Aussagenlogik (AL)

Formalisieren in der Aussagenlogik

Wörterbuch:

P: Paul besteht (die Prüfung).

Q: Kurt besteht.

R: Rolf besteht.

S: Paul studiert fleissig.

M: Paul ist müde.

(1) *Rolf besteht nicht.* ~R

(2) *Kurt wird nicht durchfallen.* ~~Q

(3) *Kurt und Paul bestehen beide.* Q & P

(4) *Kurt besteht, aber Paul nicht.* Q & ~P

(5) *Paul besteht, ausser wenn er müde ist.* ~M ⊃ P

(6) *Weder Paul noch Kurt bestehen.* ~(P ∨ Q)

(7) *Paul besteht nur, wenn er nicht müde ist.* P ⊃ ~M

(8) *Paul und Kurt bestehen nicht beide.* ~(P & Q)

(9) *Paul besteht dann, und nur dann, wenn er fleissig studiert hat und nicht müde ist.* P ≡ (S & ~M)

(10) *Paul besteht, falls Kurt es tut; andernfalls besteht keiner von ihnen.* (Q ⊃ P) & (~Q ⊃ (~P & ~Q))

(11) *Mindestens zwei (von den dreien) bestehen.* (P & Q) ∨ [(P & R) ∨ (Q & R)]

(äussere Klammern wurden weggelassen)

Aussagenlogik: Formale Syntax

Die *Syntax* (i.e.S.) oder Grammatik einer Logik hat die Aufgabe, die Menge der grammatisch wohlgeformten Ausdrücke der verschiedenen syntaktischen Kategorien zu bestimmen.

In der Aussagenlogik (AL) kommen wir mit einer syntaktischen Kategorie aus, nämlich der der *Aussage* oder *Formel* der AL. Die andern Zeichen, die in solchen Formeln vorkommen können, wie z.B. Satzkonnektive und Klammern, werden also als *synkategorematisch* eingeführt betrachtet, d. h. man betrachtet letztere nicht als Zeichen einer eigenen Kategorie, die eine selbständige Bedeutung haben, sondern nur als im Kontext (von Aussagen) bedeutungsvoll.

	Negation	Konjunktion	Disjunktion	Konditional	Bikonditional
Formalisierung	~A	(A & B)	(A ∨ B)	(A ⊃ B)	(A ≡ B)
Namen der Komponenten	A ist der negierte Satz (Formel)	A ist linkes Konjunkt B ist rechtes Konjunkt	A ist linkes Disjunkt B ist rechtes Disjunkt	A ist das Antezedenz B ist das Konsequent	A ist die linke Seite B ist die rechte Seite
Standard-formulierung	Es ist nicht der Fall, dass A	A und B	A oder B	Wenn A, dann B	A genau dann, wenn B
Varianten	Es ist falsch, dass A Nicht A Keineswegs A	Sowohl A als auch B A und auch B A, aber B A, obwohl B A, dennoch B A, und weiterhin B A, und zusätzlich B	Entweder A oder B A, andernfalls B A, oder auch B	Wenn A, B Falls A, (dann) B B, wenn A A nur, wenn B B, falls A B, vorausgesetzt A A ist hinreichende Bedingung für B B ist notwendige Bedingung für A B ist sine qua non für A	A gdw B A dann, und nur dann, wenn B A ist äquivalent mit B A ist notwendige und hinreichende Bedingung für B

Formalisierungshilfe für die aussagenlogischen Konnektive

Die Grammatik einer Sprache spezifiziert man durch die Angabe

1) ihres *Vokabulars*, d.h. der zulässigen Zeichen (kategorematische sowie auch synkategorematische)
2) der *grammatischen Regeln* (Formationsregeln), die die einzelnen Kategorien von Ausdrücken definieren.

Vokabular der AL

1) Eine Menge von *Satzparametern* SP (auch *Aussagenvariablen* genannt): P_0, P_1, …

   Konvention: wir lassen in der Praxis alle Grossbuchstaben des lateinischen Alphabets mit und ohne Subskripte zu; also A, B, C, …, Z, A_1, B_1, …
2) Satzkonnektive: ~, &, ∨, ⊃, ≡
3) Gruppierungszeichen: (,)

   Konvention: wir lassen auch Klammern anderer Art zu, um die Lesbarkeit zu erleichtern; also auch [,] und {, }.

Leider ist die Notation in der Logik nicht standardisiert. Es sollte jedoch keine großen Schwierigkeiten bereiten, zwischen den verschiedenen notationellen Varianten hin- und herzuwechseln.

Formationsregeln

1) Jeder Satzparameter der AL ist eine Formel der AL (*atomare Formel*).

2) Wenn A eine Formel der AL ist, dann ist auch $\sim A$ eine Formel der AL (*Negation*).

3) Wenn A und B Formeln sind, dann ist auch $(A \,\&\, B)$ eine Formel der AL (*Konjunktion*).

4) Wenn A und B Formeln sind, dann ist auch $(A \vee B)$ eine Formel der AL (*Disjunktion*).

5) Wenn A und B Formeln sind, dann ist auch $(A \supset B)$ eine Formel der AL (*Konditional*).

6) Wenn A und B Formeln sind, dann ist auch $(A \equiv B)$ eine Formel der AL (*Bikonditional*).

7) Nur die nach Regeln 1–6 gebildeten Ausdrücke sind Formeln der AL.

Die nach Regeln 2–6 gebildeten Formeln sind komplexe Formeln, deren Teile wir ihre *Komponenten* nennen. So hat z.B. ein Konditional $(A \supset B)$ die Komponenten A (*Antezedenz*) und B (*Konsequent*). Das Hufeisen ist in dieser Formel das *Hauptkonnektiv*.

Schreibkonvention: Wir verwenden $A, B, C, \ldots$ als Metavariablen für Formeln der AL allgemein, während $P, Q, R, \ldots$ als Metavariablen für atomare Formeln, also Satzparameter, verwendet werden. $\Gamma, \Delta, \ldots$ werden als Metavariablen für Mengen von Formeln verwendet.

Notationelle Varianten

- für die Negation: $\neg A$, $\overline{A}$
- für die Konjunktion: $(A \wedge B)$, $(A \,.\, B)$
- für den Konditional: $(A \Rightarrow B)$, $(A \rightarrow B)$
- für den Bikonditional: $(A \Leftrightarrow B)$, $(A \leftrightarrow B)$

Aussagenlogik: Formale Semantik

Sei *SP* die (abzählbar unendliche) Menge der Satzparameter („Aussagenvariablen“) der Sprache.

Die Menge der Wahrheitswerte ist $\{W, F\}$.

(Manchmal wird auch $\{0,1\}$ als die Menge der Wahrheitswerte verwendet.)

Eine Belegung V ist eine Funktion von SP in $\{W, F\}$ (d.h. $V \in \{W, F\}^{SP}$). Eine Belegung V weist also jedem Satzparameter genau einen der beiden Wahrheitswerte *W(ahr)* oder *F(alsch)* zu.

Semantische Regeln der AL

Atomare Formeln:

1) Der Wahrheitswert von atomaren Formeln ist durch die Belegung V festgelegt.

Komplexe Formeln:

2) Sei A von der Form $\sim B$. Dann ist $V(A) = W$ wenn $V(B) = F$. Sonst ist $V(A) = F$.

3) Sei A von der Form $(B \& C)$. Dann ist $V(A) = W$ wenn $V(B) = W$ und $V(C) = W$. Sonst ist $V(A) = F$.

4) Sei A von der Form $(B \vee C)$. Dann ist $V(A) = W$ wenn $V(B) = W$ oder $V(C) = W$. Sonst ist $V(A) = F$.

5) Sei A von der Form $(B \supset C)$. Dann ist $V(A) = W$ wenn $V(B) = F$ oder $V(C) = W$. Sonst ist $V(A) = F$.

6) Sei A von der Form $(B \equiv C)$. Dann ist $V(A) = W$ wenn $V(B) = V(C)$. Sonst ist $V(A) = F$.

Beispiel

Betrachten wir als Beispiel das Konditional ~(P & Q) ⊃ ~R. Der Wahrheitswert dieser Formel hängt von den Wahrheitswerten der beiden Teilformeln ~(P & Q) und ~R ab, deren Wahrheitswerte wiederum von den Wahrheitswerten der Teilformeln (P & Q) bzw. R abhängen; der Wahrheitswert von (P & Q) hängt wiederum von den Wahrheitswerten der Satzparameter P und Q ab. Die Wahrheitswerte der Satzparameter P, Q und R sind durch die Belegung gegeben.

Sei *V* eine Belegung, die dem Satzparameter P den Wahrheitswert *F* und Q und R den Wahrheitswert *W* zuweist. Unter dieser Belegung ist (P & Q) falsch und daher ~(P & Q) wahr; die Negation ~R ist wahr und somit die gesamte Formel falsch. Dies lässt sich leicht tabellarisch darstellen:

P	Q	R	(P & Q)	~ (P & Q)	~ R	~(P & Q) ⊃ ~R
F	W	W	F	W	F	F

I.d.R. verwenden wir eine kompaktere Notation, in der wir die Teilformeln einer komplexen Formel nicht separat aufführen und stattdessen den Wahrheitswert der Teilformeln unter den jeweiligen Konnektiven notieren:

P	Q	R	~	(P	&	Q)	⊃	~	R
F	W	W	W		F		**F**	F	

(Fett: Hauptkonnektiv)

Wahrheitstafeln

Oft sind wir nicht am semantischen Wert einer Formel unter einer bestimmten Belegung interessiert, sondern an allen Werten, die die Formeln unter allen verschiedenen Belegungen annehmen kann. Eine entsprechende *Wahrheitstafel* lässt sich leicht konstruieren, indem die Werte der Satzparameter systematisch alterniert werden:

	P	Q	R	~	(P	&	Q)	⊃	~	R
V_1	W	W	W	F	W	W	W	**W**	F	W
V_2	W	W	F	F	W	W	W	**W**	W	F
V_3	W	F	W	W	W	F	F	**F**	F	W
V_4	W	F	F	W	W	F	F	**W**	W	F
V_5	F	W	W	W	F	F	W	**F**	F	W
V_6	F	W	F	W	F	F	W	**W**	W	F
V_7	F	F	W	W	F	F	F	**F**	F	W
V_8	F	F	F	W	F	F	F	**W**	W	F

Semantische Begriffe

Def. (AL-erfüllt). Eine Belegung *V AL-erfüllt* eine Formel *A* gdw *V(A) = W*.

Def. (AL-erfüllbar). Eine Formel *A* ist *AL-erfüllbar* gdw es gibt eine Belegung *V*, die *A* AL-erfüllt.

Def. (AL-wahr, AL-gültig). Eine Formel *A* ist *AL-wahr* gdw *V(A) = W*, für jede Belegung *V*.

D.h. jede Belegung *V* AL-erfüllt *A*.

Def. (AL-falsch, AL-kontradiktorisch). Eine Formel *A* ist *AL- falsch* gdw *V(A) = F*, für jede Belegung *V*.

D.h. keine Belegung V AL-erfüllt *A*.

Def. (AL-nichtdeterminiert, AL-kontingent). Eine Formel *A* ist *AL-nichtdeterminiert* gdw es gibt eine Belegung *V* so dass gilt *V(A) = W* und es gibt eine Belegung *V'* so dass gilt *V'(A) = F*.

Def. (AL-äquivalent). Zwei Formeln *A* und *B* sind *AL-äquivalent* gdw *V(A) = V(B)*, für jede Belegung *V*.

Def. (AL-erfüllt eine Menge von Formeln). Eine Belegung *V AL-erfüllt (simultan) eine Menge von Formeln* der AL, *Γ*, gdw für alle *B* ∈ *Γ*, *V(B) = W*.

D.h. eine Belegung erfüllt eine Menge von Formeln gdw sie jedes Element der Menge erfüllt.

Def. (AL-simultan erfüllbar, AL-konsistent). Eine Menge von Formeln der AL, *Γ*, ist *AL-(simultan) erfüllbar* gdw es gibt eine Belegung *V* so dass gilt: *V* AL-erfüllt *Γ.*

Eine Menge von Formeln, *Γ*, heisst *AL nicht-erfüllbar* (*AL-inkonsistent*) gdw *Γ* nicht AL-erfüllbar ist.

Def. (AL-impliziert). Eine Menge *Γ* von Formeln *AL-impliziert* eine Formel *A* (*A* folgt aussagenlogisch aus *Γ*, *Γ* ⊨ *A*) gdw für jede Belegung *V* die *Γ* AL-erfüllt gilt: *V(A) = W*.

D.h. es gibt keine Belegung V, die jede Formel in *Γ* erfüllt, jedoch *A* nicht erfüllt.

Def. (AL-gültige Argumente). Ein Argument $A_1, \ldots, A_n / \therefore B$ ist *AL-gültig* gdw $\{A_1, \ldots, A_n\} \vDash B$.

D.h. ein Argument ist logisch gültig gdw die Prämissen die Konklusion logisch implizieren, wenn es also nicht sein kann, dass die Prämissen wahr sind, während die Konklusion falsch ist.

Ein Argument, das nicht gültig ist, heisst *AL-ungültig*.

Semantische Bäume

Angenommen, wir wollen zeigen, dass eine gegebene Menge von Formeln (simultan) erfüllbar ist. Bei der Wahrheitstafel-Methode würden wir systematisch („blind“) alle möglichen Belegungen für die in den Formeln vorkommenden Satzparameter durchprobieren und prüfen, ob eine der Belegungen alle Formeln der Menge erfüllt. Bei der Baum-Methode hingegen versuchen wir, zielgerichtet eine Belegung für die Satzparameter zu *rekonstruieren*, die die Formelmenge erfüllt: Wir nehmen an, dass es eine Belegung gibt, die die gegebene Formelmenge erfüllt, und schließen auf die Wahrheitswerte der Satzparameter, die in der Formelmenge vorkommen. Ergibt sich dabei ein Widerspruch, dann ist die Formelmenge nicht erfüllbar.

Betrachten wir als konkretes Beispiel die Menge

$$\{\sim P \;\&\; \sim\sim Q,\; P \vee \sim R,\; Q \supset R\}$$

Wir stellen diese Menge wie folgt dar:

1	~P & ~~Q	
2	P ∨ ~R	
3	Q ⊃ R	

Wir suchen eine Belegung, die die Formeln auf den Zeilen 1–3 simultan erfüllt. Diese Belegung muss nicht zuletzt die Konjunktion auf Zeile 1 erfüllen. Aus der Definition der semantischen Regel für Konjunktionen folgt, dass jede Belegung, die ~P & ~~Q erfüllt, auch die beiden Konjunkte ~P und ~~Q erfüllen muss:

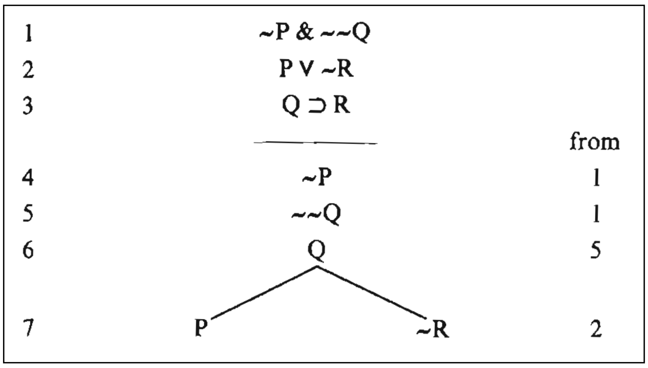
1	~P & ~~Q	
2	P ∨ ~R	
3	Q ⊃ R	
	———	from
4	~P	1
5	~~Q	1

Mit anderen Worten: Wenn eine Belegung die Formeln auf den Zeilen 1–3 simultan erfüllt, muss sie auch alle fünf Formeln simultan erfüllen.

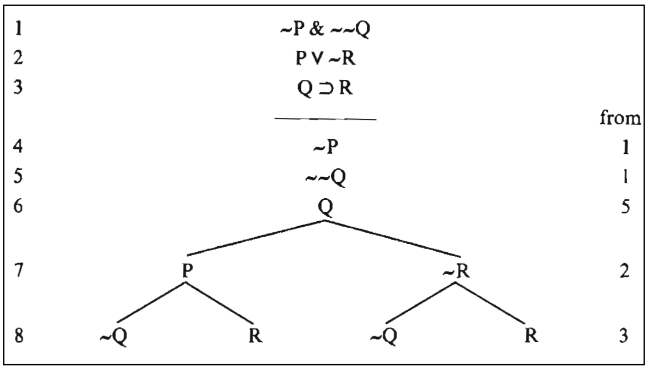
Betrachten wir als nächstes die (doppelte) Negation auf Zeile 5: Jede Belegung, die ~~Q erfüllt, muss auch Q erfüllen:

1	~P & ~~Q	
2	P ∨ ~R	
3	Q ⊃ R	
	———	from
4	~P	1
5	~~Q	1
6	Q	5

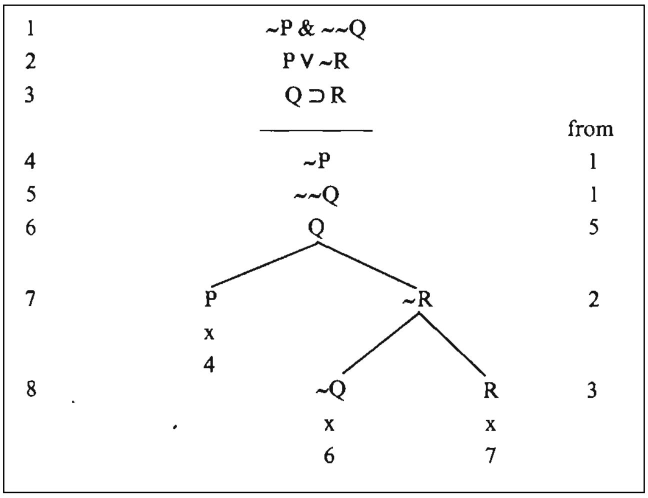
Für die Disjunktion auf Zeile 2 gilt: Eine Belegung, die P ∨ ~R erfüllt, muss mindestens eines der beiden Disjunkte P und ~R erfüllen. Wir stellen das durch eine Verzweigung dar:



Zum Schluss betrachten wir das Konditional auf Zeile 3: Wenn eine Belegung Q ⊃ R erfüllt, dann muss entweder das Antezedenz Q unter der Belegung falsch (d.h. ~Q wahr) oder das Konsequent R wahr sein:



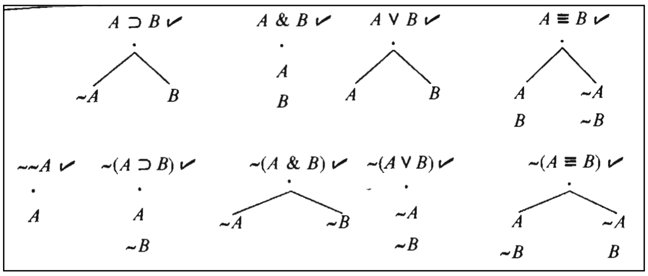
Die Pfade von der Wurzel zu den Blättern repräsentieren Formelmengen. Wir stellen fest, dass alle vier Pfade (bzw. Mengen) jeweils einen Widerspruch enthalten. Solche Pfade nennen wir geschlossen und markieren sie mit x. Tatsächlich hätten wir bereits nach hinzufügen von P auf Zeile 7 wegen ~P auf Zeile 4 den Pfad schließen können:



Nicht geschlossenen Pfade nennen wir offen.

Alle Pfade des Baums für unser Beispiel sind geschlossen (enthalten einen Widerspruch). Das heißt, dass alle Formelmengen, die der Baum repräsentiert, nicht erfüllbar sind, und damit insbesondere auch nicht die ursprüngliche Formel-Menge {~P & ~~Q, P ∨ ~R, Q ⊃ R}.

Regeln



Die zweite Regel (für Konjunktionen) lesen wir als: Wenn auf einem Pfad eine noch nicht expandierte (bearbeitete) Konjunktion vorkommt, dann füge beide Konjunkte am Ende aller offenen Pfade unterhalb dieses Vorkommens der Konjunktion hinzu und markiere die Formel als bearbeitet.

Die erste Regel (für Konditionale) lesen wir als: Wenn ein (noch nicht expandiertes) Konditional auf einem Pfad vorkommt, dann erweitere alle offenen Pfade unterhalb dieses Vorkommens des Konditionals um eine Verzweigung und füge das Antezendenz dem linken und das Konsequent dem rechten Zweig hinzu (und markiere die Formel.)

Alle anderen Regeln lesen wir entsprechend.

Konstruktion von Bäumen

(1) Wähle eine noch nicht expandierte (komplexe) Formel A und erweitere alle offenen Pfade unterhalb dieses Vorkommens von A um die Formeln, die durch die entsprechende Regel hinzugefügt werden. Markiere die Formel als bearbeitet.

(2) Sobald auf einem Pfad ein Satzparameter *P* und seine Negation ~*P* vorkommen, markiere den Pfad als geschlossen.

Wenn alle Pfade des Baums geschlossen sind, dann ist die Formelmenge nicht simultan erfüllbar. Enthält ein (vollständig expandierter) Baum einen offenen Pfad, dann kann man von diesem Pfad eine Belegung ablesen, die die Formelmenge erfüllt: Setze V(*P*) = W für alle Satzparameter *P* auf dem Pfad und V(*P*) = F für alle negierten Satzparameter ~*P*.

Anwendung der Methode auf AL-Begriffe:

AL-erfüllbar: Eine Menge $\{A_1, \ldots, A_n\}$ von Formeln ist *AL-erfüllbar*, wenn der Baum für $\{A_1, \ldots, A_n\}$ mindestens einen (vollständig expandierten) offenen Pfad enthält. Sind alle Pfade des Baums geschlossen, dann ist die Menge nicht erfüllbar.

AL-impliziert ($\Gamma \vDash$ A): Eine Menge $\Gamma=\{A_1, \ldots, A_n\}$ von Formeln *AL-impliziert* eine Formel B, wenn alle Pfade des Baums für $\{A_1, \ldots, A_n, \sim B\}$ geschlossen sind.

AL-wahr, AL-falsch: Eine Formal *A* ist *AL-wahr*, wenn alle Pfade des Baums für $\{\sim A\}$ geschlossen sind. *A* ist *AL-falsch*, wenn alle Pfade des Baums für $\{A\}$ geschlossen sind.

AL-äquivalent: Zwei Formeln *A* und *B* sind *AL-äquivalent*, wenn alle Pfade des Baums für $\{\sim(A \equiv B)\}$ geschlossen sind.

Anmerkung zur Sprechweise: Wir reden hier von „dem“ Baum für eine Formelmenge. Tatsächlich können – abhängig von der Reihenfolge, in der die Formeln bearbeitet werden – im allgemeinen verschiedenen Bäume für eine Formel-Menge konstruiert werden, die aber alle äquivalent sind.

Natürliches Schliessen

Das System S_{AL}, ein Natürliches-Schliessen System für die Aussagenlogik, besteht aus insgesamt 12 Regeln. Davon sind 10 sogenannte Intelim-Regeln (Introduktions- und Eliminationsregeln für Formeln mit bestimmten Konnektiven als Hauptkonnektiv) sowie 2 Regeln, die nicht mit Konnektiven zu tun haben:

a) die Regeln der Hypotheseneinführung (kurz Hyp), und

b) der Reiteration (kurz R).

Beachte: In Leblanc & Wisdom findet sich keine Regel Hyp; aus systematischen Gründen ist es jedoch von Vorteil eine solche Regel zu haben. Dann kann jeder Zeile in einer Derivation eine Begründung zugewiesen werden.

Wir führen zunächst einige Begriffe ein.

Derivation

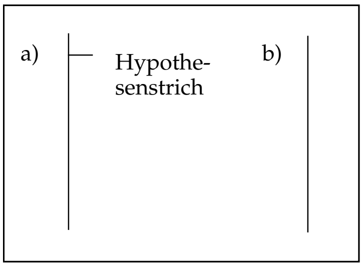
Die Objekte, die in *SAL* konstruiert werden, heissen *Derivationen*. Eine Derivation besteht aus einer Derivationslinie mit einer endlichen Anzahl von Gegenständen unmittelbar rechts von der Derivationslinie.

Derivationslinie

Es gibt zwei Arten von Derivationslinien:

a) mit Hypothesenstrich

b) ohne Hypothesenstrich.



Der Hypothesenstrich dient dazu, die durch die Hyp-Regel eingeführten Aussagen von den durch andere Regeln gewonnenen Aussagen optisch abzugrenzen.

Gegenstände einer Derivation

Es gibt weiterhin zwei Arten von Gegenständen (*items*) einer Derivation:

a) Formeln

b) Derivationen

wobei Derivationen, die selbst wieder *items* einer anderen Derivation sind, unmittelbare *Subderiva-*

tionen genannt werden. Eine Derivation kann also eine Struktur von ineinandergeschachtelten Derivationen und Formeln sein.

Subderivation

Auf solch einer Struktur kann eine Beziehung $\mathcal{D}$ ist Subderivation von $\mathcal{D}'$ rekursiv wie folgt definiert werden:

1) Jede Derivation $\mathcal{D}$ ist Subderivation von sich selbst ($\mathcal{D}$ sub $\mathcal{D}$).

2) Wenn $\mathcal{D}$ Subderivation von $\mathcal{D}'$ ist und $\mathcal{D}'$ ist ein *item* von $\mathcal{D}''$, dann ist $\mathcal{D}$ auch Subderivation von $\mathcal{D}''$.

Sonst steht nichts in der Beziehung der Subderivation.

Ein einfacher Test zur Feststellung, ob zwischen zwei Derivationen die Relation der Subderivation besteht, geht wie folgt.

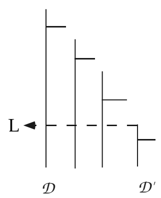
Frage: Ist $\mathcal{D}'$ Subderivation von $\mathcal{D}$ ($\mathcal{D}'$ sub $\mathcal{D}$)?

Methode:

1) Ziehe vom oberen Ende der Derivationslinie von $\mathcal{D}'$ eine horizontale Linie nach links ins „Unendliche".

2) Stelle fest, ob die Linie die Derivationslinie von $\mathcal{D}$ schneidet.

Wenn ja, denn ist $\mathcal{D}'$ sub $\mathcal{D}$; sonst nicht.

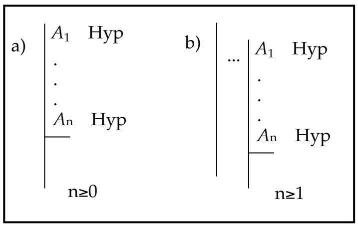


Die Regeln von S_{AL}

Die Regel der Hypotheseneinführung (Hyp):

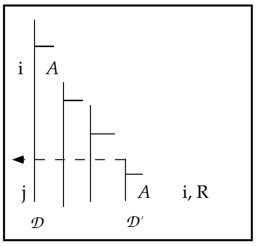
a) Um eine Derivation zu beginnen, kann man $n \geq 0$ Hypothesen einführen, indem man eine Derivationslinie eröffnet und die n Formeln untereinander und unmittelbar rechts von der Derivationslinie und oberhalb des Hypothesenstrichs schreibt. Wenn $n = 0$, dann ist eine Derivationslinie ohne Hypothesenstrich zu verwenden.

b) In einer schon angefangenen Derivation kann man jederzeit $n \geq 1$ Hypothesen einführen, indem man eine neue Derivationslinie eröffnet und die n Formeln untereinander und unmittelbar rechts von der Derivationslinie und oberhalb des Hypothesenstrichs schreibt.



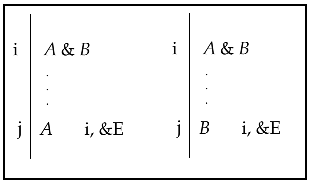
Die Regel der Reiteration (R)

Man kann eine in einer Derivation $\mathcal{D}$ schon gewonnene Formel A zu einem späteren Zeitpunkt in einer Subderivation $\mathcal{D}'$ von $\mathcal{D}$ wiederholen.

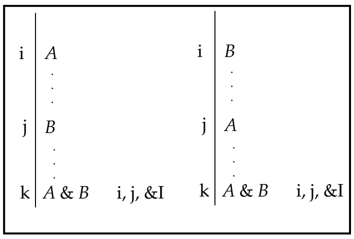


Die Regeln für die Konjunktion

&Eliminierung (&E): Wenn man in einer Derivation $\mathcal{D}$ eine Konjunktion A & B hat, dann kann man zu einem späteren Zeitpunkt (in einer Subderivation $\mathcal{D}'$ von $\mathcal{D}$) eines der beiden Konjunkte, A oder B, hinschreiben.



&Introduktion (&I): Wenn man in einer Derivation $\mathcal{D}$ eine Formel A und in einer Derivation $\mathcal{D}'$ eine Formel B hat, so kann man zu einem späteren Zeitpunkt die Konjunktion von A und B, A & B, in einer Derivation $\mathcal{D}''$ hinschreiben, die sowohl zu $\mathcal{D}$ als auch zu $\mathcal{D}'$ subordiniert ist.



Kommentar zu den Regeln

a) Man unterscheidet bei einer Regel zwischen *Prämissen* und *Konklusion*. Jede Regel, ausser die

Hypotheseneinführung, hat mindestens eine Prämisse und genau eine Konklusion, wobei die Konklusion immer eine Formel der AL sein muss. So sind z.B. Reiteration und &Eliminierung 1-Prämissen-Regeln, während &Introduktion eine 2-Prämissen-Regel ist.

b) Bei mehr-Prämissen-Regeln ist die Reihenfolge der Prämissen irrelevant; sie können auch auf mehrere Derivationen verteilt sein.

c) Die Konklusion kommt immer nach den Prämissen (d.h. sie steht auf einer späteren Zeile als alle Prämissen-Zeilen) und sie kann nur in einer Derivation stehen, die zu allen Derivationen, in denen eine Prämisse vorkommt, subordiniert ist.

Sei eine *Prämissenderivation* eine Derivation, in der eine Prämisse einer Regel steht, und die *Konklusionsderivation* die Derivation, in der die Konklusion der Regel steht.

Dann bedeutet die in c) formulierte Beschränkung:

Eine Anwendung einer Regel ist nur korrekt, wenn die Konklusionsderivation zu allen Prämissen- derivationen subordiniert ist.

Um die graphische Darstellung der Regeln nicht unnötig kompliziert zu machen, stehen Prämissen und Konklusion ab den Regeln für & alle in einer Derivation. Es sollte jedoch klar sein, dass dies eine Vereinfachung ist.

Regeln für den Konditional

Die Eliminierungsregel für den Konditional ist unser altbekannter Modus Ponens, während die Introduktionsregel dem Konditionalen Beweis entspricht:

⊃Eliminierung (⊃E): Wenn man in einer Derivation einen Konditional, $A \supset B$, hat und dessen Antezedens, A, dann kann man zu einem späteren Zeitpunkt das Konsequens des Konditionals, B, hinschreiben.

i	$A \supset B$	
	⋮	
j	A	
	⋮	
k	B	i, j, ⊃E

⊃Introduktion (⊃I): Wenn man in einer Derivation eine (Sub-)Derivation hat, deren einzige Hypothese A und deren letzte Zeile B ist, dann kann man zu einem späteren Zeitpunkt den Konditional $A \supset B$ hinschreiben.

i	A	Hyp
	$\vdots$	
j	B	
k	$A \supset B$	i-j, $\supset$I

Beachte die Notation „i–j“ in der Rechtfertigung von ⊃I, anstelle von „i, j“: die Bindestrich-Notation soll anzeigen, dass es sich bei der Prämisse um eine Derivation handelt, die sich von Zeile i bis zur Zeile j erstreckt. Dies ist also eine 1-Prämissen-Regel!

Regeln für die Negation

Die Eliminierungsregel für die Negation ist die „doppelte Negation“, während die Introduktionsregel unser altbekannter indirekter Beweis (Reductio ad absurdum) ist.

~Eliminierung (~E): Wenn man in einer Derivation eine doppelte Negation – also eine Formel der Form $\sim\sim A$ – hat, so kann man zu einem späteren Zeitpunkt die Formel A hinschreiben.

i	$\sim\sim A$	
	$\vdots$	
j	A	i, ~E

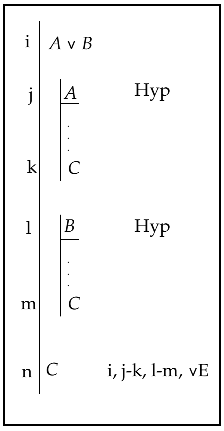
~Introduktion (~I): Wenn man in einer Derivation $\mathcal{D}$ eine (Sub-)Derivation hat, deren einzige Hypothese A und die zwei Zeilen B und $\sim B$ enthält, dann kann man zu einem späteren Zeitpunkt in $\mathcal{D}$ (oder einer Subderivation von $\mathcal{D}$) die Formel $\sim A$ hinschreiben.

i	A	Hyp
	$\vdots$	
	B	
	$\vdots$	
j	$\sim B$	
k	$\sim A$	i-j, ~I

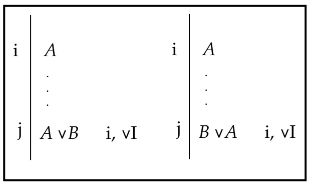
Regeln für die Disjunktion

Die Eliminierungsregel für die Disjunktion ist die altbekannte Fallunterscheidung oder das Dilemma. Die Introduktionsregel heisst auch oft Addition.

∨Eliminierung (∨E): Wenn man in einer Derivation eine Disjunktion $A \vee B$ hat und zwei (Sub-)Derivationen, deren eine als einzige Hypothese *A* und als letzte Zeile *C* hat und deren andere als einzige Hypothese *B* und als letzte Zeile *C* hat, so kann man zu einem späteren Zeitpunkt die Formel *C* hinschreiben.



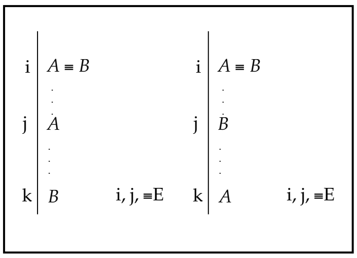
∨Introduktion (∨I): Wenn man in einer Derivation eine Formel *A* hat, so kann man zu einem späteren Zeitpunkt die Formel *A* ∨ *B* oder auch *B* ∨ *A* hinschreiben.



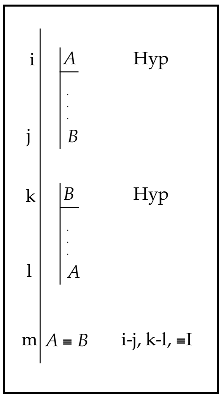
Regeln für den Bikonditional

Die Eliminierungsregel ist – wie man erwarten würde – ein „bidirektionaler" Modus Ponens; und die Introduktionsregel ist entsprechend eine zweifache ⊃Introduktion.

≡Eliminierung (≡E): Wenn man in einer Derivation einen Bikonditional *A* ≡ *B* hat und zusätzlich eine seiner Seiten, also *A* bzw. *B*, so kann man zu einem späteren Zeitpunkt die jeweils andere Seite des Bikonditionals, also *B* bzw. *A*, hinschreiben.



≡Introduktion (≡I): Wenn man in einer Derivation zwei (Sub-)Derivationen hat, deren eine als einzige Hypothese *A* und als letzte Zeile *B* hat und deren andere als einzige Hypothese *B* und als letzte Zeile *A* hat, so kann man zu einem späteren Zeitpunkt den Bikonditional *A* ≡ *B* hinschreiben.

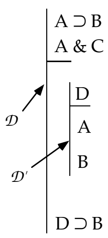


Beweistheoretische Begriffe von S_{AL}

Im Folgenden einige Begriffe der Beweistheorie von S_{AL}. Obwohl, wie man schnell sieht, eine enge Beziehung zu bestimmten semantischen Begriffen besteht (so z.B. zwischen Ableitbarkeit und logischer Folgerung), müssen die zwei „Begriffsarten“ streng unterschieden werden: Beweistheoretische Begriffe sind *syntaktische* Begriffe, d.h. beziehen sich nur auf die Struktur der Ausdrücke, und haben von ihrer Bedeutung (Definition) her mit den semantischen Begriffen nichts zu tun. Die oben angedeutete Korrespondenz zwischen beweistheoretischen (syntaktischen) und semantischen Begriffen ist nicht trivial und erfordert einen relativ komplizierten Beweis (Korrektheit und Vollständigkeit von S_{AL}).

Wir gehen vom Begriff der Derivation in S_{AL} aus, wie er oben definiert wurde. Die *Hypothesen einer Derivation* sind nur diejenigen Formeln, die Gegenstände der Derivation sind und oberhalb des Hypothesenstrichs der zugehörigen Derivationslinie stehen. (Also insbesondere nicht die Hypothesen von Subderivationen der Derivation.)

Im folgenden Beispiel sind A ⊃ B und A & C die einzigen Hypothesen der Derivation $\mathcal{D}$, und $\mathcal{D}$ ist die einzige Hypothese der Derivation $\mathcal{D}'$.



Def. (Ableitung). Eine *Ableitung in* S_{AL} ist eine Derivation, in der jede Zeile durch eine der Regeln von S_{AL} gewonnen wurde.

Def. (Abkleitung von A aus Γ). Eine *Ableitung in* S_{AL} *von A aus* Γ ist eine Ableitung in S_{AL} , deren Hypothesen alle in Γ sind und deren letzter Gegenstand (*item*) A ist.

Im obigen Beispiel ist $\mathcal{D}$ eine Ableitung von D ⊃ B aus {A ⊃ B, A & C}. Subderivationen sind Ableitungen nicht nur aus ihrer eigenen Hypothesenmenge, sondern aus der Vereinigungsmenge der eigenen Hypothesen mit allen Hypothesen *superordinierter* Derivationen. $\mathcal{D}'$ im Beispiel ist also eine Ableitung von B aus {D} ∪ {A ⊃ B, A & C}.

Def. (ableitbar). Eine Formel A ist *ableitbar in* S_{AL} *aus* Γ ($\Gamma \vdash A$) gdw es gibt eine Ableitung in S_{AL} von A aus Γ.

Def. (Beweis). Ein *Beweis in* S_{AL} *von* A ist eine Ableitung in S_{AL} von A aus ∅.

Def. (beweisbar). Eine Formel A ist *beweisbar in* S_{AL} (Notation: $\vdash A$) gdw es gibt einen Beweis in S_{AL} von A. Man sagt in diesem Fall auch: A ist ein *Theorem* oder *Satz* von S_{AL}.

Def. (Konsistenz von Mengen). Eine Menge von Formeln Γ ist *konsistent in* S_{AL} (widerspruchsfrei) gdw es gibt eine Formel A, die nicht aus Γ ableitbar ist (nicht $\Gamma \vdash A$).

Dies ist ein syntaktischer Konsistenzbegriff und muss sorgfältig vom semantischen Konsistenzbegriff (Erfüllbarkeit) unterschieden werden.

Aus dieser Definition ergibt sich, dass eine Menge von Formeln Γ *inkonsistent in* S_{AL} ist gdw sie nicht konsistent in S_{AL} ist, was wiederum bedeutet, dass aus einer solchen inkonsistenten Menge *alle* Formeln ableitbar sind, insbesondere A und $\sim A$, für jede Formel A.

Def. (gültige Argumente). Ein Argument $A_1, \ldots, A_n$ / $\therefore B$, ist *gültig in* S_{AL} gdw $\{A_1, \ldots, A_n\} \vdash B$

(d.h. wenn die Konklusion aus den Prämissen in SAL ableitbar ist).

Def. (äquivalent). Zwei Formeln A und B sind *äquivalent in* S_{AL} gdw $\{A\} \vdash B$ und $\{B\} \vdash A$

(d.h. wenn sie auseinander in S_{AL} ableitbar sind).

Strategien für die Regelanwendung

(i) Wenn du einen Konditional $A \supset B$ gewinnen willst, dann nimm zuvor das Antezedenz A als Hypothese in einer Subderivation an und versuche das Konsequent B in der Subderivation abzuleiten; dann kann der Konditional mit ⊃I gewonnen werden.

(ii) Wenn du eine Konjunktion A & B gewinnen willst, dann versuche zuvor jedes der beiden Konjunkte A und B unabhängig voneinander zu gewinnen; dann kann die Konjunktion mit &I gewonnen werden.

(iii) Wenn du eine Disjunktion $A \vee B$ gewinnen willst, dann versuche zuvor das eine (A) oder andere (B) Disjunkt abzuleiten; dann kann die Disjunktion mit ∨= gewonnen werden.

(iv) Wenn du eine Negation $\sim A$ gewinnen willst, dann nimm zuvor den negierten Satz A als Hypothese in einer Subderivation an und versuche, daraus irgendeine Kontradiktion B und $\sim B$ in der Subderivation abzuleiten; dann kann die Negation mit ~I gewonnen werden.

(v) Wenn du einen Bikonditional $A \equiv B$ gewinnen willst, dann versuche zuvor, in zwei Subderivationen, B aus der Hypothese A und A aus der Hypothese B abzuleiten; dann kann der Bikonditional mit ≡I gewonnen werden.

(vi) Wenn du eine Formel A gewinnen willst und dabei nicht vorankommst, dann versuche zuvor – als letztes Mittel – in einer Subderivation mit $\sim A$ als Hypothese eine Kontradiktion B und $\sim B$ abzuleiten; dann kann die doppelte Negation $\sim\sim A$ mit ~I gewonnen werden, und daraus wiederum A mit ~E gewonnen werden.

(vii) Wenn du eine Formel C gewinnen willst und die vorher abgeleiteten Formeln enthalten eine Disjunktion $A \vee B$, dann versuche zuvor C in jeweils zwei Subderivationen mit je A und B als Hypothesen abzuleiten; dann kann C mit 2E gewonnen werden.

(viii) Wann immer eine Eliminations-Regel angewandt werden kann, wende sie an.

(ix) Reiteriere hemmungslos. Man kann dabei nichts falsch machen, höchstens unnötig reiterieren.

(x) Keep cool!!