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 8. April 1299 fielen in Hamburg 17,5 mm Niederschlag.	??
• Am 8. April 2008 schien in Hamburg die Sonne länger als 4 Stunden.	??
• Am 8. 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.

Zum Selbststudium: Aussagen – Die Basis für Logik-Systeme

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**.

Zum Selbststudium

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).

  ➔ Siehe das vorigen Kapitel "Sprachen & Grammatiken: Einführung"

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

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**
$(A \wedge B)$	734 ist durch 3 teilbar und die Quersumme von 74 ist durch 3 teilbar.	**falsch**
$(A \vee B)$	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**
$(B \wedge A)$	Die Quersumme von 74 ist durch 3 teilbar und 734 ist durch 3 teilbar.	**falsch**
$(A \wedge A)$	734 ist durch 3 teilbar und 734 ist durch 3 teilbar.	**falsch**
$(A \vee \neg A)$	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

Aussagenlogik und Prädikatenlogik

Verschiedene Objektsprachen

- $\mathcal{L}_{AL}$: Die Objektsprache der Aussagenlogik
  - Beispiele: A, ¬A, (A ∨ C), (A ⇒ B), (A ⇔ (A ∨ C))
  - Die kleinsten Einheiten (atomaren Formeln) sind Aussagensymbole (A, B, C)
  - Aus ihnen werden mit Junktoren (∨, ∧, ¬, ⇒, ⇔) komplexe Formeln gebildet.
- $\mathcal{L}_{PL}$: Die Objektsprache der Prädikatenlogik
  - Anreicherung der Aussagenlogik
  - Atomare Formeln haben eine interne Struktur: P(a), P(x), R(a, b)
    Sie werden aus Prädikatssymbolen (P, R) und Termen (a, b, x) gebildet
  - Quantoren bilden zusätzliche Formeln: ∀x (R(x, a) ⇒ P(x)), ∃x (P(x) ∧ 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.

Alphabet der symbolischen Aussagenlogik

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 $\mathcal{As}_{AL}$
- den Junktoren: ¬, ∧, ∨, ⇒, ⇔
- 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	¬	Negation	*nicht …*
zweistellig	∧	Konjunktion	*… und …*
zweistellig	∨	Disjunktion	*… oder …*
zweistellig	⇒	Implikation	*wenn …, dann …*
zweistellig	⇔	Biimplikation	*… genau dann, wenn …*

Zum Selbststudium

Aussagensymbole

- Für die Menge der Aussagensymbole ist eigentlich nur folgendes wichtig:
  - Es sind höchstens abzählbar 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.
  - 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, das sie nicht viel Platz brauchen und gut auszusprechen sind.

Formeln der symbolischen Aussagenlogik

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

Definition 2.2

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

1. Alle Aussagensymbole sind (atomare) Formeln. Beispiele: A, B, C, D,…
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)

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

Die Sprache der Aussagenlogik

- Die induktive Definition der wohlgeformten Formeln (Def. 2.2) stellt eine Methode dar, eine Sprache $\mathcal{L}_{AL} \subseteq {\Sigma_{AL}}^*$ zu spezifizieren.
- Eine andere Methode für die Spezifikation von $\mathcal{L}_{AL}$ besteht in der Verwendung von formalen Grammatiken (➔ voriges Kapitel und Teil 3 der Vorlesung).

Zum Selbststudium: Dialekte der Logik

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

- Klammerkonventionen
- Menge der Aussagensymbole
- 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 $\overline{A}$ anstelle der Negation $\neg A$ 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

Formeln und andere Zeichenketten

Formeln	**Zeichenketten, die keine Formeln sind**
A	¬
¬B	) A ∧ B (
(A ∧ B)	(∧ A B)
((A ∧ B) ∨ (A ∧ ¬C))	(A ∧ B ∨ A ∧ ¬C)
((A ∧ B) ⇒ C)	¬(A)

Eine Vereinfachung: Das äußerste Klammerpaar kann weggelassen werden

Formeln	**Zeichenketten, die keine Formeln sind**
A ∧ B	(A ∧ B ∨ A ∧ ¬C)
A ⇔ (C ∨ B)	((A ∧ 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 zu zerlegen:

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

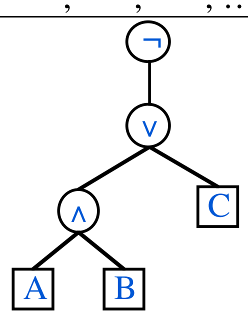
Definitionen 2.3

- 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:** $F := \neg((A \wedge B) \vee \neg C)$
  - Teilformeln: $\neg((A \wedge B) \vee \neg C)$, $((A \wedge B) \vee \neg C)$, $(A \wedge B)$, A, B, $\neg C$, C
  - keine Teilformeln: D, $B) \vee \neg C)$, $(C \Rightarrow (A \wedge B))$, $(B \wedge A)$
- 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
$(A \wedge \neg C)$	$\wedge$	Konjunktion
$((A \wedge B) \vee (A \wedge \neg C))$	$\vee$	Disjunktion
$((A \wedge B) \Rightarrow (A \vee C))$	$\Rightarrow$	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	$\boxed{A}$
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.	

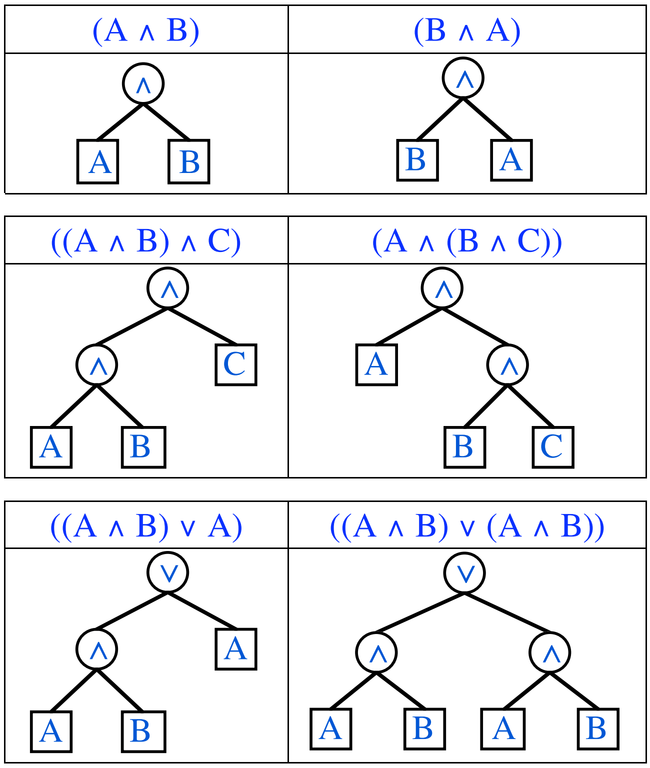
- Die Bäume werden mit der Wurzel oben und den Blättern unten gezeichnet.
- $\neg$ hat einen Nachfolger, die anderen Junktor-Knoten haben zwei Nachfolger.
- Klammern kommen in den Bäumen nicht vor.
- Kommt eine Teilformel mehrfach in der Formel vor, dann kommt der entsprechende Teilbaum auch mehrfach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

Strukturbäume: Beispiele

Reihenfolge

Klammerung

doppelte Teilformeln

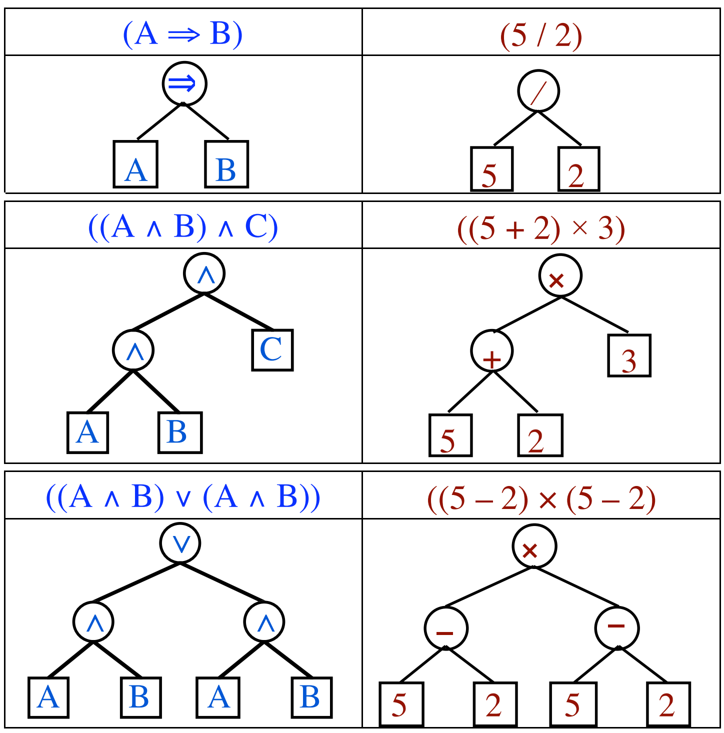


Strukturbäume: Logische Formeln ➠ Arithmetische Ausdrücke

Reihenfolge

Klammerung

Doppelte Zeichenketten



"Kontextfreie Grammatik" für Formeln der Aussagenlogik

Vokabular der terminalen Symbole
(Symbole, die in der Formel vorkommen):
$\Sigma = \{A, B, C, D, \ldots, \neg, \wedge, \vee, \Rightarrow, \Leftrightarrow,), (\}$

nicht-terminales Symbol
(Symbole, die für die Erzeugung der Formeln benötigt werden, aber nicht in der Formel vorkommen):
$N = \{S\}$

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 $S \to A_i$ ($A_i \in \mathcal{A}s_{AL}$).

Regeln (Produktionen):

$$
\begin{aligned}
P = \{\ & S \to A, \\
& S \to B, \\
& S \to C, \\
& S \to D, \\
& \ldots, \\
& S \to \neg S, \\
& S \to (S \wedge S), \\
& S \to (S \vee S), \\
& S \to (S \Rightarrow S), \\
& S \to (S \Leftrightarrow S)\ \}
\end{aligned}
$$

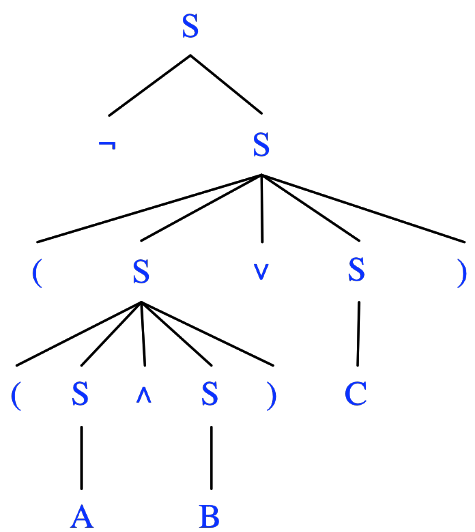
"Kontextfreie Grammatik" für AL-Formeln: Ableitungen und Strukturbäume

Komplexe Formel $\neg((A \wedge B) \vee C)$

Ableitung :

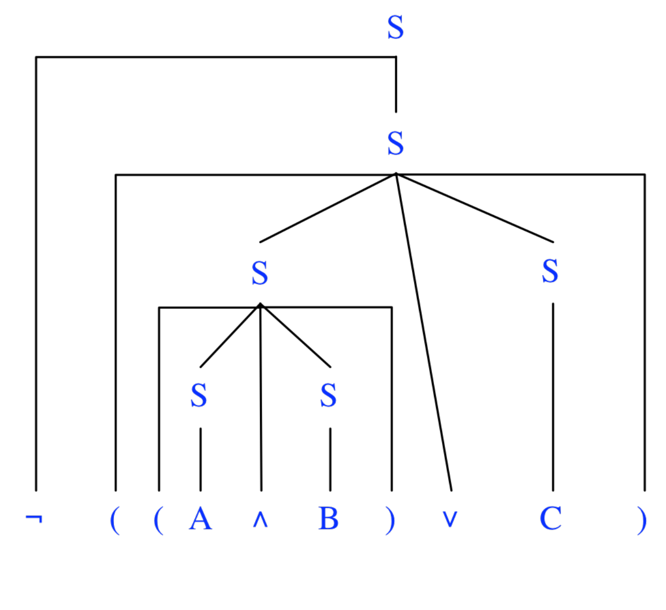
$$
\begin{aligned}
S \quad & \Rightarrow \neg S \\
& \Rightarrow \neg (S \vee S) \\
& \Rightarrow \neg ((S \wedge S) \vee S) \\
& \Rightarrow \neg ((A \wedge S) \vee S) \\
& \Rightarrow \neg ((A \wedge B) \vee S) \\
& \Rightarrow \neg ((A \wedge B) \vee C)
\end{aligned}
$$

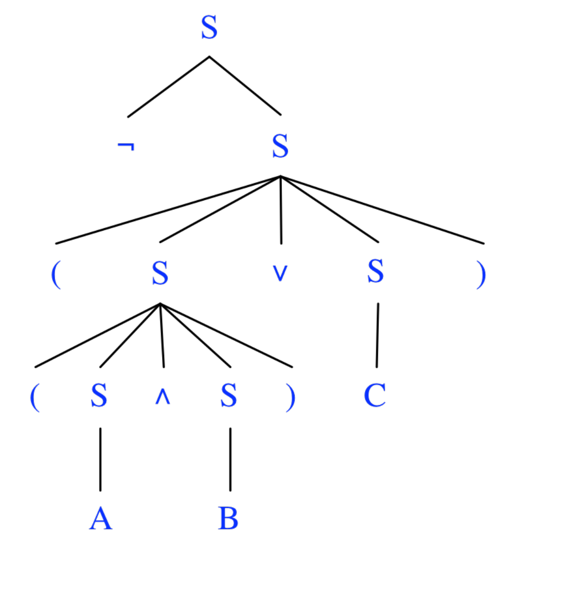
Anmerkung:
Die Reihenfolge der Ableitung von A, B und C ist nicht zwingend "von links nach rechts"



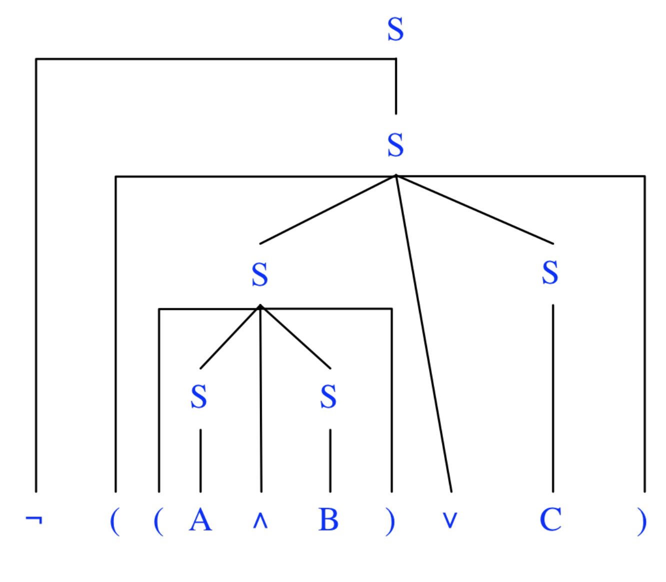
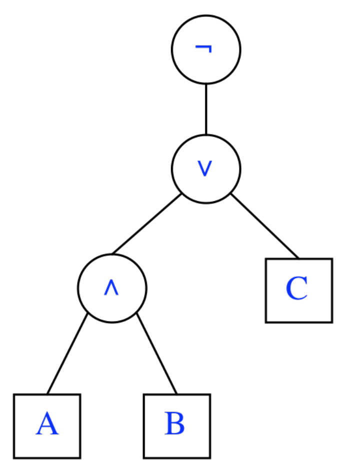
KFG-Strukturbäume: Designvarianten

Komplexe Formel ¬((A ∧ B) ∨ C)



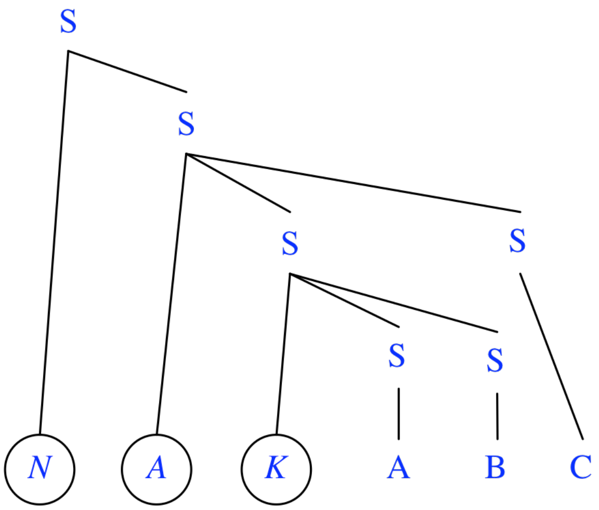


Vergleich der Strukturbäume für ¬((A ∧ B) ∨ C)

$\mathcal{L}_{AL}$ Spezifikation mit KF-Regeln	Induktive Spezifikation von $\mathcal{L}_{AL}$
	
Junktor-Knoten sind strukturell nicht-ausgezeichnete Blätter	Junktor-Knoten sind strukturell ausgezeichnet

Aussagenlogik: Polnische Notation

- Geht zurück auf Jan Łukasiewicz
- $\Sigma_{AL} = \mathcal{A}s_{AL} \cup \{N, K, A, C, E\}$ keine Klammern!!

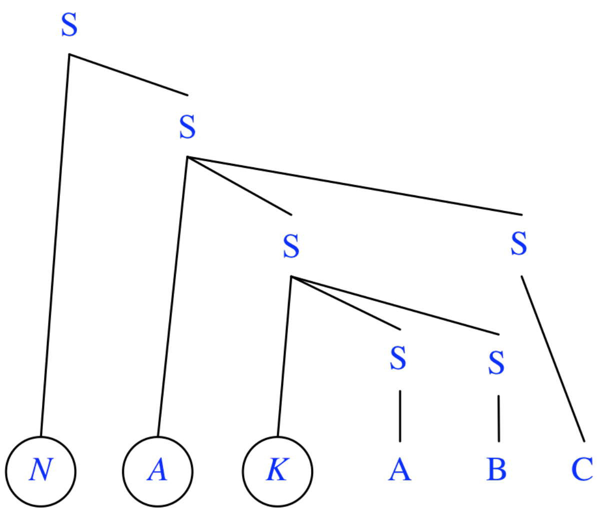


PN -Symbol	Standard-Symbol	Produktions-regel
N	$\neg$	$S \rightarrow N\ S$
K	$\wedge$	$S \rightarrow K\ S\ S$
A	$\vee$	$S \rightarrow A\ S\ S$
C	$\Rightarrow$	$S \rightarrow C\ S\ S$
E	$\Leftrightarrow$	$S \rightarrow E\ S\ S$

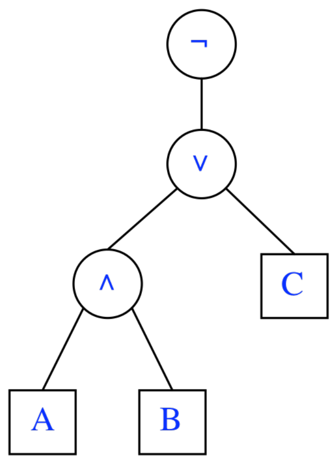
$N\ A\ K$ A B C $\Leftrightarrow$ $\neg((A \wedge B) \vee C)$

Vergleich der Strukturbäume

Polnische Notation $N\ A\ K$ A B C



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



Polnische Notation	Standardnotation
Junktor-Knoten sind strukturell ausgezeichnet: Linkeste Töchter eines nicht-unären S-Knotens	Junktor-Knoten sind strukturell ausgezeichnet
Präfix-Notation der Junktoren	Infix-Notation der binären Junktoren

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 ($\mathcal{A}s_{AL}$) 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)

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.

Zum Selbststudium: Die Menge der aussagenlogischen Formeln

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:

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

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] $\mathcal{As}_{AL} \subseteq \mathbf{M}$ (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 $\mathcal{As}_{AL} \subseteq \mathcal{L}_{AL}$, 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: $\mathcal{F}_0 = \mathcal{A}s_{AL}$ (Def. 2.4) und
$\mathcal{A}s_{AL} \subseteq \mathbf{M}$ (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 $\mathcal{L}_{AL}$

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

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

Erläuterung

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

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 $A \in \mathcal{As}_{\mathrm{AL}}$ ist $f(A) = f_{\mathcal{As}}(A)$.

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

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

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

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

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:

$grad_{\mathcal{As}}$: $\mathcal{As}_{AL} \rightarrow \mathbb{N}_0$,	$grad_{\mathcal{As}}(A) = 0$	für alle $A \in \mathcal{As}_{AL}$
$grad_{\neg}$: $\mathbb{N}_0 \rightarrow \mathbb{N}_0$,	$grad_{\neg}(n) = n + 1$	für alle $n \in \mathbb{N}_0$
$grad_{\wedge}$: $\mathbb{N}_0 \times \mathbb{N}_0 \rightarrow \mathbb{N}_0$,	$grad_{\wedge}(n, m) = n + m + 1$	für alle $n, m \in \mathbb{N}_0$

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

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

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

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

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

Jede Formel hat einen eindeutig bestimmten Grad.

Beispiel: *grad*(¬((A ∧ B) ∨ C))

grad(¬((A ∧ B) ∨ C))

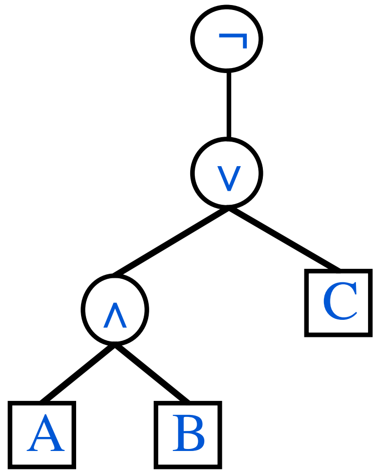
= *grad*(((A ∧ B) ∨ C)) + 1

= (*grad*((A ∧ B)) + *grad*(C) + 1) + 1

= ((*grad*(A) + *grad*(B) + 1) + *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:

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

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

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

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

Dann existiert genau eine Funktion *tiefe*: $\mathcal{L}_{AL} \rightarrow \mathbb{N}_0$, 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:

$tiefe(\neg F) = tiefe_{\neg}(tiefe(F)) = tiefe(F) + 1$

$tiefe((F \wedge G)) = tiefe_{\wedge}(tiefe(F), tiefe(G)) = max(tiefe(F), tiefe(G)) + 1$

$tiefe((F \vee G)) = tiefe_{\vee}(tiefe(F), tiefe(G)) = max(tiefe(F), tiefe(G)) + 1$

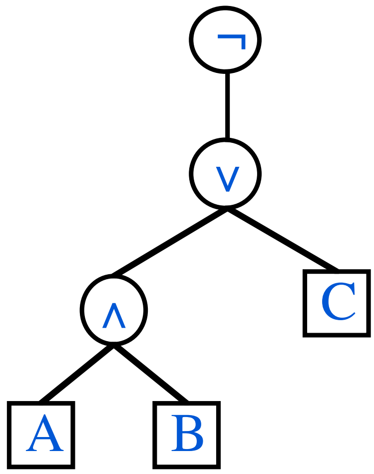
$tiefe((F \Rightarrow G)) = tiefe_{\Rightarrow}(tiefe(F), tiefe(G)) = max(tiefe(F), tiefe(G)) + 1$

$tiefe((F \Leftrightarrow G)) = tiefe_{\Leftrightarrow}(tiefe(F), tiefe(G)) = max(tiefe(F), tiefe(G)) + 1$

Jede Formel hat eine eindeutig bestimmte Tiefe.

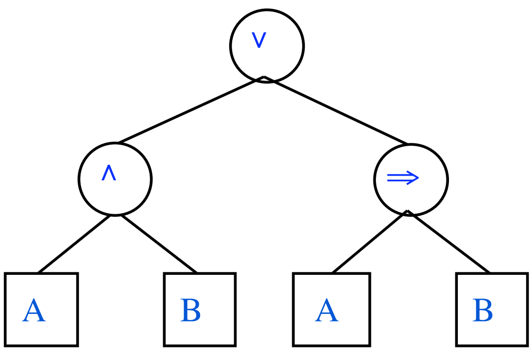
Zum Selbststudium: Beispiel: ***tiefe***(¬((A ∧ B) ∨ C))

tiefe(¬((A ∧ B) ∨ C))
= *tiefe*(((A ∧ B) ∨ C)) + 1
= (*max*(*tiefe*((A ∧ B)), *tiefe*(C)) + 1) + 1
= (*max*(*max*(*tiefe*(A), *tiefe*(B)) + 1, *tiefe*(C)) + 1) + 1
= (*max*(*max*(0, 0) + 1, 0) + 1) + 1
= (*max*(1, 0) + 1) + 1
= (1 + 1) + 1
= 3



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

Vergleichen Sie die Funktionen *grad* und *tiefe* an Hand der Berechnungen für die Formel ((A ∧ B) ∨ (A ⇒ 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"