einzelnen Schritte als formale Beweise rekonstruieren lassen
● weisen immer alle Details aus ● sind algorithmisch prüfbar ● sind für Menschen nicht verständlich ● verschleiern Beweisideen	● können sich in ihrer Detaillierung nach dem Ansprechpartner richten ● sind verständlich, wenn die richtige Detaillierung gewählt wurde ● können zwischen ‚Beweisidee' und Beweisausführung unterscheiden

Definition 6.14: Korrektheit & Vollständigkeit

Ein aussagenlogischer Kalkül $\mathcal{C}$ heißt genau dann

- *a-korrekt* (ableitungskorrekt), falls für alle Formelmengen **M** und alle Formeln G gilt: wenn **M** $\vdash_{\mathcal{C}}$ G, dann **M** $\models$ G (Also $\mathrm{Abl}_{\mathcal{C}}(\mathbf{M}) \subseteq \mathcal{Fol}(\mathbf{M})$)
- *b-korrekt* (beweiskorrekt), falls für alle Formeln G gilt: wenn $\vdash_{\mathcal{C}}$ G, dann $\models$ G. (Also $\mathrm{Abl}_{\mathcal{C}}(\{\}) \subseteq \mathcal{Taut}_{\mathrm{AL}}$)
- *a-vollständig* (ableitungsvollständig), falls für alle Formelmengen **M** und alle Formeln G gilt: wenn **M** $\models$ G, dann **M** $\vdash_{\mathcal{C}}$ G. (Also $\mathcal{Fol}(\mathbf{M}) \subseteq \mathrm{Abl}_{\mathcal{C}}(\mathbf{M})$)
- *b-vollständig* (beweisvollständig), falls für alle Formeln G gilt: wenn $\models$ G, dann $\vdash_{\mathcal{C}}$ G. (Also $\mathcal{Taut}_{\mathrm{AL}} \subseteq \mathrm{Abl}_{\mathcal{C}}(\{\})$)

Beobachtung zu 6.14

- Jeder a-korrekte Kalkül ist b-korrekt und jeder a-vollständige Kalkül ist b-vollständig.
- Ein b-vollständiger Kalkül, der die Regel Modus Ponens enthält, ist a-vollständig.
- Ein b-korrekter Kalkül ist a-korrekt, wenn für ihn das Deduktionstheorem gilt.

  ➔ Für logische Kalküle wird üblicherweise nachgewiesen:
  Deduktionstheorem (→6.15), B-Korrektheit, B-Vollständigkeit, Verfügbarkeit von MP

Zum Selbststudium

Vollständigkeit und Korrektheit

- Normalerweise werden diese Begriffe einfach verwendet, ohne a- oder b-Zusatz. Gemeint ist dann meistens a-Korrektheit und a-Vollständigkeit. Wir haben die Unterscheidung hier getroffen, um den Stellenwert des Deduktionstheorems deutlicher zu machen.

Widerlegungskalküle

- Verschiedene Beweisverfahren, z.B. Resolution (kommt noch in dieser Vorlesung) und Tableau-Verfahren (Masterstudium: FGI-3) sind vorwiegend dazu geeignet, festzustellen, ob eine Formelmenge erfüllbar ist oder nicht.
- Diese Verfahren sind in der Regel nicht a- oder b-vollständig, aber wir haben ja schon gesehen, wie man Folgerbarkeits- und Tautologiefragen in Erfüllbarkeitsfragen umwandeln kann.

Ein aussagenlogischer Kalkül C heißt genau dann

- *w-korrekt* (widerlegungskorrekt), falls für alle Formelmengen **M** und eine Kontradiktion K gilt: wenn $\mathbf{M} \vdash_C \mathsf{K}$, dann ist **M** unerfüllbar.
- *w-vollständig* (widerlegungsvollständig), falls für alle Formelmengen **M** und eine Kontradiktion K gilt: wenn **M** unerfüllbar ist, dann $\mathbf{M} \vdash_C \mathsf{K}$.

Satz 6.15 (Deduktionstheorem für $\mathcal{H}$)

Seien **M** eine Formelmenge und F und G Formeln mit $\mathbf{M} \cup \{F\} \vdash_{\mathcal{H}} G$,

dann gilt: $\mathbf{M} \vdash_{\mathcal{H}} F \Rightarrow G$

- Speziell für $\mathbf{M} = \emptyset$: Wenn $\{F\} \vdash_{\mathcal{H}} G$, dann $\vdash_{\mathcal{H}} F \Rightarrow G$.
- Wenn G ableitbar ist aus den Annahmen **M** und F, dann ist aus **M** ableitbar: $F \Rightarrow G$.
- Das Deduktionstheorem stellt die systematische Beziehung zwischen Ableitbarkeit (Relation zwischen Formeln) und Implikation (Junktor) her.

Struktur des Beweises

Sei $G_1, \ldots, G_n$ eine Ableitung von G aus $\mathbf{M} \cup \{F\}$; es gilt also: $G_n = G$.

Beweis erfolgt durch vollständige Induktion über i, d.h. es wird für alle Schritte der Ableitung gezeigt: Wenn $\mathbf{M} \cup \{F\} \vdash_{\mathcal{H}} G_i$, dann $\mathbf{M} \vdash_{\mathcal{H}} F \Rightarrow G_i$ für $1 \leq i \leq n$.

Dafür nehmen wir die Ableitung von $G_1, \ldots, G_n$ und formen sie zu Ableitungen für $F \Rightarrow G_i$ um.

Beweis des Deduktionstheorems (1)

Induktionsanfang: i = 1: Ein-Schritt Ableitungen aus **M** $\cup$ {F}

- G_1 ist Axiom: Wir ergänzen die Ableitung wie folgt

(1)	G_1	Axiom	wie zuvor
(1.1)	$G_1 \Rightarrow (F \Rightarrow G_1)$	Axiom H_1	$sub_{1.1}(A) = G_1$, $sub_{1.1}(B) = F$
(1.2)	$F \Rightarrow G_1$	(1), (1.1) MP	$sub_{1.2}(A) = G_1$, $sub_{1.2}(B) = (F \Rightarrow G_1)$

- $G_1 \in$ **M**: Wir ergänzen die Ableitung wie folgt

(1)	G_1	aus **M**	wie zuvor
(1.1)	$G_1 \Rightarrow (F \Rightarrow G_1)$	Axiom H_1	$sub_{1.1}(A) = G_1$, $sub_{1.1}(B) = F$
(1.2)	$F \Rightarrow G_1$	(1), (1.1) MP	$sub_{1.2}(A) = G_1$, $sub_{1.2}(B) = (F \Rightarrow G_1)$

- $G_1 = F$: Wir können obige 5-schrittige Ableitung für $D \Rightarrow D$ umformen, indem wir an Stelle von D überall F einsetzen.

(1.5)	$F \Rightarrow F$	

Beweis des Deduktionstheorems (2)

Induktionsannahme

- Bedingungen des Theorems erfüllt für alle i < m

Induktionsschritt

- Drei Fälle können auftreten:
  - G_m ist ein Axiom ➔ Behandlung wie Induktionsanfang
  - $G_m \in \mathbf{M}$ oder $G_m = F$ ➔ Behandlung wie Induktionsanfang
  - G_m ergibt sich durch MP aus G_i und $G_k = (G_i \Rightarrow G_m)$ (i, k < m)

    Nach der Induktionsannahme gibt es Ableitungen für $(F \Rightarrow G_i)$ und für $(F \Rightarrow G_k)$
  - In der Ableitung also schon enthalten

(i.n)	$F \Rightarrow G_i$	…
(k.n')	$F \Rightarrow (G_i \Rightarrow G_m)$	…

  - Ergänzung um

(m.1)	$(F \Rightarrow G_i) \Rightarrow ((F \Rightarrow (G_i \Rightarrow G_m)) \Rightarrow (F \Rightarrow G_m))$	Axiom H_2
(m.2)	$(F \Rightarrow (G_i \Rightarrow G_m)) \Rightarrow (F \Rightarrow G_m)$	(i.n) (m.1) MP
(m.3)	$F \Rightarrow G_m$	(k.n') (m.2) MP

Zum Selbststudium: Ergänzung

Hier ergänzend die Substitutionen,

- die nicht in die Tabelle passten.

(m.1)	$(F \Rightarrow G_i) \Rightarrow ((F \Rightarrow (G_i \Rightarrow G_m)) \Rightarrow (F \Rightarrow G_m))$	Axiom H_2 $sub_{m.1}(A) = F$, $sub_{m.1}(B) = G_i$, $sub_{m.1}(C) = G_m$
(m.2)	$(F \Rightarrow (G_i \Rightarrow G_m)) \Rightarrow (F \Rightarrow G_m)$	(i.n) (m.1) MP $sub_{m.2}(A) = (F \Rightarrow G_i)$, $sub_{m.2}(B) = (F \Rightarrow (G_i \Rightarrow G_m)) \Rightarrow (F \Rightarrow G_m)$
(m.3)	$F \Rightarrow G_m$	(k.n') (m.2) MP $sub_{m.3}(A) = (F \Rightarrow (G_i \Rightarrow G_m))$, $sub_{m.3}(B) = F \Rightarrow G_m$

Korrektheit & Vollständigkeit von $\mathcal{H}$

Satz 6.16: Korrektheitstheorem für $\mathcal{H}$

- Der Hilbertkalkül $\mathcal{H}$ ist korrekt (a-korrekt und b-korrekt).

Ohne Beweis an dieser Stelle. Es reicht zu zeigen, dass die Axiome wirklich Tautologien sind, auf die Korrektheit des Modus Ponens zu verweisen und damit b-Korrektheit zu begründen. Das Deduktionstheorem 6.15 liefert dann a-Korrektheit.

Satz 6.17: Vollständigkeitstheorem für $\mathcal{H}$

- Der Hilbertkalkül $\mathcal{H}$ ist vollständig (a-vollständig und b-vollständig).

Ohne Beweis an dieser Stelle.

➔ Es gibt Kalküle für die Aussagenlogik, die vollständig und korrekt sind.

Andere Kalküle der Aussagenlogik

Weitere Deduktionssysteme

- Natürliches Schließen
- Sequenzen-Kalkül
- Gentzen-Kalkül

  (werden beispielhaft im Masterstudium in FGI-3 behandelt.)

Widerlegungssysteme

- Zielsetzung: Verfahren, die die syntaktische Struktur von Formeln berücksichtigen, um Mengen von Formeln als unerfüllbar nachzuweisen. (➔ Inkonsistenz)
  - Tableau-Systeme (Wird im Masterstudium in FGI-3 behandelt.)

  ➔ Resolution (wird in Kap. 8 behandelt für die Aussagenlogik und in Kap. 11 für die Prädikatenlogik behandelt.)

Zur Ergänzung: Konsistenz und Inkonsistenz – Definition

Definitionen 6.18

1. Eine Menge von zwei Formeln $\{F, \neg F\}$ heißt *Kontradiktionspaar* oder *Paar komplementärer Formeln*.
2. Eine Formelmenge **M** heißt genau dann *inkonsistent* bzgl. eines Kalküls $\mathcal{C}$ ($\mathbf{M} \vdash_{\mathcal{C}}$), falls beide Elemente eines Kontradiktionspaars aus **M** mit $\mathcal{C}$ ableitbar sind,
   d.h. es gibt eine Formel F, so dass sowohl $\mathbf{M} \vdash_{\mathcal{C}} F$ als auch $\mathbf{M} \vdash_{\mathcal{C}} \neg F$ gilt.
   Anderenfalls heißt **M** *konsistent* bzgl. $\mathcal{C}$.

Zur Ergänzung: Inkonsistenz – Unerfüllbarkeit

Satz 6.19

Wenn **M** inkonsistent bzgl. eines a-korrekten Kalküls ist, dann ist **M** unerfüllbar.

Beweis

- Wenn **M** inkonsistent bzgl. $\mathcal{C}$ ist, dann gibt es F mit: $\mathbf{M} \vdash_{\mathcal{C}} \mathsf{F}$ und $\mathbf{M} \vdash_{\mathcal{C}} \neg\mathsf{F}$.
- Da Ableitbarkeit mit $\mathcal{C}$ korrekt ist, gilt: $\mathbf{M} \models \mathsf{F}$ und $\mathbf{M} \models \neg\mathsf{F}$.
- Wäre **M** erfüllbar, müsste es eine Belegung geben, die sowohl F als auch $\neg\mathsf{F}$ wahr macht, aber dies widerspricht der Definition für Belegungen (3.1).
- Also ist **M** unerfüllbar.

Satz 6.20

Wenn **M** unerfüllbar und $\mathcal{C}$ ein a-vollständiger Kalkül ist, dann ist **M** inkonsistent bzgl. $\mathcal{C}$.

Beweis

- Da **M** unerfüllbar ist, folgt jede Formel aus **M**, also auch die Formeln A und $\neg\mathsf{A}$.
- Da Ableitbarkeit mit $\mathcal{C}$ vollständig ist, gilt: $\mathbf{M} \vdash_{\mathcal{C}} \mathsf{A}$ und $\mathbf{M} \vdash_{\mathcal{C}} \neg\mathsf{A}$.

Wichtige Konzepte in diesem Foliensatz

- (Uniforme Formel-) Substitution (in der Prädikatenlogik werden wir noch einen anderen Typ von Substitution kennen lernen: Uniforme Variablen-Substitution)
- Inferenzregel, allgemein und als wichtiges Beispiel den Modus Ponens
  - Anwendbarkeit einer Inferenzregel, Ableitbarkeit mit einer Inferenzregel
  - Korrektheit einer Inferenzregel
- Kalkül
  - Axiom, Ableitung in einem Kalkül, (formaler) Beweis in einem Kalkül
  - Korrektheit und Vollständigkeit eines Kalküls
  - Konsistenz, Inkonsistenz bzgl. eines Kalküls
  - Hilbert-System als Beispiel für einen Kalkül
  - Deduktionstheorem (für den Hilbert-Kalkül)

Aussagenlogik: Hornformeln

- Spezielle Formelklasse
  - Teilklasse der Formeln in Konjunktiver Normalform

    Zur Erinnerung: Wenn F in KNF ist, kann F als eine Konjunktion von Klauseln, d.h., als eine Konjunktion von Disjunktionen von Literalen, angesehen werden.
  - Charakterisierbar über Implikation als Hauptoperator

    ➔ Grundlage für Logik-Programmierung (z.B. Prolog ➔ SE-3, Datalog ➔ GWV)

    [➔ Hornformeln auf der Basis der Prädikatenlogik]
  - Die Klasse der Hornformeln [➔ Def. 7.1] ist nur eine Teilklasse der Formeln der Aussagenlogik:

    Es gibt sogar Formeln der Aussagenlogik, zu denen es keine äquivalenten Hornformeln gibt [➔ Folie 7.14-7.16].
  - Hornformeln erlauben eine besonders effiziente Berechnung der Prüfung von Erfüllbarkeit und Unerfüllbarkeit [➔ Folie 7.17-7.27].

Grundproblem: Erfüllbarkeit von Formelmengen

Gegeben: Eine Menge von Formeln M.

Frage: Ist M erfüllbar? (Ist es möglich, dass **alle** Formeln in **M** unter derselben Interpretation wahr sind?)

Beobachtung

- Wenn **M** nur Literale enthält, können wir die Frage leicht beantworten (s. Wie auch Erfüllbarkeit von dualen Klauseln.)
  - Beispiel: **M** = {¬A, C, B, ¬D}

  ➔ Existieren komplementäre Literale in **M**?
- Wenn **M** Literale und Konjunktionen von Literalen enthält, gilt dasselbe.
  - Beispiel: **M** = {¬A ∧ C, B ∧ ¬D}
- Was, wenn **M** Disjunktionen von Literalen enthält?
  - Beispiel: **M** = {¬A ∨ C, B ∨ D, A ∨ ¬B, ¬C ∨ ¬D}

  ➔ Im schlimmsten Fall muss man verschiedene Belegungen ausprobieren.
  - Beispiel: **M** = {¬A ∨ C, B, A ∨ ¬B, ¬C ∨ ¬D}

  ➔ Im günstigen Fall kann man die Belegung direkt bestimmen:
  $\mathcal{A}$(B) = **1**, $\mathcal{A}$(A) = **1**, $\mathcal{A}$(C) = **1**, $\mathcal{A}$(D) = **0**.

Hornlogik erlaubt nur, ‚günstige Fälle' auszudrücken !

Definition 7.1 (Hornklausel, Hornformel)

- Eine Klausel K ist genau dann eine *Hornklausel*, falls K **höchstens ein** positives Literal enthält.
- Eine Formel F in konjunktiver Normalform, bei der jede Klausel eine Hornklausel ist, ist eine *Hornformel*.

Beispiele

Hornklauseln		keine Hornklauseln
A	$\neg A$	
$\neg A \vee B$	$\neg A \vee \neg B$	$A \vee B$
$\neg A \vee B \vee \neg C$		$\neg A \vee B \vee C$
$A \vee \neg B \vee \neg C \vee \neg D$		$A \vee \neg B \vee C \vee \neg D$

Typen von Hornklauseln

Eine Hornklausel K enthält **höchstens ein** positives Literal.

➔ Es gibt **drei Typen** von Hornklauseln:

1. K enthält **negative** Literale **und ein positives** Literal: *Regeln*

   $K = \neg A_1 \vee \neg A_2 \vee \ldots \vee \neg A_i \vee A_k$,

2. K enthält **nur negative** Literale: *Beschränkungen*

   $K = \neg A_1 \vee \neg A_2 \vee \ldots \vee \neg A_i$

3. K enthält **nur ein positives** Literal: *Fakten*

   $K = A_1$

Zum Selbststudium: Grammatik für Hornformel (KNF-Spezialfall)

Vokabular der terminalen Symbole (Symbole, die in der Formel vorkommen): $\Sigma_{\mathrm{HFK}} = \mathcal{A}s_{\mathrm{AL}} \cup \{\neg, \wedge, \vee, \,), \,(\,\}$ **nicht-terminales Symbol** (Symbole, die für die Erzeugung der Formeln benötigt werden, aber nicht in der Formel vorkommen): $N_{\mathrm{HFK}} = \{\mathrm{HFK}, \mathrm{HKl}, \mathrm{L}, \mathrm{As}\}$ **Startsymbol:** HFK Anmerkung: HFK: Hornformel KNF-Schreibweise HKl: Horn-Klausel L: Literal NL: negatives Literal As: Aussagesymbol (positives Literal)	**Regeln (Produktionen):** P = { HFK → (HFK ∧ HFK) , HFK → HKl , HKl → (NL ∨ HKl) , HKl → (HKl ∨ NL) , HKl → L , NL → ¬As , L → NL , L → As , As → A , As → B , As → C , As → D , … }

Eine „Normalform“ für Hornformeln

Definition 7.2

Eine Hornklausel, in der kein Aussagensymbol mehrfach auftritt, heißt *reduziert*.
Eine Hornformel H heißt genau dann *reduziert*, wenn

1. alle Klauseln in H reduziert sind, und wenn
2. alle Klauseln in H verschieden sind.

Beobachtung

- Wenn in einer Hornklausel K ein Paar komplementärer Literale auftritt, dann ist K eine Tautologie. (Dies kann nur bei Typ 1 der Fall sein).
- Wenn in einer Hornklausel K kein Paar komplementärer Literale auftritt, dann gibt es eine zu K äquivalente reduzierte Hornklausel.

➔ Anwendung der Idempotenz

Satz 7.3 (Existenz einer reduzierten Hornformel)

Für jede nicht allgemeingültige Hornformel existiert eine äquivalente reduzierte Hornformel.
Beweis: Zur Übung.

Exkurs: Syntaxerweiterung: $\top$ und $\perp$

Zur Erinnerung

- Sind F und G *Tautologien*, so gilt: $F \equiv G$.
- Sind F und G *unerfüllbar*, so gilt: $F \equiv G$.

➔ Als Vertreter für die Äquivalenzklasse

- der *Tautologien* führen wir die atomaren Formel $\top$ (*top*) ein.
- der *unerfüllbaren Formeln* führen wir die atomaren Formel $\perp$ (*bottom*) ein.

1. $\top$ und $\perp$ sind *logische Konstanten* , also keine Aussagesymbole.
2. Für alle Belegungen $\mathcal{A}$ wird festgesetzt: $\mathcal{A}(\top) = \mathbf{1}$, $\mathcal{A}(\perp) = \mathbf{0}$
3. Dann gelten folgende Äquivalenzen:

$\neg\top \equiv \perp$	$\neg\perp \equiv \top$
$\top \wedge F \equiv F$ für alle F	$\top \vee F \equiv \top$ für alle F
$\perp \wedge F \equiv \perp$ für alle F	$\perp \vee F \equiv F$ für alle F

4. Für Implikationen gilt:

$\top \Rightarrow F \equiv F$	für alle F	$F \Rightarrow \perp \equiv \neg F$	für alle F
$F \Rightarrow \top \equiv \top$	für alle F	$\perp \Rightarrow F \equiv \top$	für alle F

Zum Selbststudium: Syntaxerweiterung: $\top$ und $\bot$

Im folgenden arbeiten wir also mit einer erweiterten Logiksprache

Definition 7x

Es sei $\mathcal{A}s_{\text{AL}}$ eine endliche oder abzählbar unendliche Menge von Aussagesymbolen, so dass $\{\neg, \wedge, \vee, \Rightarrow, \Leftrightarrow, \top, \bot,), (\} \cap \mathcal{A}s_{\text{AL}} = \emptyset$.

Die *wohlgeformten Ausdrücke der Aussagenlogik (Formeln) mit* $\top, \bot$ sind induktiv definiert:

1. $\top, \bot$ und alle Aussagensymbole aus $\mathcal{A}s_{\text{AL}}$ sind (atomare) Formeln.
2. Falls F und G Formeln sind, so sind $(F \wedge G)$, $(F \vee G)$, $(F \Rightarrow G)$ und $(F \Leftrightarrow G)$ (*komplexe*) Formeln.
3. Falls F eine Formel ist, so ist auch $\neg F$ eine (*komplexe*) Formel.
4. Es gibt keine anderen Formeln, als die, die durch endliche Anwendung der Schritte 1–3 erzeugt werden.

- Die Menge aller aussagenlogischen Formeln bezeichnen wir als $\mathcal{L}_{\text{AL}}{}^{\top\bot}$.

➔ Für $\mathcal{L}_{\text{AL}}{}^{\top\bot}$ müssen bei der strukturellen Induktion im Induktionsanfang auch die Fälle $\top$ und $\bot$ behandelt werden.

Hornklauseln – Implikationsschreibweise

Zu jede Hornklausel gibt es eine äquivalente Formel mit Hauptoperator $\Rightarrow$, so dass in der linken Teilformel kein Junktor außer der Konjunktion auftritt und rechts nur ein Aussagensymbol oder $\perp$.

- Fall 1: K enthält negative Literale und ein positives Literal: *Regeln*

  $$K = \neg A_1 \vee \neg A_2 \vee \ldots \vee \neg A_i \vee A_k$$

  dann kann K in der folgenden Weise umgeformt werden:

  $$K \equiv (\neg A_1 \vee \neg A_2 \vee \ldots \vee \neg A_i) \vee A_k$$

  $$\equiv \neg(A_1 \wedge A_2 \wedge \ldots \wedge A_i) \vee A_k \qquad \equiv (A_1 \wedge A_2 \wedge \ldots \wedge A_i) \Rightarrow A_k$$

- Fall 2: K enthält nur negative Literale: *Beschränkungen*

  $$K = \neg A_1 \vee \neg A_2 \vee \ldots \vee \neg A_i$$

  $$\equiv \neg(A_1 \wedge A_2 \wedge \ldots \wedge A_i) \qquad \equiv (A_1 \wedge A_2 \wedge \ldots \wedge A_i) \Rightarrow \perp$$

  [wegen $\neg F \equiv F \Rightarrow \perp$ für alle F]

- Fall 3: K enthält nur ein positives Literal: *Fakten*

  $$K = A_1 \qquad \equiv T \Rightarrow A_1$$

  [wegen $F \equiv T \Rightarrow F$ für alle F]

Z. Selbststudium: Grammatik für Hornformel (Implikationsschreibweise)

Vokabular der terminalen Symbole (Symbole, die in der Formel vorkommen): $\Sigma_{\mathrm{HFI}} = \mathcal{As}_{\mathrm{AL}} \cup \{\wedge, \Rightarrow, \bot, \top,), (\}$ **nicht-terminales Symbol** (Symbole, die für die Erzeugung der Formeln benötigt werden, aber nicht in der Formel vorkommen): $N_{\mathrm{HFI}} = \{\mathrm{HFI}, \mathrm{HI}, \mathrm{L}, \mathrm{As}\}$ **Startsymbol:** HF Anmerkung: HFI: Hornformel Implikationsschreibweise HI: Horn-Implikation KAt: Konjunktion von Atomen As: Aussagesymbol (positives Literal)	**Regeln (Produktionen):** $P = \{$ HFI $\rightarrow$ HI $\wedge$ HFI , HFI $\rightarrow$ HI , HI $\rightarrow$ (KAt $\Rightarrow$ As) , HI $\rightarrow$ (KAt $\Rightarrow$ $\bot$) , HI $\rightarrow$ ($\top$ $\Rightarrow$ As) , KAt $\rightarrow$ As $\wedge$ KAt , KAt $\rightarrow$ As , As $\rightarrow$ A , As $\rightarrow$ B , As $\rightarrow$ C , As $\rightarrow$ D , … $\}$

Exkurs: Hornklauseln – Logik-Programmierung

Die drei Typen von Hornklauseln entsprechen drei Grundtypen von Ausdrücken der Logik-Programmierung (➔ SE-3, Regelsprachen mit definiten Klauseln: Datalog ➔ GWV):

Fakten (z.B. einer Wissensbasis)

- Hans ist der Sohn von Peter: ISTSOHNVON(HANS, PETER).
- Peter ist der Sohn von August: ISTSOHNVON(PETER, AUGUST).
- Fall 3: nur ein positives Literal. A_k . entspricht $T \Rightarrow A_k$

Regeln (z.B. einer Wissensbasis)

- Wenn Hans Sohn von Peter und Peter Sohn von Ben ist, dann ist Hans Enkel von Ben:
  ISTENKELVON(HANS, BEN) :– ISTSOHNVON(HANS, PETER), ISTSOHNVON(PETER, BEN).
- Fall 1: negative Literale (rechts!) und ein positives Literal (links!)
  A_k :– A_1 , A_2 , … , A_i . entspricht $(A_1 \wedge A_2 \wedge \ldots \wedge A_i) \Rightarrow A_k$

Anfragen / Ziele (Modelliert über Beschränkungen)

- Ist Hans Enkel von Ben und Ben Sohn von Peter? :
  ? :– ISTENKELVON(HANS, BEN), ISTSOHNVON(BEN, PETER).
- Fall 2: nur negative Literale. ? :– A_1 , … , A_i . entspricht $(A_1 \wedge \ldots \wedge A_i) \Rightarrow \perp$

[".", ":–", "," und "?" sind Symbole der Logik-Programmierung]

Zum Selbststudium (insbesondere im Zusammenhang mit SE 3)

Logik-Programme, die auf Aussagenlogik basieren

- sind sehr ausdrucksschwach, da sie keine Variablen verwenden können
- bei der Behandlung der Prädikatenlogik wird die Beziehung zu PROLOG noch enger.

Dass Anfragen als Klauseln mit negativen Literalen zu verstehen sind,

- ergibt sich aus Satz 5.9 (→ $F \models G$ GDW. $F \wedge \neg G$ unerfüllbar ist.)
- Eigentlich will man ja wissen, ob aus den Fakten und Regeln die Anfrage folgt (wobei die Teile der Anfrage konjunktiv verknüpft werden).
- Dazu fügt man die Negation der Anfrage mit den Fakten und Regeln zusammen und prüft, ob sich daraus ein Widerspruch ergibt. Ist dies der Fall, dann folgt die Anfrage aus dem Rest.
- Werden mehrere Anfragen gleichzeitig beantwortet, dann ergibt sich aus einem gefundenen Widerspruch, dass (mindestens) eine der Anfragen aus den Fakten und Regeln folgt.

Zum Selbststudium: Logik-Programm und Hornformeln

- B, C und G sind Fakten.
- Wenn C, dann auch A und wenn G dann auch D.
- Ergibt sich daraus A, B und D? oder ergibt sich E?
  - Modelliert als Test: Wird die Beschränkung $(\neg A \vee \neg B \vee \neg D)$ bzw. $\neg E$ verletzt?

Fakten	B .	Ziele	? :– A , B , D .
	C .		? :– E .
	G .		
Regeln	A :– C .		
	D :– G .		

$$B \wedge C \wedge G \wedge (\neg C \vee A) \wedge (\neg G \vee D) \wedge (\neg A \vee \neg B \vee \neg D) \wedge \neg E$$

umgeformt als Konjunktion von Implikationen

$$\equiv \quad (T \Rightarrow B) \wedge (T \Rightarrow C) \wedge (T \Rightarrow G) \wedge (C \Rightarrow A) \wedge (G \Rightarrow D) \wedge$$
$$((A \wedge B \wedge D) \Rightarrow \bot) \wedge (E \Rightarrow \bot)$$

Gesucht

Ein einfaches Verfahren, dass für **jedes** Logik-Programm die (Un)erfüllbarkeit prüft.

Was ist durch Hornformeln ausdrückbar?

- Es gibt Formeln der Aussagenlogik, zu denen es keine äquivalente Hornformel gibt.
- Der entsprechende Wahrheitswertverlauf ist nicht in Hornform ausdrückbar.
- Hornformel-Logik ist gegenüber der Aussagenlogik eingeschränkt (in ihrer Ausdrucksstärke).

Beispiel: Es gibt zu $A \vee B$ keine äquivalente Hornformel.

Es gibt keine zu $A \vee B$ äquivalente Hornformel

- Wenn es eine zu $A \vee B$ äquivalente Hornformel H^* gäbe, so würde es hierzu eine äquivalente reduzierte Hornformel H geben. [Satz 7.3: Existenz reduzierter Hornformel]
- Diese äquivalente reduzierte Hornformel H müsste – falls sie ausschließlich aus den Aussagesymbolen A und B aufgebaut ist – aus den *(Horn-)Klauseln*

  $A \quad B \quad \neg A \quad \neg B \quad A \vee \neg B \quad \neg A \vee B \quad \neg A \vee \neg B$

  gebildet sein, d.h. eine Konjunktion aus einigen dieser Klauseln sein.

	A	B	$A \vee B$	$\neg A$	$\neg B$	$A \vee \neg B$	$\neg A \vee B$	$\neg A \vee \neg B$
$\mathcal{A}_1$	1	1	1	0	0	1	1	0
$\mathcal{A}_2$	1	0	1	0	1	1	0	1
$\mathcal{A}_3$	0	1	1	1	0	0	1	1
$\mathcal{A}_4$	0	0	0	1	1	1	1	1

- Aus der Tafel lassen sich zwei Begründungen dafür ablesen, dass es keine zu $A \vee B$ äquivalente Hornformel H gebildet aus den Aussagesymbolen A und B gibt.
  - Jede der Hornklauseln wird unter mindestens einer der Belegung $\mathcal{A}_1$, $\mathcal{A}_2$ oder $\mathcal{A}_3$ zu **0** ausgewertet. Da sich die Bewertung **0** in einer Konjunktion durchsetzt, kann keine der Klauseln zu H gehören, die ja – wie $A \vee B$ – zu **1** ausgewertet werden soll.
  - Nur A und B werden von $\mathcal{A}_4$ zu **0** ausgewertet. Wäre aber A oder B Konjunkt von H, dann würde für H auch bei $\mathcal{A}_2$ oder $\mathcal{A}_3$ eine **0** entstehen.

Zum Selbststudium: Es gibt keine zu A ∨ B äquivalente Hornformel

➔ Können andere Aussagesymbole dazu beitragen, dass eine zu A ∨ B äquivalente Hornformel gebildet wird?

Gesetzt den Fall, es kommt ein weiteres Aussagesymbol C ins Spiel.

- Es können 10 neue Hornklauseln über A, B und C gebildet werden (zusätzlich zu C und ¬C).

	A	B	C	A ∨ B	A ∨ ¬C	B ∨ ¬C	¬A ∨ C	¬A ∨ ¬C	¬B ∨ C	¬B ∨ ¬C	A ∨ ¬B ∨ ¬C	¬A ∨ B ∨ ¬C	¬A ∨ ¬B ∨ C	¬A ∨ ¬B ∨ ¬C
$\mathcal{A}_1$	1	1	0	1	1	1	0	1	0	1	1	1	0	1
$\mathcal{A}_2$	1	0	0	1	1	1	0	1	1	1	1	1	1	1
$\mathcal{A}_3$	0	1	0	1	1	1	1	1	0	1	1	1	1	1
$\mathcal{A}_4$	0	0	0	0	1	1	1	1	1	1	1	1	1	1
$\mathcal{A}'_1$	1	1	1	1	1	1	1	0	1	0	1	1	1	0
$\mathcal{A}'_2$	1	0	1	1	1	0	1	0	1	1	1	0	1	1
$\mathcal{A}'_3$	0	1	1	1	0	1	1	1	1	0	0	1	1	1
$\mathcal{A}'_4$	0	0	1	0	0	0	1	1	1	1	1	1	1	1

- Weitere Überlegungen wie zuvor.

Erfüllbarkeitstest für Hornformeln: Markierungsalgorithmus

Eingabe: Eine Hornformel F in Implikationsschreibweise

- **Grundidee**: Markiere Aussagesymbole, die in jeder Belegung, die F erfüllt, auf **1** abgebildet werden (müssen).
  1. Versehe jedes Vorkommen eine Aussagensymbols A_i in F mit einer Markierung, falls es in F eine Teilformel der Form $(T \Rightarrow A_i)$ gibt.
  2. WHILE es gibt in F eine Teilformel G der Form

     [1] $(A_1 \wedge A_2 \wedge \ldots \wedge A_n) \Rightarrow A_i$ oder der Form

     [2] $(A_1 \wedge A_2 \wedge \ldots \wedge A_n) \Rightarrow \perp$,

     $n \geq 1$, wobei $A_1,\ldots, A_n$ bereits markiert sind (und A_i noch nicht markiert ist)

     DO IF G hat die Form [1]

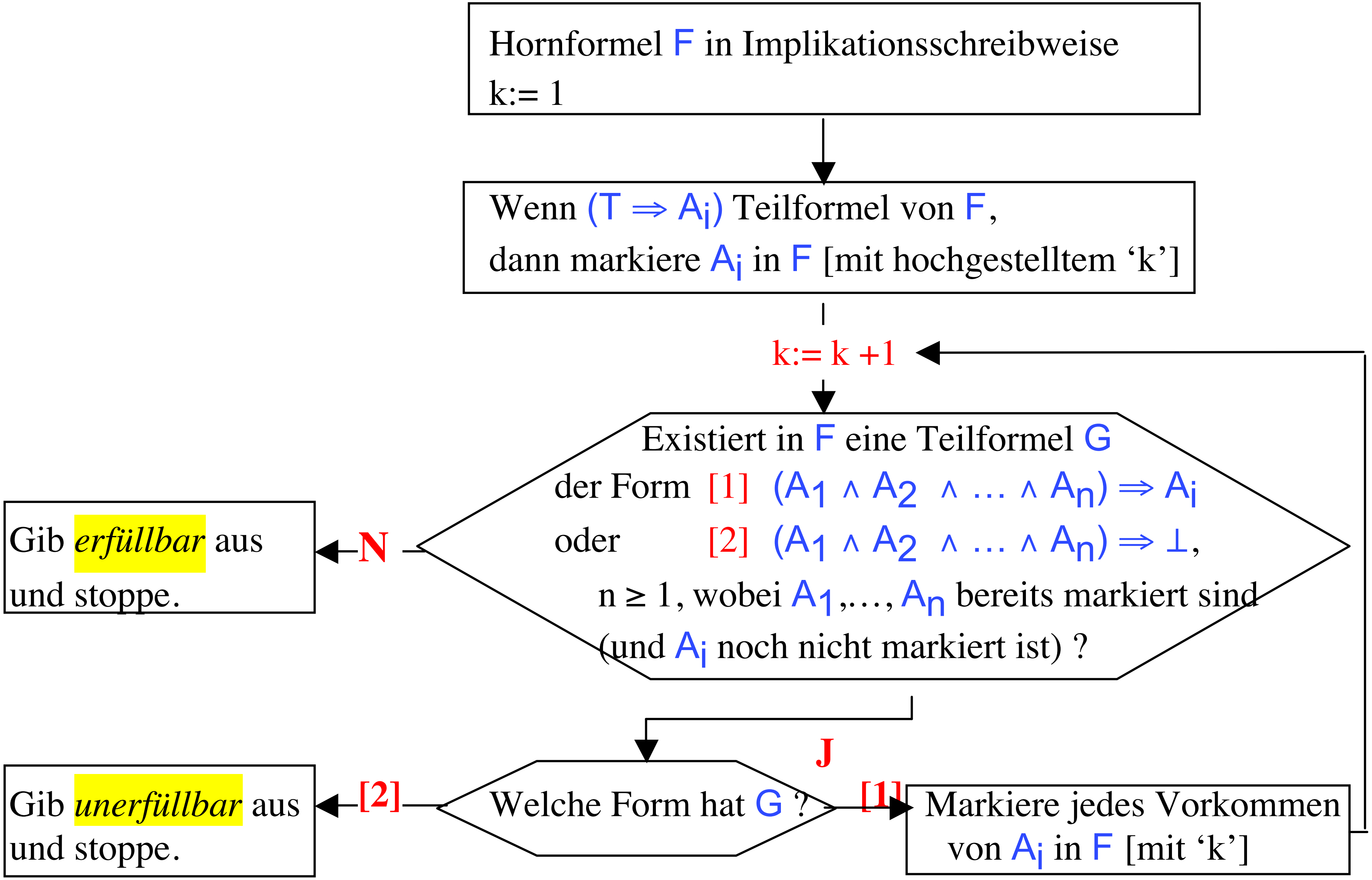
     THEN markiere jedes Vorkommen von A_i in F

     ELSE gib *unerfüllbar* aus und stoppe.
  3. Gib *erfüllbar* aus und stoppe.

     Die erfüllende Belegung wird durch die Markierung gegeben:

$$\mathcal{A}(A_i) = \begin{cases} \mathbf{1}, \text{ falls } A_i \text{ eine Markierung besitzt} \\ \mathbf{0}, \text{ sonst} \end{cases}$$

Markierungsalgorithmus: Struktur des Ablaufs



Erfüllbarkeitstest für Hornformeln: Beispiel-1

$$(\neg A \vee \neg B \vee E) \wedge C \wedge A \wedge (\neg C \vee B) \wedge (\neg A \vee D) \wedge (\neg C \vee \neg D \vee A) \wedge \neg E \wedge \neg G$$

Als Konjunktion von Implikationen:

$$((A \wedge B) \Rightarrow E) \wedge (T \Rightarrow C) \wedge (T \Rightarrow A) \wedge (C \Rightarrow B) \wedge$$
$$(A \Rightarrow D) \wedge ((C \wedge D) \Rightarrow A) \wedge (E \Rightarrow \bot) \wedge (G \Rightarrow \bot)$$

Markierung 1. Runde:

$$((^1A \wedge B) \Rightarrow E) \wedge (T \Rightarrow {}^1C) \wedge (T \Rightarrow {}^1A) \wedge ({}^1C \Rightarrow B) \wedge$$
$$({}^1A \Rightarrow D) \wedge (({}^1C \wedge D) \Rightarrow {}^1A) \wedge (E \Rightarrow \bot) \wedge (G \Rightarrow \bot)$$

Markierung 2. Runde:

$$(({}^1A \wedge {}^2B) \Rightarrow E) \wedge (T \Rightarrow {}^1C) \wedge (T \Rightarrow {}^1A) \wedge ({}^1C \Rightarrow {}^2B) \wedge$$
$$({}^1A \Rightarrow {}^2D) \wedge (({}^1C \wedge {}^2D) \Rightarrow {}^1A) \wedge (E \Rightarrow \bot) \wedge (G \Rightarrow \bot)$$

Markierung 3. Runde:

$$(({}^1A \wedge {}^2B) \Rightarrow {}^3E) \wedge (T \Rightarrow {}^1C) \wedge (T \Rightarrow {}^1A) \wedge ({}^1C \Rightarrow {}^2B) \wedge$$
$$({}^1A \Rightarrow {}^2D) \wedge (({}^1C \wedge {}^2D) \Rightarrow {}^1A) \wedge ({}^3E \Rightarrow \bot) \wedge (G \Rightarrow \bot)$$

➔ unerfüllbar

Erfüllbarkeitstest für Hornformeln: Beispiel-2

$$F := (A \lor \neg B) \ \land \ (\neg C \lor \neg A \lor D) \ \land \ (\neg A \lor \neg B) \ \land \ D \ \land \ \neg E$$

als Konjunktion von Implikationen:

$$(B \Rightarrow A) \ \land \ ((C \land A) \Rightarrow D) \ \land \ ((A \land B) \Rightarrow \bot) \ \land \ (T \Rightarrow D) \ \land \ (E \Rightarrow \bot)$$

Markierung:

$$(B \Rightarrow A) \land ((C \land A) \Rightarrow {}^{1}D) \land ((A \land B) \Rightarrow \bot) \land (T \Rightarrow {}^{1}D) \land (E \Rightarrow \bot)$$

Einteilung der Klauseln nach Form [1] und Form [2]:

[1]	$B \Rightarrow A$	[2]	$(A \land B) \Rightarrow \bot$
	$(C \land A) \Rightarrow {}^{1}D$		$E \Rightarrow \bot$

➔ erfüllbar

Erfüllende Belegung:

	A	B	C	D	E	F
$\mathcal{A}$	**0**	**0**	**0**	**1**	**0**	**1**

Zum Selbststudium

Welches Ergebnis liefert der Markierungsalgorithmus

- Wenn es keine Klauseln der Form $(T \Rightarrow A_i)$ gibt?
- Wenn es keine Klauseln der Form $(A_1 \wedge A_2 \wedge \ldots \wedge A_n) \Rightarrow \perp$ gibt?

Begründen Sie, warum dieses Ergebnis korrekt ist.

- Greifen Sie dazu auch die Erläuterungen zu 7.4 und 7.6 zu.

Was passiert,

- Wenn es keine Klauseln der Form $(A_1 \wedge A_2 \wedge \ldots \wedge A_n) \Rightarrow A$ gibt?

Korrektheit & Terminierung des Markierungsalgorithmus

Satz 7.4: Der Markierungsalgorithmus für Hornformeln ist korrekt.
Er stoppt nach spätestens n Markierungsschritten.

(n = Anzahl der Aussagensymbole in F)

Anmerkungen

1. Der Markierungsalgorithmus setzt voraus, dass die Eingabe eine Hornformel ist.
2. Korrektheit: Der Markierungsalgorithmus gibt bei erfüllbaren Hornformeln *erfüllbar* und bei unerfüllbaren Hornformeln *unerfüllbar* aus.
3. Der Erfüllbarkeitstest durch Markierung benötigt maximal n Markierungsschritte, während eine Berechnung der Wahrheitswerttafel gegebenenfalls 2^n Belegungen berücksichtigen muss.

➔ Markierungsalgorithmus korrespondiert zur Fragebeantwortung in der Logik-Programmierung (➔ SE-3).

Terminierung des Markierungsalgorithmus

Voraussetzungen: F ist eine Formel, in der genau n Aussagensymbole vorkommen

Beweis

Im ersten Schritt (1.) werden einige Aussagensymbole markiert, dann ist der Schritt abgeschlossen.
In jedem weiteren Markierungsschritt (2. / WHILE-Schleife) wird mindestens ein weiteres Aussagensymbol markiert.
Ist keine weitere Markierung möglich, so wird die WHILE-Schleife verlassen.

➔ Maximal n Markierungsschritte.
Dann stoppt das Verfahren mit der Ausgabe „*erfüllbar*" oder „*unerfüllbar*".

➔ Markierungsalgorithmus hat *linearen Aufwand*.

Korrektheit der Markierung: Beweis (1)

Lemma 7.5: Wenn $\mathcal{A}$ ein Modell für die Hornformel F ist,
so gilt für die durch den Algorithmus markierten Symbole A_i: $\mathcal{A}(A_i) = \mathbf{1}$.

Voraussetzungen: Def. 3.1, 3.3, 4.18, 7.1

Beweis

Es sei $\mathcal{A}$ ein Modell für die Hornformel F.

Dann macht $\mathcal{A}$ auch alle Konjunkte von F wahr, also die Hornklauseln, aus denen F aufgebaut ist.

Behauptung

$\mathcal{A}$ macht die im Markierungsschritt k markierten Aussagensymbole wahr.
Vollständige Induktion über die Nr. des Markierungsschrittes, in dem A_i markiert wurde.

Induktionsanfang

In Markierungsschritt Nr. 1 (das ist Schritt 1 des Algorithmus) werden nur Aussagensymbole A_i markiert, für die eine Klausel $(T \Rightarrow A_i)$ in F existiert.

$\mathcal{A}$ macht $(T \Rightarrow A_i)$ wahr und $(T \Rightarrow A_i)$ ist zudem mit A_i äquivalent ist. Also macht $\mathcal{A}$ A_i wahr.

Korrektheit der Markierung: Beweis (2)

Induktionsannahme

Für alle $j < k$ gelte: $\mathcal{A}$ macht die im Markierungsschritt Nr. j markierten Aussagensymbole wahr.

Induktionsschritt

Der k-te Markierungsschritt erfolgt innerhalb der WHILE-Schleife. Dabei werden nur solche Aussagensymbole A_i markiert, für die eine Klausel $(A_1 \wedge A_2 \wedge \ldots \wedge A_n) \Rightarrow A_i$ existiert, und $A_1, A_2, \ldots, A_n$ wurden schon (in einem früheren Schritt) markiert.

Da $\mathcal{A}$ $A_1, A_2, \ldots, A_n$ (Induktionsannahme) und die Klausel $(A_1 \wedge A_2 \wedge \ldots \wedge A_n) \Rightarrow A_i$ wahr macht (s.o.), macht $\mathcal{A}$ auch A_i wahr.

Resümee

In jedem Markierungsschritt gilt, dass die markierten Aussagensymbole durch $\mathcal{A}$ wahr gemacht werden.

Korrektheit der Klassifikation „*unerfüllbar*“: Beweis

Lemma 7.6a: Wenn „*unerfüllbar*“ ausgegeben wird, so ist die Eingangsformel F unerfüllbar.

Beweis

Die Klassifikation „*unerfüllbar*“ wird ausgegeben (Schritt 2 „ELSE“), wenn die Klausel $(A_1 \wedge A_2 \wedge \ldots \wedge A_n) \Rightarrow \bot$ in F enthalten ist, wobei $n \geq 1$ und $A_1, \ldots, A_n$ bereits markiert sind.

Beweis durch Widerspruch

Annahme: In F ist die Klausel $(A_1 \wedge A_2 \wedge \ldots \wedge A_n) \Rightarrow \bot$ enthalten, der Markierungsalgorithmus markiert $A_1, \ldots, A_n$ und trotzdem gibt es ein Modell $\mathcal{A}$ von F.

Da $\mathcal{A}(F) = \mathbf{1}$, ist für $1 \leq i \leq n$ auch: $\mathcal{A}(A_i) = \mathbf{1}$, da die A_i markiert werden. (s. Lemma 7.5)

Dann ist auch: $\mathcal{A}(A_1 \wedge A_2 \wedge \ldots \wedge A_n) = \mathbf{1}$ (Def 3.1)

Und damit $\mathcal{A}((A_1 \wedge A_2 \wedge \ldots \wedge A_n) \Rightarrow \bot) = \mathbf{0}$ also auch $\mathcal{A}(F) = \mathbf{0}$. (Def 3.1)

Also Da $\mathcal{A}(F) = \mathbf{1}$ und $\mathcal{A}(F) = \mathbf{0}$ nicht beides gelten kann, ist die Annahme widersprüchlich und es kann keine erfüllende Belegung für F geben, wenn der Markierungsalgorithmus „*unerfüllbar*“ ausgibt.

Korrektheit der Klassifikation „*erfüllbar*“: Beweis

Lemma 7.6b: Wenn in Schritt 3 „*erfüllbar*“ ausgegeben wird, so liefert die Markierung ein Modell $\mathcal{A}$ für F, gemäß $\mathcal{A}(A_i) = \begin{cases} \mathbf{1}, \text{ falls } A_i \text{ eine Markierung besitzt} \\ \mathbf{0}, \text{ sonst} \end{cases}$.

Voraussetzung: Def. 3.1, Lemma 7.5

Beweis

Die Klassifikation „*erfüllbar*“ wird ausgegeben (Schritt 3), wenn der Schritt 2 erfolgreich durchlaufen wurde. Wir müssen zeigen, dass in diesem Fall $\mathcal{A}$ alle Klauseln wahr macht.

Sei K eine beliebige Klausel aus F.

- Falls K eine atomare Formel ist, so wird K in Schritt 1 markiert. Es gilt: $\mathcal{A}(K) = \mathbf{1}$.
- Falls K die Form $(A_1 \wedge A_2 \wedge \ldots \wedge A_n) \Rightarrow B$ hat, so können 2 Fälle auftreten:
- Alle A_i und B sind markiert (wg. Schritt 2), dann gilt: $\mathcal{A}(K) = \mathbf{1}$.
- Mindestens ein A_i $(1 \leq i \leq n)$ ist nicht markiert, also: $\mathcal{A}(A_i) = \mathbf{0}$.
  Dann gilt aber auch: $\mathcal{A}(K) = \mathbf{1}$.
- Falls K die Form $(A_1 \wedge A_2 \wedge \ldots \wedge A_n) \Rightarrow \perp$ hat, so ist mindestens ein A_i $(1 \leq i \leq n)$ nicht markiert, denn anderenfalls wäre *unerfüllbar* ausgegeben worden.
  Also: $\mathcal{A}(A_i) = \mathbf{0}$ und deswegen $\mathcal{A}(K) = \mathbf{1}$.

Zum Selbststudium: Hornformeln – Logik-Programmierung:

„Logik-Programm“

Fakten	B .	Regeln	A :– C .	Ziele	? :– A , B , D .
	C .		D :– G .		? :– E .
	G .				

als Konjunktion von Implikationen: $(T \Rightarrow B) \wedge (T \Rightarrow C) \wedge (T \Rightarrow G) \wedge$
$(C \Rightarrow A) \wedge (G \Rightarrow D) \wedge ((A \wedge B \wedge D) \Rightarrow \bot) \wedge (E \Rightarrow \bot)$

Erfüllbarkeitstest für Hornformeln – Logik-Programmierung

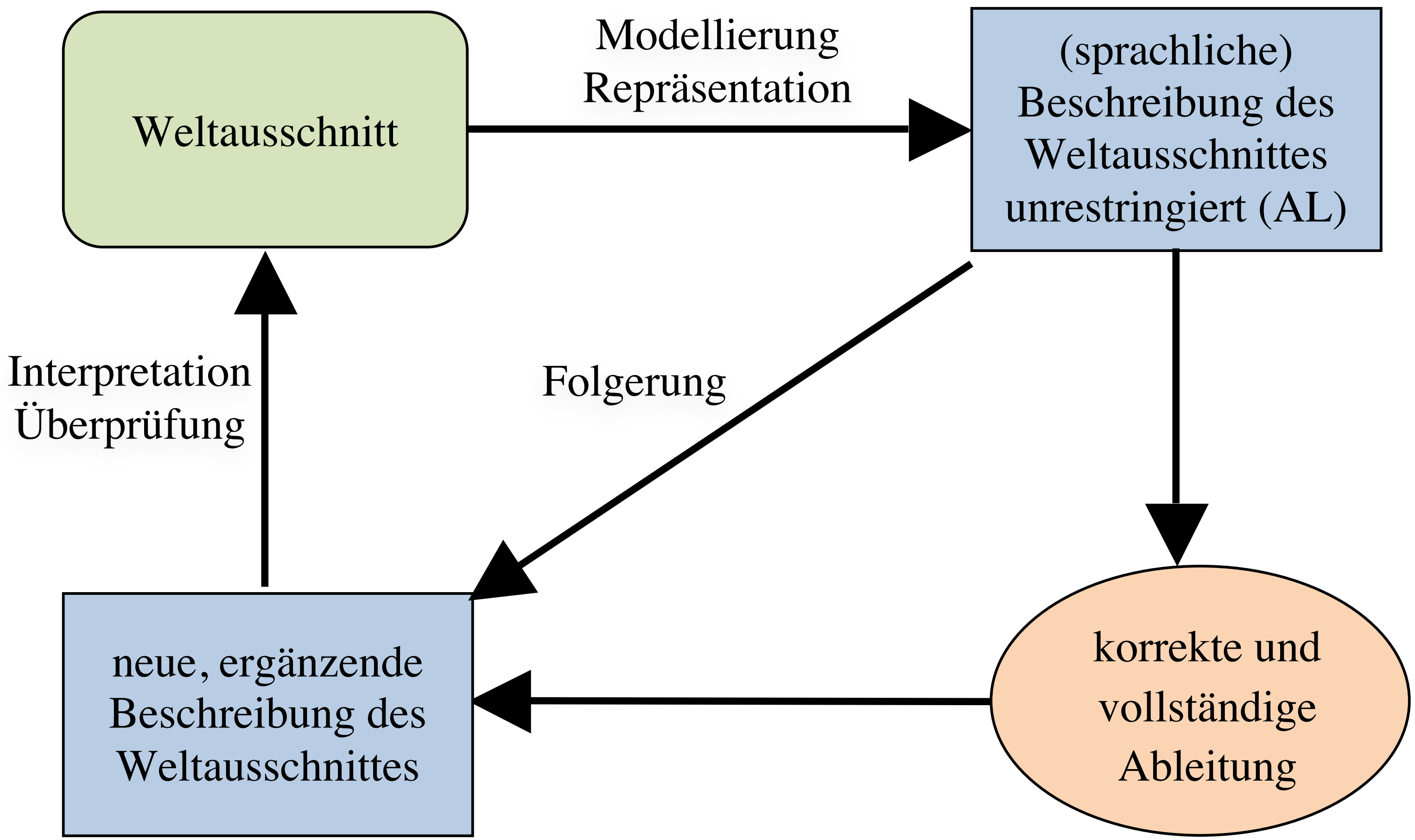
Markierung 1. Runde: $(T \Rightarrow {}^1B) \wedge (T \Rightarrow {}^1C) \wedge (T \Rightarrow {}^1G) \wedge$
$({}^1C \Rightarrow A) \wedge ({}^1G \Rightarrow D) \wedge ((A \wedge {}^1B \wedge D) \Rightarrow \bot) \wedge (E \Rightarrow \bot)$

Markierung 2. Runde: $(T \Rightarrow {}^1B) \wedge (T \Rightarrow {}^1C) \wedge (T \Rightarrow {}^1G) \wedge$
$({}^1C \Rightarrow {}^2A) \wedge ({}^1G \Rightarrow {}^2D) \wedge (({}^2A \wedge {}^1B \wedge {}^2D) \Rightarrow \bot) \wedge (E \Rightarrow \bot)$

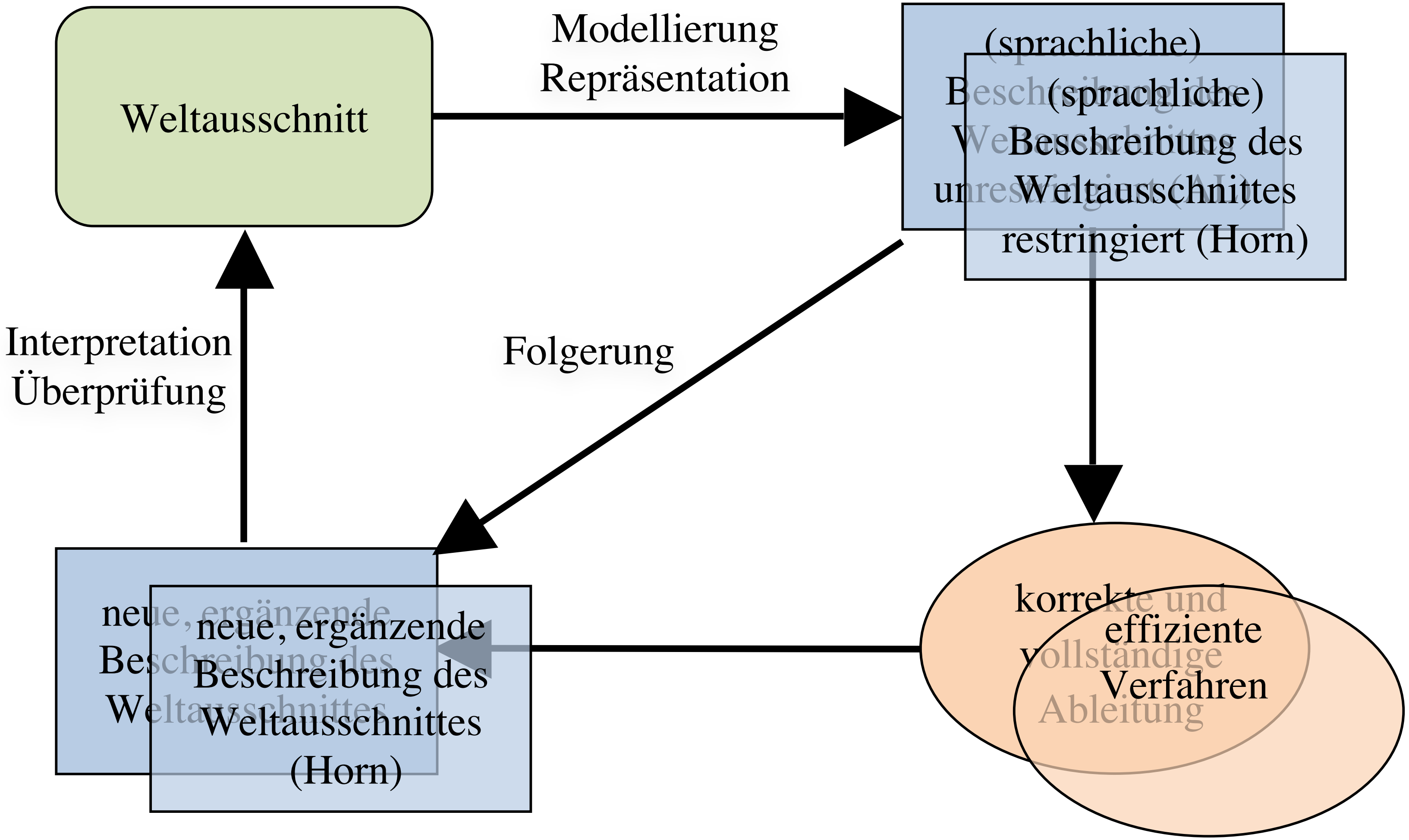
➔ unerfüllbar

➔ $(A \wedge B \wedge D)$ folgt aus den Fakten und Regeln.

Modellierung, Folgerung und Ableitung in Logik



Modellierung, Folgerung und Ableitung in Hornlogik



Wichtige Konzepte dieser Vorlesung

- Hornklausel, Hornformel, KNF-Darstellung, Implikationsschreibweise
- Logische Konstanten ($\top$: top, $\perp$: bottom)
- Beschränkte Ausdrucksfähigkeit von Hornformeln (es gibt Wahrheitswertverläufe, zu denen es keine Hornformeln gibt)
- Markierungsalgorithmus: Grundidee, Funktionsweise, Termination, Korrektheit

Aussagenlogik: Resolution

Resolution

- ist Widerlegungsverfahren (speziell) für Klauselmengen
- basiert also auf Formeln in Konjunktiver Normalform
- ist nicht auf Hornformeln beschränkt aber für Hornformeln auch sehr effizient.

- *Resolventenbildung* – Anwendung der *Resolventenregel* (Inferenzregel)

Zur Erinnerung

- Klauseln sind Disjunktionen von Literalen.
- KNF-Formeln sind Konjunktionen von Klauseln.

- Bei der Resolution ist es angenehm
  - Klauseln als Mengen von Literalen und
  - KNF-Formeln als Mengen von Klauseln darzustellen.

Vorbemerkung zur Resolution

Korrekte Inferenzregeln in Klauseldarstellung

- Modus ponens (MP): $\dfrac{A,\ A \Rightarrow B}{B}$ $\qquad$ $\dfrac{A,\ \neg A \vee B}{B}$
- Modus tollens (MT): $\dfrac{\neg B,\ A \Rightarrow B}{\neg A}$ $\qquad$ $\dfrac{\neg B,\ \neg A \vee B}{\neg A}$
- Disjunktiver Syllogismus (DS): $\dfrac{\neg B,\ A \vee B}{A}$ $\qquad$ $\dfrac{\neg B,\ A \vee B}{A}$
- Disjunktiver Syllogismus (DS): $\dfrac{\neg A,\ A \vee B}{B}$ $\qquad$ $\dfrac{\neg A,\ A \vee B}{B}$
- Hypothetischer Syllogismus (HS): $\dfrac{A \Rightarrow B,\ B \Rightarrow C}{A \Rightarrow C}$ $\qquad$ $\dfrac{\neg A \vee B,\ \neg B \vee C}{\neg A \vee C}$

➔ Resolution verallgemeinert diese Regeln (und viele mehr) unter Verwendung einer geeigneten Darstellung (Mengendarstellung)

➔ Resolution ist ein korrektes Ableitungsverfahren.

Struktur der Vorlesung

Resolutionsregel

- Definition der zugrunde liegenden Mengendarstellung
- Die Resolutionsregel
- Beispiele der Anwendung
- Korrektheit der Resolution
    - Resolutionslemma

Definition Resolutionsableitung

- (Widerlegungs-)Korrektheit
- (Widerlegungs-)Vollständigkeit

Resolutionsalgorithmus

Verfeinerungen des Verfahrens (Ein Ausblick)

- P- / N-Resolution
- lineare Resolution, Stützmengenresolution
- Einheitsresolution

Mengendarstellung von KNF und Klauseln (1)

Definition 8.0

Ist $K = (\bigvee_{i=1}^{m} L_i)$ eine Klausel, dann nennen wir

$\mathbf{K} = \{L_1, \ldots, L_m\}$ die *Mengendarstellung* von K.

Ist $F = (\bigwedge_{i=1}^{n} (\bigvee_{k=1}^{mi} L_{i,k}))$ eine aussagenlogische Formel in konjunktiver Normalform, dann ist

$\mathbf{F} = \{\{L_{1,1}, \ldots, L_{1,m1}\}, \ldots, \{L_{n,1}, \ldots, L_{n,mn}\}\}$ die *Mengendarstellung* von F.

Die Mengendarstellung signalisieren wir durch Fettdruck der Variablen:

$\mathbf{K}$, $\mathbf{K_1}$, $\mathbf{K_2}$, …, $\mathbf{R}$: Klausel in Mengendarstellung

$\mathbf{F}$: KNF-Formel in Mengendarstellung

Wir passen die Wahrheitswertberechnung auf die Mengendarstellung an:

$\mathcal{A}(\mathbf{K}) = \text{Maximum}(\{\mathcal{A}(L) \mid L \in \mathbf{K}\})$ $\qquad$ $\mathcal{A}(\mathbf{F}) = \text{Minimum}(\{\mathcal{A}(\mathbf{K}) \mid \mathbf{K} \in \mathbf{F}\})$

Wir passen den Äquivalenzbegriff auf die Mengendarstellung an:

$\mathbf{F_1} \equiv \mathbf{F_2}$ genau dann, wenn $\mathcal{A}(\mathbf{F_1}) = \mathcal{A}(\mathbf{F_2})$ für alle Belegungen $\mathcal{A}$.

Tritt die leere Menge in der Rolle einer Klausel auf, dann wird sie als *leere Klausel* bezeichnet und durch $\square$ symbolisiert.

➔ Die leere Klausel $\square$ ist die Mengendarstellung zu $\perp$ (konstante Kontradiktion)

Entsprechend legen wir fest: $\mathcal{A}(\square) = \mathbf{0}$

Beispiel

Formel in KNF

$F \quad = (\neg A \vee \neg B \vee \neg D) \wedge \neg E \wedge (\neg C \vee A) \wedge C \wedge B \wedge (\neg G \vee D) \wedge G$

$\quad\;\; = K_1 \wedge K_2 \wedge K_3 \wedge K_4 \wedge K_5 \wedge K_6 \wedge K_7$

Mengendarstellungen der Klauseln

➔ $K_1 = \{\neg A, \neg B, \neg D\}$ $K_2 = \{\neg E\}$ $K_3 = \{\neg C, A\}$

$K_4 = \{C\}$ $K_5 = \{B\}$ $K_6 = \{\neg G, D\}$ $K_7 = \{G\}$

Mengendarstellung der Formel

➔ $F = \{K_1, K_2, K_3, K_4, K_5, K_6, K_7\}$

$= \{\{\neg A, \neg B, \neg D\}, \{\neg E\}, \{\neg C, A\}, \{C\}, \{B\}, \{\neg G, D\}, \{G\}\}$

➔ In der Mengendarstellung sind die Junktoren nicht mehr explizit.

Achtung: Leere Menge in der Mengendarstellung

Bislang

- haben wir Mengen nur als Mengen von Formeln benutzt, wobei wir eine implizite Konjunktion der enthaltenen Formeln als die Standardinterpretation gesetzt haben.
- Eine Formelmenge wird von einer Belegung nur dann nicht wahr gemacht, wenn mindestens eine Formel enthalten ist, die von der Belegung falsch gemacht wird. Entsprechend wird auch die leere Formelmenge von jeder Interpretation wahr gemacht

In der Mengendarstellung brauchen wir eigentlich 2 leere Mengen

- Die leere (Literal-)Menge als Klausel betrachtet wollen wir implizit disjunktiv interpretieren.
- Einen Literal-Menge als Klausel betrachtet wird von einer Belegung genau dann wahr gemacht, wenn mindestens ein enthaltenes Literal von ihr wahr gemacht wird. Entsprechend soll die leere Literal-Menge (leere Klausel) von allen Belegungen zu falsch ausgewertet werden.

Die Mengenlehre stellt aber nur eine leere Menge zur Verfügung.

- Der Kontext muss uns immer die Information geben, ob die leere Menge gerade für die leere Formelmenge oder für die leere Klausel steht.
- Deshalb haben wir das Symbol $\square$ eingeführt. Es steht immer für die leere Klausel !

Bemerkung zur Mengendarstellung

- Für jede Formel in KNF ist die Mengendarstellung eindeutig.
- Die Umkehrung gilt nicht: verschiedene KNF-Formeln können dieselbe Mengendarstellung haben.
- Die Zulässigkeit der Mengendarstellung beruht auf den Gesetzen der Assoziativität, Kommutativität, Idempotenz

Satz (ohne Nummer)

Klauseln bzw. Formeln mit derselben Mengendarstellung sind äquivalent.
(Beweis zur Übung.)

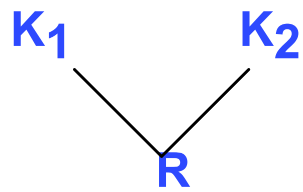
Resolutionsregel / Resolventenregel

Erinnerung: Schreibkonvention für *komplementäre Literale*: $\overline{L} = \begin{cases} A, & \text{falls } L = \neg A \\ \neg A, & \text{falls } L = A \end{cases}$

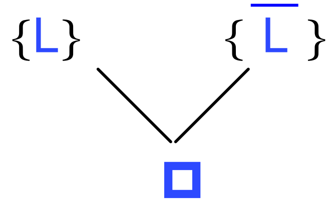
Definition 8.1 (Resolvente)

Seien K_1 und K_2 Klauseln in Mengendarstellung und sei L ein Literal mit $L \in K_1$ und $\overline{L} \in K_2$. Dann heißt die Literalmenge $R = (K_1 - \{L\}) \cup (K_2 - \{\overline{L}\})$ *Resolvente* von K_1 und K_2 (bzgl. L).

- Darstellung als Diagramm:



- Resolventenbildung als Ableitung: *Resolution* $\{K_1, K_2\} \vdash_{\text{res}} R$
- Falls $K_1 = \{L\}$ und $K_2 = \{\overline{L}\}$, so ist die Resolvente leer ($R = \emptyset$) also die *leere Klausel* $\square$.
- Darstellung als Diagramm:



Resolution: Beispiele (1)

Gegeben: Eine Formel in KNF

$A \wedge (\neg A \vee B)$ ist KNF zu $A \wedge (A \Rightarrow B)$

Resolutionsableitung

als Baum von Klauseln in der Mengendarstellung:

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

$\{B\}$

als Baum von Klauseln

A $(\neg A \vee B)$

B

Zur Erinnerung

$A \wedge (\neg A \vee B) \models B$

$\{A, (A \Rightarrow B)\} \vdash_{MP} B$

Resolution: Beispiele (2)

Gegeben:

$(A \vee B) \wedge (\neg A \vee C)$

Resolutionsableitung

$\{A, B\}$ $\quad$ $\{\neg A, C\}$

$\{B, C\}$

Zur Erinnerung

$(\neg A \Rightarrow B) \wedge (A \Rightarrow C) \models (B \vee C)$

$(\neg A \vee B) \wedge (\neg A \vee C)$

$\{\neg A, B\}$ $\quad$ $\{\neg A, C\}$

kein Paar komplementärer Literale !

keine Resolventenbildung !

Resolutionslemma 8.2

Sei F eine Formel in KNF, dargestellt als Klauselmenge. Ferner sei R eine Resolvente zweier Klauseln K_1 und K_2 in F bezüglich des Literals L,

d.h. $\mathsf{R} = (\mathsf{K}_1 - \{\mathsf{L}\}) \cup (\mathsf{K}_2 - \{\overline{\mathsf{L}}\})$.

Dann sind F und $\mathsf{F} \cup \{\mathsf{R}\}$ äquivalent.

Voraussetzungen: Def. 3.1, 8.0, 8.1

Beweis

Sei $\mathcal{A}$ eine Belegung. Zu zeigen: $\mathcal{A}(\mathsf{F}) = \mathbf{1}$ GDW. $\mathcal{A}(\mathsf{F} \cup \{\mathsf{R}\}) = \mathbf{1}$

Falls $\mathcal{A}(\mathsf{F} \cup \{\mathsf{R}\}) = \mathbf{1}$, dann auch $\mathcal{A}(\mathsf{F}) = \mathbf{1}$.

Es sei $\mathcal{A}(\mathsf{F}) = \mathbf{1}$. Zu zeigen ist: $\mathcal{A}(\mathsf{F} \cup \{\mathsf{R}\}) = \mathbf{1}$ insbesondere $\mathcal{A}(\mathsf{R}) = \mathbf{1}$

Für alle Klausel $\mathsf{K} \in \mathsf{F}$ gilt: $\mathcal{A}(\mathsf{K}) = \mathbf{1}$, also auch für K_1 und K_2.

1. Fall: $\mathcal{A}(\mathsf{L}) = \mathbf{0}$: Wegen $\mathcal{A}(\mathsf{K}_1) = \mathbf{1}$ gilt: $\mathcal{A}(\mathsf{K}_1 - \{\mathsf{L}\}) = \mathbf{1}$. Also $\mathcal{A}(\mathsf{R}) = \mathbf{1}$.

2. Fall: $\mathcal{A}(\overline{\mathsf{L}}) = \mathbf{0}$: Wegen $\mathcal{A}(\mathsf{K}_2) = \mathbf{1}$ gilt: $\mathcal{A}(\mathsf{K}_2 - \{\overline{\mathsf{L}}\}) = \mathbf{1}$. Also: $\mathcal{A}(\mathsf{R}) = \mathbf{1}$.

Corollar 8.3

- Resolventenbildung ist *korrekt*, d.h., wenn $\mathbf{M} \vdash_{res} \mathsf{R}$, dann $\mathbf{M} \vDash \mathsf{R}$.

Definition 8.4 (Resolventenmengen)

Sei **F** eine Formel in KNF, dargestellt als Klauselmenge.

$$\mathrm{Res}(\mathbf{F}) := \mathbf{F} \cup \{\mathbf{R} \mid \mathbf{R} \text{ ist Resolvente zweier Klauseln in } \mathbf{F}\}$$

Dies wird induktiv fortgesetzt durch:

$$\mathrm{Res}^0(\mathbf{F}) := \mathbf{F}$$

$$\mathrm{Res}^{n+1}(\mathbf{F}) := \mathrm{Res}(\mathrm{Res}^n(\mathbf{F})) \quad n \geq 0$$

$$\mathrm{Res}^*(\mathbf{F}) := \bigcup_{n\geq 0} \mathrm{Res}^n(\mathbf{F})$$

Die Bildung von Resolventenmengen entspricht der Bildung von Mengen ableitbarer Formeln in Kalkülen der Aussagenlogik (vgl. Def. 6. 8).

Sei **M** eine Formelmenge und $\mathcal{C}$ ein Kalkül:

$$\mathrm{Abl}_{\mathcal{C}}(\mathbf{M}) := \mathbf{M} \cup \{\mathrm{F} \mid \mathbf{M} \vdash_{\mathcal{C}} \mathrm{F}\} \qquad \text{in einem Schritt aus } \mathbf{M} \text{ ableitbare Formeln}$$

$$\mathrm{Abl}_{\mathcal{C}}^0(\mathbf{M}) := \mathbf{M}$$

$$\mathrm{Abl}_{\mathcal{C}}^{n+1}(\mathbf{M}) := \mathrm{Abl}_{\mathcal{C}}(\mathrm{Abl}_{\mathcal{C}}^n(\mathbf{M})) \quad n \geq 0$$

$$\mathrm{Abl}_{\mathcal{C}}^*(\mathbf{M}) := \bigcup_{n\geq 0} \mathrm{Abl}_{\mathcal{C}}^n(\mathbf{M})$$

Resolution: Beispiel (3)

Gegeben: Eine Formel in KNF

$(\neg C \vee A) \wedge C \wedge (\neg A \vee \neg B \vee \neg D) \wedge B \wedge (\neg G \vee D) \wedge G \wedge \neg E$

Resolutionsableitung (mehrschrittig)

$\mathrm{Res}^0(\mathbf{F}) = \mathbf{F}$ $\{\neg C, A\}$ $\{C\}$ $\{\neg A, \neg B, \neg D\}$ $\{B\}$ $\{\neg G, D\}$ $\{G\}$ $\{\neg E\}$

$\mathrm{Res}^1(\mathbf{F})$ $\{A\}$ $\{\neg A, \neg D\}$ $\{D\}$

$\mathrm{Res}^2(\mathbf{F})$ $\{\neg A\}$

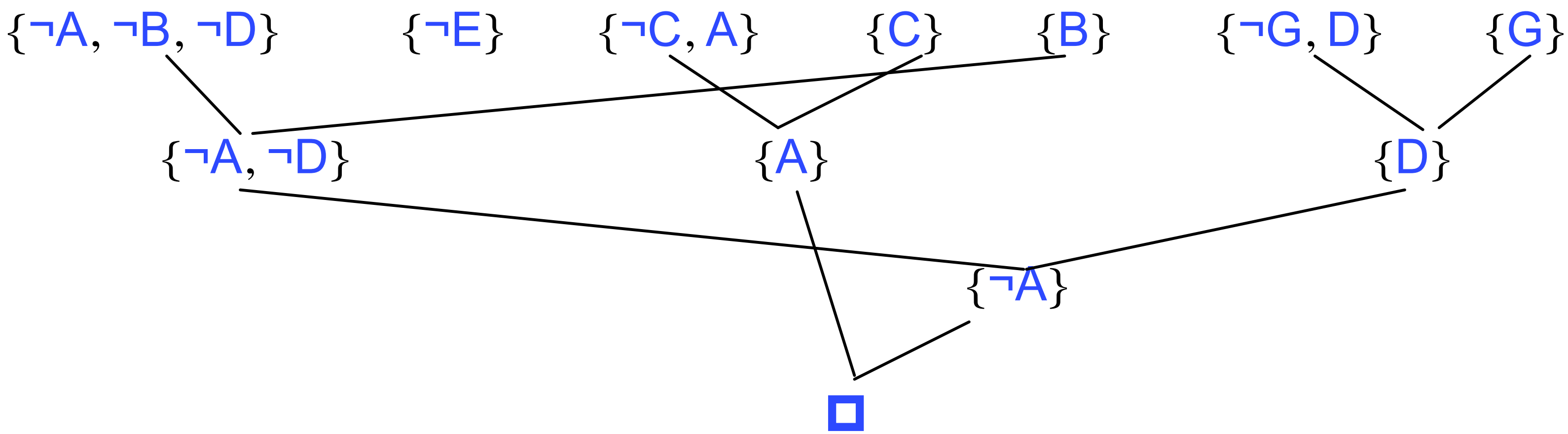
$\mathrm{Res}^3(\mathbf{F})$ $\square$

Resolution: Beispiel (4)

Gegeben: Die gleiche Formel in anderer Anordnung der Klauseln

$$(\neg A \vee \neg B \vee \neg D) \wedge \neg E \wedge (\neg C \vee A) \wedge C \wedge B \wedge (\neg G \vee D) \wedge G$$

Resolutionsableitung (bei Beibehaltung der Reihenfolge aus der KNF):



➔ Die Anordnung der Klauseln in der Basiszeile, beeinflusst nicht das Ergebnis, aber die Übersichtlichkeit der Resolutionsableitung.

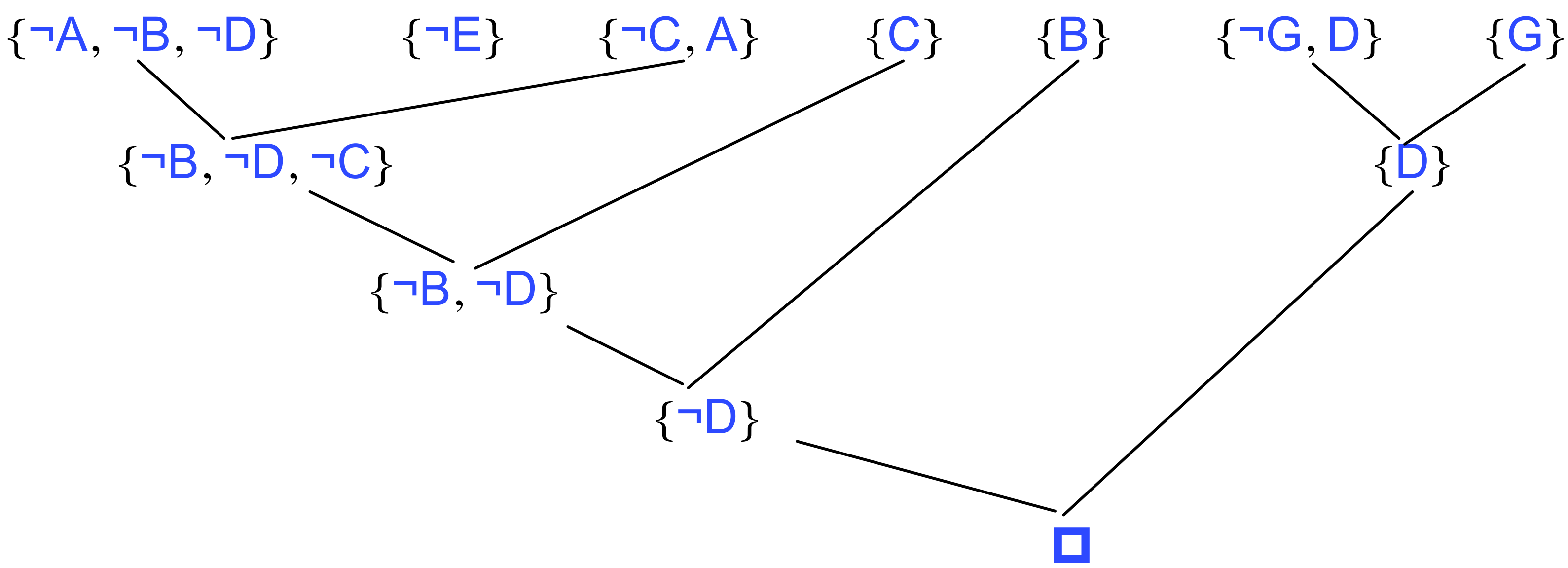
➔ Es müssen nicht alle Klauseln an der Ableitung beteiligt sein.

Resolution: Beispiel (5)

Gegeben: Die gleiche Formel in KNF

$(\neg A \vee \neg B \vee \neg D) \;\wedge\; \neg E \;\wedge\; (\neg C \vee A) \;\wedge\; C \;\wedge\; B \;\wedge\; (\neg G \vee D) \;\wedge\; G$

Eine andere Resolutionsableitung:



Resolventenmengen: Äquivalenz

Lemma 8.5 (Äquivalenz der Resolventenmengen)

Sei $\mathbf{F}$ eine Formel in KNF, dargestellt als Klauselmenge.

Dann gilt: $\mathbf{F} \equiv \mathrm{Res}(\mathbf{F})$ und $\mathbf{F} \equiv \mathrm{Res}^*(\mathbf{F})$

Beweis

$\mathbf{F}$ ist eine endliche Menge von Klauseln, daher gibt es eine endliche Menge von Klauselpaaren (mit einer endlichen Anzahl von Literalen), auf die die Resolventenregel angewendet werden kann.

Somit gibt es ein $n \geq 0$, so dass $\mathbf{R_1}, \ldots, \mathbf{R_n}$ eine Aufzählung aller Resolventen zweier Klauseln aus $\mathbf{F}$ ist.

Dann gilt: $\mathrm{Res}(\mathbf{F}) = ((\ldots((\mathbf{F} \cup \{\mathbf{R_1}\}) \cup \{\mathbf{R_2}\})\ldots) \cup \{\mathbf{R_n}\})$

Aus dem Resolutionslemma 8.2 ergibt sich (mit vollständiger Induktion):

$$\mathbf{F} \equiv \mathbf{F} \cup \{\mathbf{R_1}\} \equiv (\mathbf{F} \cup \{\mathbf{R_1}\}) \cup \{\mathbf{R_2}\} \equiv \ldots \equiv \mathrm{Res}(\mathbf{F})$$

➔ Entsprechend lässt sich hieraus (mit vollständiger Induktion) zeigen:

$$\mathbf{F} \equiv \mathrm{Res}^1(\mathbf{F}) \equiv \mathrm{Res}^2(\mathbf{F}) \equiv \ldots \equiv \mathrm{Res}^*(\mathbf{F})$$

Lemma 8.5.1

Ist **F** eine Klauselmenge und $\mathbf{K} \in \mathrm{Res}^*(\mathbf{F})$, dann gibt es eine endliche Teilmenge $\mathbf{G} \subseteq \mathbf{F}$, so dass $\mathbf{K} \in \mathrm{Res}^*(\mathbf{G})$.

Voraussetzungen: Def. 8.0, 8.4

Beweis

(Interessant ist natürlich nur der Fall, dass **F** selbst eine unendliche Menge ist.)

Es sei **F** eine Klauselmenge und $\mathbf{K} \in \mathrm{Res}^*(\mathbf{F})$.

Nach Def. 8.4 gibt es dann ein n, so dass $\mathbf{K} \in \mathrm{Res}^n(\mathbf{F})$.

Zu zeigen ist also (mit vollständiger Induktion):

Für alle n und $\mathbf{K} \in \mathrm{Res}^n(\mathbf{F})$, gibt es eine endliche Teilmenge $\mathbf{G} \subseteq \mathbf{F}$, so dass $\mathbf{K} \in \mathrm{Res}^n(\mathbf{G})$.

Induktionsanfang

Ist $n = 0$, dann ist $\mathbf{K} \in \mathrm{Res}^0(\mathbf{F}) = \mathbf{F}$ und mit $\mathbf{G} = \{\mathbf{K}\} \subseteq \mathbf{F}$ haben wir die gesuchte Menge.

Induktionsannahme

Für alle $i < n$ und $\mathbf{K} \in \mathrm{Res}^i(\mathbf{F})$, gibt es eine endliche Teilmenge $\mathbf{G} \subseteq \mathbf{F}$, so dass $\mathbf{K} \in \mathrm{Res}^i(\mathbf{G})$.

Zum Selbststudium: Fortsetzung

Induktionsschritt

Es sei $K' \in Res^n(F) = Res\,(Res^{n-1}(F))$

$= Res^{n-1}(F) \cup \{R \mid R$ ist Resolvente zweier Klauseln in $Res^{n-1}(F)\}$ (Def. 8.4)

Fall 1: Ist $K' \in Res^{n-1}(F)$, dann ist die Induktionsannahme auf K' anwendbar.

Fall 2: Ist K' Resolvente zweier Klauseln (K_1 und K_2) in $Res^{n-1}(F)$, dann ist die Induktionsannahme auf K_1 und K_2 anwendbar.

Demnach gibt es zwei endliche Teilmengen $G_1, G_2 \subseteq F$, so dass $K_1 \in Res^{n-1}(G_1)$ und $K_2 \in Res^{n-1}(G_2)$.

Damit ist dann

$K' \in \{R \mid R$ ist Resolvente zweier Klauseln in $Res^{n-1}(G_1 \cup G_2)\} \subseteq Res^n(G_1 \cup G_2)$

und $G_1 \cup G_2$ ist endlich.

Resolutionssatz 8.6

Eine Klauselmenge **F** ist genau dann unerfüllbar, wenn $\square \in \mathrm{Res}^*(F)$, d.h. $F \vdash_{res} \square$

Voraussetzungen: Def. 3.3, 3.4, 8.0, 8.4, Lemma 8.5, 8.5.1

Beweis (1. Teil: w-Korrektheit)

Sei $\square \in \mathrm{Res}^*(F)$. Zu zeigen ist, dass **F** unerfüllbar ist.

Nach Lemma 8.5.1 gibt es eine endliche Teilmenge $G \subseteq F$, so dass $\square \in \mathrm{Res}^*(G)$.

Da $\square \in \mathrm{Res}^*(G)$ ist $\mathrm{Res}^*(G)$ unerfüllbar.

Nach Lemma 8.5 ist $G \equiv \mathrm{Res}^*(G)$, also ist auch **G** unerfüllbar
und mit $G \subseteq F$ ist **F** unerfüllbar.

Beweis (2. Teil: w-Vollständigkeit):

Sei **F** unerfüllbar. Zu zeigen: $\square \in \mathrm{Res}^*(F)$.

Ist **F** unendlich und unerfüllbar, dann hat **F** eine endliche unerfüllbare Teilmenge (Endlichkeitssatz 5.18).
Daher reicht es, den Beweis für endliche Klauselmengen zu führen.

Beweis der Widerlegungsvollständigkeit (2)

Grundidee

- Die Klauseln aus F werden in zwei Teilmengen F_0^+ und F_1^+ sortiert, so dass gilt
  - $F_0^+ \cup F_1^+ \subseteq F$
  - In F_0^+ kommt A_{n+1} nur als positives Literal vor.
  - In F_1^+ kommt A_{n+1} nur als negatives Literal vor.
  - (Klauseln mit komplementären A_{n+1} aus F werden ignoriert.)
- und es wird gezeigt
  - $F_0^+ \vdash_{res} \square$ oder $F_0^+ \vdash_{res} A_{n+1}$
  - $F_1^+ \vdash_{res} \square$ oder $F_1^+ \vdash_{res} \neg A_{n+1}$
  - Und damit kann ggf. mit einem letzten Resolutionsschritt $F \vdash_{res} \square$ gezeigt werden.
- Auf dem Weg dorthin
  - werden F_0 und F_1 definiert (durch Streichung der A_{n+1}-Literale aus F_0^+ und F_1^+)
  - und es wird gezeigt, dass $F_0 \vdash_{res} \square$ und $F_1 \vdash_{res} \square$.
  - Die hierbei zugrundeliegenden Ableitungen werden dann zu den erforderlichen Ableitungen auf der Basis von F_0^+ und F_1^+ umgestaltet.

Beweis der Widerlegungsvollständigkeit (3)

Zu zeigen: Für jede endliche und unerfüllbare Klauselmenge **F** ist $\square \in \mathrm{Res}^*(\mathbf{F})$.

Beweis erfolgt durch vollständige Induktion über die Anzahl der Aussagensymbole in **F**.

Induktionsanfang (n = 0):

Wenn es keine Aussagensymbole in **F** gibt und **F** unerfüllbar ist, dann ist $\mathbf{F} = \{\square\}$ also gilt auch $\square \in \mathrm{Res}^*(\mathbf{F})$.

Induktionsannahme: Es sei $n \in \mathbb{N}$ so gewählt, dass für jede unerfüllbare Klauselmenge **G** über den Aussagensymbolen $A_1, \ldots, A_n$ gilt, $\square \in \mathrm{Res}^*(\mathbf{G})$.

Induktionsschritt

Sei nun **F** eine unerfüllbare Klauselmenge über den atomaren Formeln $A_1, \ldots, A_n, A_{n+1}$. Aus **F** werden zwei neue Klauselmengen $\mathbf{F_0}$ und $\mathbf{F_1}$ gebildet, in denen A_{n+1} nicht vorkommt.

Wenn **K** in **F**	Dann $\mathbf{K_0}$ in $\mathbf{F_0}$	Dann $\mathbf{K_1}$ in $\mathbf{F_1}$
K enthält nur Formeln aus $A_1, \ldots, A_n$	$\mathbf{K_0} := \mathbf{K}$	$\mathbf{K_1} := \mathbf{K}$
K enthält A_{n+1} und $\neg A_{n+1}$	—	—
K enthält A_{n+1}	$\mathbf{K_0} := \mathbf{K} - \{A_{n+1}\}$	—
K enthält $\neg A_{n+1}$	—	$\mathbf{K_1} := \mathbf{K} - \{\neg A_{n+1}\}$

Beispiel zur Konstruktion von F_0 und F_1

Sei $F = \{\{A, \neg B\}, \{B, C\}, \{\neg A, D\}, \{\neg D\}, \{\neg C\}\}$ $\quad A_4 = D$

Konstruktion von F_0 und F_1:

Wenn K in F		
	Dann in F_0	
K enthält nur Formeln aus A, B, C	K	$\{A, \neg B\}, \{B, C\}, \{\neg C\}$
K enthält D	$K - \{D\}$	$\{\neg A\}$
K enthält $\neg D$	—	
	Dann in F_1	
K enthält nur Formeln aus A, B, C	K	$\{A, \neg B\}, \{B, C\}, \{\neg C\}$
K enthält D	—	
K enthält $\neg D$	$K - \{\neg D\}$	$\square$

➔ $F_0 = \{\{A, \neg B\}, \{B, C\}, \{\neg A\}, \{\neg C\}\}$

$F_1 = \{\{A, \neg B\}, \{B, C\}, \square, \{\neg C\}\}$

Beweis der Widerlegungsvollständigkeit (4)

- Hilfssatz: Unter den gegebenen Voraussetzungen sind $\mathbf{F_0}$ und $\mathbf{F_1}$ beide unerfüllbar
- Beweis: Annahme: Es gibt eine Belegung $\mathcal{A}$ für $A_1, \ldots, A_n$, die $\mathbf{F_0}$ erfüllt.
  - Konstruktion einer Fortsetzung von $\mathcal{A}$ für $\mathbf{F}$:
  - $\mathcal{A}_0(B) = \begin{cases} \mathcal{A}(B) \text{ falls } B \in \{A_1, \ldots, A_n\} \\ \mathbf{0}, \quad \text{falls } B = A_{n+1} \end{cases}$
  - $\mathcal{A}_0$ wäre eine erfüllende Belegung für $\mathbf{F}$, im Widerspruch zu den Annahmen.
  - Analog ergibt sich, dass auch $\mathbf{F_1}$ unerfüllbar ist (unter Betrachtung der Fortsetzung $\mathcal{A}_1(A_{n+1}) = \mathbf{1}$ einer erfüllenden Belegung $\mathcal{A}$).

➔ Auf $\mathbf{F_0}$ und $\mathbf{F_1}$ trifft die Induktionsannahme zu, d.h. es gilt:
$\square \in \text{Res}^*(\mathbf{F_0})$ und $\square \in \text{Res}^*(\mathbf{F_1})$

Als nächstes wird aus den entsprechenden Ableitungen eine Resolutionsableitung für $\square \in \text{Res}^*(\mathbf{F})$ konstruiert

Beispiel

$\mathbf{F_0} = \{\{A, \neg B\}, \{B, C\}, \{\neg A\}, \{\neg C\}\}$

$\mathbf{F_1} = \{\{A, \neg B\}, \{B, C\}, \square, \{\neg C\}\}$

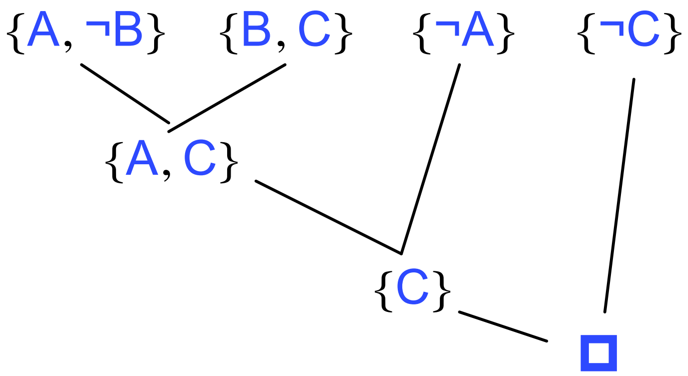
Beispiel Resolutionsableitung zu F_0 und F_1

$\mathbf{F} = \{\{A, \neg B\}, \{B, C\}, \{\neg A, D\}, \{\neg D\}, \{\neg C\}\}$

$\mathbf{F_0} = \{\{A, \neg B\}, \{B, C\}, \{\neg A\}, \{\neg C\}\}$

$\mathbf{F_1} = \{\{A, \neg B\}, \{B, C\}, \square, \{\neg C\}\}$

Resolutionsableitung für $\mathbf{F_0}$



$\mathbf{F_1}$ (hier ist nichts zu tun)

$\square$

Beweis der Widerlegungsvollständigkeit (5)

Wir konnten feststellen, dass $\square \in \mathrm{Res}^*(F_0)$.

- Das heißt, es gibt Klauseln $K_1, \ldots, K_m$, so dass $K_m = \square$ und für $1 \leq i \leq m$ gilt: $K_i \in F_0$ oder K_i ist Resolvente von Klauseln K_a, K_b mit $a, b < i$.
- Einige der K_i der ersten Sorte entstanden aus F durch Streichen von A_{n+1}. Für diese Klauseln wird die ursprüngliche Klausel wiederhergestellt:

(*) $K_i^+ = K_i \cup \{A_{n+1}\}$.

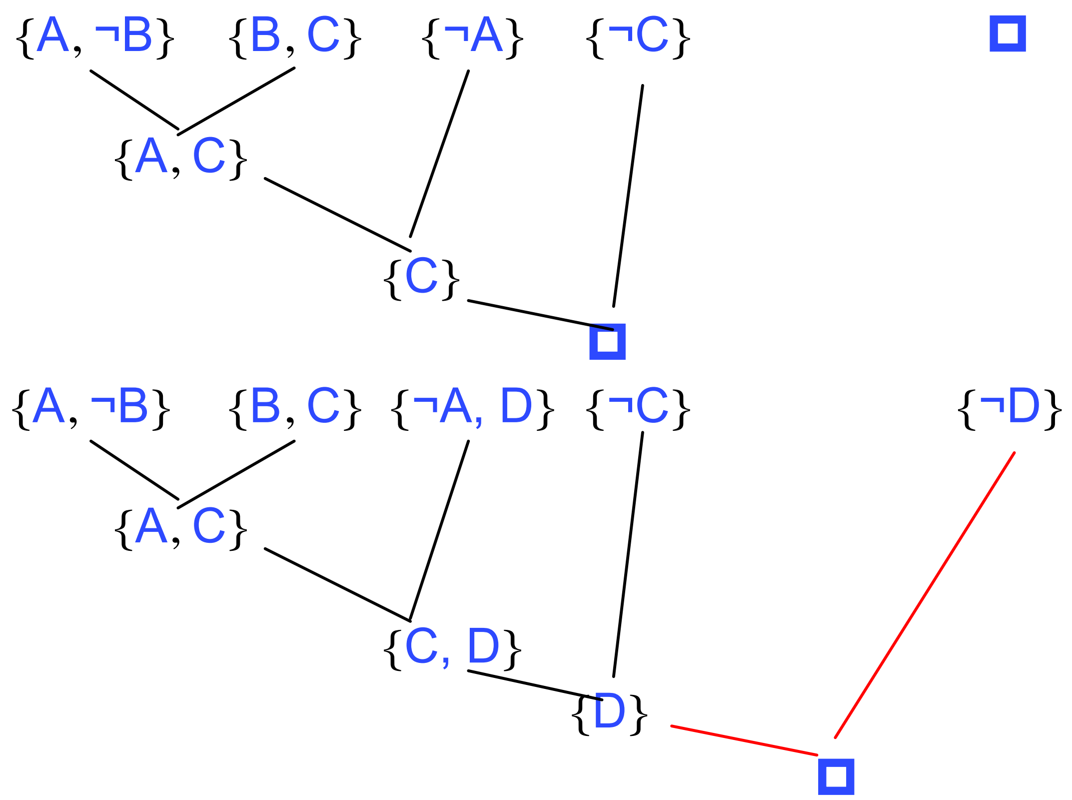
- Einige der K_i der zweiten Sorte sind Resolventen, die direkt oder indirekt auf Klauseln beruhen, die durch Streichen von A_{n+1} aus F erzeugt wurden.
- Für diese Klauseln wird eine neue Klausel entsprechend (*) definiert.
- Für alle anderen Klauseln wird $K_i^+ = K_i$ gesetzt.
- Für $1 \leq i \leq m$ gilt: $K_i^+ \in F$ oder K_i^+ ist Resolvente von Klauseln K_a^+, K_b^+ mit $a, b < i$. Insbesondere ist $K_m^+ \in \mathrm{Res}^*(F)$ und $K_m^+ = \{A_{n+1}\}$ oder $K_m^+ = \square$
- Entsprechend ergibt sich aus $\square \in \mathrm{Res}^*(F_1)$: $\{\neg A_{n+1}\} \in \mathrm{Res}^*(F)$ oder $\square \in \mathrm{Res}^*(F)$
- Also haben wir schon $\square \in \mathrm{Res}^*(F)$ oder es ergibt sich durch einen weiteren Resolutionsschritt: $\{\{A_{n+1}\}, \{\neg A_{n+1}\}\} \vdash_{res} \square$, also $\square \in \mathrm{Res}^*(F)$

Beispiel: Rekonstruktion der Resolutionsableitung von F

$\mathbf{F} = \{\{A, \neg B\}, \{B, C\}, \{\neg A, D\}, \{\neg D\}, \{\neg C\}\}$

$\mathbf{F_0} = \{\{A, \neg B\}, \{B, C\}, \{\neg A\}, \{\neg C\}\}$ $\qquad$ $\mathbf{F_1} = \{\{A, \neg B\}, \{B, C\}, \square, \{\neg C\}\}$

Resolutionsableitung für $\mathbf{F_0}$ $\qquad\qquad$ $\mathbf{F_1}$



Resolutionsalgorithmus

Eingabe: Eine Formel **F** in KNF (als Klauselmenge), d.h. eine **endliche** Klauselmenge !

REPEAT
    G := **F**;
    F := Res(**F**);
    UNTIL (□ ∈ **F**) OR (**F** = **G**)
IF □ ∈ **F** THEN „**F** ist unerfüllbar"
    ELSE „**F** ist erfüllbar"

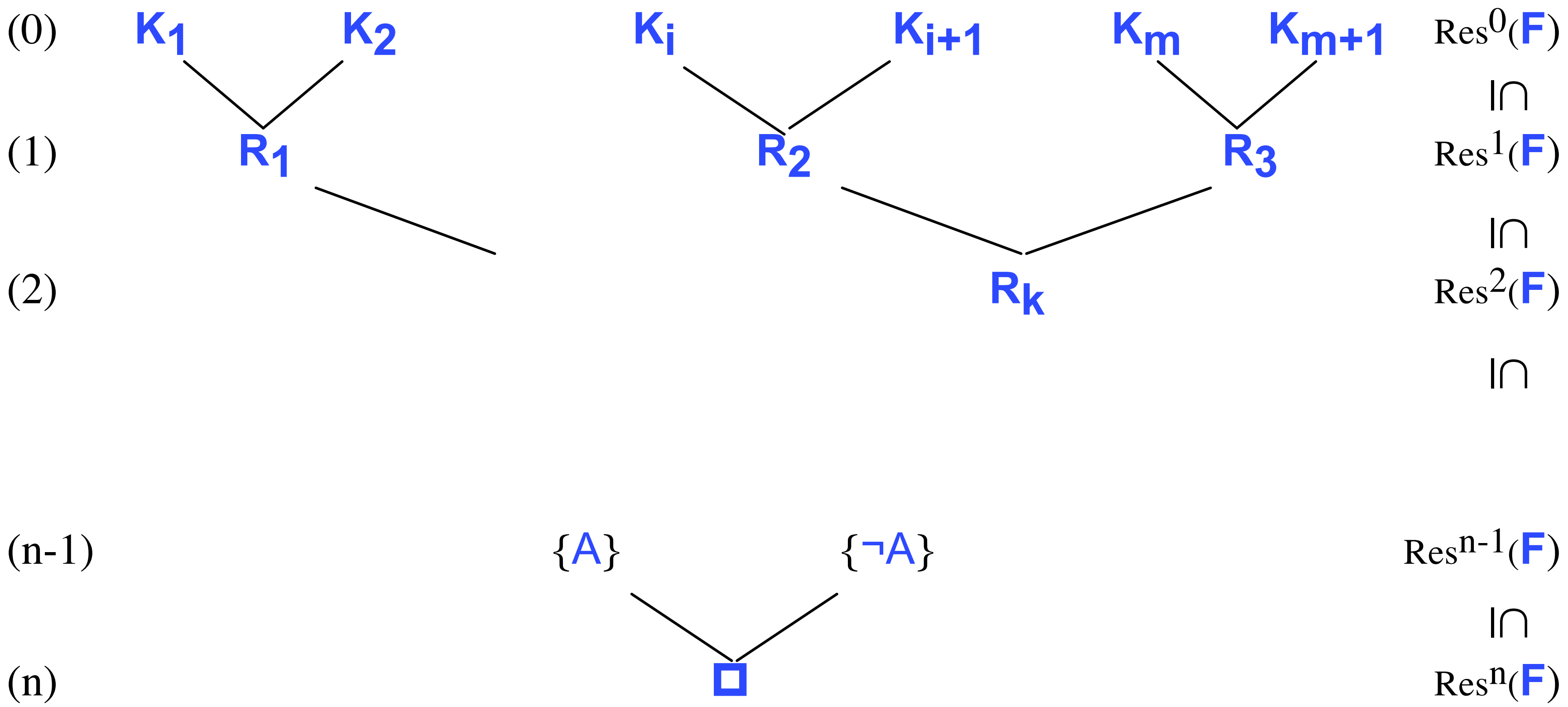
- Bei n Aussagensymbolen gibt es maximal 4^n Klauseln.
- **F** = **G** Prüfung sichert den Abbruch, wenn keine neuen Resolventen mehr gefunden werden.
- Aufwand des Resolutionsalgorithmus' ist exponentiell.

    ➔ Entwicklung effizienterer Resolutionsalgorithmen, die jedoch nicht vollständig sind.

- Wenn (Un)Erfüllbarkeit einer unendlichen Klauselmenge **M** zu prüfen ist, wird der Resolutionsalgorithmus auf Folgen endlicher Teilmengen angewendet. Wenn **M** erfüllbar ist, dann bricht dieses Verfahren nie ab !

Struktur von Resolutionsableitungen zu $\square$

$\{K_1, \ldots, K_{m+1}\} \vdash_{res} \square$



Verfeinerung der Resolution

Resolution – Komplexitätsprobleme

- „Kombinatorische Explosion“ bei der Erzeugung aller Resolventen
- Keine Sicherstellung der Terminierung bei nicht-endlichen Klauselmengen
  (→ Resolution in der Prädikatenlogik)

Lösungsansätze

Auswahlstrategien

- Heuristische Regeln für Auswahl zu resolvierender Klauseln, wobei aber (notfalls) alle Resolventen gebildet werden können.
- Theoretisch unklar, in welchem Maße derartige Strategien wirkungsvoll sind.

Auswahlrestriktionen

- Einschränkung der Resolutionsmöglichkeiten, d.h. gewisse Resolutionsschritte werden ausgeschlossen.
  ➔ Die w-Vollständigkeit eines derartig modifizierten Kalküls muss untersucht werden.
  (Vgl. Schöning Kap. 2.6)
- Korrektheit ist bei allen Ansätzen sichergestellt, da die Resolventenbildung korrekt ist.

Auswahlstrategie bei der Resolution

Beispiel: Präferenz für kleine Klauseln

Erzeuge möglichst kleine Klauseln, wähle Klauseln mit möglichst wenig Elementen.

- ➔ Immer sinnvoll, wenn der Mensch resolvieren soll.
- ➔ Geeignet, den Resolutionsaufwand zu verringern, da die Zielklausel keine Elemente enthält.

- Aber in schwierigen Fällen kann es nötig sein, auch sehr große Klauseln zu erzeugen, bevor man zur leeren Klausel kommt.

Restriktionen: P-Resolution / N-Resolution

Definition 8.7

- Im P-Resolutionskalkül darf nur dann die Resolvente aus K_1 und K_2 gebildet werden, wenn eine der beiden Klauseln ausschließlich positive Literale enthält.
- Im N-Resolutionskalkül darf nur dann die Resolvente aus K_1 und K_2 gebildet werden, wenn eine der beiden Klauseln ausschließlich negative Literale enthält

➔ P-Resolution und N-Resolution sind w-vollständig.

- Bei der P-Resolution entstehende Resolventen haben ein negatives Literal weniger als jede der eingehenden Formeln.
- Entsprechend reduziert N-Resolution positive Literale.

Zur Übung zu beweisen

- Enthalten alle Klauseln der Klauselmenge F mindestens ein negatives Literal, dann ist F erfüllbar.
- Enthalten alle Klauseln der Klauselmenge F mindestens ein positives Literal, dann ist F erfüllbar.

P-Resolution / N-Resolution – Beispiel

Sei **F** = {{¬A, B, C}, {¬A, B, D}, {¬C, E}, {¬E, ¬B}{E}, {A}, {¬E, ¬D}}

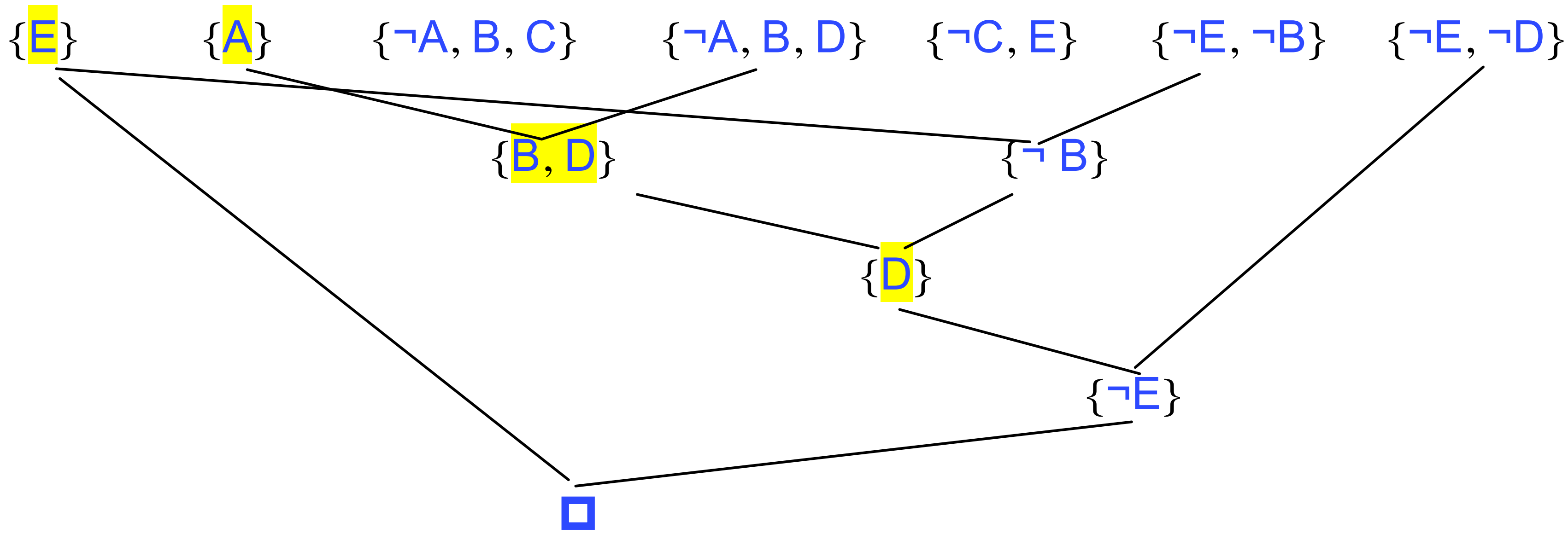
Systematische Bildung einer Teilmenge von Res(**F**):

K_1	K_2	R	Typ von **R**
{¬A, B, C}	{¬C, E}	{¬A, B, E}	
{¬A, B, C}	{¬E, ¬B}	{¬A, ¬E, C}	
{¬A, B, C}	{A}	{B, C}	P
{¬A, B, D}	{¬E, ¬B}	{¬A, ¬E, D}	
{¬A, B, D}	{A}	{B, D}	P
{¬A, B, D}	{¬E, ¬D}	{¬A, B, ¬E}	
{¬C, E}	{¬E, ¬B}	{¬C, ¬B}	N
{¬C, E}	{¬E, ¬D}	{¬C, ¬D}	N
{¬E, ¬B}	{E}	{¬B}	N
{E}	{¬E, ¬D}	{¬D}	N

➔ Anzahl der Resolutionsmöglichkeiten der ersten Stufe: 10
bei P-Resolution 4
bei N-Resolution 7

$\mathbf{F} = \{\{\neg A, B, C\}, \{\neg A, B, D\}, \{\neg C, E\}, \{\neg E, \neg B\}, \{E\}, \{A\}, \{\neg E, \neg D\}\}$

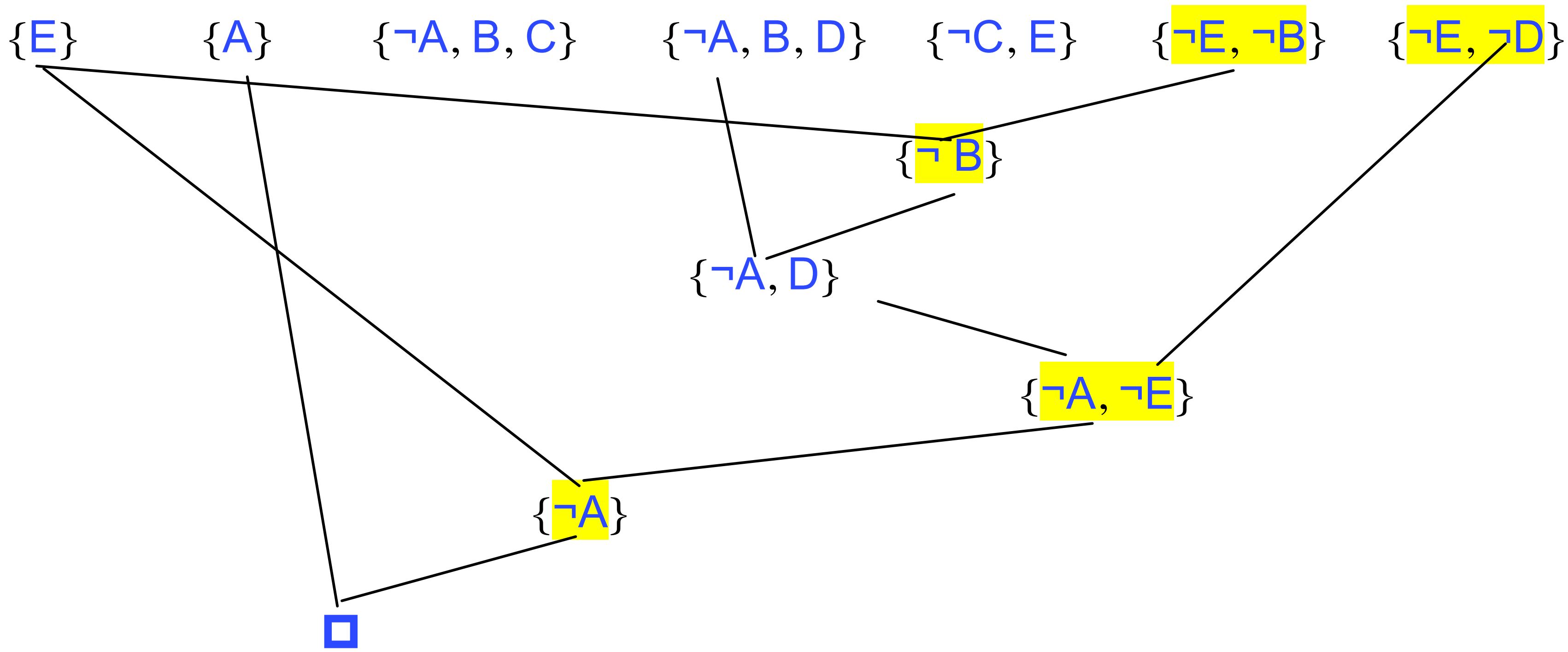
P-Resolution



N-Resolution – Beispiel

F = {{¬A, B, C}, {¬A, B, D}, {¬C, E}, {¬E, ¬B}, {E}, {A}, {¬E, ¬D}}

N-Resolution



Definition 8.8

Eine Resolutionsableitung $K_1, K_2, \ldots, K_n$ ist *linear basierend auf der Klausel* $K \in F$, falls gilt:

$K_1 = K$

Für $i > 1$ gilt: K_i ist die Resolvente aus K_{i-1} und einer Klausel B_{i-1}, wobei B_{i-1} entweder Element von F oder $B_{i-1} = K_j$, mit $j < i$.

(B_{i-1} wird *Seitenklausel* genannt)

Eine Klauselmenge F ist *linear resolvierbar basierend auf der Klausel* K, falls eine lineare Resolutionsableitung basierend auf K zur leeren Klausel existiert.

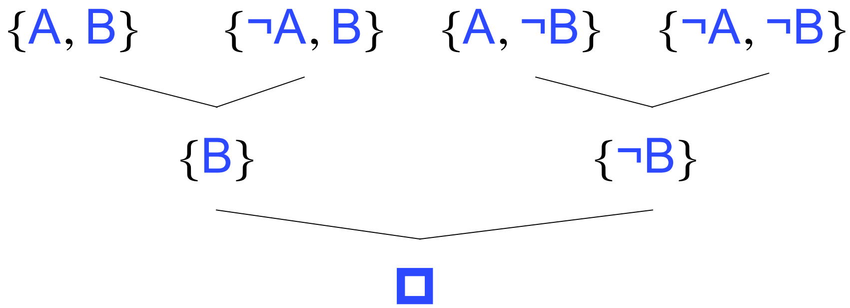
Lineare Resolution

- führt gegebenenfalls zu längeren Ableitungen
- reduziert jedoch die Möglichkeiten der Resolutionsbildung.

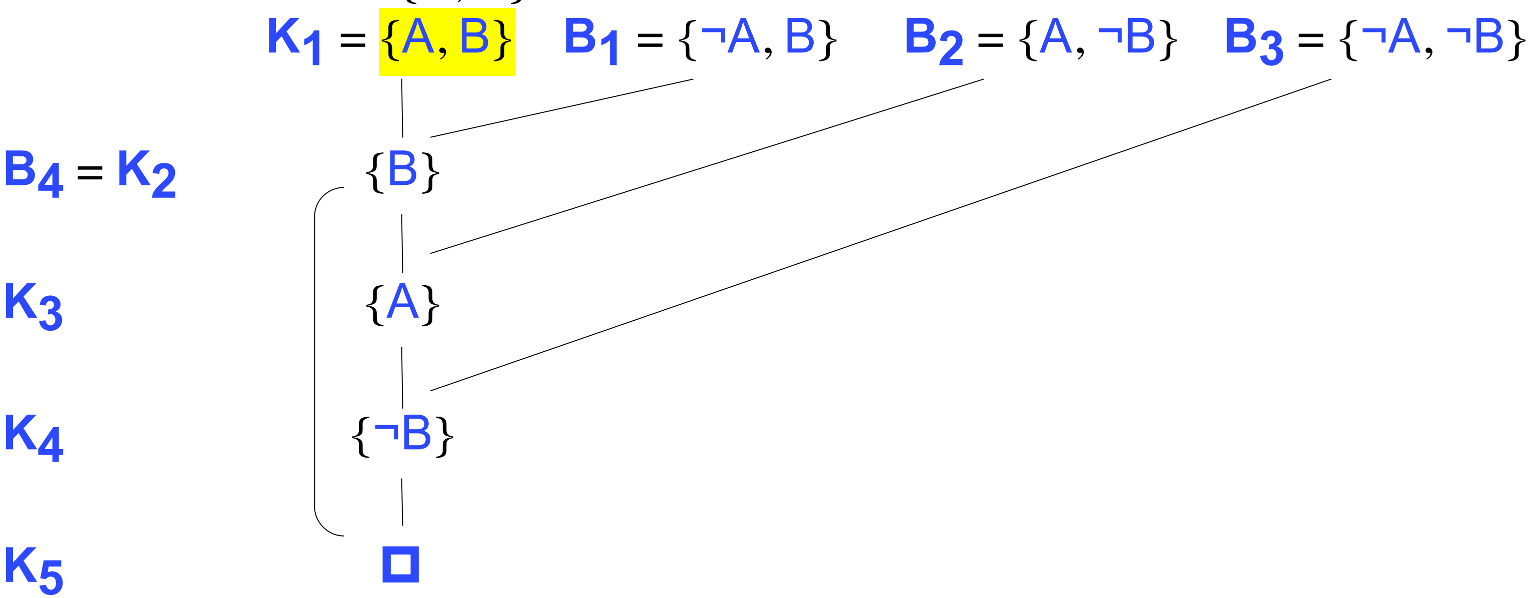
➔ Lineare Resolution ist w-vollständig, jedoch hängt der Erfolg von der Wahl von K ab.

Lineare Resolution – Beispiel

nicht linear



linear basierend auf {A, B}



Stützmengenrestriktion (Set of support)

Ausgangssituation

- Gegeben eine Klauselmenge **F** und eine Teilmenge **T** $\subseteq$ **F**, so dass **F** – **T** erfüllbar ist.
- Eine *Resolutionsableitung bzgl. der Stützmenge* **T** ist eine Resolutionsableitung, bei der niemals zwei Klauseln aus **F** – **T** miteinander resolviert werden,
  d.h. bei jeder Resolution ist eine Klausel aus **T** direkt oder indirekt beteiligt.
- Stützmengenresolution ist linear, wenn | **T** | = 1.
- Stützmengenresolution ist besonders vorteilhaft, wenn | **T** | klein und somit | **F** – **T** | groß ist.

➔ Stützmengenresolution ist w-vollständig (wenn **F** – **T** erfüllbar ist).

Anwendungsfall

- Gegeben eine konsistente Wissensbasis / Datenbank in KNF: **F'**.
- Aufgabe: Eine Datenbankanfrage G. Zu prüfen ist **F'** $\models$ G
- Sei **G'** eine Mengendarstellung zu ¬G. Zu prüfen ist, ob **F** = **F'** $\cup$ **G'** erfüllbar ist.
- Stützmengenrestriktion mit **T** = **G'** , **F** = **F'** $\cup$ **G'** und **F'** = **F** – **T** ist gesteuert durch die Anfrage und sucht nicht nach Widersprüchen in der Datenbank.

Einheitsresolution (Unit resolution)

Definition 8.9

Bei der Resolventenbildung im Kalkül der *Einheitsresolution* muss mindestens eine der Elternklauseln einelementig sein.

- Idee der Einheitsresolution: Die Anzahl der Literale verringert sich bei der Resolventenbildung.
- Einheitsresolution ist *nicht* w-vollständig.

  $\{A, B\} \quad \{\neg A, B\} \quad \{A, \neg B\} \quad \{\neg A, \neg B\}$

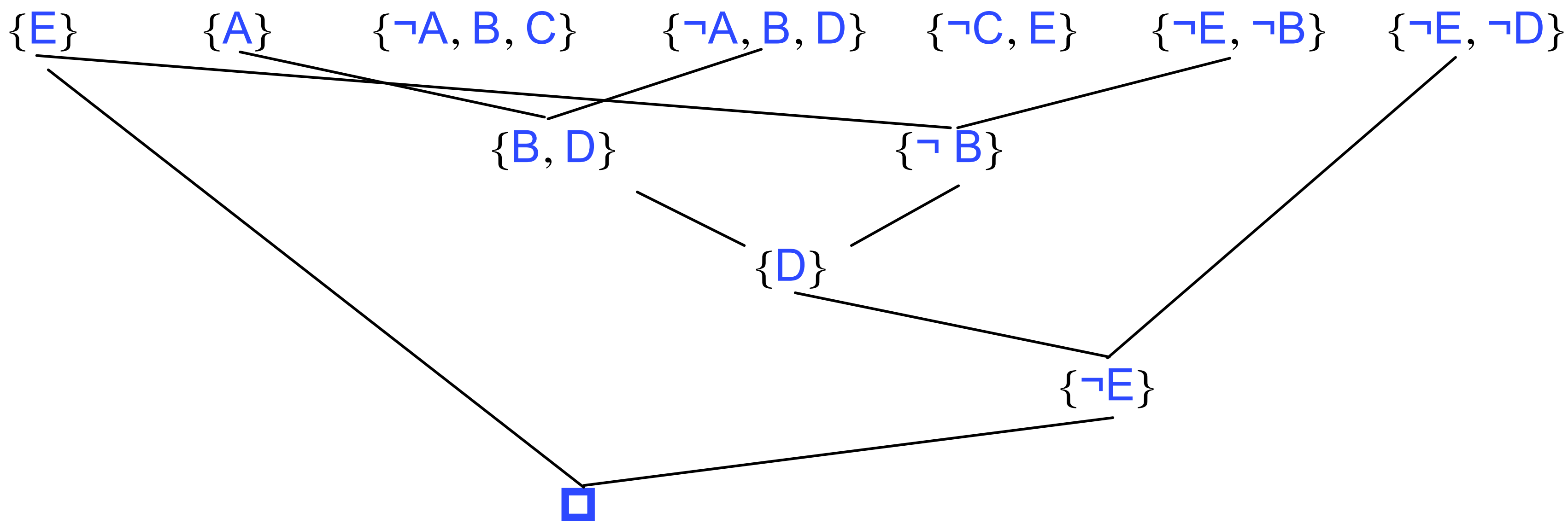
  kann nicht durch Einheitsresolution behandelt werden.

- Einheitsresolution ist w-vollständig, falls eine **Hornformel** auf Erfüllbarkeit zu prüfen ist.

Einheitsresolution – Beispiel

F = {{¬A, B, C}, {¬A, B, D}, {¬C, E}, {¬E, ¬B}, {E}, {A}, {¬E, ¬D}}

Einheitsresolution



Übersicht: Resolution

Aussagenlogische Resolution

Prädikatenlogische Resolution

- aussagenlogische Resolution + Abbildung von prädikatenlogischen Formeln auf aussagenlogische Formelmenge (Bildung von Grundinstanzen)
- oder Allgemeine Resolution mit Unifikation
- w-Vollständigkeit der aussagenlogischen und der prädikatenlogischen Resolution

---

Restriktionen – alle auch für den aussagenlogischen Fall

- w-Vollständigkeit muss bewiesen werden (Modifikation des Resolutionssatzes)
- Einige w-vollständige Kalküle
  - P-Resolution / N-Resolution: Systematische Reduktion negativer bzw. positiver Literale
  - Lineare Resolution / Stützmengenresolution:
    - gezieltes Bearbeiten – für die Widerlegung – aussichtsreicher Klauseln
    - nur w-vollständig bei geeigneter Wahl der Stützmenge

Wichtige Konzepte dieser Vorlesung

- Mengendarstellung für konjunktive Normalformen (Mengen von Mengen von Literalen)
- Leere Klausel ($\square$)
- Resolutionsregel, Korrektheit, Resolventenmengen von Formeln
- Resolution als Widerlegungsverfahren (w-Korrektheit und w-Vollständigkeit)
- Resolutionsalgorithmus
- Auswahlstrategien: P-Resolution, N-Resolution, (lineare Resolution, Stützmengenresolution, Einheitsresolution)

Prädikatenlogik

Von der Aussagenlogik zur Prädikatenlogik

- Prädikatenlogik ist eine Erweiterung der Aussagenlogik
  - Syntax der Prädikatenlogik
  - Semantik der Prädikatenlogik

Was unterscheidet die Prädikatenlogik von der Aussagenlogik?

- Größere Ausdrucksstärke: feinere Differenzierungen

➔ Größerer Aufwand bei Berechnung semantischer Eigenschaften und Beziehungen

Von der Aussagenlogik zur Prädikatenlogik: Motivation

Aussagenlogik

- Aussagen sind die kleinsten bedeutungstragenden Einheiten

Beispiele

- Hamburg ist eine Stadt.
- München ist eine Stadt.
- Hamburg liegt nördlich von München.
- München liegt südlich von Hamburg.

Atomare Formeln AL Aussagensymbole	Atomare Formeln PL Interne Struktur
Shh	S1(hh)
Sm	S1(m)
Nhh_m	N(hh, m)
Sm_hh	S2(m, hh)

Prädikatenlogik

- berücksichtigt die interne Struktur von Aussagen und erlaubt es, zusätzlich bestimmte Beziehungen zwischen 'Objekten' zum Ausdruck zu bringen.

➔ Sprache mit größerer Ausdruckskraft.

- Konsequenzen (z.B. im Datenbankbereich):
  Systematische Beziehungen zwischen Aussagen und Fragen:
  - Welche Städte liegen südlich von Hamburg und nördlich von München?
  - Frankfurt liegt südlich von Hamburg und nördlich von München.

Die kleinsten Einheiten der Prädikatenlogik

… haben verschiedene syntaktische Kategorien

- *Terme* repräsentieren Objekte.
- *Prädikatensymbole* (Relationssymbole) repräsentieren Eigenschaften und Relationen.
- *Formeln* repräsentieren Aussagen.

PL-Repräsentation	Syntaktische Kategorie	Übersetzungsschlüssel	Syntaktische Kategorie
hh	Konstante, Term	Hamburg	Name
m	Konstante, Term	München	Name
S1	Prädikatensymbol	ist eine Stadt	Prädikat
N	Prädikatensymbol	liegt nördlich von	
S2	Prädikatensymbol	liegt südlich von	

PL-Repräsentation	Syntaktische Kategorie	Übersetzung	
S1(hh)	(atomare) Formel	Hamburg ist eine Stadt.	Satz
S1(m)	(atomare) Formel	München ist eine Stadt.	Satz
N(hh, m)	(atomare) Formel	Hamburg liegt nördlich von München.	Satz
S2(m, hh)	(atomare) Formel	München liegt südlich von Hamburg.	Satz

Terme: Eindeutige Objektbenennungen (Namen)

Konstanten: atomare Terme

PL-Repräsentation	syntaktische Kategorie	Übersetzung
hh	Konstante, (atomarer) Term	Hamburg
m	Konstante, (atomarer) Term	München
d	Konstante, (atomarer) Term	Deutschland

Funktionssymbole

hs	einstelliges Funktionssymbol	die Hauptstadt von
ewz	einstelliges Funktionssymbol	die Einwohnerzahl von
abst	zweistelliges Funktionssymbol	der Abstand zwischen

Komplexe Terme: Funktionssymbole, kombiniert mit der richtigen Anzahl von Termen

hs(d)	(komplexer) Term	die Hauptstadt von Deutschland
abst(hh, m)	(komplexer) Term	der Abstand zwischen Hamburg und München
ewz(hh)	(komplexer) Term	die Einwohnerzahl von Hamburg
ewz(hs(d))	(komplexer) Term	die Einwohnerzahl der Hauptstadt von Deutschland

Prädikatensymbole (Relationssymbole)

- kombiniert mit der richtigen Anzahl von Termen ergeben (atomare) Formeln.

PL-Repräsentation	syntaktische Kategorie	Übersetzung
S1	einstelliges Prädikatensymbol	ist eine Stadt
S1(hh)	(atomare) Formel	Hamburg ist eine Stadt.
N	zweistelliges Prädikatensymbol	liegt nördlich von
N(hh, m)	(atomare) Formel	Hamburg liegt nördlich von München.
Gr	zweistelliges Prädikatensymbol	ist größer als
Gr(hh, m)	(atomare) Formel	Hamburg ist größer als München.

- Komplexe Terme können in Formeln auftreten

Gr(ewz(hh), ewz(m))	Die Einwohnerzahl von Hamburg ist größer als die Einwohnerzahl von München.
Gr(ewz(hs(d)), ewz(hh))	Die Hauptstadt von Deutschland hat mehr Einw. als HH.
¬Gr(abst(hh, m), abst(m, hh))	Der Abstand zwischen HH und München ist nicht größer als der Abstand zwischen München und HH.
Gr(abst(hh, m), abst(m, hs(d)))	Der Abstand zwischen Hamburg und München ist größer als der Abstand zwischen München und der Hauptstadt von Deutschland.

Prädikatensymbole – Relationale Datenbanksysteme

PL-Formel:

Flugverb(ab8862, ham, zrh, 240507, 06:30, 07:50)

Übersetzung:

Flug 'AB8862' fliegt am 24.05.07 von Hamburg nach Zürich; Abflug ist um 6:30, Ankunft um 7:50.

Anmerkungen:

- Flugverb ist ein 6-stelliges Prädikaten- / Relationensymbol
- ab8862, ham, zrh, 240507, 06:30, 07:50 sind Konstanten

Relationale Datenbank

- Variablenfreie PL-Formeln zu einem Prädikatensymbol werden in Tabellen zusammengeführt, z.B

Flugverb	Flight-No.	Date	Origin	Time Depart.	Destination	Time Arriv.
	ab8862	240507	ham	06:30	zrh	07:50
	ab8780	240507	ham	15:15	zrh	16:35
	ab8862	250507	ham	06:30	zrh	07:50

Quantoren und Variablen

- erlauben Aussagen über Objekte ohne eindeutige Bezugnahme

$\exists x\ (S1(x) \wedge N(x, m))$	Es gibt eine Stadt, die nördlich von München liegt.
$\neg\exists x\ (S1(x) \wedge S2(x, x))$	Keine Stadt liegt südlich von sich selbst.
$\exists x\ (S1(x) \wedge Gr(ewz(x), ewz(hh)))$	Eine Stadt hat mehr Einwohner als Hamburg.
$\neg\exists x\ (S1(x) \wedge Gr(ewz(x), ewz(hs(d))))$	Keine Stadt hat mehr Einwohner als die Hauptstadt von Deutschland.
$\forall x\ \forall y\ ((S1(x) \wedge (S1(y) \wedge N(y, x))) \Rightarrow S2(x, y))$	Jede Stadt liegt südlich von jeder Stadt, die nördlich von ihr liegt.
$\forall y\ \forall x\ ((S1(y) \wedge (S1(x) \wedge N(y, x))) \Rightarrow S2(x, y))$	Wenn eine Stadt nördlich von einer zweiten Stadt liegt, dann liegt die zweite Stadt südlich von der ersten.
$(S1(hh) \wedge N(hh, m)) \Rightarrow \exists x\ (S1(x) \wedge N(x, m))$	Wenn Hamburg eine Stadt ist und nördlich von München liegt, dann gibt es eine Stadt, die nördlich von München liegt.

Prädikatensymbole – Datenbanksysteme (Fortsetzung)

Variablen

- spielen eine zentrale Rolle in der Formulierung von Anfragen
  - Flugverb(x, ham, zrh, 240507, y, z)
    kann als Aufforderung verstanden werden, ein Modell der Formel zu finden, unter geeigneter Belegung von x, y, und z.
  - Diese Sichtweise wird verwendet
    - in der Logischen Programmierung (Prolog) → Modul SE-3
    - bei Datenbankanfragen (SQL: Structured Query Language) → Modul GDB

Quantoren

- werden in Standard-Datenbankensystemen in der Regel nicht explizit verwendet
  - sie sind implizit in die Verfahren der Antwortfindung integriert
- spielen in Deduktiven Datenbanken & Wissensbasierten Systemen eine Rolle, z.B. um Regularitäten der Domäne zu modellieren:
  Der Übergang von Flug_1 zu Flug_2 klappt im Flughafen X, wenn die Abflugzeit mehr als 90 Minuten nach der Ankunftszeit liegt. (Formulierung in PL zur Übung.)

Syntax der Prädikatenlogik: Das Alphabet

Definition 9.1

Das Alphabet der Prädikatenlogik besteht aus

- einer abzählbaren Menge $\mathcal{V}_{PL}$, dem Inventar verfügbarer Symbole ($\mathcal{V} \approx$ Vokabular).

  Diese unterteilt sich in

  - einer Menge von Variablen: x, y, x_i mit $i = 1, 2, \ldots$
  - einer Menge von Konstanten: $a, b, c, a_i, \ldots$
  - einer Menge von Funktionssymbolen: $f, g, h, f_i, \ldots$
  - einer Menge von Prädikatensymbolen (Relationssymbolen): $P, Q, R, P_i, \ldots$
  - einer Menge von Aussagensymbolen: $A, A_i, \ldots$

- den logischen Symbolen
  - Junktoren: $\neg, \wedge, \vee, \Rightarrow, \Leftrightarrow$
  - Quantoren: $\forall, \exists$
- den Hilfssymbolen Klammern und Komma:), (, ,
- Jedem Funktions- und jedem Prädikatensymbol ist eindeutig eine Stelligkeit $k > 0$ zugewiesen.

Zum Selbststudium

- Konstanten entsprechen Funktionssymbolen der Stelligkeit 0.
- Aussagensymbole entsprechen Prädikatensymbolen der Stelligkeit 0. Es ist nicht nötig, Aussagensymbole in der Prädikatenlogik zu verwenden.
- Wird dasselbe Symbol / Wort mit unterschiedlicher syntaktischen Kategorie (Stelligkeit) verwendet, dann verhält sich die Logik blind gegenüber der oberflächlichen Überein-stimmung und behandelt die unterschiedlichen Verwendungen unabhängig voneinander.

Bezeichnungskonventionen

- Als verfügbare Symbole werden oft auch Zeichenketten (Wörter) verwendet, insbesondere, wenn eine Assoziation mit einer bestimmten Bedeutung beabsichtigt ist.
- Wir verwenden bestimmte Buchstaben als (Meta-)Variablen/Platzhalter, deren Wert folgendes sein kann

PL-Variablen	u, v, w, x, y, z	Kleinbuchstaben (Ende des Alphabets)
Konstanten	a, b, c	Kleinbuchstaben (Anfang des Alphabets)
Funktionssymbole	f, g, h	Kleinbuchstaben (Bereich f – h)
Prädikatensymbole	P, Q, R	Großbuchstaben (Bereich P – S)
Formeln	F, G, H	Großbuchstaben (Bereich F – H)
Terme	t, r, s	Kleinbuchstaben (Bereich r – t)

Syntax der Prädikatenlogik: Terme von $\mathcal{L}_{PL}$

Definition 9.2 (Terme der Prädikatenlogik)

Gegeben sei ein Inventar verfügbarer Symbole $\mathcal{V}_{PL}$.

Die Menge der prädikatenlogischen Terme ist wie folgt definiert:

T-1 Jede Variable ist ein Term.

T-2 Jede Konstante ist ein Term.

T-3 Falls f ein Funktionssymbol der Stelligkeit k ist, und falls $t_1, \ldots, t_k$ Terme sind, so ist $f(t_1, \ldots, t_k)$ ein Term.

T-4 Es gibt keine anderen Terme, als die, die durch endliche Anwendung der Schritte T-1 bis T-3 erzeugt werden.

- Der Term t ist Teilterm des Terms t', wenn t beim Aufbau von t' (Schritt T-3) verwendet wurde.

Beispiel: $ewz(hs(d))$ ≈ „Einwohnerzahl der Hauptstadt von Deutschland“

Bemerkung zu Def. 9.2

- Entsprechend der Definition lassen sich
  - induktive Beweise über die Struktur der Terme führen
  - Funktionen rekursiv über die Struktur der Terme definieren.

Zum Selbststudium: Prinzip der strukturellen Induktion für Terme der Prädikatenlogik

Um zu beweisen, dass eine Behauptung $\mathcal{B}(t)$ für jeden Term t der Prädikatenlogik gilt, genügt es, folgende Schritte durchzuführen:

Induktionsanfang: Man zeigt, dass $\mathcal{B}(a)$ für jede Konstante a gilt, und dass $\mathcal{B}(x)$ für jede Variable x gilt.

Induktionsannahme: Man nimmt an, dass f ein beliebiges Funktionssymbol ist, benennt die Stelligkeit von f als k und nimmt an, dass $t_1, \ldots, t_k$ Terme sind, die alle die Behauptung erfüllen, also, dass $\mathcal{B}(t_1), \ldots, \mathcal{B}(t_k)$ gelten.

Induktionsschritt: Man zeigt, dass dann auch $\mathcal{B}(f(t_1, \ldots, t_k))$ gilt.

Syntax der Prädikatenlogik: Formeln von $\mathcal{L}_{PL}$

Definition 9.3 (Formeln der Prädikatenlogik)
Gegeben sei ein Inventar verfügbarer Symbole $\mathcal{V}_{PL}$.
Die Menge der prädikatenlogischen Formeln ($\mathcal{L}_{PL}$) ist wie folgt definiert:

F-1 Jedes Aussagensymbol ist eine (atomare) Formel.

F-2 Falls P ein Prädikatensymbol der Stelligkeit k ist, und falls $t_1, \ldots, t_k$ prädikatenlogische Terme sind, so ist $P(t_1, \ldots, t_k)$ eine (atomare) Formel.

F-3 Falls F und G Formeln sind, so sind auch folgende Zeichenketten (komplexe) Formeln: $\neg F$, $(F \wedge G)$, $(F \vee G)$, $(F \Rightarrow G)$, $(F \Leftrightarrow G)$

F-4 Ist x eine Variable und F eine Formel, so sind $\exists x\, F$ und $\forall x\, F$ (komplexe) Formeln.

F-5 Es gibt keine anderen Formeln, als die, die durch endliche Anwendung der Schritte F-1 bis F-4 erzeugt werden.

- Die Formel F ist Teilformel der Formel H, wenn F beim Aufbau von H (Schritt F-3, F-4) verwendet wurde.
- Der Term t ist Teilterm der Formel H, wenn t beim Aufbau von H (Schritt F-2) verwendet wurde oder Teilterm eines Terms ist, für den dies gilt.
- Bezüglich der Formeln $\exists x\, F$ und $\forall x\, F$ bezeichnen wir x als die Quantorenvariable.

Zum Selbststudium: Prinzip der strukturellen Induktion für prädikatenlogische Formeln

Bemerkung zu Def. 9.3

- Entsprechend der Definition lassen sich
  - induktive Beweise über die Struktur der prädikatenlogischen Formeln führen
  - Funktionen rekursiv über die Struktur der prädikatenlogischen Formeln definieren.

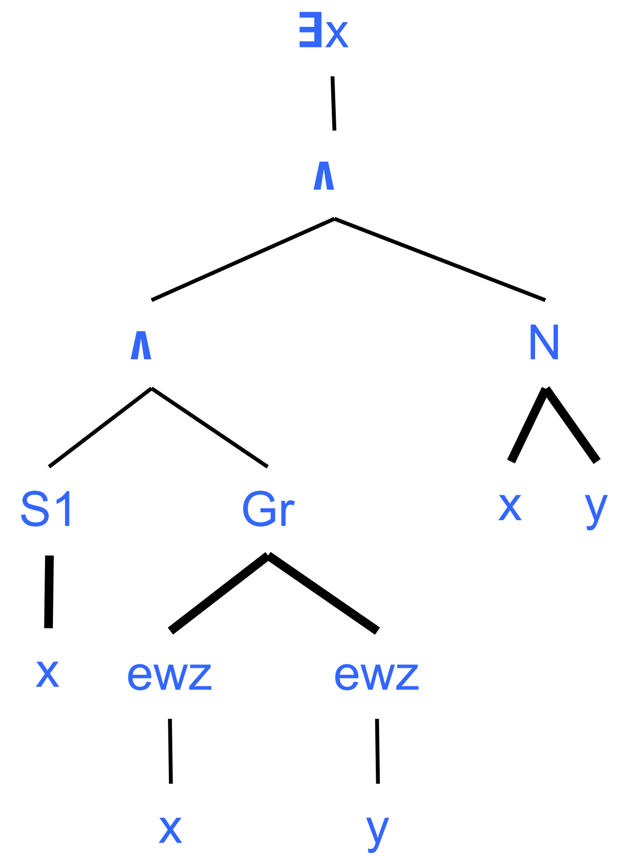
Um zu beweisen, dass eine Behauptung $\mathcal{B}(F)$ für jede Formel $F \in \mathcal{L}_{PL}$ gilt, genügt es, folgende Schritte durchzuführen:

Induktionsanfang: Man zeigt, dass $\mathcal{B}(F)$ für jede atomare Formel F gilt, also für die Aussagensymbole und für alle Formeln der Form $P(t_1, \ldots, t_k)$, wobei P ein k-stelliges Prädikatensymbol ist und $t_1, \ldots, t_k$ Terme sind.

Induktionsannahme: Man nimmt an, dass F und G Formeln sind, für die $\mathcal{B}(F)$ und $\mathcal{B}(G)$ gelten, und dass x eine Variable ist.

Induktionsschritt: Man zeigt, dass dann auch $\mathcal{B}(\neg F)$, $\mathcal{B}((F \wedge G))$, $\mathcal{B}((F \vee G))$, $\mathcal{B}((F \Rightarrow G))$, $\mathcal{B}((F \Leftrightarrow G))$, $\mathcal{B}(\exists x\ F)$ und $\mathcal{B}(\forall x\ F)$ gelten.

Prädikatenlogische Strukturbäume

	$\exists x\ ((S1(x) \land\ Gr(ewz(x), ewz(hh)))\land\ N(x, y))$
Quantoren bilden innere Knoten mit einem Nachfolger. Diese Knoten sind auch mit der zugehörigen Quantorenvariable gekennzeichnet. Prädikatensymbole und Funktionssymbole bilden innere Knoten. Die Zahl der Nachfolger dieser Knoten ist die Stelligkeit des Symbols. Konstanten, Variablen und Aussagensymbole bilden Blätter. Runde Knoten: logische Symbole Eckige Knoten: verfügbare Symbole	

Zum Selbststudium

Darüber hinaus:

- Häufig ist es nützlich, zweistellige Funktionssymbole oder Prädikatensymbole zwischen die zugehörigen Terme zu schreiben. In der Mathematik ist dies z.B. die Standardkonvention für Operationssymbole für Addition, Multiplikation etc. und Relationen wie Gleichheit, größer, kleiner etc. Um diese Notation, die sog. Infixschreibweise, zuzulassen, muss die Syntaxdefinition der Logik-Sprache geeignet geändert werden.

Prädikate und Argument-Reihenfolge

Hamburg	ist eine Stadt					
hh	S1					
	hh					S1(hh)

Hamburg	liegt nördlich von	München	
hh	N	m	
	hh, m		N(hh, m)

Hamburg	ist eine Stadt		nördlich von	München	
(Hamburg	ist eine Stadt	und	liegt nördlich von	München)	
hh	S1	∧	N	m	
	hh		hh, m		S1(hh) ∧ N(hh, m)

Hamburg	ist eine Stadt,	die	nördlich von	München	liegt
hh	S1		N	m	
	hh		hh, m		S1(hh) ∧ N(hh, m)

Übersetzungen Deutsch → PL: Existenzquantor (1)

$\exists x\ S1(x)$

Es gibt	eine Stadt
(Es gibt etwas,	das ist eine Stadt)
$\exists$	$S1$
x	x

$\exists x\ (S1(x) \wedge G(x))$

Es gibt	eine große		Stadt
(Es gibt etwas,	das ist groß	und	eine Stadt)
$\exists$	G	$\wedge$	$S1$
x	x		x

$\exists x\ (S1(x) \wedge N(x, m))$

Eine	Stadt		liegt nördlich von	München
(Es gibt etwas,	das ist eine Stadt	und	liegt nördlich von	München.)
$\exists$	$S1$	$\wedge$	N	m
x	x		x, m	

Übersetzungen Deutsch → PL: Existenzquantor (2)

$\neg\exists x\ E1(x)$

Es gibt kein	Einhorn
(Es gibt nichts, das ein Einhorn ist)	
$\neg\exists$	$E1$
x	x

$\neg\exists x\ (S1(x) \wedge N(x, m))$

Keine	Stadt		liegt nördlich von	München	
(Es gibt nichts,	das eine Stadt ist	und	nördlich von	München	liegt.)
$\neg\exists$	$S1$	$\wedge$	N	m	
x	x		x, m		

$\neg\exists x\ (S1(x) \wedge N(x, x))$

Keine	Stadt		liegt nördlich von	sich selbst	
(Es gibt nichts,	das eine Stadt ist	und	nördlich von	sich selbst	liegt.)
$\neg\exists$	$S1$	$\wedge$	N		
x	x		x, x		

Übersetzungen Deutsch → PL: Allquantor

$\forall x\ T(x)$

Alles	ist toll
(Für jedes gilt:	es ist toll)
$\forall$	T
x	x

$\forall x\ (S1(x) \Rightarrow G(x))$

Jede	Stadt		ist groß
(wenn etwas	eine Stadt ist,	dann	ist es groß)
(Für jedes gilt:	wenn es eine Stadt ist,	dann	ist es groß)
$\forall$	S1	$\Rightarrow$	G
x	x		x

$\forall x\ (S1(x) \Rightarrow N(x, m))$

Jede	Stadt		liegt nördlich von	München
(wenn etwas	eine Stadt ist,	dann	liegt es nördlich von	München)
(Für jedes gilt:	wenn es eine Stadt ist,	dann	liegt es nördlich von	München)
$\forall$	S1	$\Rightarrow$	N	m
x	x		x, m	

Übersetzungen Deutsch → PL

Beispiel: $\forall x\ (M1(x) \Rightarrow \exists y\ M2(y,x))$

Jeder	Mensch		hat eine	Mutter
(Wenn etwas	ein Mensch ist,	dann	hat er eine	Mutter)
(Für jedes gilt:	wenn es ein Mensch ist,	dann	hat es eine	Mutter)
$\forall$	M1	$\Rightarrow$	$\exists$	M2
x	x		y	y, x

Aber:

Jeder / jede hat eine Mutter bzw. Alle haben eine Mutter

werden auch übersetzt in: $\forall x\ (M1(x) \Rightarrow \exists y\ M2(y,x))$

denn:

- das Deutsche stellt *sortale* Bedingungen daran, welche Arten von Entitäten in der Beziehung ‚_ ist Mutter von _' stehen können, damit der entsprechende Satz *sinnvoll* ist.
- die Standard-Prädikatenlogik stellt keine entsprechenden Bedingungen an die Variablen

➔ Sortale Bedingungen müssen in der Standard-Prädikatenlogik explizit durch einstellige Prädikatensymbole formuliert werden.

z.B.: Jeder / jede → M1(x)

➔ Sortenlogik ≈ Prädikatenlogik mit sortierten Variablen
[wird in FGI-3 (Masterstudium) behandelt]

Zum Selbststudium: Sortenlogik

Sorten

- ermöglichen es, Kategorien der Domäne zu berücksichtigen und hierbei auch unterschiedliche Rollen, die Entitäten der Domäne spielen, einzubeziehen, z.B.
  in Flugverb(ab8862, ham, zrh, 240507, 06:30, 07:50)
  - treten vier Sorten auf: 'flightnumber', 'airport', 'date', 'time'
  - die Argumente der Sorte
    - 'airport' tritt in zwei Rollen auf: 'origin' und 'destination'
    - 'time' tritt in zwei Rollen auf: 'departure' und 'arrival'
- Entsprechend differenzierte Analysen sind Grundlage für
  - die Strukturierung einer Domäne in Relationalen Datenbanken → Modul GDB
  - die Darstellung von Regularitäten in Wissensbasierten Systemen (& Deduktiven Datenbanken) → Modul GWV
- ermöglichen es, gewisse Schritte des automatischen Beweisens effizienter zu realisieren. [wird – nicht regelmässig – in FGI-3 (Masterstudium) behandelt]

Funktions- und Prädikatensymbole

Welche Übersetzung

‚ist (die) Hauptstadt von'	→	Hs	zweistelliges Prädikatensymbol	?
‚Hauptstadt von'	→	hs	einstelliges Funktionssymbol	

➔ Bedingung:
Mit einem Funktionssymbol gebildete Terme benennen eindeutig ein Objekt.

Beispiele

	Darstellung mit einstelligem Funktionssymbol	Darstellung mit zweistelligem Prädikatensymbol
Die Hauptstadt von Frankreich liegt an der Seine.	Liegt_an(hs(fr), s)	∃x (Liegt_an(x, s) ∧ Hs(x, fr))
Paris ist die Hauptstadt von Frankreich.	p = hs(fr)	Hs(p, fr)

➔ Statt einstelliger Funktionssymbole können auch zweistellige Prädikatensymbole verwendet werden, aber: dies führt zu anderen prädikatenlogischen Darstellungen.

➔ Bei Verwendung von Funktionssymbolen ist ein zusätzliches logisches Symbol = (Identität) nützlich. ➔ Prädikatenlogik mit Identität (nicht Gegenstand dieser Vorlesung, wird in FGI-3 behandelt.)

Funktions- und Prädikatensymbole (2)

	Funktionale Schreibweise	Relationale Schreibweise
Paris ist die Haupt-stadt von Frankreich.	$p = hs(fr)$	$Hs(p, fr)$
	Fordert für jede Konstante a die Eindeutigkeit des Funktionswertes $hs(a)$.	Ermöglicht die Existenz mehrerer Objekte b_1 und b_2, die zu einer Konstante a in der "Hauptstadt-Relation" Hs stehen können: $Hs(b_1, a)$ und x.

Jede n-stellige Funktion kann auch als (n+1)-stellige Relation dargestellt werden.

- Dabei geht die explizite Funktionale Abhängigkeit verloren.
- In Relationalen Datenbanken spielt "Funktionale Abhängigkeit" eine wichtige Rolle.
  → Kap. 7. Logischer DB-Entwurf im Modul GDB

Freie Variablen einer Formel

Definition 9.4 (freie Variablen von Termen und Formeln)

Wir definieren eine Abbildung FV von Termen und Formeln auf Mengen von Variablen rekursiv wie folgt:

$FV(a) = \{\ \}$	für jede Konstante a
$FV(x) = \{\ x\ \}$	für jede Variable x
$FV(f(t_1, \ldots, t_k)) = FV(t_1) \cup \ldots \cup FV(t_k)$	für jedes k-stellige Funktionssymbol f und alle Terme $t_1, \ldots, t_k$
$FV(A) = \{\ \}$	für jedes Aussagensymbol A
$FV(P(t_1, \ldots, t_k)) = FV(t_1) \cup \ldots \cup FV(t_k)$	für jedes k-stellige Prädikatensymbol P und alle Terme $t_1, \ldots, t_k$
$FV(\neg F) = FV(F)$	für jede Formel F
$FV((F \wedge G)) = FV((F \vee G)) =$ $FV((F \Rightarrow G)) = FV((F \Leftrightarrow G)) =$ $FV(F) \cup FV(G)$	für beliebige Formeln F und G
$FV(\exists x\ F) = FV(\forall x\ F) = FV(F) \setminus \{\ x\ \}$	für jede Formel F und jede Variable x

Für jede Formel F nennen wir die Elemente von $FV(F)$ die *in F frei vorkommenden Variablen*.

Gebundenes und freies Vorkommen von Variablen

Definition 9.5 (Skopus / Gebundene / freie Variable)

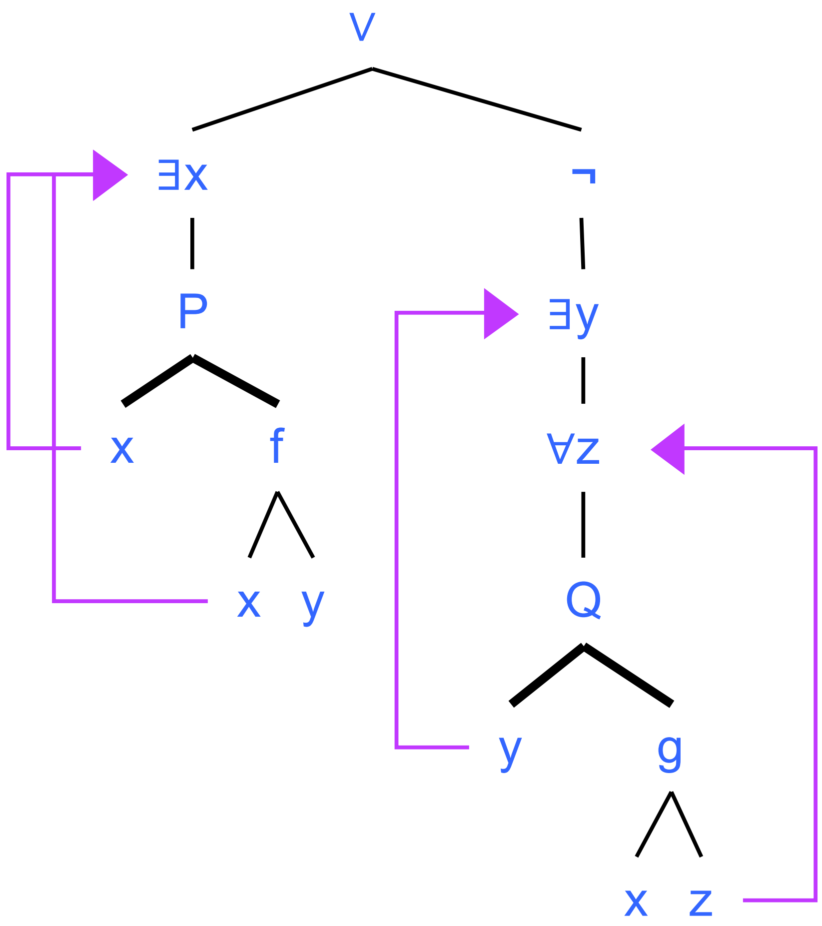
- Zur Erinnerung: Wenn F eine Formel und x eine Variable ist, dann sind auch ∃x F und ∀x F Formeln (s. Def. 9.3.4).
  Bezüglich der Formeln ∃x F und ∀x F bezeichnen wir F auch als den Skopus des Quantors und x als die Quantorenvariable.
- Eine Variable x, die *im Skopus* eines Quantors (∃x bzw. ∀x) mit der Quantorenvariable x frei vorkommt, ist durch den Quantor in dieser Position *gebunden*.
- Eine Variable x, die als Teilterm in einer Formel F enthalten aber in dieser Position durch keinen Quantor gebunden ist, ist in dieser Position frei in F.
- Eine Formel ohne freies Vorkommen von Variablen heißt *geschlossen*.
  Geschlossene Formeln werden auch als prädikatenlogische Aussagen bezeichnet.

Beispiel: Variablenbindung durch Quantoren

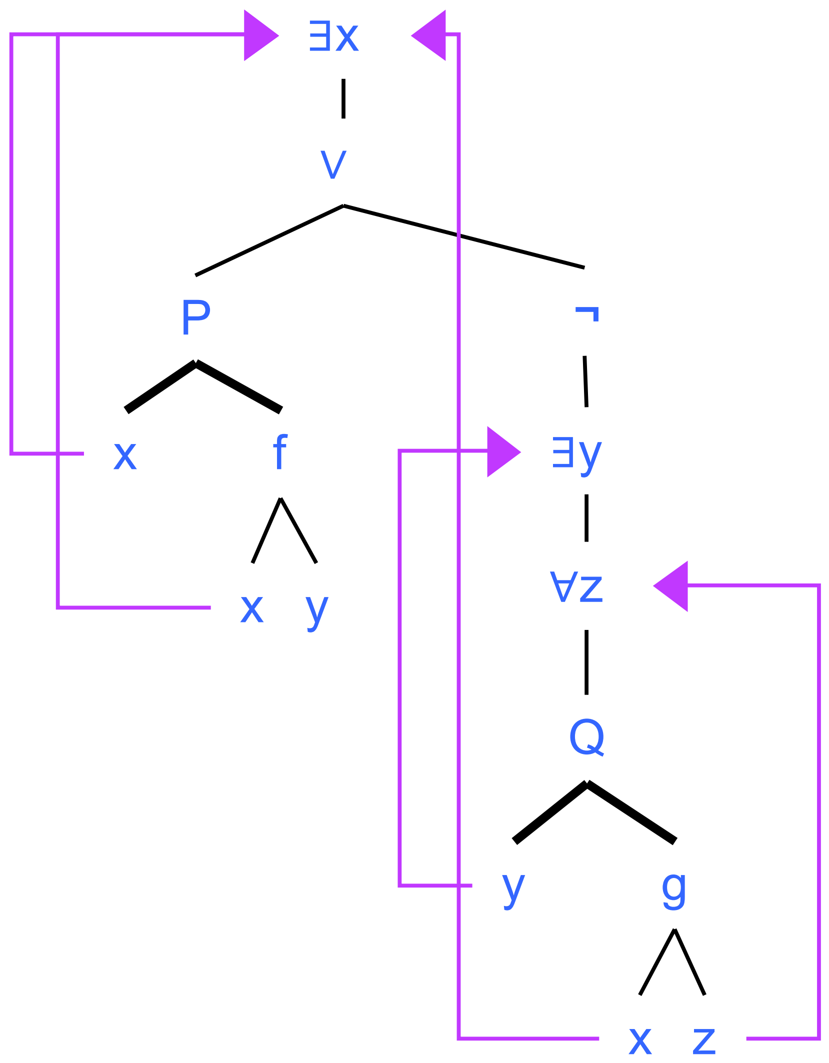
$$\overbrace{\forall x}^{\text{Quantor}} \quad \overbrace{(P(x) \lor R(x, y))}^{\text{Skopus}}$$

gebunden frei

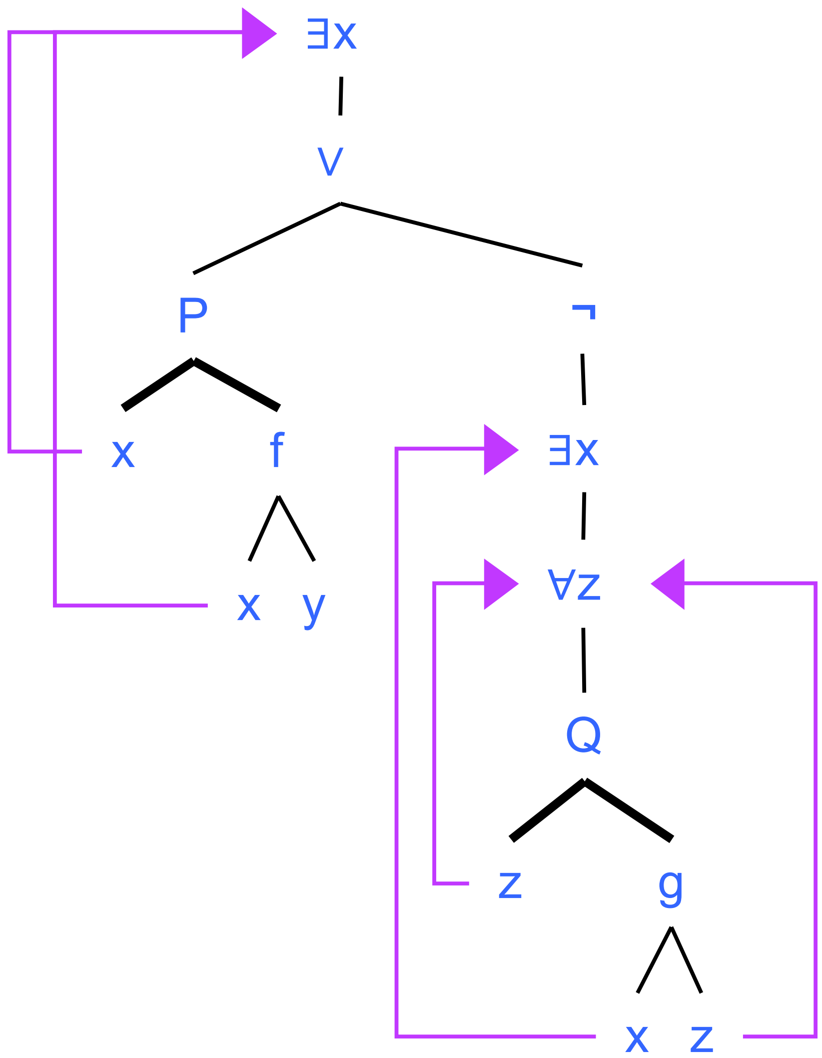
Beispiel: $\exists x\ P(x, f(x, y)) \vee \neg\exists y\ \forall z\ Q(y, g(x, z))$



Beispiel: $\exists x\ (P(x, f(x, y)) \vee \neg\exists y\ \forall z\ Q(y, g(x, z)))$



Beispiel: $\exists x\,(P(x, f(x, y)) \vee \neg\exists x\,\forall z\,Q(z, g(x, z)))$



Auswertung von Formeln in der Aussagenlogik ($\mathcal{L}_{\mathbf{AL}}$)

- Formeln werden auf Wahrheitswerte abgebildet.
- Aussagensymbole sind die frei interpretierbaren Bestandteile (verfügbare Symbole).
- Aussagensymbolen werden durch Belegungen ausgewertet.
- Die Auswertung komplexer Formeln bestimmt sich aus einer Belegung und festen Regeln für die Auswertung der Junktoren.

Auswertung von Ausdrücken in der Prädikatenlogik ($\mathcal{L}_{\mathbf{PL}}$)

- Terme werden auf (beliebige) Objekte einer Grundmenge abgebildet.
- Zusätzliche frei interpretierbare Bestandteile (verfügbare Symbole):
  (freie) Variablen, Konstanten, Funktionssymbole und Prädikatensymbole
- Sie werden durch Interpretationen bezüglich einer Grundmenge (Domäne, Universum) ausgewertet.
- Zusätzliche Regeln für die Auswertung von komplexen Termen, atomaren Formeln und für komplexe Formeln mit Quantoren
- Quantoren beeinflussen die Auswertung der von ihnen gebundenen Variablen.

Semantik: Aussagenlogik – Prädikatenlogik (Forts.)

- Die verfügbaren Symbole der prädikatenlogischen Ausdrücke werden mit Hilfe der Grundmenge (Domäne, Universum) interpretiert:
  - freie Variablen und Konstanten durch Objekte der Grundmenge
  - Funktionssymbole durch Funktionen über der Grundmenge
  - Prädikatensymbole durch Mengen über der Grundmenge
  - Aussagensymbole durch Wahrheitswerte.
  - Komplexe Ausdrücke (Terme, Formeln) werden hierauf aufbauend ausgewertet.

➔ Auch in der Semantik der Prädikatenlogik spielen Wahrheitswerte eine zentrale Rolle.

➔ Wahrheitswerte atomarer Formeln ergeben sich durch die Interpretation über der Grundmenge (Domäne).

Semantik der Prädikatenlogik: Strukturen

Definition 9.6 (Struktur)
Eine *Struktur* ist ein Paar $\mathcal{A} = (\mathsf{U}, \mathsf{I})$, wobei

- U *(Universum, Grundmenge, Domäne)* eine beliebige, nicht leere Menge ist und
- I (*Interpretation*, *Auswertung*, *valuation*) eine Abbildung ist.
  - Der Definitionsbereich von I ist $\mathcal{V}_{\text{PL}}$ (Vokabular der Prädikatenlogik).

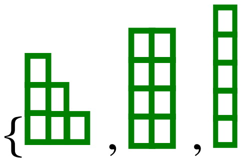
    I bildet
    - Variablen und Konstanten auf Elemente der Grundmenge U,
    - k-stellige Funktionssymbolen auf k-stellige Funktionen über U,
    - k-stellige Prädikatensymbole auf Mengen von k-Tupeln über U und
    - Aussagensymbole auf Wahrheitswerte ab.

- Wenn mehrere Strukturen $\mathcal{A}$ und $\mathcal{B}$ bzw. $\mathcal{A}_1$ und $\mathcal{A}_2$, gleichzeitig betrachtet werden sollen, schreiben wir auch $\mathcal{A} = (\mathsf{U}_{\mathcal{A}}, \mathsf{I}_{\mathcal{A}})$ und $\mathcal{B} = (\mathsf{U}_{\mathcal{B}}, \mathsf{I}_{\mathcal{B}})$; bzw. $\mathcal{A}_1 = (\mathsf{U}_1, \mathsf{I}_1)$ und $\mathcal{A}_2 = (\mathsf{U}_2, \mathsf{I}_2)$, um die Universen und Interpretationen auseinanderzuhalten.

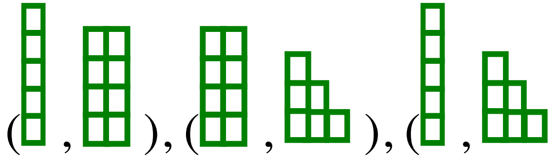
Beispiel (1): Interpretationen für $\mathcal{L}_{PL}$

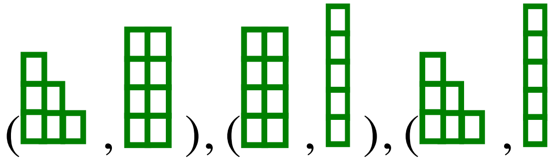
Konstanten: a, b, c   Variable: x

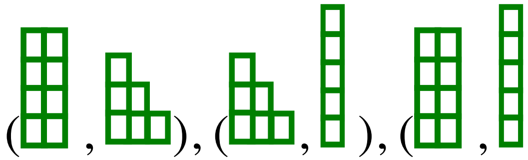
Prädikatensymbol: größer (2-stellig)

Die Domäne U = {}

Relationen über der Domäne U:

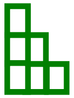
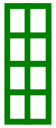
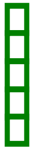
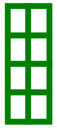
H = {  }

B = {  }

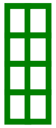
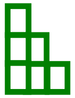
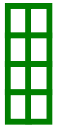
V = {  }

Beispiel (1): Interpretationen für $\mathcal{L}_{\mathbf{PL}}$ (Forts. – 1)

$\mathcal{A}_1 = (\mathsf{U}, \mathsf{I}_1)$

$\mathsf{I}_1(\mathsf{a}) =$  $\mathsf{I}_1(\mathsf{b}) =$  $\mathsf{I}_1(\mathsf{c}) =$  $\mathsf{I}_1(\mathsf{x}) =$  $\mathsf{I}_1(\text{größer}) = \mathsf{H}$

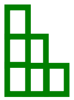
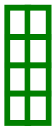
$\mathcal{A}_2 = (\mathsf{U}, \mathsf{I}_2)$

$\mathsf{I}_2(\mathsf{a}) =$  $\mathsf{I}_2(\mathsf{b}) =$  $\mathsf{I}_2(\mathsf{c}) =$  $\mathsf{I}_2(\mathsf{x}) =$  $\mathsf{I}_2(\text{größer}) = \mathsf{H}$

:

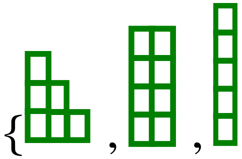
:

$\mathcal{A}_4 = (\mathsf{U}, \mathsf{I}_4)$

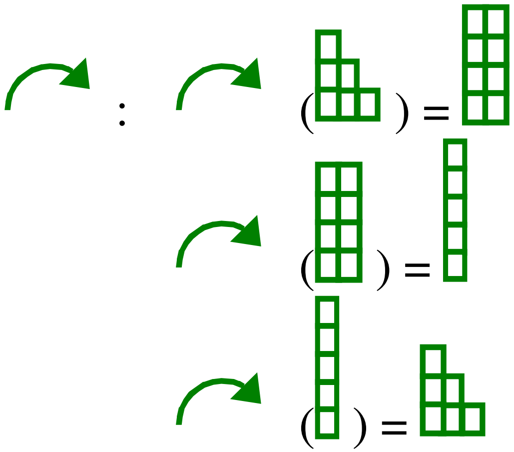
$\mathsf{I}_4(\mathsf{a}) =$  $\mathsf{I}_4(\mathsf{b}) =$  $\mathsf{I}_4(\mathsf{c}) =$  $\mathsf{I}_4(\mathsf{x}) =$  $\mathsf{I}_4(\text{größer}) = \mathsf{V}$

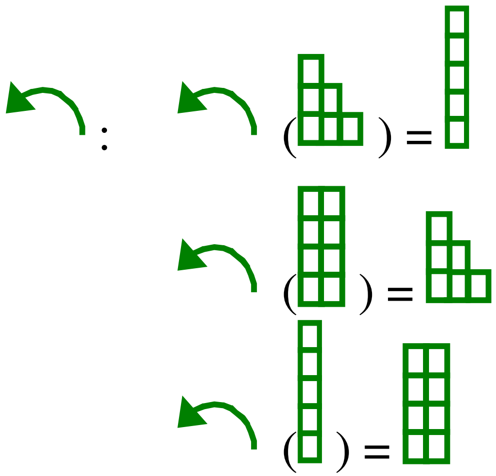
Beispiel (1): Interpretationen für $\mathcal{L}_{\mathbf{PL}}$ (Forts. – 2)

Konstanten: a, b, c   Variable: x
Prädikatensymbol: größer (2-stellig)
Funktionssymbol: nachfolger (1-stellig)

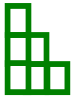
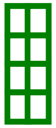
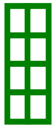
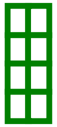
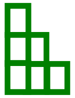
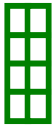
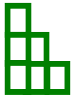
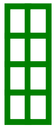
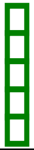
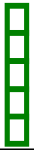
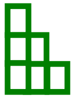
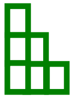
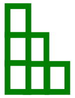
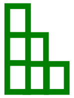
Die Domäne U = {  }

Funktionen über der Domäne U:





Tabellarische Darstellung verschiedener Interpretationen über einer Domäne

		a	b	c	x	größer	nachfolger	…
$\mathcal{A}_1$	I_1					H		
$\mathcal{A}_2$	I_2					H		
	:							
$\mathcal{A}_4$	I_4					V		
$\mathcal{A}_5$	I_5					H		

Entsprechend zu Wahrheitstafeln.
Aber: Es gibt unendlich viele Interpretationen, die Tafel kann also nicht vollständig werden.

Bei den Ausdrücken unseres Vokabulars stimmen $\mathcal{A}_1$ und I_1 immer genau überein.

Definition 9.7 (Auswertung komplexer Terme und atomarer Formeln)

Sei F eine Formel und $\mathcal{A} = (U, I)$ eine Struktur.

I wird rekursiv zu einer Auswertung fortgesetzt, für die wir dann einfach $\mathcal{A}$ schreiben.

- Rekursive Definition von $\mathcal{A}$ für Terme:

T*-1 Für jede Variable x ist: $\mathcal{A}(x) = I(x)$.

T*-2 Für jede Konstante a ist: $\mathcal{A}(a) = I(a)$.

T*-3 Für jede Liste von k Termen $t_1, \ldots, t_k$ und jedes k-stellige Funktionssymbol ist:

$$\mathcal{A}(f(t_1, \ldots, t_k)) = I(f)(\mathcal{A}(t_1), \ldots, \mathcal{A}(t_k))$$

- $I(f)$ ist eine k-stellige Funktion über U. Diese Funktion wird auf die Liste von Objekten angewendet, die bei der Auswertung der Terme entsteht.

- Rekursive Definition von $\mathcal{A}$ für atomare Formeln:

F*-1 Für jedes Aussagensymbol A ist: $\mathcal{A}(A) = I(A)$

F*-2 Für jede Liste von k Termen $t_1, \ldots, t_k$ und jedes k-stellige Prädikatensymbol P ist:

$$\mathcal{A}(P(t_1, \ldots, t_k)) = \begin{cases} \mathbf{1}, \text{ falls } (\mathcal{A}(t_1), \ldots, \mathcal{A}(t_k)) \in I(P) \\ \mathbf{0}, \text{ sonst} \end{cases}$$

Zum Selbststudium: Alternative Schreibweise

U. Schöning (vgl. Abschnitt 2.1) verwendet eine „Abkürzende Schreibweise“:

$P^{\mathcal{A}}$ statt $I(P)$, $f^{\mathcal{A}}$ statt $I(f)$, $x^{\mathcal{A}}$ statt $I(x)$

- Damit ergibt sich dann z.B.:

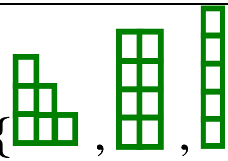
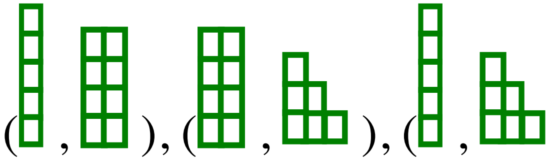
T*-3 Für jede Liste von k Termen $t_1, \ldots, t_k$ und jedes k-stellige Funktionssymbol ist:

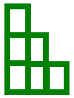
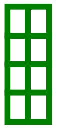
$$\mathcal{A}(f(t_1, \ldots, t_k)) = f^{\mathcal{A}}(\mathcal{A}(t_1), \ldots, \mathcal{A}(t_k))$$

F*-2 Für jede Liste von k Termen $t_1, \ldots, t_k$ und jedes k-stellige Prädikatensymbol P ist:

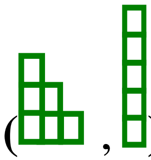
$$\mathcal{A}(P(t_1, \ldots, t_k)) = \begin{cases} \mathbf{1}, \text{ falls } (\mathcal{A}(t_1), \ldots, \mathcal{A}(t_k)) \in P^{\mathcal{A}} \\ \mathbf{0}, \text{ sonst} \end{cases}$$

Beispiel (2): Auswertung für $\mathcal{L}_{\mathbf{PL}}$

Das Vokabular	$\mathcal{A}_1 = (\mathsf{U}, \mathsf{I}_1)$
Konstanten: a, b, c Prädikatensymbol: größer (2-stellig) Funktionssymbol: nachfolger (1-stellig)	Die Domäne U = {  } H = { ()}

	a	b	c	größer	nachfolger	größer(a, c)	größer(b, a)
$\mathcal{A}_1$				H		??	

$\mathcal{A}_1$(größer(a, c)) = ?

$(\mathcal{A}_1(\mathsf{a}), \mathcal{A}_1(\mathsf{c}))$ = () $\notin$ H

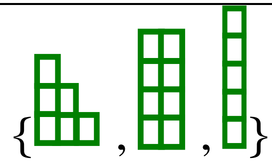
also: $\mathcal{A}_1$(größer(a, c)) = **0**

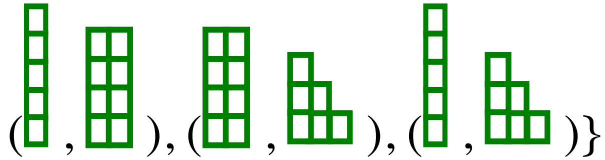
Beispiel (2): Auswertung für $\mathcal{L}_{\mathrm{PL}}$ (Forts. – 1)

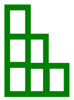
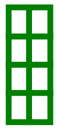
Das Vokabular

Konstanten: a, b, c
Prädikatensymbol: größer (2-stellig)
Funktionssymbol: nachfolger (1-stellig)

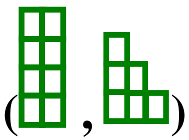
$\mathcal{A}_1 = (U, I_1)$

Die Domäne U = 

H = 

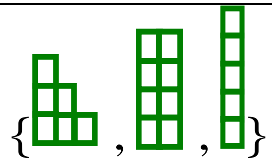
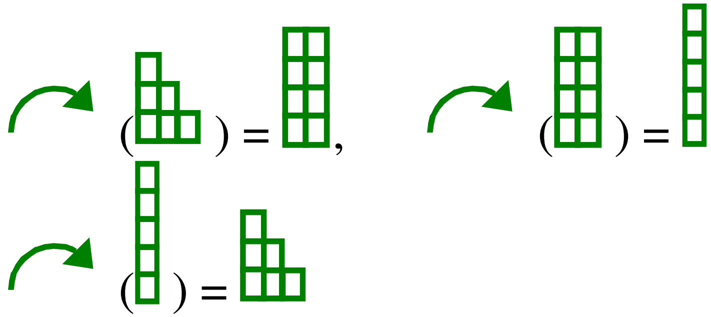
	a	b	c	größer	nachfolger	größer(a, c)	größer(b, a)
$\mathcal{A}_1$				H		**0**	??

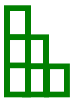
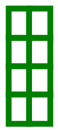
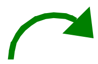
$\mathcal{A}_1$(größer(b, a)) = ?

$(\mathcal{A}_1(b), \mathcal{A}_1(a)) =$  $\in$ H

also: $\mathcal{A}_1$(größer(b, a)) = **1**

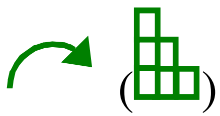
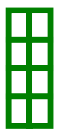
Beispiel (2): Auswertung für $\mathcal{L}_{PL}$ (Forts. – 2a)

Das Vokabular	
Konstanten: a, b, c Prädikatensymbol: größer (2-stellig) Funktionssymbol: nachfolger (1-stellig)	$\mathcal{A}_1 = (U, I_1)$ Die Domäne U =  

	a	b	c	größer	nachfolger	nachfolger(a)	nachfolger(c)
$\mathcal{A}_1$				H		??	??

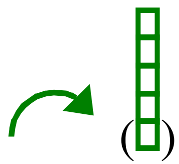
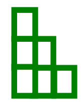
$\mathcal{A}_1$(nachfolger(a)) = ?

= $\mathcal{A}_1$(nachfolger)($\mathcal{A}_1$(a))

=  = 

$\mathcal{A}_1$(nachfolger(c)) = ?

= $\mathcal{A}_1$(nachfolger)($\mathcal{A}_1$(c))

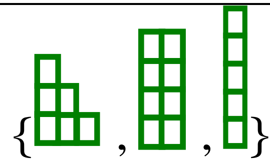
=  = 

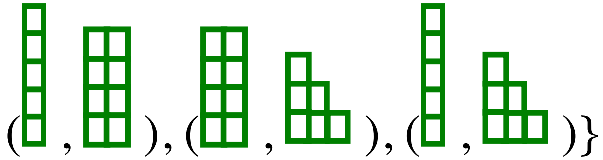
Beispiel (2): Auswertung für $\mathcal{L}_{PL}$ (Forts. – 2b)

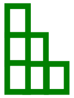
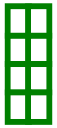
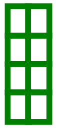
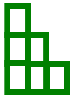
Das Vokabular

Konstanten: a, b, c
Prädikatensymbol: größer (2-stellig)
Funktionssymbol: nachfolger (1-stellig)

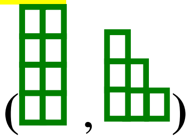
$\mathcal{A}_1 = (U, I_1)$

Die Domäne U = 

H = 

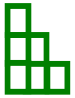
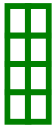
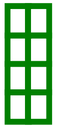
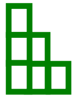
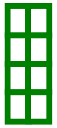
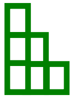
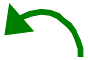
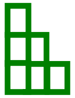
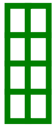
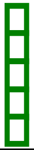
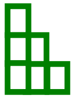
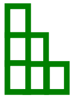
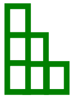
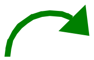
	a	b	c	größer	nachfolger	nachfolger(a)	nachfolger(c)	größer(nachfolger(a), nachfolger(c))
$\mathcal{A}_1$				H				??

$\mathcal{A}_1$(größer(nachfolger(a), nachfolger(c))) = ?

($\mathcal{A}_1$(nachfolger(a)), $\mathcal{A}_1$(nachfolger(c))) =  $\in$ H

also: $\mathcal{A}_1$(größer(nachfolger(a), nachfolger(c))) = **1**

Zum Selbstausfüllen

	a	b	c	größer	nachfolger	größer (a, c)	größer (b, a)	nach-folger (a)	nach-folger (c)	größer(nachfolger(a), nachfolger(c))
$\mathcal{A}_1$				H		**0**	**1**			**1**
$\mathcal{A}_2$				H						
$\mathcal{A}_4$				V						
$\mathcal{A}_5$				H						

Definition 9.8 (Auswertung komplexer Formeln ohne Quantoren)

Fortsetzung der rekursiven Definition der Auswertung von Formeln (genauso wie in der Aussagenlogik):

F*-3 Für alle Formeln F, G, und alle Strukturen $\mathcal{A}$, ist

$$\mathcal{A}(\neg F) = \begin{cases} \mathbf{1}, \text{ falls } \mathcal{A}(F) = \mathbf{0} \\ \mathbf{0}, \text{ sonst} \end{cases}$$

$$\mathcal{A}((F \wedge G)) = \begin{cases} \mathbf{1}, \text{ falls } \mathcal{A}(F) = \mathbf{1} \text{ und } \mathcal{A}(G) = \mathbf{1} \\ \mathbf{0}, \text{ sonst} \end{cases}$$

$$\mathcal{A}((F \vee G)) = \begin{cases} \mathbf{1}, \text{ falls } \mathcal{A}(F) = \mathbf{1} \text{ oder } \mathcal{A}(G) = \mathbf{1} \\ \mathbf{0}, \text{ sonst} \end{cases}$$

$$\mathcal{A}((F \Rightarrow G)) = \begin{cases} \mathbf{1}, \text{ falls } \mathcal{A}(F) = \mathbf{0} \text{ oder } \mathcal{A}(G) = \mathbf{1} \\ \mathbf{0}, \text{ sonst} \end{cases}$$

$$\mathcal{A}((F \Leftrightarrow G)) = \begin{cases} \mathbf{1}, \text{ falls } \mathcal{A}(F) = \mathcal{A}(G) \\ \mathbf{0}, \text{ sonst} \end{cases}$$

Definition 9.9 (x-Varianten von Strukturen)

Sei $\mathcal{A} = (U, I)$ eine Struktur, $d \in U$, und x eine Variable.

$\mathcal{A}_{[x/d]} = (U, I_{[x/d]})$ ist diejenige Struktur, die x als d interpretiert und ansonsten komplett mit $\mathcal{A}$ übereinstimmt. Es gilt also für alle verfügbaren Symbole τ:

$$\mathcal{A}_{[x/d]}(\tau) = I_{[x/d]}(\tau) = \begin{cases} d, \text{ falls } \tau = x \\ I(\tau), \text{ sonst} \end{cases}$$

- Weicht eine Struktur, wie die hier gebildete, höchstens bezüglich der Interpretation einer Variable x von der Struktur $\mathcal{A}$ ab, dann heißt sie x-Variante zu $\mathcal{A}$.

Beispiel (3): x-Varianten

Variable: x

Domäne U = { , , }

Bei endlichen Domänen lassen sich alle Varianten in einer Tabelle auflisten.

Variiert werden nur Variablen.

	a	b	c	x	größer	nachfolger
$\mathcal{A}_1$					H	
$\mathcal{A}_1[x/\;]$					H	
$\mathcal{A}_1[x/\;]$					H	
$\mathcal{A}_1[x/\;]$					H	

$\mathcal{A}_1 = \mathcal{A}_1[x/\;]$ Es ist aber unschädlich, sich in solchen Tabellen zu wiederholen.

Definition 9.10 (Auswertung komplexer Formeln mit Quantoren):

F*-4 Für jede Variable x, jede Formel F und jede Struktur $\mathcal{A}$ ist

$$\mathcal{A}(\forall x\ F) = \begin{cases} 1, & \text{falls für alle } d \in U \text{ gilt: } \mathcal{A}_{[x/d]}(F) = 1 \\ 0, & \text{sonst} \end{cases}$$

$$\mathcal{A}(\exists x\ F) = \begin{cases} 1, & \text{falls es ein } d \in U \text{ gibt mit: } \mathcal{A}_{[x/d]}(F) = 1 \\ 0, & \text{sonst} \end{cases}$$

➔ Auf den Skopus des Quantors werden alle Varianten der Auswertung angewendet.

➔ Bei der Auswertung einer quantifizierten Formel ist der ursprüngliche Wert der Quantorenvariable unerheblich.

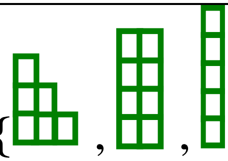
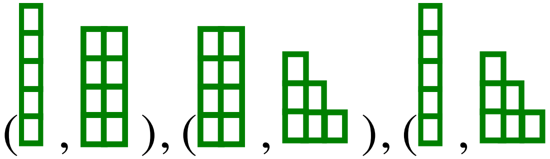
➔ Die Werteberechnung kann nicht nur ‚von unten nach oben' erfolgen.

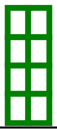
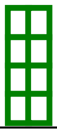
Dieses schließt die Auswertung der prädikatenlogischen Formeln ab.

Beobachtung zu Definitionen 9.7, 9.8 und 9.10

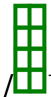
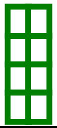
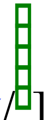
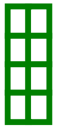
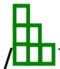
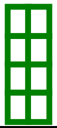
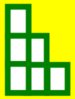
Stimmen zwei Strukturen bzgl. des Universums und bzgl. der Interpretationen aller verfügbaren Symbole und freien Variablen der Formel überein, dann liefern sie auch für die Formel denselben Wert.

Beispiel (4): Auswertung von Formeln mit Quantoren

Das Vokabular	
Konstanten: b Prädikatensymbol: größer (2-stellig) Variable: x	$\mathcal{A}_1 = (U, I_1)$ Die Domäne U = {} H = { ()}

	b	x	größer	größer(x, b)	∃x größer(x, b)
$\mathcal{A}_1$			H		??

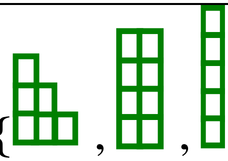
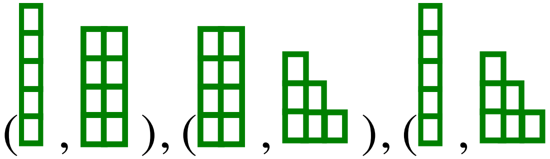
$\mathcal{A}_1$(∃x größer(x, b)) = ?

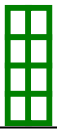
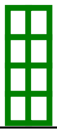
$\mathcal{A}_1$[x/]			H	**0**
$\mathcal{A}_1$[x/]			H	**1**
$\mathcal{A}_1$[x/]			H	**0**

$$\mathcal{A}(\exists x\ F) = \begin{cases} \mathbf{1}, \text{ falls es ein } d \in U \text{ gibt} \\ \quad \text{mit: } \mathcal{A}_{[x/d]}(F) = \mathbf{1} \\ \mathbf{0}, \text{ sonst} \end{cases}$$

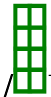
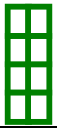
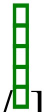
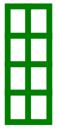
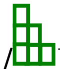
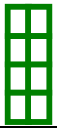
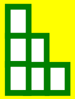
➔ $\mathcal{A}_1$(∃x größer(x, b)) = **1** denn es gibt ein d ∈ U mit: $\mathcal{A}_{1[x/d]}$(größer(x, b)) = **1**

Beispiel (4): Auswertung von Formeln mit Quantoren (Forts. – 1)

Das Vokabular Konstanten: b Prädikatensymbol: größer (2-stellig) Variable: x	$\mathcal{A}_1 = (U, I_1)$ Die Domäne U = {  } H = { ()}

	b	x	größer	größer(x, b)	∀x größer(x, b)
$\mathcal{A}_1$			H		??

$\mathcal{A}_1$(∀x größer(x, b)) = ?

$\mathcal{A}_{1[x/}$$_{]}$			H	**0**
$\mathcal{A}_{1[x/}$$_{]}$			H	**1**
$\mathcal{A}_{1[x/}$$_{]}$			H	**0**

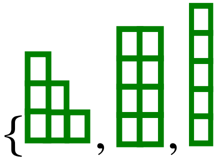
$$\mathcal{A}(\forall x\ F) = \begin{cases} \mathbf{1}, \text{ falls für alle } d \in U \\ \quad \text{gilt: } \mathcal{A}_{[x/d]}(F) = \mathbf{1} \\ \mathbf{0}, \text{ sonst} \end{cases}$$

➔ $\mathcal{A}_1$(∀x größer(x, b)) = **0** denn es gilt nicht für alle d ∈ U: $\mathcal{A}_{1[x/d]}$ (größer(x, b)) = **1**

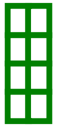
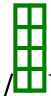
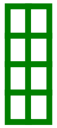
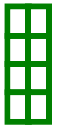
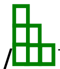
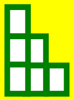
Beispiel (4): Tabellenform

Variable: x

$$\mathcal{A}(\forall x\ F) = \begin{cases} 1, \text{ falls für alle } d \in U \text{ gilt: } \mathcal{A}_{[x/d]}(F) = 1 \\ 0, \text{ sonst} \end{cases}$$

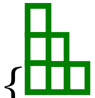
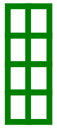
Domäne U = {}

$$\mathcal{A}(\exists x\ F) = \begin{cases} 1, \text{ falls es ein } d \in U \text{ gibt mit: } \mathcal{A}_{[x/d]}(F) = 1 \\ 0, \text{ sonst} \end{cases}$$

	b	c	x	größer	größer (x, b)	∃x größer (x, b)	∀x größer (x, b)	größer (x, c)	∃x größer (x, c)	¬größer(x, c)	∀x ¬größer(x, c)
$\mathcal{A}_1$				H		1	0		0		1
$\mathcal{A}_{1[x/$$]}$				H	0			0		1	
$\mathcal{A}_{1[x/$$]}$				H	1			0		1	
$\mathcal{A}_{1[x/$$]}$				H	0			0		1	

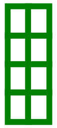
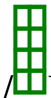
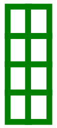
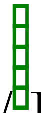
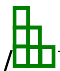
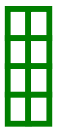
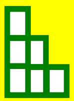
Beispiel (4): Tabellenform (Forts. – 1)

Variable: x

Domäne U = {, , }

Beachte: Für alle Strukturen $\mathcal{A}$, Variablen x, Formeln F und d, e $\in$ U gilt: $\mathcal{A}_{[x/d][x/e]} = \mathcal{A}_{[x/e]}$ (s. Def. 9.9, 9.10)

$\mathcal{A}_{[x/d]}(\forall x\ F) = \mathcal{A}(\forall x\ F)$ und $\mathcal{A}_{[x/d]}(\exists x\ F) = \mathcal{A}(\exists x\ F)$

	b	c	x	größer	größer(x, b)	∃x größer(x, b)	∀x größer(x, b)	größer(x, c)	∃x größer(x, c)	¬größer(x, c)	∀x ¬größer(x, c)
$\mathcal{A}_1$				H	**0**	**1**	**0**	**0**	**0**	**1**	**1**
$\mathcal{A}_1$[x/]				H	**0**	**1**	**0**	**0**	**0**	**1**	**1**
$\mathcal{A}_1$[x/]				H	**1**	**1**	**0**	**0**	**0**	**1**	**1**
$\mathcal{A}_1$[x/]				H	**0**	**1**	**0**	**0**	**0**	**1**	**1**

Interpretation: Variablen und Quantoren

Freie Variablen

- verhalten sich bei der Interpretation wie Konstanten.
- $\mathcal{A}_1$(größer(x, b)): In der Bestimmung des Wahrheitswertes wird x durch $\mathcal{A}_1$ ausgewertet.

Gebundene Variablen

- Der bindende Quantor schirmt die Variable gewissermaßen gegen die Außenwelt ab. Die zu verwendende Interpretation wird durch den Quantor variiert.
- $\mathcal{A}_1$(∃x größer(x, b)): die Interpretation der Variable x wird systematisch variiert und der Skopus größer(x, b) durch alle Varianten von $\mathcal{A}_1$ ausgewertet.
- Über den Quantor werden die Ergebnisse der Variation des Variablenwertes für den Skopus (größer(x, b)) zusammengefaßt.
- Quantoren schaffen einen neuen Namensraum für die Quantorenvariable.

Gemischte Vorkommen

- Tritt in einer Formel dieselbe Variable frei und gebunden auf, dann werden sie in verschiedenen Positionen unterschiedlich interpretiert.
- Nutzen zwei Quantoren dieselbe Variable, dann erfolgt die Variation unabhängig voneinander.

Modell, Gültigkeit, Unerfüllbarkeit

Definition 9.11 (ganz analog zur Aussagenlogik)

- Eine Struktur $\mathcal{A}$ heißt genau dann *Modell* für F, wenn gilt:
  $\mathcal{A}(\mathsf{F}) = \mathbf{1}$
- In diesen Fällen sagen und schreiben wir auch:
  - $\mathcal{A}$ macht F wahr
  - F gilt in $\mathcal{A}$
  - $\mathcal{A} \models \mathsf{F}$
- Eine Struktur $\mathcal{A}$ ist ein *Modell* für eine Formelmenge **M** GDW. $\mathcal{A}$ ein *Modell* für jede Formel $\mathsf{F} \in \mathbf{M}$ ist.

- Eine Formel F (eine Formelmenge **M**) heißt *erfüllbar*, falls ein Modell $\mathcal{A}$ für F (bzw. **M**) existiert, anderenfalls heißt F (bzw. **M**) *unerfüllbar*.
- Eine Formel F (eine Formelmenge **M**) heißt *falsifizierbar*, falls eine Struktur $\mathcal{A}$ kein Modell für F (bzw. **M**) ist.
- Eine Formel F (eine Formelmenge **M**) heißt (allgemein-) *gültig*, falls jede Struktur $\mathcal{A}$ ein Modell für F (bzw. **M**) ist.
  - F ist *logisch wahr*
  - $\models \mathsf{F}$
  - Die Definition der Gültigkeit beruht auf *allen Strukturen mit beliebigen Domänen und Interpretationen*.

Definition 9.12 (ganz analog zur Aussagenlogik)

- Zwei Formeln G und F sind *(logisch) äquivalent*, falls für jede Struktur $\mathcal{A}$ gilt: $\mathcal{A}(F) = \mathcal{A}(G)$.
- Eine Formel G *folgt* *(logisch)* aus einer Formel F (einer Formelmenge **M**) falls für jede Struktur $\mathcal{A}$ gilt:
  Wenn $\mathcal{A}$ ein Modell für F (bzw. **M**) ist, dann ist $\mathcal{A}$ auch ein Modell für G.
  - $F \models G$ bzw. $\mathbf{M} \models G$

Gültigkeit in der Prädikatenlogik (1)

Beispiele

- $\forall x\, P(x) \vee \neg\forall x\, P(x)$
  - ist das Resultat der Substitution mit $sub(A) = \forall x\, P(x)$, angewendet auf die aussagenlogische Tautologie $(A \vee \neg A)$.

Satz 9.13

Wenn in einer aussagenlogischen Tautologie prädikatenlogische Formeln (uniform) substituiert werden, ergibt dies eine gültige Formel der Prädikatenlogik.
ohne Beweis, aber vgl. Satz 6.2

- $\forall x\, (P(x) \vee \neg P(x))$
  - ist nicht das Resultat eine Substitution einer aussagenlogischen Tautologie.
  - ist aber eine gültige Formel der Prädikatenlogik. (ist noch zu zeigen!!!!)
- $\forall x\, P(x) \Rightarrow \exists y\, P(y)$
  - ist eine gültige Formel der Prädikatenlogik. (ist noch zu zeigen!!!!)

Gültigkeit von $\forall x\, (P(x) \vee \neg P(x))$

Beweis

Vorbemerkung: Nach Def.9.8 gilt für alle Strukturen $\mathcal{A}'$:

$$\mathcal{A}'(P(x) \vee \neg P(x)) = \begin{cases} \mathbf{1}, \text{ falls } \mathcal{A}'(P(x)) = \mathbf{1} \text{ oder } \mathcal{A}'(P(x)) = \mathbf{0} \\ \mathbf{0}, \text{ sonst} \end{cases}$$

Also: $\mathcal{A}'(P(x) \vee \neg P(x)) = \mathbf{1}$, (maW: $P(x) \vee \neg P(x)$ ist eine Tautologie).

Es sei $\mathcal{A} = (U, I)$ eine beliebige Struktur.

Die x-Varianten von $\mathcal{A}$ liefern (entsprechend der Vorbemerkung) für diese Formel alle den Wahrheitswert $\mathbf{1}$.

Entsprechend Def. 9.10 ist dann $\mathcal{A}(\forall x\, (P(x) \vee \neg P(x))) = \mathbf{1}$.

Da die Wahl von $\mathcal{A}$ nicht eingeschränkt war, ist also $\forall x\, (P(x) \vee \neg P(x))$ gültig.

Gültigkeit von $\forall x\ P(x) \Rightarrow \exists y\ P(y)$

Beweis

Es sei $\mathcal{A} = (U, I)$ eine beliebige Struktur.

Wenn $\mathcal{A}(\forall x\ P(x) \Rightarrow \exists y\ P(y)) = \mathbf{0}$ ist, dann ist $\mathcal{A}(\forall x\ P(x)) = \mathbf{1}$ und $\mathcal{A}(\exists y\ P(y)) = \mathbf{0}$.

$\mathcal{A}(\exists y\ P(y)) = \mathbf{0}$ kann nur gelten, wenn für jedes $d \in U$ die y-Variante $\mathcal{A}_{[y/d]}$ von $\mathcal{A}$ folgendes liefert: $\mathcal{A}_{[y/d]}(P(y)) = \mathbf{0}$, also $\mathcal{A}_{[y/d]}(y) = d \notin \mathcal{A}_{[y/d]}(P) = \mathcal{A}(P)$. Dafür muss $\mathcal{A}(P) = \emptyset$ sein.

$\mathcal{A}(\forall x\ P(x)) = \mathbf{1}$ kann nur gelten, wenn für jedes $e \in U$ die x-Variante $\mathcal{A}_{[x/e]}$ von $\mathcal{A}$ folgendes liefert: $\mathcal{A}_{[x/e]}(P(x)) = \mathbf{1}$, also $\mathcal{A}_{[x/e]}(x) = e \in \mathcal{A}_{[x/e]}(P) = \mathcal{A}(P)$. Dafür muss $\mathcal{A}(P) = U$ sein.

Nach Definition 9.6 ist aber das Universum einer Struktur nicht leer. Also muss für alle Strukturen $\mathcal{A}(\forall x\ P(x) \Rightarrow \exists y\ P(y)) = \mathbf{1}$ gelten.

Semantik der Prädikatenlogik
Das Problem der systematischen Berücksichtigung aller Strukturen

Auswertung (einer Formel) in der Aussagenlogik

- $\mathcal{L}_{\mathrm{AL}}$ enthält nur Formeln mit endlicher Länge,
  d.h. jede Formel enthält nur eine endliche Anzahl von atomaren Formeln.
- Die Bewertung erfolgt über der Menge der Wahrheitswerte $\{\mathbf{0}, \mathbf{1}\}$
- Es gibt nur endlich viele zu unterscheidende Belegungen $\mathcal{A}_i$ für eine Formel $F \in \mathcal{L}_{\mathrm{AL}}$.

Auswertung (einer Formel) in der Prädikatenlogik

- Da es für eine Formel $F \in \mathcal{L}_{\mathrm{PL}}$ unendlich viele Strukturen $\mathcal{A}$ gibt,
  ist eine systematische, erschöpfende Berechnung aller Auswertungen der Formel F grundsätzlich nicht möglich.
- Es werden andere Techniken benötigt, um semantische Eigenschaften zu prüfen.

Erfüllbarkeit, Folgerung: Einige wichtige Theoreme

Die folgenden Sätze gelten entsprechend zur Aussagenlogik:

Satz 9.14: Eine Formel F ist genau dann gültig, wenn $\neg F$ unerfüllbar ist.
Satz 9.15: $F \models G$ genau dann, wenn $F \wedge \neg G$ unerfüllbar ist.

- Durch Satz 9.14 wird die Gültigkeit von Formeln auf die Unerfüllbarkeit von Formeln zurückgeführt.
- Durch Satz 9.15 wird Folgerbarkeit auf Unerfüllbarkeit zurückgeführt.

- Entsprechend der Aussagenlogik werden mechanische Verfahren zur Prüfung der Unerfüllbarkeit untersucht.

  ➔ Deduktions- und Widerlegungsverfahren.

Prädikatenlogik → Aussagenlogik: Semantische Perspektive

Reduktion der Prädikatenlogik auf die Aussagenlogik:

- Wenn kein Prädikatensymbol (mit Stelligkeit > 0) in einer prädikatenlogischen Sprache vorhanden ist, so sind alle atomaren Formeln Aussagensymbole.
  - Terme, Variablen und Funktionssymbole, aber insbesondere auch Quantoren sind in einer derartigen prädikatenlogischen Sprache überflüssig.
  - Die prädikatenlogische Semantik wird zur aussagenlogischen Semantik reduziert.
- Wenn in einer prädikatenlogischen Sprache weder Variablen noch Quantoren auftreten, ist ebenfalls eine Reduktion auf die Aussagenlogik möglich.
  - Die atomaren Formeln einer derartigen prädikatenlogischen Sprache können als atomare Formeln einer aussagenlogischen Sprache angesehen werden.
  - Auch in diesem Fall wird die prädikatenlogische Semantik zur aussagenlogischen Semantik reduziert.
  - ABER: Die zusätzliche Ausdrucksstärke der Prädikatenlogik geht dabei verloren!

Prädikatenlogik 2. Stufe

Prädikatenlogik 1. Stufe

- Im bisher vorgestellten Typ der Prädikatenlogik (Prädikatenlogik 1. Stufe) gibt es:
  - eine Art von Variablen, nämlich *Individuenvariablen*, die Terme sind,
  - eine Art von Quantoren, nämlich solche, die Individuenvariablen binden.

➔ Semantik für Formeln mit Quantoren (x-Varianten)

Jenseits der Prädikatenlogik 1. Stufe

- Erweiterung zu Quantifizierungen über Funktionen, Eigenschaften und Relationen
  - zusätzliche Arten von Variablen: Funktions-, Eigenschafts- und Relationsvariablen, denen eine eindeutige Stelligkeit zugewiesen ist.
  - die Quantoren, binden Variablen aller Typen.

➔ Erweiterung der Semantik für Formeln mit Quantoren ist notwendig.

- Vergrößerung der Ausdrucksstärke:
  - Peanos Axiomatisierung der Natürlichen Zahlen

$\forall P\ ((P(1) \land \forall n\ (P(n) \Rightarrow P(n+1))) \Rightarrow \forall n\ P(n))$ [„Induktionsaxiom“]

Zum Selbststudium: Funktionssymbole

Wann nutzt man Funktionssymbole, wann Relationssymbole

- Funktionssymbole in der Logiksprache sind im wesentlichen an zwei Stellen nützlich
  - Formalisierung der Mathematik
  - Zur Definition von Verarbeitungstechniken und Beweisen über die Logik (Wir werden später sog. Skolemfunktionssybole nutzen und damit gewisse Existenzquantoren überflüssig machen.)
- Die Übersetzung natürlichsprachlicher Ausdrücke in die Logik erfordert eigentlich nie die Nutzung von Funktionssymbolen.
- Bei der einfachen Logik, die wir hier nutzen, sind Funktionssybole auch etwas tückisch, da man sie mit jedem Term kombinieren kann und dann ein neuer Term entsteht, der wieder etwas bezeichnet. Das geht in der Mathematik für arithmetische Operationen ganz gut, aber schon die formale Erfassung der Tatsache, dass man nichts durch 0 teilen darf, ist eine echte Herausforderung der logischen Modellierung, mit der wir uns hier lieber nicht befassen wollen.

Wichtige Konzepte dieser Vorlesung

Syntax der Prädikatenlogik

- Variable, Quantor, Konstante, Funktionssymbol, Prädikatensymbol / Relationssymbol, Term, Stelligkeit
- Teilterm, Teilformel, Strukturbaum
- freie Variable (in einer Formel), durch Quantor gebundene Variable (in einer Position), Quantorenvariable, Skopus eines Quantors, geschlossene Formel

Semantik der Prädikatenlogik

- Struktur, Universum / Domäne, Interpretation / Auswertung
- Bestimmung des Wertes einer Formel durch eine Struktur / Auswertung einer Formel
- x-Variante einer Struktur
- Modell, erfüllbar, falsifizierbar, unerfüllbar (allgemein-)gültig
- Äquivalenz, Folgerung

Prädikatenlogik: Normalformen

Normalformen basierend auf Äquivalenz

- Äquivalenz in der Prädikatenlogik
- Übertragung aussagenlogischer Äquivalenzen
- Formeln mit Quantoren: Äquivalenzen in der Prädikatenlogik
- Substitution (Überführungslemma)
  - Gebundene Umbenennung von Variablen
- Pränexform

Normalformen basierend auf Erfüllbarkeitsäquivalenz

- Erfüllbarkeitsäquivalenz vs. Äquivalenz
- Bindung freier Variablen
- Skolemisierung
- Skolemform und Klauselnormalform
  - Normalform, die Grundlage für Resolutionsbeweise ist [→ Kap. 11]

Äquivalenz in der Prädikatenlogik

zur Erinnerung: Definition 9.12 (ganz analog zur Aussagenlogik)

Zwei Formeln F und G sind *(logisch)* *äquivalent*, falls für jede Struktur $\mathcal{A}$ gilt:
$\mathcal{A}(F) = \mathcal{A}(G)$.

Satz 10.1

Wenn in zwei äquivalenten aussagenlogischen Formeln prädikatenlogische Formeln (uniform) für die Aussagensymbole substituiert werden, ergibt dies zwei äquivalente Formeln der Prädikatenlogik.

ohne Beweis, vgl. Satz 6.3

➔ Die Äquivalenzen der Aussagenlogik gelten auch in der Prädikatenlogik.

Beispiel zu 10.1

Kommutativität von $\wedge$: $(\forall x\, P(x) \wedge \exists y\, Q(y)) \equiv (\exists y\, Q(y) \wedge \forall x\, P(x))$

Elimination von $\Rightarrow$: $(\forall x\, P(x) \Rightarrow \exists y\, Q(y)) \equiv (\neg\forall x\, P(x) \vee \exists y\, Q(y))$

➔ Zusätzliche prädikatenlogische Äquivalenzen ergeben sich aus den semantischen Beziehungen zwischen Quantoren.

Wichtige Äquivalenzen der Prädikatenlogik

Satz 10.2: Seien F und G beliebige prädikatenlogische Formeln.

1. $\neg \forall x\, F \equiv \exists x\, \neg F$
   $\neg \exists x\, F \equiv \forall x\, \neg F$

   Dualität von $\forall$ und $\exists$

2. Falls x in G nicht frei vorkommt

   $G \equiv \forall x\, G \equiv \exists x\, G$

   leere Quantifikation

   $(\forall x\, F \wedge G) \equiv \forall x\, (F \wedge G)$
   $(\forall x\, F \vee G) \equiv \forall x\, (F \vee G)$
   $(\exists x\, F \wedge G) \equiv \exists x\, (F \wedge G)$
   $(\exists x\, F \vee G) \equiv \exists x\, (F \vee G)$
   $(G \Rightarrow \forall x\, F) \equiv \forall x\, (G \Rightarrow F)$
   $(G \Rightarrow \exists x\, F) \equiv \exists x\, (G \Rightarrow F)$
   $(\forall x\, F \Rightarrow G) \equiv \exists x\, (F \Rightarrow G)$
   $(\exists x\, F \Rightarrow G) \equiv \forall x\, (F \Rightarrow G)$

   Skopuserweiterung
   wenn damit keine neuen Variablen-Bindungen entstehen

3. $(\forall x\, F \wedge \forall x\, G) \equiv \forall x\, (F \wedge G)$
   $(\exists x\, F \vee \exists x\, G) \equiv \exists x\, (F \vee G)$

   Distributivität von
   Allquantor bzgl. Konjunktion
   Existenzquantor bzgl. Disjunktion

4. $\forall x\, \forall y\, F \equiv \forall y\, \forall x\, F$
   $\exists x\, \exists y\, F \equiv \exists y\, \exists x\, F$

   Vertauschung der *Quantorenreihenfolge*
   nur bei gleichen Quantoren

Beweis einiger Äquivalenzen

Satz 10.2.2.1: Wenn x nicht frei in G vorkommt, dann $G \equiv \forall x\, G$ und $G \equiv \exists x\, G$.

Beweis

Sei $\mathcal{A} = (U, I)$ eine Struktur.

$\mathcal{A}(G) = 1$ GDW. für alle $d \in U$ gilt: $\mathcal{A}_{[x/d]}(G) = 1$ [da x nicht frei in G]

GDW. $\mathcal{A}(\forall x\, G) = 1$ [Auswertung des Allquantors]

$\mathcal{A}(G) = 1$ GDW. für ein $d \in U$ gilt: $\mathcal{A}_{[x/d]}(G) = 1$ [da U nicht leer und x nicht frei in G]

GDW. $\mathcal{A}(\exists x\, G) = 1$ [Auswertung des Existenzquantors]

Satz 10.2.2.2: Wenn x nicht frei in G vorkommt, dann $(\forall x\, F \wedge G) \equiv \forall x\, (F \wedge G)$.

Beweis

Sei $\mathcal{A} = (U, I)$ eine Struktur.

$\mathcal{A}(\forall x\, F \wedge G) = 1$ GDW. $\mathcal{A}(\forall x\, F) = 1$ und $\mathcal{A}(G) = 1$ [Auswertung der Konjunktion]

GDW. für alle $d \in U$ gilt: $\mathcal{A}_{[x/d]}(F) = 1$ und $\mathcal{A}(G) = 1$ [Auswertung des Allquantors]

GDW. für alle $d \in U$ gilt: $\mathcal{A}_{[x/d]}(F) = 1$ und $\mathcal{A}_{[x/d]}(G) = 1$ [da x nicht frei in G]

GDW. für alle $d \in U$ gilt: $\mathcal{A}_{[x/d]}(F \wedge G) = 1$ [Auswertung der Konjunktion]

GDW. $\mathcal{A}(\forall x\, (F \wedge G)) = 1$ [Auswertung des Allquantors]

Satz 10.3

a) $(\forall x\ F \vee \forall x\ G) \not\equiv \forall x\ (F \vee G)$

$(\exists x\ F \wedge \exists x\ G) \not\equiv \exists x\ (F \wedge G)$

b) $\forall x\ \exists y\ F \not\equiv \exists y\ \forall x\ F$

Beweisidee:

Durch Gegenbeispiele
zu a) Domäne: natürliche Zahlen: $F \approx x$ ist ungerade, $G \approx x$ ist gerade
zu b) Domäne: natürliche Zahlen: $F \approx y$ ist direkter Nachfolger von x

Zum Selbststudium: Äquivalenzen – Nicht-Äquivalenzen

- Um Äquivalenz von Formeln F und G zu beweisen, ist es notwendig, beliebige Strukturen $\mathcal{A}$ (d.h. insbesondere über beliebigen Domänen) zu berücksichtigen, genauer (vgl. Def. 9.12), zu zeigen, dass für jede Struktur $\mathcal{A}$ gilt:
  $\mathcal{A}(F) = \mathcal{A}(G)$.
  [Dies geschieht in den Beweisen zu Satz 10.2].
- Um die Nicht-Äquivalenz der Formeln F und G zu beweisen, reicht es aus, eine Struktur $\mathcal{A}$ zu finden, für die $\mathcal{A}(F) \neq \mathcal{A}(G)$.
  [Für Satz 10.3 gibt die Beweisidee den Hinweis auf entsprechende Strukturen; ein ausgearbeiteter Beweis für 10.3 beinhaltet sowohl die explizite Spezifikation der Strukturen als auch den Nachweis der Verschiedenheit der Auswertung, d.h. die Begründung für $\mathcal{A}(F) \neq \mathcal{A}(G)$.]

- Entsprechendes gilt für Beweise von
  - Unerfüllbarkeit – Nicht-Unerfüllbarkeit
  - Folgerbarkeit – Nicht-Folgerbarkeit

Ersetzbarkeitstheorem (Prädikatenlogik)

Satz 10.4

Seien F und G äquivalente Formeln und sei H eine Formel mit Teilformel F.
Sei H' eine Formel, die aus H durch Ersetzung von F durch G hervorgeht.

Dann gilt: $H' \equiv H$.

Beweis: s. Satz 4.9 (folgt dem Prinzip der strukturellen Induktion):
Beim Induktionsschritt sind zusätzlich Formeln mit Quantoren zu berücksichtigen.
Der entscheidende Punkt ist, zu zeigen:

Hilfssatz 10.5

Wenn H_1 und H_1' äquivalente Formeln sind, dann $\exists x\, H_1 \equiv \exists x\, H_1'$ und $\forall x\, H_1 \equiv \forall x\, H_1'$.

Beweis

Sei $H_1 \equiv H_1'$ und $\mathcal{A}$ ein Modell von $\exists x\, H_1$. Dann gibt es ein $d \in U_{\mathcal{A}}$, so dass $\mathcal{A}_{[x/d]}(H_1) = \mathbf{1}$. Da $H_1 \equiv H_1'$, ist $\mathcal{A}_{[x/d]}(H_1') = \mathbf{1}$ und damit ist $\mathcal{A}$ ein Modell von $\exists x\, H_1'$.
Der zweite Teil des Beweises funktioniert analog.

Beispiel: Äquivalenz-Umformungen

	$(\neg(\exists x\, P(x, y) \lor \forall z\, Q(z)) \land \exists w\, P(f(a), w))$	
$\equiv$	$((\neg\exists x\, P(x, y) \land \neg\forall z\, Q(z)) \land \exists w\, P(f(a), w))$	de Morgan
$\equiv$	$((\forall x\, \neg P(x, y) \land \exists z\, \neg Q(z)) \land \exists w\, P(f(a), w))$	[1] Dualität
$\equiv$	$(\exists w\, P(f(a), w) \land (\forall x\, \neg P(x, y) \land \exists z\, \neg Q(z)))$	Kommutativität $\land$
$\equiv$	$\exists w\, (P(f(a), w) \land (\forall x\, \neg P(x, y) \land \exists z\, \neg Q(z)))$	[2] Skopus von $\exists w$
$\equiv$	$\exists w\, (P(f(a), w) \land \forall x\, (\neg P(x, y) \land \exists z\, \neg Q(z)))$	[2] Skopus von $\forall x$
$\equiv$	$\exists w\, (\forall x\, (\neg P(x, y) \land \exists z\, \neg Q(z)) \land P(f(a), w))$	Kommutativität $\land$
$\equiv$	$\exists w\, (\forall x\, (\exists z\, \neg Q(z) \land \neg P(x, y)) \land P(f(a), w))$	Kommutativität $\land$
$\equiv$	$\exists w\, (\forall x\, \exists z\, (\neg Q(z) \land \neg P(x, y)) \land P(f(a), w))$	[2] Skopus von $\exists z$
$\equiv$	$\exists w\, \forall x\, \exists z\, (\neg Q(z) \land \neg P(x, y) \land P(f(a), w))$	[2] Skopus von $\forall x$ und $\exists z$

➔ Quantorenreihenfolge nach der Umformung hängt von der Reihenfolge der Umformungen ab

➔ Skopuserweiterung nach [2] ist nur möglich, wenn in der neu eingeschlossenen Formel nicht die Variable auftritt, die durch den betroffenen Quantor gebunden ist.

Normalform für Prädikatenlogik

Definition 10.6

- Eine Formel F heißt *bereinigt*, falls es keine Variable in F gibt, die in der Formel sowohl gebunden als auch frei vorkommt und falls alle Quantoren in F unterschiedliche Quantorenvariablen aufweisen.
- Eine Formel F heißt pränex (oder in Pränexform), falls sie die folgende Form aufweist: $F = \mathcal{Q}_1 y_1 \mathcal{Q}_2 y_2 \ldots \mathcal{Q}_n y_n\, G$, wobei $\mathcal{Q}_i \in \{\exists, \forall\}$, $n \geq 0$, die y_i Variablen sind und in G keine Quantoren vorkommen. $\mathcal{Q}_1 y_1 \mathcal{Q}_2 y_2 \ldots \mathcal{Q}_n y_n$ heißt (Quantoren-) *Präfix*, G ist die *Matrix* von F.
- Ist eine Formel G bereinigt, in Pränexform und äquivalent zur Formel F, dann nennen wir G eine BPF zu F.

Das nächste Ziel

- Satz 10.10: Zu jeder Formel gibt es eine BPF

Dafür erforderlich

- Möglichkeit der Umbenennung von Variablen
  - Definition Variablensubstitution
  - Überführungslemma (semantische Effekte der Variablensubstitution)
  - Satz zur gebundenen Umbenennung

Variablen-Substitution (1)

Definition 10.7 (einfache (Variablen-) Substitution)

Seien F eine Formel, x eine Variable und t ein Term. $F_{[x/t]}$ ist die Formel, die sich aus F ergibt, wenn jedes freie Vorkommen der Variablen x in F durch den Term t ersetzt wird. $[x/t]$ wird als *Substitution* bezeichnet.

Explizit als rekursive Funktion definiert:

Eine einfache *(Variablen-) Substitution* ist eine rekursiv definierte Funktion $[x/t]$.

- Die Variable x wird auf den Term t abgebildet.
- Alle anderen Variablen und Konstanten werden auf sich selbst abgebildet.
- Für komplexe Terme gilt: $[x/t](f(t_1, \ldots, t_k)) = f([x/t](t_1), \ldots, [x/t](t_k))$
- Für atomare Formeln gilt: $[x/t](P(t_1, \ldots, t_k)) = P([x/t](t_1), \ldots, [x/t](t_k))$
- Für komplexe Formeln gilt:

  $[x/t](\neg F) = \neg[x/t](F)$

  $[x/t]((F \wedge G)) = ([x/t](F) \wedge [x/t](G))$ $\quad$ $[x/t]((F \vee G)) = ([x/t](F) \vee [x/t](G))$

  $[x/t]((F \Rightarrow G)) = ([x/t](F) \Rightarrow [x/t](G))$ $\quad$ $[x/t]((F \Leftrightarrow G)) = ([x/t](F) \Leftrightarrow [x/t](G))$

  $[x/t](\forall x\, F) = \forall x\, F$ $\quad$ $[x/t](\exists x\, F) = \exists x\, F$

  $[x/t](\forall y\, F) = \forall y\, [x/t](F)$ $\quad$ $[x/t](\exists y\, F) = \exists y\, [x/t](F)$, wobei $x \neq y$.
- Statt $[x/t](F)$ wird auch $F_{[x/t]}$ geschrieben.

Beispiel: einfache Variablensubstitution

$F = \forall y\ (\exists z\ P(x, y, z) \vee \forall x\ P(x, y, z))$

$t = g(a)$

$\sigma_1 = [x / g(a)]$

$$
\begin{aligned}
\sigma_1(F) &= \sigma_1(\forall y\ (\exists z\ P(x, y, z) \vee \forall x\ P(x, y, z)))\\
&= \forall y\ \sigma_1((\exists z\ P(x, y, z) \vee \forall x\ P(x, y, z)))\\
&= \forall y\ (\sigma_1(\exists z\ P(x, y, z)) \vee \sigma_1(\forall x\ P(x, y, z)))\\
&= \forall y(\exists z\ \sigma_1(P(x, y, z)) \vee \forall x\ P(x, y, z))\\
&= \forall y(\exists z\ P(\sigma_1(x), \sigma_1(y), \sigma_1(z)) \vee \forall x\ P(x, y, z))\\
&= \forall y\ (\exists z\ P(g(a), y, z) \vee \forall x\ P(x, y, z))
\end{aligned}
$$

$\sigma_2 = [y / g(a)]$

$\sigma_2(F) = \forall y\ (\exists z\ P(x, y, z) \vee \forall x\ P(x, y, z))$

$\sigma_3 = [z / g(b)]$

$\sigma_3(F) = \forall y\ (\exists z\ P(x, y, z) \vee \forall x\ P(x, y, g(b)))$

$\sigma_1(\sigma_3(F)) = \forall y\ (\exists z\ P(g(a), y, z) \vee \forall x\ P(x, y, g(b)))$

Definition 10.7x ((Variablen-) Substitution)

Eine *(Variablen-) Substitution* ist eine rekursiv definierte Funktion σ.

- Variablen x_i werden auf Terme $\sigma(x_i)$ abgebildet.
- Konstanten a_i werden auf sich selbst abgebildet.
- Für komplexe Terme gilt: $\sigma(f(t_1, ..., t_k)) = f(\sigma(t_1), ..., \sigma(t_k))$
- Für atomare Formeln gilt: $\sigma(P(t_1, ..., t_k)) = P(\sigma(t_1), ..., \sigma(t_k))$
- Für komplexe Formeln gilt:

  $\sigma(\neg F) = \neg\sigma(F)$

  $\sigma((F \wedge G)) = (\sigma(F) \wedge \sigma(G))$ $\qquad$ $\sigma((F \vee G)) = (\sigma(F) \vee \sigma(G))$

  $\sigma((F \Rightarrow G)) = (\sigma(F) \Rightarrow \sigma(G))$ $\qquad$ $\sigma((F \Leftrightarrow G)) = (\sigma(F) \Leftrightarrow \sigma(G))$

  $\sigma(\forall x\, F) = \forall x\, \sigma_x(F)$ $\qquad$ $\sigma(\exists x\, F) = \exists x\, \sigma_x(F)$
- Dabei ist σ_x diejenige Substitution, die sich von σ nur dadurch unterscheidet, dass die Variable x auf sich selbst abgebildet wird.

➔ Durch eine (Variablen-)Substitution wird jede Formel F auf eine Formel $\sigma(F)$ abgebildet, wobei *jedes* Vorkommen einer freien Variablen x_i in F durch den entsprechenden Term $\sigma(x_i)$ ersetzt wird.

➔ Quantorenvariablen und durch sie gebundene Vorkommen von Variablen werden durch die Variablen-Substitution nicht ersetzt.

Beispiel: Substitution

$F = \forall y\, (\exists z\, P(x, y, z) \vee \forall x\, P(x, y, z))$

$\sigma_4 =$ [$x / g(a), z / g(a)$] : die beiden Variablen werden parallel substituiert !

$\sigma_{4y} = \sigma_4$

$\sigma_{4x} = [z / g(a)]$

$\sigma_{4z} = [x / g(a)]$

$$
\begin{aligned}
\sigma_4(F) &= \sigma_4(\forall y\, (\exists z\, P(x, y, z) \vee \forall x\, P(x, y, z))) \\
&= \forall y\, \sigma_{4y}((\exists z\, P(x, y, z) \vee \forall x\, P(x, y, z))) \\
&= \forall y\, \sigma_4((\exists z\, P(x, y, z) \vee \forall x\, P(x, y, z))) \\
&= \forall y\, (\sigma_4(\exists z\, P(x, y, z)) \vee \sigma_4(\forall x\, P(x, y, z))) \\
&= \forall y(\exists z\, \sigma_{4z}(P(x, y, z)) \vee \forall x\, \sigma_{4x}(P(x, y, z)\,)) \\
&= \forall y(\exists z\, P(\sigma_{4z}(x), \sigma_{4z}(y), \sigma_{4z}(z)) \vee \forall x\, P(\sigma_{4x}(x), \sigma_{4x}(y), \sigma_{4x}(z))) \\
&= \forall y\, (\exists z\, P(g(a), y, z) \vee \forall x\, P(x, y, g(a)))
\end{aligned}
$$

Satz 10.8 (Überführungslemma)

Sei F eine Formel, x eine Variable und t ein Term, der keine in F gebundene Variable enthält. Dann gilt für jede Struktur $\mathcal{A}$:

$$\mathcal{A}(F_{[x/t]}) = \mathcal{A}_{[x/\mathcal{A}(t)]}(F)$$

Beweis: siehe Schöning: Übung 58.

- Das Überführungslemma stellt den Zusammenhang zwischen Substitutionen und Modellen (bzw. Auswertungen) her:

$\mathcal{A}(F_{[x/t]})$	Auswertung der Formel, die aus F durch Substitution von x durch t hervorgeht
$\mathcal{A}_{[x/\mathcal{A}(t)]}(F)$	Auswertung von F bezüglich der x-Variante, in der x zu $\mathcal{A}(t)$ ausgewertet wird.

$$\begin{array}{ccccccc} x & [x/t] \rightarrow & t & \qquad & F & [x/t] \rightarrow & F_{[x/t]} \\ \downarrow \mathcal{A}_{[x/\mathcal{A}(t)]} & & \downarrow \mathcal{A} & & \downarrow \mathcal{A}_{[x/\mathcal{A}(t)]} & & \downarrow \mathcal{A} \\ \mathcal{A}_{[x/\mathcal{A}(t)]}(x) & = & \mathcal{A}(t) & & \mathcal{A}_{[x/\mathcal{A}(t)]}(F) & = & \mathcal{A}(F_{[x/t]}) \end{array}$$

Gebundene Umbenennung

Satz 10.9 (gebundene Umbenennung)

Sei F eine Formel und y eine Variable, die in F nicht vorkommt.

Dann gilt: $\exists x\ F \equiv \exists y\ F_{[x/y]}$ und $\forall x\ F \equiv \forall y\ F_{[x/y]}$

Beweis

Sei $\mathcal{A}$ eine Struktur.

$\mathcal{A}(\exists y\ F_{[x/y]}) = 1$

GDW. es ein $d \in U$ gibt, so dass $\mathcal{A}_{[y/d]}(F_{[x/y]}) = 1$ [Quantor-Interpretation]

GDW. es ein $d \in U$ gibt, so dass $\mathcal{A}_{[y/d][x/d]}(F) = 1$ [Überführungslemma]

GDW. $\mathcal{A}_{[y/d]}(\exists x\ F) = 1$ [Quantor-Interpretation]

GDW. $\mathcal{A}(\exists x\ F) = 1$ [da y nicht frei in F vorkommt]

$\mathcal{A}(\forall y\ F_{[x/y]}) = 1$

GDW. für alle $d \in U$ gilt $\mathcal{A}_{[y/d]}(F_{[x/y]}) = 1$ [Quantor-Interpretation]

GDW. für alle $d \in U$ gilt $\mathcal{A}_{[y/d][x/d]}(F) = 1$ [Überführungslemma]

GDW. für alle $d \in U$ gilt $\mathcal{A}_{[x/d]}(F) = 1$ [da y nicht frei in F vorkommt]

GDW. $\mathcal{A}(\forall x\ F) = 1$ [Quantor-Interpretation]

Achtung: Umbenennung *freier* Variablen erhält die Äquivalenz nicht !

Existenz einer äquivalenten bereinigten Pränexform

Satz 10.10 (Existenz einer äquivalenten Formel in BPF)
Für jede Formel F gibt es eine äquivalente Formel G in bereinigter Pränexform.

Beweis
Vorbemerkung:
Gemäß Satz 4.10 gibt es zu jeder Formel eine äquivalente Formel, in der $\Leftrightarrow$ nicht vorkommt.
Der Beweis setzt entsprechend voraus, dass $\Leftrightarrow$ in F nicht vorkommt.

Induktionsanfang
Wenn F eine atomare Formel ist, dann hat F Pränexform mit leerem Präfix (n = 0).
Induktionsannahme
Es seien F_1 und F_2 Formeln, zu denen es äquivalente bereinigte Pränexformen $\mathcal{Q}_1 x_1 \mathcal{Q}_2 x_2 \ldots \mathcal{Q}_n x_n\ G_1$ bzw. $\mathcal{Q}'_1 y_1 \mathcal{Q}'_2 y_2 \ldots \mathcal{Q}'_m y_m\ G_2$ gibt.
Aufgrund von Satz 10.9 können wir weiterhin voraussetzen, dass $x_i \neq y_j$ für alle i und j sind und dass keine in F_1 bzw. F_2 freien Variable zu den x_i bzw. y_j gehört.
Induktionsschritt: nächste Folie.

Pränexform: Induktionsschritt

Zur Erinnerung: $F_1 \equiv \mathcal{Q}_1x_1\mathcal{Q}_2x_2\ldots\,\mathcal{Q}_nx_n\; G_1$ und $F_2 \equiv \mathcal{Q}'_1y_1\mathcal{Q}'_2y_2\ldots\,\mathcal{Q}'_my_m\; G_2$

Sei $\overline{\mathcal{Q}} = \begin{cases} \exists, \text{ falls } \mathcal{Q} = \forall \\ \forall, \text{ falls } \mathcal{Q} = \exists \end{cases}$.

Dann gilt: $\neg F_1 \equiv \overline{\mathcal{Q}}_1x_1\overline{\mathcal{Q}}_2x_2\ldots\,\overline{\mathcal{Q}}_nx_n\;\neg G_1$ [Satz 10.2.1]

$F_1 \wedge F_2 \equiv \mathcal{Q}_1x_1\mathcal{Q}_2x_2\ldots\,\mathcal{Q}_nx_n\; G_1 \wedge \mathcal{Q}'_1y_1\mathcal{Q}'_2y_2\ldots\,\mathcal{Q}'_my_m\; G_2$

$\equiv \mathcal{Q}_1x_1\mathcal{Q}_2x_2\ldots\,\mathcal{Q}_nx_n\mathcal{Q}'_1y_1\mathcal{Q}'_2y_2\ldots\,\mathcal{Q}'_my_m\;(G_1 \wedge G_2)$ [Satz 10.2.2]

$F_1 \vee F_2 \equiv \mathcal{Q}_1x_1\mathcal{Q}_2x_2\ldots\,\mathcal{Q}_nx_n\; G_1 \vee \mathcal{Q}'_1y_1\mathcal{Q}'_2y_2\ldots\,\mathcal{Q}'_my_m\; G_2$

$\equiv \mathcal{Q}_1x_1\mathcal{Q}_2x_2\ldots\,\mathcal{Q}_nx_n\mathcal{Q}'_1y_1\mathcal{Q}'_2y_2\ldots\,\mathcal{Q}'_my_m\;(G_1 \vee G_2)$ [Satz 10.2.2]

$F_1 \Rightarrow F_2 \equiv \mathcal{Q}_1x_1\mathcal{Q}_2x_2\ldots\,\mathcal{Q}_nx_n\; G_1 \Rightarrow \mathcal{Q}'_1y_1\mathcal{Q}'_2y_2\ldots\,\mathcal{Q}'_my_m\; G_2$

$\equiv \overline{\mathcal{Q}}_1x_1\overline{\mathcal{Q}}_2x_2\ldots\,\overline{\mathcal{Q}}_nx_n\;\mathcal{Q}'_1y_1\mathcal{Q}'_2y_2\ldots\,\mathcal{Q}'_my_m\;(G_1 \Rightarrow G_2)$ [Satz 10.2.2]

$\exists x\, F_1 \equiv \exists x\; \mathcal{Q}_1x_1\mathcal{Q}_2x_2\ldots\,\mathcal{Q}_nx_n\; G_1$ [BPN, wenn $x \neq x_i$ für alle i]

$\forall x\, F_1 \equiv \forall x\; \mathcal{Q}_1x_1\mathcal{Q}_2x_2\ldots\,\mathcal{Q}_nx_n\; G_1$ [BPN, wenn $x \neq x_i$ für alle i]

wenn $x = x_i$ für ein i, dann kommt x nicht frei in $\mathcal{Q}_1x_1\mathcal{Q}_2x_2\ldots\,\mathcal{Q}_nx_n\; G_1$ vor und

$\exists x\, F_1 \equiv \forall x\, F_1 \equiv \mathcal{Q}_1x_1\mathcal{Q}_2x_2\ldots\,\mathcal{Q}_nx_n\; G_1$ [BPN]

Normalformen: Zwischenstand

Äquivalenzumformungen führen zu bereinigten Pränexformen

- die Matrix lässt sich in KNF oder DNF bringen.
- im Präfix können aber Allquantoren und Existenzquantoren gemischt auftreten.
- In einer Formel können sowohl freie als auch gebundene Variablen auftreten.
- Durch Äquivalenzumformungen läßt sich dieses nicht aufheben.

Die nächsten Schritte

- Bindung freier Variablen durch Existenzquantoren.
- Elimination der Existenzquantoren im Präfix durch Skolemisierung.
- Beide Umformungsschritte gewährleisten zwar nicht Äquivalenz aber (wenigstens) Erfüllbarkeitsäquivalenz.

Definition 10.11 (Erfüllbarkeitsäquivalenz)

Zwei Formeln F und G sind genau dann *erfüllbarkeitsäquivalent*, wenn gilt:
F ist erfüllbar GDW. G erfüllbar ist.

Beobachtung zu Definition 10.11

- F und G sind erfüllbarkeitsäquivalent: F ist unerfüllbar GDW. G unerfüllbar ist.
- F und G sind erfüllbarkeitsäquivalent:
  Es gibt eine erfüllende Struktur $\mathcal{A}$ zu F GDW. es eine erfüllende Struktur $\mathcal{A}'$ zu G gibt.
- Erfüllbarkeitsäquivalenz ist schwächer als Äquivalenz,
  d.h. Äquivalenz schließt Erfüllbarkeitsäquivalenz ein.
- Äquivalenz betrifft gleiche Bewertung durch alle Strukturen. Erfüllbarkeitsäquivalenz betrifft die Existenz von erfüllenden Strukturen. (Diese können verschieden sein.)

Umformungen, die Erfüllbarkeitsäquivalenz garantieren

- sind als Vorstufe von Widerlegungsverfahren nützlich ➔ Resolution (Kapitel 12)

Zur Erinnerung: Reduktion semantischer Fragen auf (Un)Erfüllbarkeit

Sätze

(gelten in der Prädikatenlogik ebenso wie in der Aussagenlogik)

- Formel F ist genau dann eine Tautologie ($\models F$), wenn $\neg F$ unerfüllbar ist.
- Formel F folgt genau dann aus Formel G ($G \models F$), wenn $(G \wedge \neg F)$ unerfüllbar ist.
- Formeln F und G sind genau dann äquivalent ($G \equiv F$), wenn $\neg(F \Leftrightarrow G)$ unerfüllbar ist.

Konsequenz

- Jedes Verfahren, das der Feststellung der Erfüllbarkeit / Unerfüllbarkeit von Formeln dient, ist auch verwendbar, um Gültigkeit, Folgerung und Äquivalenz festzustellen.
- Allerdings muss das Verfahren dazu auf geeignet gebildete Formeln angewendet werden.
- Entsprechend muss auch die Vorverarbeitung auf die geeignete (und das ist nicht immer der ursprüngliche) Formel angewendet werden.
- Nachdem die ursprüngliche Frage in eine (Un)Erfüllbarkeitsfrage bzgl. einer Formel umgewandelt wurde, können auch (unbesorgt) Umformungen beruhend auf Erfüllbarkeitsäquivalenz vorgenommen werden.

Beispiel: Erfüllbarkeitsäquivalenz

Satz 10.12

Sei F eine Formel und x eine Variable. F ist genau dann erfüllbar, wenn $\exists x\ F$ erfüllbar ist.

➔ maW. F und $\exists x\ F$ sind erfüllbarkeitsäquivalent.

Beweis

Sei $\mathcal{A}$ ein Modell von F.

Dann gilt $\mathcal{A}_{[x/\mathcal{A}(x)]}(F) = \mathcal{A}(F) = \mathbf{1}$ und damit $\mathcal{A}(\exists x\ F) = \mathbf{1}$

Sei $\mathcal{A}$ ein Modell von $\exists x\ F$.

Dann gibt es ein $d \in U$, so dass $\mathcal{A}_{[x/d]}(F) = \mathbf{1}$. Also ist auch F erfüllbar.

Satz 10.13

Zu jeder prädikatenlogischen Formel gibt es eine erfüllbarkeitsäquivalente geschlossene Formel.

Beweisidee

Sei F eine Formel und seien $x_1, \ldots, x_n$ die in F frei vorkommenden Variablen.

F und $\exists x_1 \ldots \exists x_n\ F$ sind erfüllbarkeitsäquivalent.

(Dies ist durch vollständige Induktion über n zu zeigen.)

Beispiel: Erfüllbarkeitsäquivalenz

Satz 10.14

Sei F eine Formel, x eine Variable und a eine Konstante, die nicht in F vorkommt.
Dann sind $F_{[x/a]}$ und $\exists x\, F$ erfüllbarkeitsäquivalent.

Beweis

Sei $\mathcal{A}$ ein Modell von $F_{[x/a]}$.

Dann gilt $\mathcal{A}_{[x/\mathcal{A}(a)]}(F) = \mathcal{A}(F_{[x/a]}) = \mathbf{1}$ und damit $\mathcal{A}(\exists x\, F) = \mathbf{1}$

Sei $\mathcal{B}$ ein Modell von $\exists x\, F$.

Dann gibt es ein $d \in U$, so dass $\mathcal{B}_{[x/d]}(F) = \mathbf{1}$. Die Struktur $\mathcal{A}$ sei so definiert, dass sie nur in der Interpretation von a von $\mathcal{B}$ abweicht. Hier gilt: $\mathcal{A}(a) = d$. Da a nicht in F vorkommt, spielt $\mathcal{B}(a) = \mathcal{B}_{[x/d]}(a)$ für die Bestimmung von $\mathcal{B}_{[x/d]}(F)$ keine Rolle, also ist $\mathcal{B}_{[x/d]}(F) = \mathcal{A}_{[x/d]}(F) = \mathcal{A}_{[x/\mathcal{A}(a)]}(F)$.

Wegen des Überführungslemmas gilt weiter: $\mathcal{A}_{[x/\mathcal{A}(a)]}(F) = \mathcal{A}(F_{[x/a]})$.

Also ist auch $F_{[x/a]}$ erfüllbar.

Beobachtung zu den Sätzen 10.12, 10.13, 10.14

- Die Sätze machen nur Aussagen über Existenzquantoren mit maximalem Skopus.

➔ Skolemisierung: Verallgemeinerung von Satz 10.14 für Existenzquantoren an beliebiger Position im Quantorenpräfix

Definition 10.15 (Skolemisierung, Skolemsymbol)

Es sei $F = \forall y_1 \forall y_2 \ldots \forall y_k \exists z\ G$, $k \geq 0$ und f ein k-stelliges Funktionssymbol, das nicht in F vorkommt.

Wir bilden die Formel:

$$F' = \forall y_1 \forall y_2 \ldots \forall y_k\ G_{[z/f(y1,\ \ldots,\ yk)]}$$

- f heißt *Skolemsymbol*, wenn $k > 0$ auch *Skolemfunktion*, im Fall $k = 0$ auch *Skolemkonstante*.

  [Toralf Skolem]

- Skolemsymbole sind von allen bisher verwendeten Funktionssymbolen und Konstanten verschieden.
- Die Bildung von F' heißt Skolemisierung von F.

Beobachtungen zu Definition 10.15

- F' enthält einen Existenzquantor weniger als F.
- Die durch diesen Existenzquantor gebundene Variable z tritt nicht mehr in F' auf.
- z wurde in G durch $f(y_1, \ldots, y_k)$ substituiert.
- Ist F eine Formel in BPF und F' durch Skolemisierung aus F hervorgegangen, dann ist auch F' in BPF.

Beispiele: Skolemisierung

k=0	$\exists x\ \forall y\ (vorf(x,y) \wedge weibl(x))$	Skolemkonstante e
	$\forall y\ (vorf(e, y) \wedge weibl(e))$	
k=1	$\forall y\ \exists x\ (elt(x, y) \wedge weibl(x))$	Skolemfunktion $m(y)$
	$\forall y\ (elt(m(y), y) \wedge weibl(m(y)))$	
	$\forall y\ \exists x\ (vorf(x, y) \wedge weibl(x))$	Skolemfunktion $f(y)$
	$\forall y\ (vorf(f(y), y) \wedge weibl(f(y)))$	
k=2	$\forall x\ \forall y\ \exists z\ (gr\_elt(x, y) \Rightarrow (elt(x, z) \wedge elt(z, y)))$	
	$\forall x\ \forall y\ (gr\_elt(x, y) \Rightarrow (elt(x, g(x, y)) \wedge elt(g(x, y),y)))$	Skolemfunktion $g(x, y)$

➔ Vorsicht: Die Einführung der Skolemsymbole besagt noch nichts über die Interpretation der Formeln, die Skolemsymbol enthalten.

Satz 10.16

Sei F eine Formel und F' durch Skolemisierung aus F hervorgegangen.
Dann sind F und F' erfüllbarkeitsäquivalent.

Beweis

Der Fall der Einführung von Skolemkonstanten wurde in Satz 10.14 behandelt. Für den allgemeinen Fall:

➔ siehe Schöning (s. 67f.)

Definition 10.17

Eine Formel F ist in Skolemform (… ist eine Skolemformel), wenn sie

- in BPF (bereinigter Pränexform) ist
- geschlossen ist (keine freien Variablen aufweist) und
- keinen Existenzquantor enthält.

Eine Formel F ist in Klauselnormalform, wenn sie

- in Skolemform ist und
- ihre Matrix in KNF ist.

➔ In Skolemformen und Klauselnormalformen sind alle Variablen allquantifiziert.

➔ Die Matrix enthält alle relevante Information.

Satz 10.18 (Existenz einer erfüllbarkeitsäquivalenten Skolemformel)

Für jede Formel in BPF existiert eine erfüllbarkeitsäquivalente Skolemformel.

Beweisidee

Nach Satz 10.13 können die freien Variablen durch Existenzquantoren gebunden werden und auf die entstehende Formel kann mehrfach Skolemisierung angewendet werden, bis kein Existenzquantor mehr auftritt. Alle Schritte gewährleisten Erfüllbarkeitsäquivalenz (und Erfüllbarkeitsäquivalenz ist transitiv.)

Verfahren zur Erstellung der Skolemform

Sei F in BPF.

$F_0 := F$

i := 0

while F_i enthält einen Existenzquantor **do**

begin

Es ist $F_i = \forall y_1 \forall y_2 \ldots \forall y_k \exists z\, G_i$

Wähle ein neues k-stelliges Funktionssymbol und setze:

$F_{i+1} = \forall y_1 \forall y_2 \ldots \forall y_k\, G_i[z/f(y_1, \ldots, y_k)]$

i := i + 1

end

Satz 10.18 (Ex. einer erfüllbarkeitsäquivalenten Klauselnormalform)

Für jede Formel existiert eine erfüllbarkeitsäquivalente Formel in Klauselnormalform.

Beweisidee

	Umformungsschritte		
F	beliebige Form		
	Bereinigung: Umbenennen der gebundenen Variablen		
F_1	bereinigt	$F \equiv F_1$	Satz 10.9
	Bindung aller freien Variablen durch Existenzquantoren		
F_2	bereinigt, geschlossen	F_1 erfüllbarkeitsäquivalent F_2	Satz 10.13
	Erstellung einer Pränexform		
F_3	BPF, geschlossen	$F_2 \equiv F_3$	Satz 10.10
	Skolemisierung		
F_4	Skolemform	F_3 erfüllbarkeitsäquivalent F_4	Satz 10.18
	Umformung der Matrix in KNF		
F_5	Klauselnormalform	$F_4 \equiv F_5$	Satz 4.23

Also sind auch F und F_5 sind erfüllbarkeitsäquivalent.

Ziel: Mengendarstellung der Klauselnormalform (1)

Klauselnormalform

$$F = \forall y_1 \forall y_2 \ldots \forall y_p \left(\bigwedge_{i=1}^{n} \left(\bigvee_{k=1}^{mi} L_{i,k} \right) \right)$$

- Keine Existenzquantoren, keine freien Variablen.
- kein Allquantor ist einem Junktor untergeordnet.

Distributivität

- $(\forall x\, G \wedge \forall x\, H) \equiv \forall x\, (G \wedge H)$

Also

$$F \equiv \left(\bigwedge_{i=1}^{n} \forall y_1 \forall y_2 \ldots \forall y_p \left(\bigvee_{k=1}^{mi} L_{i,k} \right) \right)$$

- In dieser Darstellung ist keine Konjunktion einen Allquantor untergeordnet und kein Allquantor einer Disjunktion.
- Durch Mengenbildung können wir die Konjunktion implizit darstellen:

$$\{\, \forall y_1 \forall y_2 \ldots \forall y_p \left(\bigvee_{k=1}^{mi} L_{i,k} \right) \mid i \in \{1, \ldots, n\} \,\}$$

Ziel: Mengendarstellung der Klauselnormalform (2)

Klauselnormalform: $F = \forall y_1 \forall y_2 \ldots \forall y_p \left(\bigwedge_{i=1}^{n} \left(\bigvee_{k=1}^{mi} L_{i,k} \right) \right)$

F **ist repräsentierbar durch:** $\{ \forall y_1 \forall y_2 \ldots \forall y_p \left(\bigvee_{k=1}^{mi} L_{i,k} \right) \mid i \in \{1, \ldots, n\} \}$

- Keine Existenzquantoren, keine freien Variablen.
- Alle Variablen sind allquantifiziert.
- Alle Quantoren haben Klausel (und ggf. All-Quantoren-Präfix) als Skopus.

Reihenfolge der Allquantoren ist egal, leere Quantifikation ist unnötig

- $\forall x\ \forall y\ G \equiv \forall y\ \forall x\ G$
- Falls x in G nicht frei vorkommt gilt: $G \equiv \forall x\ G$

➔ In der Mengendarstellung kann man die Quantoren weglassen, da man alles weiß, was man zu ihrer Rekonstruktion benötigen könnte.

➔ Namensübereinstimmung von Variablen in verschiedenen Klausel ist irrelevant.

F **ist repräsentierbar durch:** $\{ \left(\bigvee_{k=1}^{mi} L_{i,k} \right) \mid i \in \{1, \ldots, n\} \}$

Mengendarstellung von Klauselnormalformen

Definition 10.19

Auch in der Prädikatenlogik werden

- atomare Formeln und ihre Negationen als Literale
- und Disjunktionen von Literalen als Klauseln bezeichnet.

Ist K eine (prädikatenlogische) Klausel, mit

$K = (\bigvee_{k=1}^{m} L_k)$, dann nennen wir

$K = \{L_1, \ldots, L_m\}$ die *Mengendarstellung* von K.

Ist F eine Klauselnormalform mit der Matrix F*, wobei

$F^* = (\bigwedge_{i=1}^{n} (\bigvee_{k=1}^{mi} L_{i,k}))$, dann nennen wir

$F = \{\{L_{1,1}, \ldots, L_{1,m1}\}, \ldots, \{L_{n,1}, \ldots, L_{n,mn}\}\}$ die *Mengendarstellung* von F.

➔ In der Mengendarstellung einer Klauselnormalform sind die Quantoren nicht mehr explizit.

Mengendarstellung von Klauselnormalformen (2)

Da in Klauselnormalformen

- keine freien Variablen und
- keine Existenzquantoren vorkommen,
- alle Allquantoren im Präfix stehen und
- die Reihenfolge von Allquantoren im Präfix semantisch unwesentlich ist,

werden alle Variablen in der Mengendarstellung behandelt, als wenn sie durch Allquantoren mit maximalem Skopus gebunden sind.

➔ das ist gleichwertig mit: Der Skopus der impliziten Allquantoren ist jeweils die Klausel

Wahrheitswertberechnung

für die Mengendarstellung von Klauselnormalformen

Klauseln

$\mathcal{A}(\mathsf{K}) =$ Maximum($\{\mathcal{A}(\mathsf{L}) \mid \mathsf{L} \in \mathsf{K}\}$)

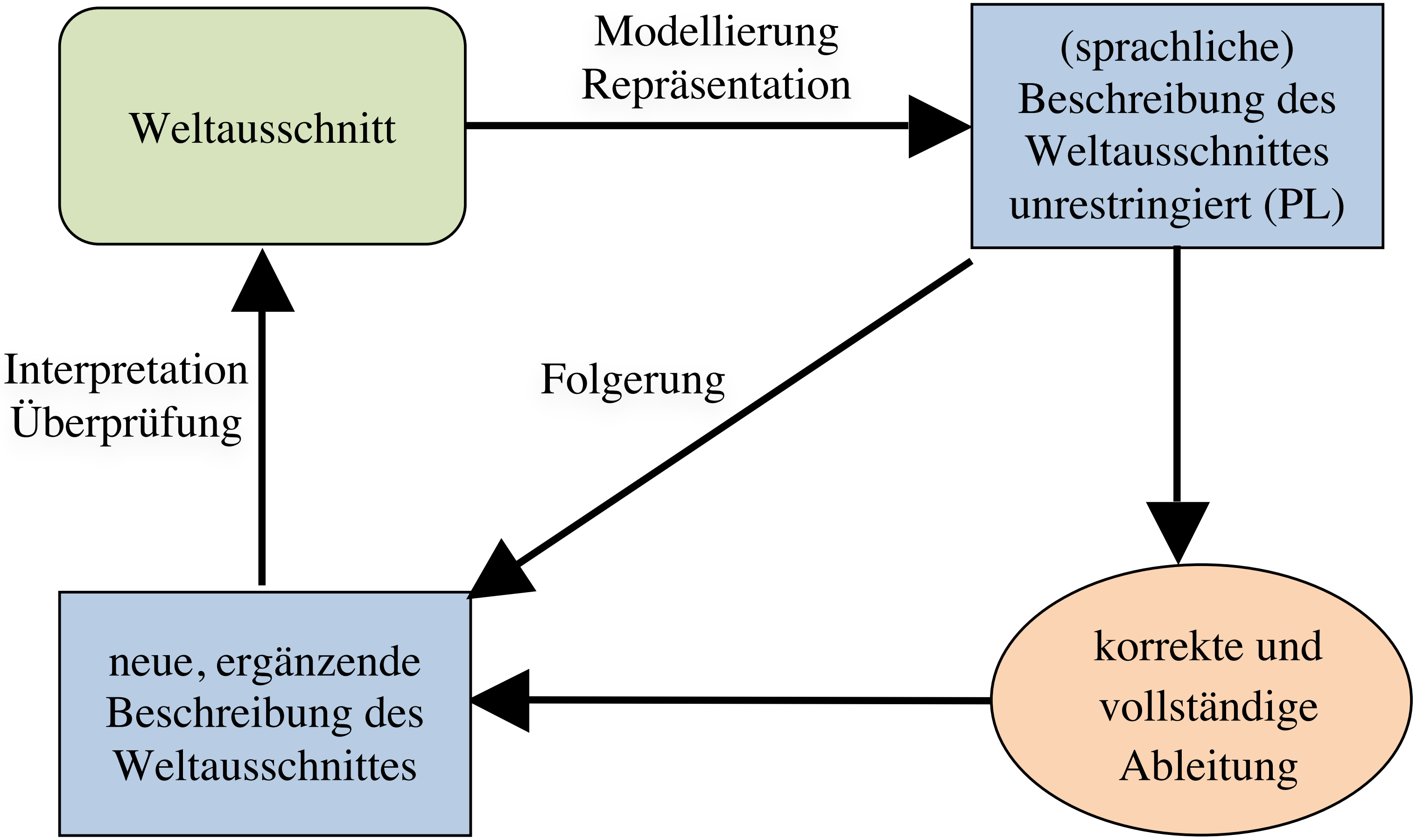
Klauselnormalformen

$\mathcal{A}(\mathsf{F}) =$ Minimum($\{\mathcal{A}'(\mathsf{K}) \mid \mathsf{K} \in \mathsf{F}$ und $\mathcal{A}'$ ist eine Struktur, die sich von $\mathcal{A}$ nur durch die Interpretation der Variablen unterscheidet$\}$)

Wichtige Konzepte dieser Vorlesung

- Äquivalenz in der Prädikatenlogik, wichtige Äquivalenzen (Dualität der Quantoren, Skopuserweiterung, Distributivität)
- Äquivalenzumformungen
- bereinigte Formeln, Pränixform, (Quantoren-)Präfix, Matrix
- Variablen-Substitution, Gebundene Umbenennung
- Erfüllbarkeitsäquivalenz (Unterschied zur Äquivalenz)
- Skolemfunktion, Skolemkonstante, Skolemisierung ($\rightarrow$ Erfüllbarkeitsäquivalalenz)
- Skolemform, Klauselnormalform, Mengendarstellung der Klauselnormalform

Modellierung, Folgerung und Ableitung in Logik



Zur Erinnerung: Resolution in der Aussagenlogik

Voraussetzung: Eingabe in KNF (Mengendarstellung)

Definition 8.1 (Resolvente)

Seien K_1 und K_2 Klauseln in Mengendarstellung und sei L ein Literal mit $L \in K_1$ und $\overline{L} \in K_2$. Dann heißt die Literalmenge $R = (K_1 - \{L\}) \cup (K_2 - \{\overline{L}\})$ *Resolvente* von K_1 und K_2 (bzgl. L). (Falls $K_1 = \{L\}$ und $K_2 = \{\overline{L}\}$, so ist die Resolvente die *leere Klausel* $\square$.)

Resolutionssatz 8.6

Eine Klauselmenge F ist genau dann unerfüllbar, wenn $\square \in \mathrm{Res}^*(F)$, d.h. $F \vdash_{res} \square$

Resolutionsalgorithmus

Eingabe: Eine Formel F in KNF (als Klauselmenge), d.h. eine **endliche** Klauselmenge !

REPEAT
    $G := F$;
    $F := \mathrm{Res}(F)$;
    UNTIL ($\square \in F$) OR ($F = G$)
IF $\square \in F$ THEN „F ist unerfüllbar“
    ELSE „F ist erfüllbar“

Bei n Aussagensymbolen gibt es maximal 4^n Klauseln.

$F = G$ Prüfung sichert den Abbruch.

Prädikatenlogik: Resolution

- Beweisverfahren, Widerlegungsverfahren für die Prädikatenlogik
  - Vorverarbeitung: Wandlung der eigentlichen Aufgabenstellung (z.B. Folgerungsprüfung) in eine Widerlegungsaufgabe
    Bildung einer (erfüllbarkeitsäquivalenten) Skolemform zu der Widerlegungsaufgabe (vgl. Kap. 10)
  - Ausgangspunkt:
    Beziehung zwischen aussagenlogischer und prädikatenlogischer Semantik
    ➔ Grundresolution
  - Erweiterung der aussagenlogischen Resolution um die Behandlung von Variablen
    ➔ Unifikation
    ➔ prädikatenlogische Resolventenbildung

➔ In dieser Vorlesung wird das Verfahren der prädikatenlogischen Resolution vorgestellt, die semantischen Hintergründe (einschließlich Korrektheits- und Vollständigkeitsbeweise) sind Gegenstand von spezialisierten Vorlesungen (insbesondere im Masterstudium).

Resolution: Von der Aussagenlogik zur Prädikatenlogik (1)

Prädikatenlogik ohne Variablen und Quantoren

- berücksichtigt die interne Struktur von Aussagen und erlaubt es, zusätzlich bestimmte Beziehungen zwischen ‘Objekten’ zum Ausdruck zu bringen.
- Engl_ersch(hp7) ≈ Harry Potter 7 ist auf englisch erschienen.
- Dt_ersch(hp7) ≈ Harry Potter 7 ist auf deutsch erschienen.
- Dt_ersch(hp7) ⇒ Engl_ersch(hp7) ≈ Wenn Harry Potter 7 auf deutsch erschienen ist, dann ist Harry Potter 7 auf englisch erschienen.
- Sei **F =** {Dt_ersch(hp7), Dt_ersch(hp7) ⇒ Engl_ersch(hp7)}
- Gilt dann: **F** $\models$ Engl_ersch(hp7) ? [Ja, ergibt sich direkt aus Def 9.8, 9.12.]
- Gilt dann: **F** $\vdash_{MP}$ Engl_ersch(hp7) ? [Ja, bei Substitution einer PL-Formel bei der Anwendung der Inferenzregeln, Def 6.5]
- Gilt dann: **F** $\vdash_{res}$ Engl_ersch(hp7) ? [Ja, wenn wir PL-Literale zulassen]

Gegeben: Dt_ersch(hp7) ∧ (Dt_ersch(hp7) ⇒ Engl_ersch(hp7))

In KNF: Dt_ersch(hp7) ∧ (¬ Dt_ersch(hp7) ∨ Engl_ersch(hp7))

Ableitbarkeit von Engl_ersch(hp7) durch Resolution?

Resolutionsableitung

Dt_ersch(hp7)   (¬ Dt_ersch(hp7) ∨ Engl_ersch(hp7))

Engl_ersch(hp7)

Da Dt_ersch(hp7) und ¬ Dt_ersch(hp7) komplementäre Literale sind.

Folgerbarkeitstest durch Resolutionsableitung: Beispiel 2

Gegeben: F_1 = Dt_ersch(hp7) ∧ (¬ Dt_ersch(hp7) ∨ Engl_ersch(hp7))

Negation von Engl_ersch(hp7): ¬ Engl_ersch(hp7)

Unerfüllbarkeitstest durch Resolution von

(Dt_ersch(hp7) (¬ Dt_ersch(hp7) ∨ Engl_ersch(hp7)) ¬ Engl_ersch(hp7)

Engl_ersch(hp7)

□

Da Dt_ersch(hp7) und ¬ Dt_ersch(hp7) sowie
Engl_ersch(hp7) und ¬ Engl_ersch(hp7)
jeweils komplementäre Literale sind.

Warum sind

- Engl_ersch(hp6) ≈ Harry Potter 6 ist auf englisch erschienen.
- Jp_ersch(hp7) ≈ Harry Potter 7 ist auf japanisch erschienen.

nicht auf entsprechende Weise als aus F nachweisbar ?

Resolution: Von der Aussagenlogik zur Prädikatenlogik (2)

Prädikatenlogik mit Quantoren

- erlaubt es, generische Aussagen über Objektbereiche zu machen, ohne Objekte einzeln zu benennen.
- Engl_ersch(hp7) ≈ Harry Potter 7 ist auf englisch erschienen.
- Dt_ersch(hp7) ≈ Harry Potter 7 ist auf deutsch erschienen.
- $\forall x$ (Dt_ersch(x) $\Rightarrow$ Engl_ersch(x)) ≈ Wenn etwas auf deutsch erschienen ist, dann ist es (auch) auf englisch erschienen.
- Sei **F** = {Dt_ersch(hp7), $\forall x$ (Dt_ersch(x) $\Rightarrow$ Engl_ersch(x))}
- Gilt dann: **F** $\vDash$ Engl_ersch(hp7) ? [Ja, ergibt sich direkt aus Def 9.8, 9.12.]
- Gilt dann: **F** $\vdash_{MP}$ Engl_ersch(hp7) ? [Nein, wir brauchen eine weitere Regel um von $\forall x$ (Dt_ersch(x) $\Rightarrow$ Engl_ersch(x)) zu Dt_ersch(hp7) $\Rightarrow$ Engl_ersch(hp7) zu kommen]
- Gilt dann: **F** $\vdash_{res}$ Engl_ersch(hp7) ? [Nicht mit der bisherigen Regel]
- Wie ist **F** $\vDash$ Engl_ersch(hp7) unter Verwendung eines angepassten Resolutionsverfahrens nachzuweisen?

Folgerbarkeitstest durch Resolutionsableitung: Beispiel 3

Eine andere Implikationsbeziehung: $\forall x\ (Dt\_ersch(x) \Rightarrow Engl\_ersch(x))$

Umformung: $\forall x\ (\neg Dt\_ersch(x) \vee Engl\_ersch(x))$

Gegeben: $F_2 = Dt\_ersch(hp7) \wedge \forall x\ (\neg Dt\_ersch(x) \vee Engl\_ersch(x))$

Negation von $Engl\_ersch(hp7)$: $\neg Engl\_ersch(hp7)$

Unerfüllbarkeitstest durch Resolution von

$(Dt\_ersch(hp7)$ $\quad \forall x\ (\neg Dt\_ersch(x) \vee Engl\_ersch(x))$ $\quad \neg Engl\_ersch(hp7)$

$(\neg Dt\_ersch(hp7) \vee Engl\_ersch(hp7))$

$Engl\_ersch(hp7)$

$\square$

➔ Die Konsequenz $(\neg Dt\_ersch(hp7) \vee Engl\_ersch(hp7))$ des All-Satzes $\forall x\ (\neg Dt\_ersch(x) \vee Engl\_ersch(x))$ wird im Verfahren benötigt, dann erhalten wir auch Komplementarität der Literale.

Exkurs: Logik ohne Variablen und Quantoren [s.a. 9-60]

Atomare Formeln der Aussagenlogik

- ungegliedert: A_1, A_2, …
- können gleich oder verschieden sein

Atomare Formeln der Prädikatenlogik ohne Variablen

- zusammengesetzt: P(a), Q(a), P(f(a)), R(a, b), R(b, a), …
- können dieselben oder verschiedene Prädikatssymbole aufweisen
- können dieselben oder verschiedene Teil-Terme haben
- dieselben Teil-Terme können in denselben oder verschiedenen Argumentpositionen der Prädikatensymbole auftreten.

Geschlossene Formeln der PL ohne Variablen / Quantoren

- Für die Bestimmung des Wahrheitswertes (einer komplexen Formel) ist nur der Wahrheitswert der atomaren Teilformeln wichtig, nicht deren Aufbau.
- Für die Erfüllbarkeit ist wichtig, ob Teilformeln gleich oder verschieden sind.
  Es ist nicht wichtig, worin Unterschiede bestehen.

➔ Mengen solcher Formeln können wie aussagenlogische Formelmengen behandelt werden

Grundsubstitution, Grundinstanz

Definition 11.1

Es sei F eine Formel in Skolemform mit Matrix F*.

- Eine Substitution, die alle freien Variablen in F* durch geschlossene (d.h. variablenfreie) Terme ersetzt, heißt *Grundsubstitution*.
- Wenn alle freien Variablen in F* durch eine Grundsubstitution ersetzt werden, heißt die resultierende Formel eine *Grundinstanz* von F bzw. F*.
- [Anm.: Wenn freie Variablen in F* durch Substitution ersetzt werden, heißt die resultierende Formel eine *Instanz* von F bzw. F*]

Beobachtung zu 11.1

Ist F eine Formel in Skolemform und G eine (Grund-)Instanz von F, dann folgt G aus F.

- Die Menge der Grundinstanzen von F spielen eine besondere Rolle für die prädikatenlogische Resolution.

Beispiel: Grundsubstitution

$F = (P(x, y, z) \vee P(z, x, z))$

$\sigma_5 = [x / g(a), z / f(a, b), y / b]$: die beiden Variablen werden parallel substituiert !

$$\begin{aligned}
\sigma_5(F) &= \sigma_5((P(x, y, z) \vee P(z, x, z))) \\
&= (\sigma_5(P(x, y, z)) \vee \sigma_5(P(z, x, z))) \\
&= (P(\sigma_5(x), \sigma_5(y), \sigma_5(z)) \vee P(\sigma_5(z), \sigma_5(x), \sigma_5(z))) \\
&= (P(g(a), b, f(a, b)) \vee P(f(a, b), g(a), f(a, b)))
\end{aligned}$$

Grundterme (Herbrand-Universum)

Definition 11.20

Sei F eine (geschlossene) Formel in Skolemform.

Die *Menge der Grundterme* (das *Herbrand-Universum*) zu F, symbolisiert als $\mathbf{G}(F)$, ist induktiv definiert durch:

1) Ist k eine in F vorkommende Konstante, dann ist $k \in \mathbf{G}(F)$.
2) Falls in F keine Konstante vorkommt, so wird eine neue Konstante a gewählt [$a \in \mathbf{G}(F)$].
3) Für jedes in F vorkommende n-stellige Funktionssymbol f und Terme $t_1, \ldots, t_n \in \mathbf{G}(F)$ ist der Term $f(t_1, \ldots, t_n)$ in $\mathbf{G}(F)$.
4) Das sind alle Elemente von $\mathbf{G}(F)$.

Beobachtungen zu Def. 11.20

- $\mathbf{G}(F)$ ist eine Menge von geschlossenen, d.h. Variablen-freien Termen.
- Alle Terme in $\mathbf{G}(F)$ sind aus den Bestandteilen von F (und ggf. der zusätzlichen Konstante a) aufgebaut.
- Alle geschlossenen Terme, die aus den Bestandteilen von F (und ggf. der zusätzlichen Konstante a) aufgebaut sind, sind in $\mathbf{G}(F)$ enthalten.
- In $\mathbf{G}(F)$ treten auch die Skolemsymbole auf.

Grundinstanzen (Herbrand-Expansion)

Definition 11.21

Sei $F = \forall y_1 \forall y_2 \ldots \forall y_n\, F^*$ eine geschlossene Formel in Skolemform [F^* die Matrix].

Für F wird die *Menge der Grundinstanzen* (*Herbrand-Expansion*) **E**(F) definiert durch:

$$\mathbf{E}(F) = \{F^*[y_1/t_1][y_2/t_2]\ldots[y_n/t_n] \mid t_1,\ldots,t_n \in \mathbf{G}(F)\}$$

- In der Matrix F^* werden alle Variablen durch Terme aus **G**(F) substituiert.
  Hierbei werden alle Substitutionsmöglichkeiten wahrgenommen.

Beispiel $F := \forall x\, \forall y\, \forall z\, P(x, f(y), g(z, x))$ $F^* = P(x, f(y), g(z, x))$

E(F) enthält

$P(a, f(a), g(a, a)) = F^*[x/a][y/a][z/a]$

$P(f(a), f(a), g(a, f(a))) = F^*[x/f(a)][y/a][z/a]$

$P(a, f(f(a)), g(a, a)) = F^*[x/a][y/f(a)][z/a]$

$P(a, f(a), g(f(a), a)) = F^*[x/a][y/a][z/f(a)]$

… …

- Die Formeln aus **E**(F) sind geschlossen und quantorenfrei.
  Sie können als – intern strukturierte – aussagenlogische Formeln aufgefasst werden.

Definition 11.3

Seien K_1, K_2 und R prädikatenlogische Klauseln von Grundinstanzen in Mengendarstellung. R heißt (Grund-)*Resolvente* von K_1 und K_2, falls gilt:

1. Es gibt ein Literal mit $L \in K_1$, so dass $\overline{L} \in K_2$ und
2. $R = (K_1 - \{L\}) \cup (K_2 - \{\overline{L}\})$

Beispiel: $F_0 = \forall x\, \forall y\, ((\neg P(x) \vee Q(y)) \wedge P(a) \wedge \neg Q(b))$

Skolemform $F = \forall x\, \forall y\, ((\neg P(x) \vee Q(y)) \wedge P(a) \wedge \neg Q(b))$

Matrix $F^* = ((\neg P(x) \vee Q(y)) \wedge P(a) \wedge \neg Q(b))$

Grundinstanzen aus **E**(F) mit Instanziierungen über $\{a, b\}$, den Konstanten in F:

$$F_1 = ((\neg P(a) \vee Q(a)) \wedge P(a) \wedge \neg Q(b))$$
$$F_2 = ((\neg P(a) \vee Q(b)) \wedge P(a) \wedge \neg Q(b))$$
$$F_3 = ((\neg P(b) \vee Q(a)) \wedge P(a) \wedge \neg Q(b))$$
$$F_4 = ((\neg P(b) \vee Q(b)) \wedge P(a) \wedge \neg Q(b))$$

Grundresolution [Forts. des Beispiels]

$F_1 = \{ \{\neg P(a), Q(a)\}, \{P(a)\}, \{\neg Q(b)\} \}$

$\{Q(a)\}$

$F_2 = \{ \{\neg P(a), Q(b)\}, \{P(a)\}, \{\neg Q(b)\} \}$

$\{\neg P(a)\}$

$\square$

$F_3 = \{ \{\neg P(b), Q(a)\}, \{P(a)\}, \{\neg Q(b)\} \}$

$F_4 = \{ \{\neg P(b), Q(b)\}, \{P(b)\}, \{\neg Q(b)\} \}$

$\{\neg P(b)\}$

Gültigkeitsprüfung mit Hilfe von Grundresolution

Ist $H = \neg\exists y\, \forall z\, [P(z, y) \Leftrightarrow \neg\exists x\, (P(z, x) \wedge P(x, z))]$ gültig?

➔ ist $\neg H \equiv F = \exists y\, \forall z\, [P(z, y) \Leftrightarrow \neg\exists x\, (P(z, x) \wedge P(x, z))]$ unerfüllbar?

Umformungen zur Erstellung einer Klauselnormalform für F:

$F = \exists y\, \forall z\, [P(z, y) \Leftrightarrow \neg\exists x\, (P(z, x) \wedge P(x, z))]$

Biimplikation eliminieren

$\equiv \exists y\, \forall z\, [(\neg P(z, y) \vee \neg\exists x\, (P(z, x) \wedge P(x, z))) \wedge (P(z, y) \vee \exists x\, (P(z, x) \wedge P(x, z)))]$

Negation nach innen

$\equiv \exists y\, \forall z\, [(\neg P(z, y) \vee \forall x\, (\neg P(z, x) \vee \neg P(x, z))) \wedge (P(z, y) \vee \exists x\, (P(z, x) \wedge P(x, z)))]$

[x/w] Umbenennung

$\equiv \exists y\, \forall z\, [(\neg P(z, y) \vee \forall x\, (\neg P(z, x) \vee \neg P(x, z))) \wedge (P(z, y) \vee \exists w\, (P(z, w) \wedge P(w, z)))]$

Pränexform

$\equiv \exists y\, \forall z\, \exists w\, \forall x\, [(\neg P(z, y) \vee (\neg P(z, x) \vee \neg P(x, z))) \wedge (P(z, y) \vee (P(z, w) \wedge P(w, z)))]$

Skolemisierung [y/a]

erfäqui $\forall z\, \exists w\, \forall x\, [(\neg P(z, a) \vee (\neg P(z, x) \vee \neg P(x, z))) \wedge (P(z, a) \vee (P(z, w) \wedge P(w, z)))]$

Skolemisierung [$w/f(z)$]

erfäqui $\forall z\, \forall x\, [(\neg P(z, a) \vee (\neg P(z, x) \vee \neg P(x, z))) \wedge (P(z, a) \vee (P(z, f(z)) \wedge P(f(z), z)))]$

KNF-Erstellung

$\equiv \forall z\, \forall x\, [(\neg P(z, a) \vee \neg P(z, x) \vee \neg P(x, z)) \wedge (P(z, a) \vee P(z, f(z))) \wedge (P(z, a) \vee P(f(z), z))]$

Gültigkeitsprüfung mit Hilfe von Grundresolution [Fortsetzung]

Matrix: $F^* = C_1 \wedge C_2 \wedge C_3$

$C_1 = (\neg P(z, a) \vee \neg P(z, x) \vee \neg P(x, z))$

$C_2 = (P(z, a) \vee P(z, f(z)))$

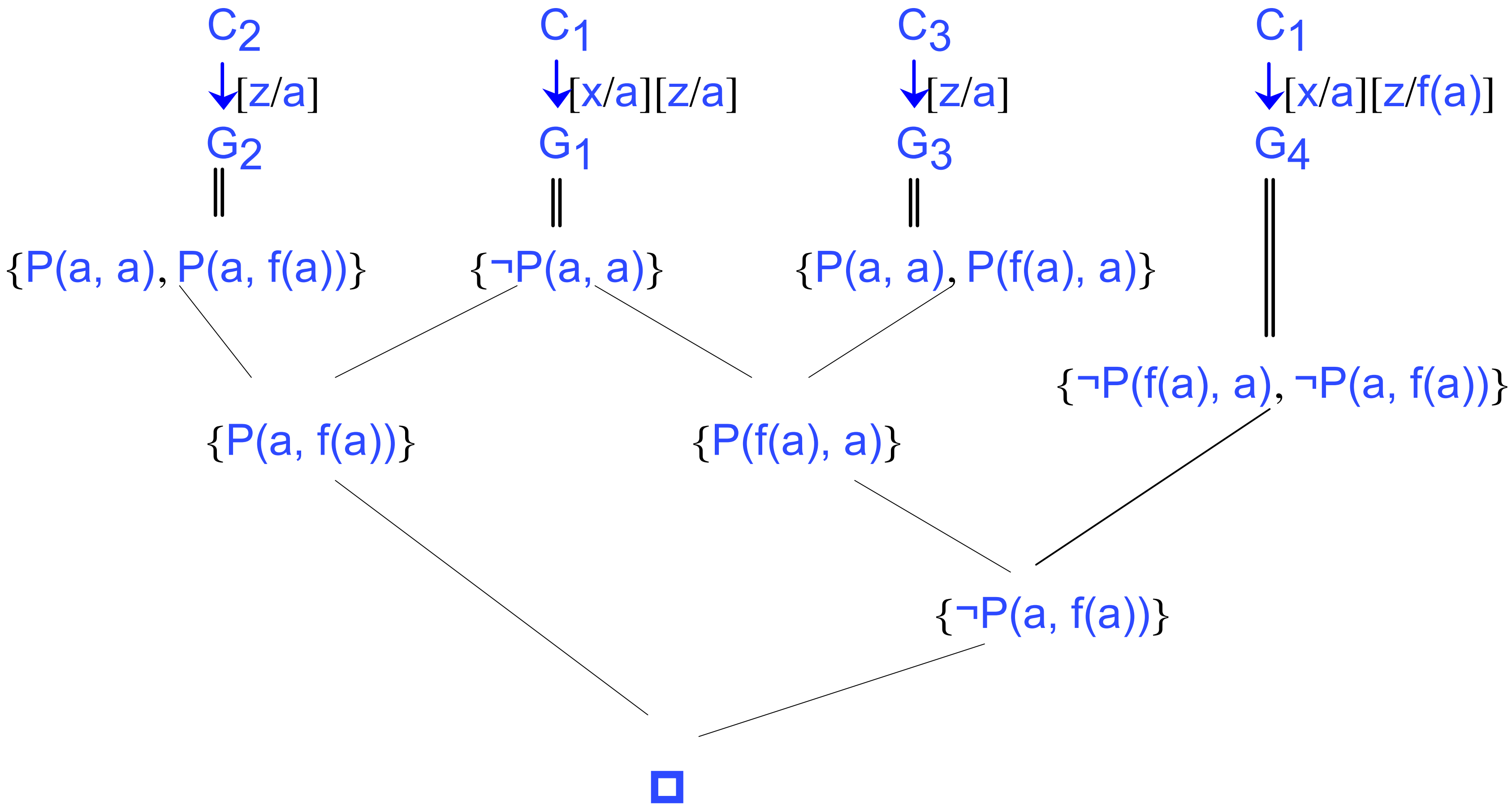
$C_3 = (P(z, a) \vee P(f(z), z))$

[Bei der Bildung der Grundinstanzen können die Klauseln individuell betrachtet werden, da die Konjunktion bei der Mengendarstellung sowieso aufgelöst wird.]

Einige Möglichkeiten der Bildung von Grundinstanzen zu den Klauseln (auf der Basis der Konstanten von F^*)

Klausel	Grundsubstitution	Grundinstanz der Klausel	Mengendarstellung
C_1	[x/a][z/a]	$G_1 = \neg P(a, a) \vee \neg P(a, a) \vee \neg P(a, a)$	$\{\neg P(a, a)\}$
C_2	[z/a]	$G_2 = P(a, a) \vee P(a, f(a))$	$\{P(a, a), P(a, f(a))\}$
C_3	[z/a]	$G_3 = P(a, a) \vee P(f(a), a)$	$\{(P(a, a), P(f(a), a))\}$
C_1	[x/a][z/f(a)]	$G_4 = \neg P(f(a), a) \vee \neg P(f(a), a) \vee \neg P(a, f(a))$	$\{\neg P(f(a), a), \neg P(a, f(a))\}$

Gültigkeitsprüfung mit Hilfe von Grundresolution [Fortsetzung]



➔ F ist unerfüllbar.

➔ H ist gültig.

Grundresolution: Zwischenfazit

Das Schema der Grundresolution

Klauseln	C_i , …	C_k
Grundsubstitutionen	$\downarrow$ $[x_i/t_i]$, …	$\downarrow$ $[x_k/t_k]$, …
Grundinstanzen zu einigen Klauseln	G_i	G_k
Resolution	$\square$	

- Zu zwei aussagenlogischen Klauseln gibt es stets nur endlich viele Resolventen.
- Zu zwei prädikatenlogischen Klauseln kann es unendlich viele Grundinstanzen geben, die die Bildung von (unendlich vielen) Resolventen erlauben.
- Der Aufwand des Grundresolutionsverfahrens ergibt sich durch den Aufwand bei der Suche nach geeigneten Grundinstanzen.
- Wünschenswerte Eigenschaft prädikatenlogischer Resolution:
  Für jedes Klauselpaar: Beschränkung auf eine endliche Menge von Resolventen, so dass im Gesamtverfahren die Ableitbarkeit der leeren Klausel gegenüber der Grundresolution nicht eingeschränkt wird.

Von der aussagenlogischen Resolventenbildung zur prädikatenlogischen Resolventenbildung

- Grundresolutionsverfahren der Prädikatenlogik separiert zwei Aspekte
  - Bildung der Grundinstanzen:
    reduziert prädikatenlogische Ausdrücke zu aussagenlogischen Ausdrücken
  - Resolution: ist ein Verfahren für aussagenlogische Ausdrücke
- Der Kern des Resolutionsprinzips:
  Resolvieren von Klauseln mit komplementären Literalen:
  - Aussagenlogik: z.B. $\{(\neg P \vee Q), P, \neg Q\} \vdash_{res} Q$
    und $\{(\neg P \vee Q), P, \neg Q, Q\} \vdash_{res} \square$
  - Prädikatenlogik: z.B. $\forall x\, \forall y\, ((\neg P(x) \vee Q(y)) \wedge P(a) \wedge \neg Q(b))$
    - In welchen Sinne sind $\neg P(x)$ und $P(a)$ komplementär?
    - Wie können etwa $\neg P(x)$ und $P(a)$ komplementär gemacht werden?
      ➔ z.B. durch geeignete Grundsubstitution.
- Prädikatenlogik allgemein:
  - In welchen Sinne sind $\neg P(x)$ und $P(y)$, bzw. $\neg Q(x)$ und $Q(f(y))$ komplementär?
  - Was ist die Resolvente zweier prädikatenlogischen Klauseln?

Von der Grundresolution zur prädikatenlogischen Resolution

- Statt Grundsubstitutionen andere Arten geeigneter Substitutionen, denn
  - Grundsubstitutionen sind sehr speziell (Festlegung auf individuelle geschlossene Terme)
  - Grundsubstitutionen erzeugen viele Grundinstanzen, die später in der Resolution nicht verwendet werden.
- Ziel: „zurückhaltende“ Substitutionen:

Beispiel: $\{P(x), \neg Q(g(x))\}$ $\qquad$ $\{\neg P(f(y))\}$

$$\downarrow [x / f(y)]$$

$$\{P(f(y)), \neg Q(g(f(y)))\} \quad \{\neg P(f(y))\}$$

$\downarrow$ (prädikatenlogische) Resolution

$$\{\neg Q(g(f(y)))\}$$

➔ keine Festlegung im Hinblick auf y.

- *Unifikation* / *unifizieren*: vereinigen, zusammenführen.

Definition 11.5

Eine Substitution σ ist genau dann ein *Unifikator* einer endlichen Menge von Ausdrücken (Termen $t_1, \dots, t_n$, bzw. Formeln $F_1, \dots, F_k$), wenn durch σ alle Ausdrücke dieser Menge auf denselben Ausdruck abgebildet werden.

D.h. $t_1\sigma = \dots = t_n\sigma$ bzw. $F_1\sigma = \dots = F_k\sigma$

- Wichtiger Spezialfall: Die Unifikation von Literalen
- Terme sind nur mit Termen unifizierbar, Formeln nur mit Formeln

Eine Menge von Ausdrücken ist *unifizierbar*, falls es einen Unifikator für diese Menge gibt.

- Beispiel: $\{x, f(y)\}$ bzw. $\{P(x), P(f(y))\}$

  $\sigma = [x / f(y)]$

  $x\sigma = f(y) = f(y)\sigma$ bzw. $P(x)\sigma = P(x\sigma) = P(f(y)) = P(f(y))\sigma$

- Anmerkung zur Definition bei U. Schöning:
  Schöning fokussiert auf den Fall der Unifikation von Literalen.

Komposition von Substitutionen

Definition 11.6

- Es seien x und y Variablen und t_1 und t_2 Terme.
- $[x/t_1]\,[y/t_2]$ bezeichnet die Substitution, die zuerst jedes freie Vorkommen von x durch t_1 ersetzt und anschließend jedes freie Vorkommen von y durch t_2 ersetzt.
- Seien σ und τ Substitutionen. Die *Komposition* von σ und τ ist die Substitution $\sigma\tau$, für die gilt: $x(\sigma\tau) = (x\sigma)\tau$, d.h.: $\sigma\tau(x) = \tau(\sigma(x))$, für alle Variablen x.

Beispiel

$F = P(x) \wedge Q(f(y))$

$F_{[x/a]} =$	$P(a) \wedge Q(f(y))$	$F_{[x/y]} =$	$P(y) \wedge Q(f(y))$	$F_{[x/g(y)]} =$	$P(g(y)) \wedge Q(f(y))$
$F_{[x/a][y/b]} =$	$P(a) \wedge Q(f(b))$	$F_{[x/y][y/a]} =$	$P(a) \wedge Q(f(a))$	$F_{[x/g(y)][y/a]} =$	$P(g(a)) \wedge Q(f(a))$

Die Relation *allgemeiner als* zwischen Substitutionen (Unifikatoren)

Definition 11.7 Seien σ_1 und σ_2 zwei Substitutionen. σ_2 ist *allgemeiner als* σ_1, falls es eine Substitution τ gibt, für die gilt: $\sigma_1 = \sigma_2\tau$.

Beispiel: $\sigma_1 = [x/ f(g(a, g(w, y)))]\ [y/ g(h(w), b))]\ [z/ h(w)]$

$\sigma_2 = [x/ f(g(x, y))]\ [y/ g(z, b)]$

$\tau = [x/ a]\ [y/ g(w, y)]\ [z/ h(w)]$

denn: $\sigma_2\tau = [x/ f(g(x, y))]\ [y/ g(z, b)]\ [x/a]\ [y/ h(z)]\ [z/ h(w)]$

	x	y	z
σ_2	$f(g(x, y))$	$g(z, b)$	
τ	$\downarrow [x/ a]$ $\searrow [y/ g(w, y)]$	$\downarrow [z/ h(w)]$	$\downarrow [z/ h(w)]$
$\sigma_2\tau$	$f(g(a, g(w, y)))$	$g(h(w), b)$	$h(w)$

➔ σ_2 ist *allgemeiner als* σ_1 im folgenden Sinne: Es ist möglich erst σ_2 auszuführen und dann durch eine weitere Substitution, τ, den Effekt zu erreichen, den σ_1 in einem Schritt erzielt.

Zum Selbststudium

Satz 11.8

(i) Seien σ_1, σ_2 und σ_3 Substitutionen. Wenn σ_3 *allgemeiner* ist als σ_2 und σ_2 *allgemeiner* ist als σ_1, dann ist σ_3 *allgemeiner* als σ_1. [Transitivität]

(ii) Für alle Substitutionen σ gilt: σ ist *allgemeiner* als σ. [Reflexivität]

(iii) Seien σ_1 und σ_2 Substitutionen. Wenn σ_1 *allgemeiner* ist als σ_2 und σ_2 *allgemeiner* ist als σ_1, dann gilt für alle Variablen x: $x\sigma_1$ und $x\sigma_2$ unterscheiden sich nur in der Wahl von Variablen.

Beispiel für 11.8.iii

$\sigma_1 = [x/\ f(y)][z\ /\ g(b,\ y)]$

$\sigma_2 = [x/\ f(u)][z\ /\ g(b,\ u)]$

mit $\tau_1 = [y/\ u]$ und $\tau_2 = [u/\ y]$ gilt $\sigma_1 = \sigma_2\ \tau_2$ und $\sigma_2 = \sigma_1\ \tau_1$

Allgemeinster Unifikator

Definition 11.9

Ein Unifikator σ einer endlichen Menge von Ausdrücken (Termen $t_1, \ldots, t_n$, bzw. Formeln $F_1, \ldots, F_k$) ist genau dann ein *allgemeinster Unifikator*, wenn er allgemeiner ist als alle Unifikatoren dieser Menge.

- Allgemeinster Unifikator σ einer Menge von Ausdrücken **M**:

$$\begin{tikzcd} \mathbf{M} \arrow[r, "\sigma"] \arrow[dr, "\sigma'"'] & \mathbf{M}\sigma \arrow[d, "\tau"] \\ & \mathbf{M}\sigma\tau = \mathbf{M}\sigma' \end{tikzcd}$$

- allgemeinster Unifikator – *most general unifier* – MGU
- Wenn σ ein allgemeinster Unifikator einer Menge **L** = $\{L_1, \ldots, L_k\}$ von Literalen ist, dann werden alle Literale aus **L** durch σ auf $L_1\sigma = \ldots = L_k\sigma$ abgebildet.

  Derartige $L_i\sigma$ sind die Formeln, die Ausgangspunkt der prädikatenlogischen Resolution sind.

Satz 11.10

Jede endliche unifizierbare Menge von Literalen besitzt einen allgemeinsten Unifikator.

Konstruktiver Beweis

- Durch die Angabe eines Unifikationsalgorithmus

  Die Idee:

  Sukzessive Konstruktion eines Unifikators

  - Ausgangspunkt: $\mathbf{L} = \{ L_1, \dots , L_n \}$
  - $\mathbf{L}$ ist noch nicht unifiziert, wenn $\mathbf{L}\sigma = \{ L_1\sigma , \dots , L_n\sigma \}$ mehr als ein Element besitzt:

    $|\, \mathbf{L}\sigma \,| = 1$ bedeutet $\mathbf{L}$ wird durch σ unifiziert.

    $|\, \mathbf{L}\sigma \,| > 1$ bedeutet $\mathbf{L}$ wird durch σ nicht unifiziert.
  - Sukzessive Konstruktion von Substitutionen σ_i:
    - Jede zusätzliche Substitution unifiziert eine Abweichung zwischen Termen
    - $|\, \mathbf{L}\sigma_1 \dots \sigma_i \,|$ wird immer kleiner. Ziel: $|\, \mathbf{L}\sigma_1 \dots \sigma_i \,| = 1$.

➔ Unifikation bezieht sich stets auf endliche Mengen.

Unifikationsalgorithmus

Eingabe: Eine nicht-leere (endliche) Menge **L** von Literalen

i := 0; σ_0:=[ ]; (≈ leere Substitution, jede Variable wird auf sich selbst abgebildet)

while | **L**σ_i | > 1 **do**

begin Suche in **L**σ_i (von links nach rechts) die erste Position, in der sich mindestens zwei Literale L_j und L_k unterscheiden.

  if Der Unterschied wird durch Junktor (Negation) oder Prädikatensymbole hervorgerufen

  then stoppe mit der Ausgabe „nicht unifizierbar"

  else **if** keiner der unterschiedlichen Terme ist eine Variable

    then stoppe mit der Ausgabe „nicht unifizierbar"

    else **begin** Sei x die Variable und t der andere Term

      if x kommt in t (als echter Teilterm) vor

      then stoppe mit der Ausgabe „nicht unifizierbar"

      else σ_{i+1}:= σ_i [x/t]; i := i+1;

    end;

end;

Gib σ_i als allgemeinsten Unifikator aus!

Unifikationsalgorithmus (Beispiel 1)

$\mathbf{L} = \{Q(x, f(a)), Q(b, y)\}$ $\sigma_0 = [\ ]$ $|\mathbf{L}\sigma_0| = 2$

1. $L_1 = Q(x, f(a))$ $L_2 = Q(b, y)$
   $t_1 = x$ ↑ $t_2 = b$ ↑ Paar unterschiedlicher Terme
   $\sigma_1 = [x/b]$

2. $L_1\sigma_1 = Q(b, f(a))$ $L_2\sigma_1 = Q(b, y)$ $|\mathbf{L}\sigma_1| = 2$
   $t_1 = f(a)$ ↑ $t_2 = y$ ↑ Paar unterschiedlicher Terme
   $\sigma_2 = [x/b]\ [y/ f(a)]$

3. $L_1\sigma_2 = Q(b, f(a))$ $L_2\,\sigma_2 = Q(b, f(a))$ $|\mathbf{L}\sigma_2| = 1$

➔ Die Literalmenge **L** wurde erfolgreich unifiziert.
➔ allgemeinster Unifikator: $[x/b]\ [y/ f(a)]$

Unifikationsalgorithmus (Beispiel 2)

1. $\mathbf{L} = \{Q(x, f(a)), P(b, y)\}$ $\qquad \sigma_0 = [\,]$ $\qquad |\,\mathbf{L}\sigma_0\,| = 2$

   $L_1 = Q(x, f(a))$ $\qquad L_2 = P(b, y)$

   unterschiedliche Prädikatensymbole

➔ Die Literalmenge **L** ist nicht unifizierbar.

2. $\mathbf{L} = \{Q(x, f(a)), Q(b, f(x))\}$

   $L_1 = Q(x, f(a))$ $\qquad L_2 = Q(b, f(x))$ $\qquad |\,\mathbf{L}\sigma_0\,| = 2$

   $t_1 = x$ $\qquad t_2 = b$ Paar unterschiedlicher Terme

   $\sigma_1 = [x/b]$

   $L_1\sigma_1 = Q(b, f(a))$ $\qquad L_2\sigma_1 = Q(b, f(b))$ $\qquad |\,\mathbf{L}\sigma_1\,| = 2$

   $t_1 = a$ $\qquad t_2 = b$ Paar unterschiedlicher Terme

   aber variablenfrei.

➔ Die Literalmenge **L** ist nicht unifizierbar.

Unifikationsalgorithmus (Beispiel 3)

$\mathbf{L} = \{Q(x, x), Q(y, f(y))\}$

$L_1 = Q(x, x)$ $L_2 = Q(y, f(y))$ $|\mathbf{L}\sigma_0| = 2$

$t_1 = x$ $t_2 = y$ Paar unterschiedlicher Terme

$\sigma_1 = [x/y]$

$L_1\sigma_1 = Q(y, y)$ $L_2\sigma_1 = Q(y, f(y))$ $|\mathbf{L}\sigma_1| = 2$

$t_1 = y$ $t_2 = f(y)$ Paar unterschiedlicher Terme

y kommt als echter Teilterm in $f(y)$ vor.

➔ Die Literalmenge **L** ist nicht unifizierbar.

Zum Selbststudium: Unifikationsalgorithmus (Beispiel 4)

$\mathbf{L} = \{\neg P(f(z, g(a, y)), h(z)), \neg P(f(f(u, v), w), h(f(a, b)))\}$

$\sigma_0 = [\ ]$ $\qquad\qquad$ $|\mathbf{L}\sigma_0| = 2$

1. $L_1 = \neg P(f(z, g(a, y)), h(z))$ $\qquad$ $L_2 = \neg P(f(f(u, v), w), h(f(a, b)))$

   $t_1 = z$ $\qquad$ $t_2 = f(u, v)$

   $\sigma_1 = [z/\ f(u, v)]$

2. $L_1\sigma_1 = \neg P(f(f(u, v), g(a, y)), h(f(u, v)))$ $\qquad$ $L_2\sigma_1 = \neg P(f(f(u, v), w), h(f(a, b)))$

   $t_1 = g(a, y)$ $\qquad$ $t_2 = w$

   $\sigma_2 = [z/\ f(u, v)]\ [w/\ g(a, y)]$

3 $L_1\sigma_2 = \neg P(f(f(u, v), g(a, y)), h(f(u, v)))$ $\qquad$ $L_2\sigma_2 = \neg P(f(f(u, v), g(a, y)), h(f(a, b)))$

   $t_1 = u$ $\qquad$ $t_2 = a$

   $\sigma_3 = [z/\ f(u, v)]\ [w/\ g(a, y)]\ [u/a]$

4. $L_1\sigma_3 = \neg P(f(f(a, v), g(a, y)), h(f(a, v)))$ $\qquad$ $L_2\sigma_3 = \neg P(f(f(a, v), g(a, y)), h(f(a, b)))$

   $t_1 = v$ $\qquad$ $t_2 = b$

   $\sigma_4 = [z/\ f(u, v)]\ [w/\ g(a, y)]\ [u/a]\ [v/b]$

5. $L_1\sigma_4 = \neg P(f(f(a, b), g(a, y)), h(f(a, b)))$ $\qquad$ $L_2\sigma_4 = \neg P(f(f(a, b), g(a, y)), h(f(a, b)))$

   $|\mathbf{L}\sigma| = 1$ $\qquad$ ➔ Die Literalmenge **L** wurde erfolgreich unifiziert.

Unifikationsalgorithmus (Termination und Korrektheit)

Termination

- Eingabe: Eine endliche Menge **L** = { L_1, … , L_n} von Literalen. Die Gesamtzahl der auftretenden Variablen ist dann auch endlich.

➔ In jedem Schritt wird
  - entweder eine Variable gefunden, die durch eine Substitution ersetzt wird (somit wird die Anzahl der Variablen in **L**σ_i kleiner),
  - oder es wird ein Paar nicht unifizierbarer Literale gefunden, und somit das Verfahren abgebrochen.

➔ Die Zahl der Schleifendurchläufe ist also durch die Gesamtzahl der in **L** auftretenden Variablen beschränkt.

Korrektheit

- Die while-Schleife wird nur dann erfolgreich verlassen, wenn | **L**σ_i | = 1, d.h. wenn ein Unifikator gefunden ist.
- Noch zu zeigen
  - Die gebildete Substitution σ ist ein allgemeinster Unifikator.
  - Wenn **L** unifizierbar ist, dann wird die while-Schleife erfolgreich durchlaufen

Zwischenfazit: Unifikation für die Resolution

Zurückhaltende Substitution

Beispiel: $\{P(x), \neg Q(g(x))\}$ $\{\neg P(f(y))\}$

$\downarrow$ $[x / f(y)]$

$\{P(f(y)), \neg Q(g(f(y)))\}$ $\{\neg P(f(y))\}$

$\downarrow$ (prädikatenlogische) Resolution

$\{\neg Q(g(f(y)))\}$

➔ keine Festlegung im Hinblick auf y.

Satz 11.10

Jede endliche unifizierbare Menge von Literalen besitzt einen allgemeinsten Unifikator.

Unifikationsalgorithmus

- Unifikatoren für unifizierbare Literalmenge sind (leicht) bestimmbar.
- Unifikatoren sind (bis auf Variablenbenennung) eindeutig bestimmt

➔ Resolution basierend auf Unifikation (im Gegensatz zu Grundsubstitution) liefert bei endlichen Klauselmengen nur endlich viele Resolventen.

Zum Selbststudium: Algorithmus liefert allgemeinsten Unifikator

Satz 11.11

Wenn **L** eine unifizierbare Menge von Literalen ist, dann ist der durch den Unifikationsalgorithmus konstruierte Unifikator σ ein allgemeinster Unifikator für **L**.

D.h.: Ist σ' ein Unifikator von **L**, dann gibt es eine Substitution τ mit: $\sigma' = \sigma\,\tau$

Hilfssatz 11.12 (Verallgemeinerung von 11.11)

Ist σ' ein Unifikator von **L**, dann gilt für jede im Unifikationsalgorithmus gebildete Substitution σ_i: $\sigma' = \sigma_i\,\sigma'$.

Beweis

Induktionsverankerung: $\sigma' = [\,]\,\sigma' = \sigma_0\,\sigma'$

Induktionsvoraussetzung: Es sei σ_i eine Substitution mit $\sigma' = \sigma_i\,\sigma'$

Induktionsschritt: $\sigma_{i+1} := \sigma_i\,[x/t]$, wobei x eine Variable und t ein Term ist und x und t in korrespondierenden Positionen in Literalen von $\mathbf{L}\sigma_i$ auftauchen, also (nach Induktionsvoraussetzung) durch σ' unifizierbar sind. Das heißt: $x\sigma' = t\sigma'$.

Für x ist $x\,[x/t]\,\sigma' = t\,\sigma' = x\,\sigma'$.

Für jede Variable $y \neq x$ ist $y\,[x/t]\,\sigma' = y\,\sigma'$, da x in y nicht vorkommt.

Also ist $[x/t]\,\sigma' = \sigma'$ und damit ist $\sigma_{i+1}\,\sigma' = \sigma_i\,[x/t]\,\sigma' = \sigma_i\,\sigma' = \sigma'$

Hilfssatz 11.13

Ist σ' ein Unifikator von **L**, dann wird die Schleife erfolgreich durchlaufen bis $|\,\mathbf{L}\sigma_i\,| = 1$.

Beweis

Da in jedem Durchlauf $\mathbf{L}\sigma_i$ durch σ' unifizierbar ist, kann der im Algorithmus gefundene Unterschied zwischen den Literalen nur in den Termen liegen.
Weiterhin muss mindestens einer der Terme eine Variable x sein (sonst haben wir abweichende Funktionssymbole / Konstanten und die Terme sind nicht unifizierbar oder wir haben den Unterschied noch nicht genau lokalisiert), den anderen Term nennen wir t.
Weiterhin muss gelten: $x\sigma' = t\sigma'$.
Wäre nun x ein echter Teilterm von t, dann wäre $x\sigma'$ auch ein echter Teilterm von $t\sigma'$, die beiden also nicht identisch. Also kommt x nicht in t vor und der Algorithmus bricht auch bei der dritten Möglichkeit nicht ab.

Prädikatenlogische Resolution: Resolventenbildung

Definition 11.14

Seien K_1, K_2 und R prädikatenlogische Klauseln in Mengendarstellung. R heißt *prädikatenlogische Resolvente* von K_1 und K_2 ($\{ K_1, K_2 \} \vdash_{res} R$) falls gilt:

1. Es gibt Variablenumbenennungen (Substitutionen) σ_1 und σ_2, so dass $K_1\sigma_1$ und $K_2\sigma_2$ keine gemeinsamen Variablen aufweisen.
2. Es gibt eine nicht leere Menge von Literalen $\{L_1, \ldots, L_m\}$ aus $K_1\sigma_1$ und eine nicht leere Menge von Literalen $\{L'_1, \ldots, L'_n\}$ aus $K_2\sigma_2$, so dass $\{ \overline{L_1}, \ldots, \overline{L_m}, L'_1, \ldots, L'_n\}$ mit dem allgemeinsten Unifikator σ unifizierbar ist.
3. $R = ((K_1\sigma_1 - \{ L_1, \ldots, L_m \}) \cup (K_2\sigma_2 - \{ L'_1, \ldots, L'_n \}))\, \sigma$

- Nach Konstruktion gilt auch: $R = (K_1\sigma_1\sigma - \{ L_1\}\sigma) \cup (K_2\sigma_2\sigma - \{ \overline{L_1} \}\sigma)$
- Die aussagenlogische Resolventenbildung ist ein Spezialfall der prädikatenlogischen Resolventenbildung (da keine Variablen auftreten):

➔ $\sigma_1 = \sigma_2 = \sigma = [\,]$

➔ $m = n = 1$

Prädikatenlogische Resolution – Beispiel zu 11.14 (1)

$K_1 = \{P(f(x)), \neg Q(z), P(z)\}$ $K_2 = \{\neg P(x), R(g(x), a)\}$

Auswahl von Literalen für die Resolution

$\{P(f(x)), \neg Q(z), P(z)\}$ $\{\neg P(x), R(g(x), a)\}$

(keine gemeinsamen Variablen)

$\sigma_1 = [\,]$ $\sigma_2 = [x/u]$

$\{P(f(x)), \neg Q(z), P(z)\}$ $\{\neg P(u), R(g(u), a)\}$

(Unifikation)

$\sigma = [z/\ f(x)][u/\ f(x)]$

$\{P(f(x)), \neg Q(f(x))\}$ $\{\neg P(f(x)), R(g(f(x)), a)\}$

(Resolution)

$\{\neg Q(f(x)), R(g(f(x)), a)\}$

Prädikatenlogische Resolution – Beispiel zu 11.14 (2)

$K_1 = \{P(f(x)), \neg Q(z), P(z)\}$ $\quad\quad$ $K_2 = \{\neg P(x), R(g(x), a)\}$

Auswahl von Literalen für die Resolution

$\{P(f(x)), \neg Q(z), P(z)\}$ $\quad\quad$ $\{\neg P(x), R(g(x), a)\}$

(keine gemeinsamen Variablen)

$\sigma_1 = [\,]$ $\quad\quad$ $\sigma_2 = [x/u]$

$\{P(f(x)), \neg Q(z), P(z)\}$ $\quad\quad$ $\{\neg P(u), R(g(u), a)\}$

(Unifikation)

$\sigma = [u/ z]$

$\{P(f(x)), \neg Q(z), P(z)\}$ $\quad\quad$ $\{\neg P(z), R(g(z), a)\}$

(Resolution)

$\{P(f(x)), \neg Q(z), R(g(z), a)\}$

Prädikatenlogische Resolution – Beispiel zu 11.14 (3)

$K_1 = \{P(f(x)), \neg Q(z), P(z)\}$ $K_2 = \{\neg P(x), R(g(x), a)\}$

Auswahl von Literalen für die Resolution

$\{P(f(x)), \neg Q(z), P(z)\}$ $\{\neg P(x), R(g(x), a)\}$

(keine gemeinsamen Variablen)

$\sigma_1 = [\,]$ $\sigma_2 = [x/u]$

$\{P(f(x)), \neg Q(z), P(z)\}$ $\{\neg P(u), R(g(u), a)\}$

(Unifikation)

$\sigma = [u/ f(x)]$

$\{P(f(x)), \neg Q(z), P(z)\}$ $\{\neg P(f(x)), R(g(f(x)), a)\}$

(Resolution)

$\{\neg Q(z), P(z), R(g(f(x)), a)\}$

Kurzdarstellung der Beispiele zu 11.14

Anstelle der ausführlichen Resolutionsgraphen oben verwenden wir im folgenden die Kurzdarstellung:

$\{P(f(x)), \neg Q(z), P(z)\}$ $\qquad$ $\{\neg P(x), R(g(x), a)\}$

$[z/ f(x)]$ $\qquad$ $[x/ f(x)]$

$\{\neg Q(f(x)), R(g(f(x)), a)\}$

$\{P(f(x)), \neg Q(z), P(z)\}$ $\qquad$ $\{\neg P(x), R(g(x), a)\}$

$[\ ]$ $\qquad$ $[x/z]$

$\{P(f(x)), \neg Q(z), R(g(z), a)\}$

$\{P(f(x)), \neg Q(z), P(z)\}$ $\qquad$ $\{\neg P(x), R(g(x), a)\}$

$[\ ]$ $\qquad$ $[x/f(x)]$

$\{\neg Q(z), P(z), R(g(f(x)), a)\}$

Beobachtung zu Beispiel zu 11.14

Abgesehen von Varianten in der Variablenbenennung haben $K_1 = \{P(f(x)), \neg Q(z), P(z)\}$ und $K_2 = \{\neg P(x), R(g(x), a)\}$ drei prädikatenlogische Resolventen

- $\{\neg Q(f(x)), R(g(f(x)), a)\}$
- $\{\neg Q(z), P(z), R(g(f(x)), a)\}$
- $\{P(f(x)), \neg Q(z), R(g(z), a)\}$

(Welche Resolvente bei der Ableitung der leeren Klausel gebildet werden muss, hängt von den anderen beteiligten Klauseln ab.)

Satz 11.15

Seien K_1 und K_2 prädikatenlogische Klauseln in Mengendarstellung. Abgesehen von Variationen in der Variablenbenennung gibt es für K_1 und K_2 nur endlich viele prädikatenlogische Resolventen.

Beweisidee

Die Resolvente wird eindeutig durch die Klauseln und die Wahl der Literalmengen in Def. 11.14.2 bestimmt. Alternativen bei der Festlegung der beteiligten Substitutionen führen nur zu Variationen bei den Variablenbenennungen.

Zum Selbststudium: Prädikatenlogische Resolutionsableitung

Definition 11.16 (Resolventenmengen; vgl. 8.4)

Sei **F** eine Formel in Klauselnormalform in Mengendarstellung.

$\mathrm{Res}(\mathbf{F}) := \mathbf{F} \cup \{ R \mid R$ ist prädikatenlogische Resolvente zweier Klauseln in $F \}$

Dies wird induktiv fortgesetzt durch:

$$\mathrm{Res}^0(\mathbf{F}) := \mathbf{F}$$

$$\mathrm{Res}^{n+1}(\mathbf{F}) := \mathrm{Res}\,(\mathrm{Res}^n(\mathbf{F})) \quad n \geq 0$$

$$\mathrm{Res}^*(\mathbf{F}) := \bigcup_{n \geq 0} \mathrm{Res}^n\,(\mathbf{F})$$

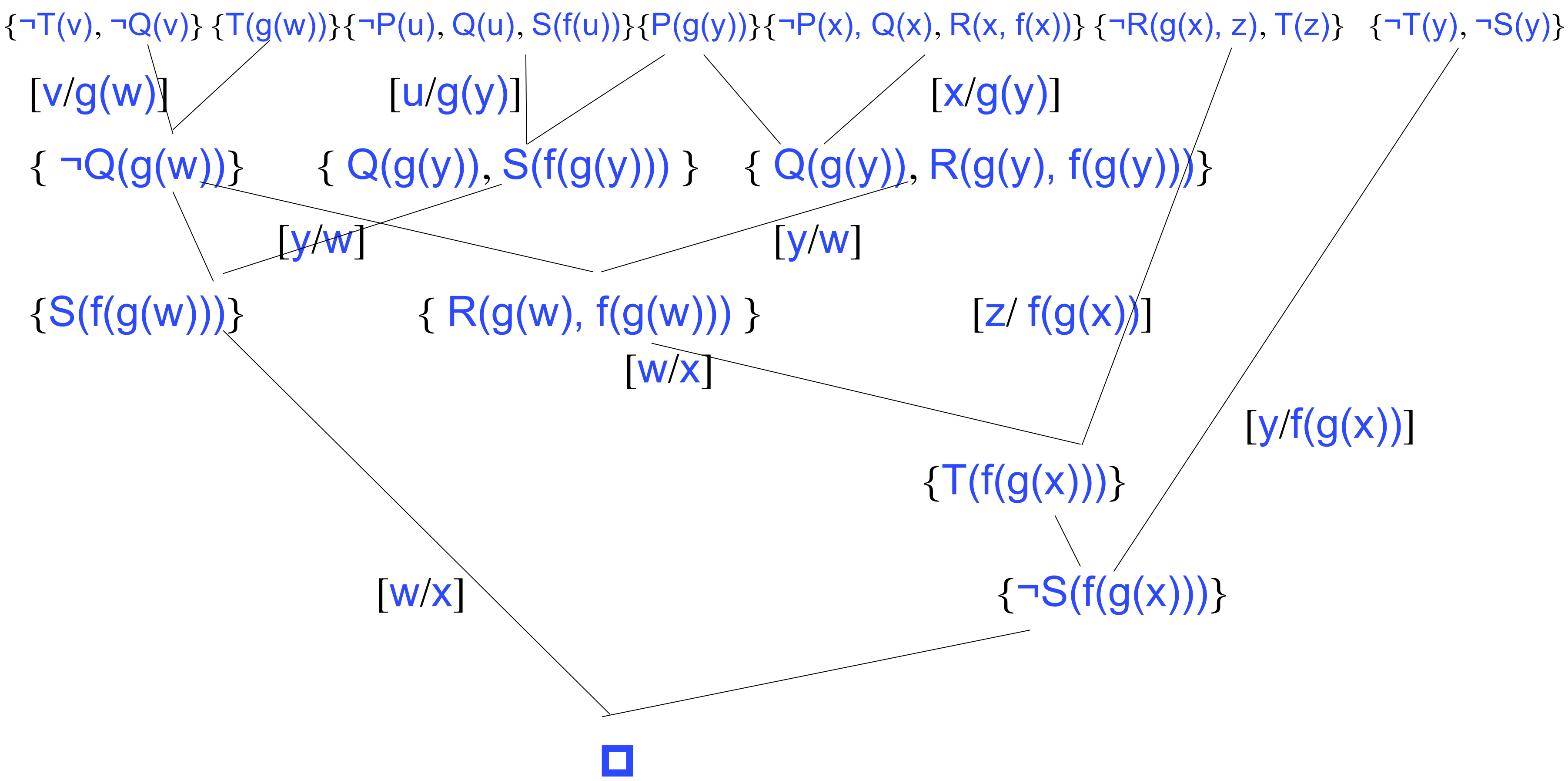
- $\square \in \mathrm{Res}^*(\mathbf{F})$ GDW. Es gibt eine Folge von Klauseln $K_1, K_2, \ldots, K_n$ derart, dass
- $K_n = \square$
- Für $i = 1, \ldots, n$ ist K_i entweder Element von **F** oder K_i ist Resolvente von K_{i1}, K_{i2} mit $i_1, i_2 < i$.

Zum Selbststudium: Prädikatenlogische Resolutionsableitung – Beispiel

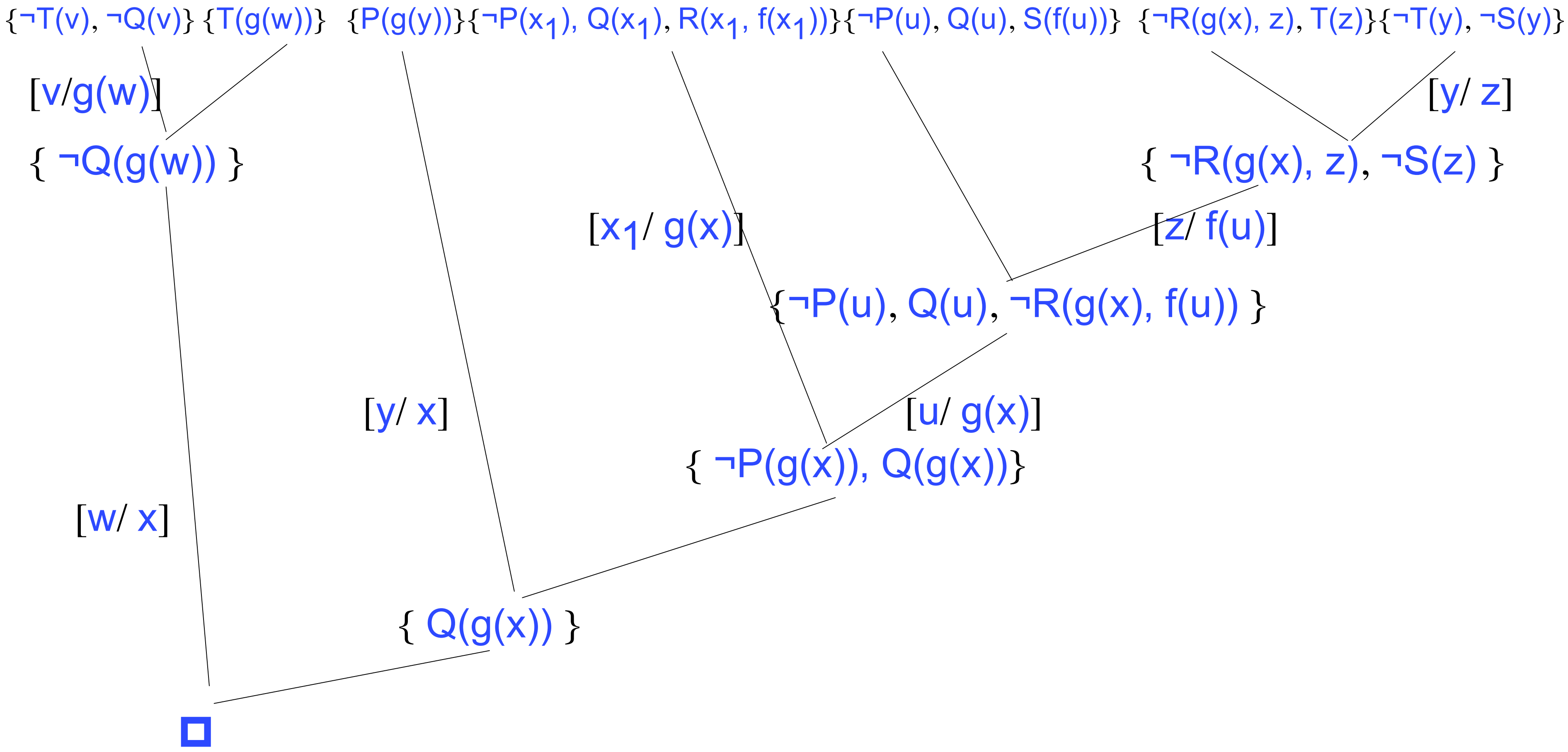
$\{\neg P(x), Q(x), R(x, f(x))\}$	$\{\neg P(u), Q(u), S(f(u))\}$	$\{T(g(w))\}$	$\{P(g(y))\}$
$\{\neg R(g(x), z), T(z)\}$	$\{\neg T(v), \neg Q(v)\}$	$\{\neg T(y), \neg S(y)\}$	

(1)	$\{T(g(w))\}$	in **F**		
(2)	$\{\neg T(v), \neg Q(v)\}$	in **F**		
(3)	$\{\neg Q(g(w))\}$	Resolvente	(1)	(2) $[v/g(w)]$
(4)	$\{\neg P(u), Q(u), S(f(u))\}$	in **F**		
(5)	$\{P(g(y))\}$	in **F**		
(6)	$\{Q(g(y)), S(f(g(y)))\}$	Resolvente	(4) $[u/g(y)]$	(5)
(7)	$\{S(f(g(w)))\}$	Resolvente	(3)	(6) $[y/w]$
(8)	$\{\neg P(x), Q(x), R(x, f(x))\}$	in **F**		
(9)	$\{Q(g(y)), R(g(y), f(g(y)))\}$	Resolvente	(5)	(8) $[x/g(y)]$
(10)	$\{R(g(w), f(g(w)))\}$	Resolvente	(3)	(9) $[y/w]$
(11)	$\{\neg R(g(x), z), T(z)\}$	in **F**		
(12)	$\{T(f(g(x)))\}$	Resolvente	(10) $[w/ x]$	(11) $[z/ f(g(x))]$
(13)	$\{\neg T(y), \neg S(y)\}$	in **F**		
(14)	$\{\neg S(f(g(x)))\}$	Resolvente	(12)	(13) $[y/ f(g(x))]$
(15)	$\square$	Resolvente	(7) $[w/ x]$	(14)

Zum Selbststudium: Resolution in der Prädikatenlogik



Resolution in der Prädikatenlogik – Resolutionsgraph



Zum Selbststudium: Lifting-Lemma

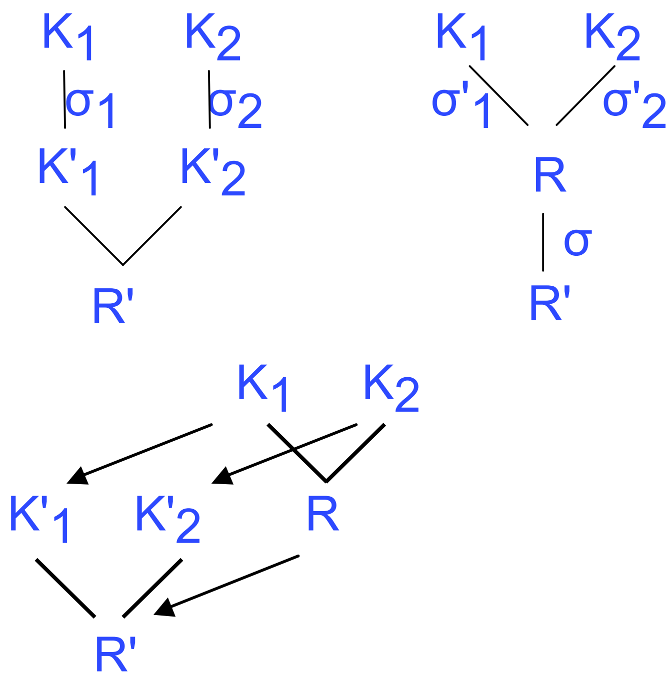
Satz 11.17 (Lifting Lemma)
Seien K_1 und K_2 zwei prädikatenlogische Klauseln und K'_1 und K'_2 zwei Grundinstanzen dieser Klauseln, die zu R' resolvierbar sind.
Dann gibt es eine prädikatenlogische Resolvente R von K_1 und K_2, so dass R' Grundinstanz von R ist.

Grundsubstitution

Grundresolution

prädikatenlogische
Resolution / Unifikation

Grundsubstitution



Beweis: siehe Schöning (Kap. 2.5)

Satz 11.18 (Resolutionssatz)

Sei F eine Aussage in Klauselnormalform und F die Mengendarstellung von F.

Dann gilt: F ist genau dann unerfüllbar, wenn $\square \in \mathrm{Res}^*(F)$.

Beweisskizze (Details siehe Schöning)

Korrektheit

Wenn $F \vdash_{res} R$, dann $F \vDash \{ R \}$, der Beweis stützt sich wesentlich auf:

Wenn $\{ K_1, K_2 \} \vdash_{res} R$, dann $\{ K_1, K_2 \} \vDash \{ R \}$

- Hier ist zu berücksichtigen:
  - die implizite Allquantifizierung von Variablen in Klauseln
  - die Unifikation (der Terme bzw. Literale)

Der Rest ergibt sich dann durch einen einfachen Schluss:

➔ Also: Wenn $\square \in \mathrm{Res}^*(F)$, dann $F \vDash \bot$, also ist dann F eine Kontradiktion.

Resolutionssatz der Prädikatenlogik [Fortsetzung]

Vollständigkeit

Sei F unerfüllbar

- Der Grundresolutionssatz sichert die Existenz einer Folge von Klauseln $K'_1, K'_2, \ldots, K'_n$, so dass
  - $K'_1, K'_2, \ldots, K'_n$ ist eine aussagenlogische Resolutionsableitung
  - $K'_n = \square$
- Zu dieser Folge von Grundinstanzklauseln existiert eine korrespondierende Folge von prädikatenlogischen Klauseln, die wir von i = 1 ausgehend konstruieren:
  - Falls K'_i, eine Grundinstanz zu einer Klausel $K \in F^*$ ist, wähle $K_i = K$.
  - Falls K'_i die Resolvente von K'_{i1} und K'_{i2} ($i_1, i_2 < i$) ist, so sind für K'_{i1} und K'_{i2} schon korrespondierende prädikatenlogische Klauseln, K_{i1} und K_{i2} festgelegt.

    Aufgrund des Lifting-Lemmas gibt es eine prädikatenlogische Resolvente R, die K'_i als Grundinstanz besitzt. Wir legen fest: $K_i = R$.

➔ $K_1, K_2, \ldots, K_n$ ist eine prädikatenlogische Resolutionsableitung, mit $K_n = \square$.

Übersicht: Resolution

- Aussagenlogische Resolution

- Prädikatenlogische Resolution

➔ Grundresolution: Bildung von Grundinstanzen + aussagenlogische Resolution

➔ Prädikatenlogische Resolution durch Unifikation

- Vollständigkeit der Grundresolution und der prädikatenlogischen Resolution

---

- Verfeinerung der Resolution
  - Restriktionen
  - Strategien
- können vom aussagenlogischen Fall in den prädikatenlogischen Fall übertragen werden

➔ Vollständigkeit muss bewiesen werden (Modifikation des Resolutionssatzes)

Von der Modelltheorie zur prädikatenlogischen Beweistheorie: Herbrands Konzeption

- Die Konzeption von Herbrand stellt spezifische Strukturen und Modelle bereit, vermittels derer Formeln in Mengen von Zeichenketten interpretiert werden.
- Auf dieser Grundlage kann die Erfüllbarkeit / Unerfüllbarkeit von Klauselnormalformen über Erfüllbarkeit / Unerfüllbarkeit von aussagenlogischen Formelmengen geprüft werden.
- Ein Test auf Erfüllbarkeit / Unerfüllbarkeit einer prädikatenlogischen Formel erfordert damit eine gezielte Suche einer endlichen unerfüllbaren Teilmenge in einer unendlichen AL-Formelmenge.
- Die Herbrand-Expansion **E**(F) einer Formel F ist eine Menge von geschlossenen und quantorenfreien Formeln. Sie korrespondiert zur Menge der Grundinstanzen von F.

➔ Die zentrale Frage ist somit:
Welche Grundsubstitutionen führen zu unerfüllbaren Teilmengen von **E**(F) ?

➔ Zur Herbrandkonzeption:
- Schöning Kapitel 2.4 und/oder Spies Kapitel 10.5
- Vorlesungen im Masterstudium

Theorem von Skolem-Herbrand-Gödel

Sei F eine geschlossene Formel in Skolemform. F ist genau dann erfüllbar, wenn die Formelmenge **E**(F) erfüllbar ist.

- Erfüllbarkeit einer geschlossenen prädikatenlogischen Formel F in Skolemform entspricht Erfüllbarkeit einer Menge von aussagenlogischen Formeln – **E**(F).

Satz von Herbrand 11.22

Sei F ein geschlossene Formel in Skolemform.
F ist genau dann unerfüllbar, wenn es eine endliche Teilmenge von **E**(F) gibt, die unerfüllbar ist.

Beweisidee

- Endlichkeitssatz der Aussagenlogik:
  Eine Menge **M** von aussagenlogischen Formeln ist genau dann erfüllbar, wenn jede endliche Teilmenge von **M** erfüllbar ist.

Zum Selbststudium: Grundresolutionsalgorithmus (vgl. Schöning Kap 2.5)

Algorithmus

Eingabe: Eine geschlossene Formel F in Skolemform mit der Matrix F^* in KNF:
[F_1, F_2, … sei eine Aufzählung von **E**(F).]

$i := 0;$ $\quad\quad M := \emptyset;$

repeat

$i := i + 1;$ $\quad\quad M := M \cup \{ F_i \};$ $\quad\quad M := \text{Res}^*(M);$

until $\square \in M$

Ausgabe: Gib „unerfüllbar“ aus und stoppe.

- Die Vorschrift (der Berechnungsschritt) $M := \text{Res}^*(M)$ terminiert für jedes i, da die Resolution einer endlichen Menge von Grundinstanzen terminiert. (Aussagenlogischer Resolutionssatz)

Satz 11.4

Bei Eingabe einer Formel F in Klauselnormalform stoppt der Grundresolutionsalgorithmus genau dann nach endlich vielen Schritten mit der Ausgabe „unerfüllbar“, wenn F unerfüllbar ist.

Kommentar zum Resolutionssatz der Prädikatenlogik

Satz 11.18 (Resolutionssatz)

Sei F eine Aussage in Klauselnormalform und F die Mengendarstellung von F.
Dann gilt: F ist genau dann unerfüllbar, wenn $\square \in \mathrm{Res}^*(F)$.

Dieser Satz behandelt nur unerfüllbare Formeln !

- w-Vollständigkeit: Ist eine Formel unerfüllbar, dann gibt es eine endliche Resolutionsableitung der leeren Klausel.
- Es gibt sogar systematische Verfahren, die garantieren, dass nach endlicher Zeit eine solche Ableitung gefunden wird,
- wobei ‘endlich’ auch sehr lange dauern kann: es gibt keine Obergrenze des Aufwandes

(Un)erfüllbarkeit in der Prädikatenlogik ist aber unentscheidbar !

- Bei erfüllbaren Formeln als Eingabe kann es passieren, dass das Verfahren nicht abbricht, da immer neue (interessante) Resolventen gebildet werden.
- Schlimmer noch: Für jede Strategie gibt es Formeln, die zu nicht abbrechenden Läufen führen, ohne dass man feststellen kann, ob sie noch abbrechen werden (→ Halteproblem).

(Un)erfüllbarkeit in der Aussagenlogik ist entscheidbar !

Wichtige Konzepte dieser Vorlesung

- Substitution, Grundsubstitution, Unifikator, allgemeinster Unifikator
- Grundinstanz, Grundresolution
- Unifikation, unifizierbar
- Unifikationsalgorithmus
- Prädikatenlogische Resolution, Resolutionsableitung
- Resolutionssatz (w-Vollständigkeit der prädikatenlogischen Resolution)