

Aussagenlogische Kalküle

Ziel:

mit Hilfe von schematischen *Regeln* sollen *alle* aus einer Formel logisch folgerbaren Formeln durch (prinzipiell syntaktische) Umformungen abgeleitet werden können.

Derartige Systeme von "Beweisregeln" werden *Kalküle* genannt.

Beispiel einer Beweisregel:

Die Regel mit Schemavariablen α, β, γ

$$\frac{\alpha \vee \beta, \beta \rightarrow \gamma, \gamma \rightarrow \alpha}{\alpha}$$

Eine Instanz mit

$$\begin{aligned}\alpha &= \neg P_1 \\ \beta &= \mathbf{0} \\ \gamma &= P_1 \wedge P_2\end{aligned}$$

$$\frac{\neg P_1 \vee \mathbf{0}, \mathbf{0} \rightarrow (P_1 \wedge P_2), (P_1 \wedge P_2) \rightarrow \neg P_1}{\neg P_1}$$

Definition für Beweisregeln, präzisiert:

Für ein $n \in \mathbb{N}$ ist eine n -stellige Regel R eine entscheidbare, $n + 1$ -stellige Relation über der Menge der Formeln.

Ist $(u_1, \dots, u_n, u_{n+1}) \in R$, so heißen

- $(u_1, \dots, u_n, u_{n+1})$ eine *Instanz* von R
- u_1, \dots, u_n die *Prämissen* dieser Instanz
- u_{n+1} die *Conclusio* dieser Instanz.

Wir schreiben meist

$$\frac{u_1, \dots, u_n}{u_{n+1}}$$

Die Instanzen von nullstelligen Regeln heißen auch *Axiome*.

Ableitungen in einem Kalkül, abstrakt:

Sei eine Formelmenge L , eine Menge M von Voraussetzungen und eine Menge von Regeln festgelegt.

Definition

Eine *Ableitung* aus M ist eine Folge (u_1, \dots, u_m) von Formeln in L , so dass für jedes $i \in \{1, \dots, m\}$ gilt:

- u_i ist Axiom; oder
- $u_i \in M$; oder
- es gibt eine Instanz

$$\frac{u_{j_1}, \dots, u_{j_n}}{u_i}$$

einer Regel mit $j_1, \dots, j_n < i$

Man beachte, dass u_1 ein Axiom oder ein Element von M sein muss.

Eine Formel A heisst *ableitbar* aus M , kurz

$$M \vdash A$$

genau dann, wenn es eine Ableitung (u_1, \dots, u_m) gibt mit $u_m = A$.

- Für $\emptyset \vdash u$ schreiben wir $\vdash u$, für $\{v\} \vdash u$ schreiben wir $v \vdash u$.
- Falls erforderlich wird der Kalkül in der Notation mit angezeigt, also $M \vdash_{\text{Kal}} u$.

Was sollte ein Kalkül leisten können?

Korrektheit: $M \vdash A \Rightarrow M \models A$

Vollständigkeit: $M \models A \Rightarrow M \vdash A$

Beispiel:

Die Regeln des Hilbertkalküls

$$\text{Ax1: } \frac{}{\alpha \rightarrow (\beta \rightarrow \alpha)}$$

$$\text{Ax2: } \frac{}{(\alpha \rightarrow (\beta \rightarrow \gamma)) \rightarrow ((\alpha \rightarrow \beta) \rightarrow (\alpha \rightarrow \gamma))}$$

$$\text{Ax3: } \frac{}{(\neg\alpha \rightarrow \neg\beta) \rightarrow (\beta \rightarrow \alpha)}$$

$$\text{Mp: } \frac{\alpha, \alpha \rightarrow \beta}{\beta} \quad (\text{Modus ponens})$$

Der Hilbertkalkül ist aber für das automatische Beweisen nicht so gut geeignet.

Der Resolutionskalkül

wird verwendet, um die *Unerfüllbarkeit* einer endlichen Menge aussagenlogischer Formeln zu beweisen.

- Zur Überprüfung, ob F eine Tautologie darstellt, kann man $\neg F$ auf Unerfüllbarkeit testen.
- Eine Formel G **folgt** aus einer Formelmenge $\{F_1, \dots, F_k\}$ klassisch gdw. $F_1 \wedge \dots \wedge F_k \wedge \neg G$ unerfüllbar ist:
Widerlegungsverfahren („Refutation“).

Exkurs:

Unerfüllbarkeit und logische Folgerung

$$\Delta \models \varphi \Leftrightarrow \Delta \cup \{\neg\varphi\} \text{ ist unerfüllbar.}$$

Modelltheoretische Argumentation:

" \rightarrow "

Wenn $\Delta \models \varphi$, dann sind alle Modelle für Δ auch Modelle für φ .
Damit ist keine dieser Interpretationen Modell für $\neg\varphi$ und
damit ist $\Delta \cup \neg\varphi$ unerfüllbar.

" \leftarrow "

Sei nun $\Delta \cup \neg\varphi$ unerfüllbar, aber Δ erfüllbar. Sei I die Interpretation, die Δ erfüllt. I erfüllt nicht $\neg\varphi$, denn sonst wäre $\Delta \cup \neg\varphi$ erfüllbar.
Damit folgt, dass I φ erfüllt. (Eine Interpretation muss entweder φ oder $\neg\varphi$ erfüllen.). Da dies für willkürliche Modelle I für Δ gilt, gilt es für alle I , die Δ erfüllen. Damit sind alle Modelle für Δ auch Modelle für φ und φ ist logische Konsequenz von Δ .

Resolutionskalkül: beruht auf der schematischen, aussagenlogischen Schlussregel der Resolution.

Sei P eine aussagenlogische Variable. *Resolutionsregel*:

$$\frac{\alpha \vee P, \beta \vee \neg P}{\alpha \vee \beta}$$

Merkmale des Resolutionskalküls

- Widerlegungskalkül
- Voraussetzung:
Alle Formeln sind in konjunktiver Normalform.
- Es gibt eine einzige Regel: die Resolutionsregel
eine Modifikation des Modus ponens.

- es sind keine logischen Axiome notwendig.

Zur Vereinfachung führen wir eine
Mengenschreibweise für Formeln in KNF ein.

Diese definiert zugleich die *Syntax des Kalküls*:

Eine *Klausel* ist eine endliche Menge von Literalen.

\square ist die *leere Klausel*.

Eine Klausel

$$\{P_1, P_2\} \quad \{P_1, \neg P_2\}$$

lesen wir als die Disjunktion der ihr angehörenden Literalen:

$$P_1 \vee P_2 \quad P_1 \vee \neg P_2$$

Die grundlegende Datenstruktur, auf der wir operieren, sind die
Mengen von Klauseln,

$$\{\{P_1, P_2\}, \{P_1, \neg P_2\}, \{\neg P_1, P_2\}, \{\neg P_1, \neg P_2\}\}$$

gelesen als Konjunktion der Klauseln.

$$(P_1 \vee P_2) \wedge (P_1 \vee \neg P_2) \wedge (\neg P_1 \vee P_2) \wedge (\neg P_1 \vee \neg P_2)$$

Bis auf die Reihenfolge der Disjunktions- bzw.
Konjunktionsglieder sind also in umkehrbar eindeutiger Weise
die Formeln in konjunktiver Normalform als Klauselmengen
wiedergegeben.

Die Mengenschreibweise bringt Vorteile hinsichtlich der Ausnutzung der Assoziativität und Kommutativität von Konjunktion und Disjunktion.

Def. "Resolvente":

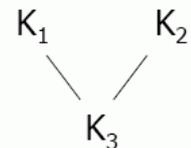
a) Seien K_1 , K_2 und K_3 Klauseln. Die Klausel K_3 heißt eine **Resolvente** von K_1 und K_2 , wenn es

ein Literal L gibt, so dass die folgenden beiden Bedingungen erfüllt sind:

1) Es gilt $L \in K_1$ und $\neg L \in K_2$. Ist hierbei $L = \neg A$ ein negatives Literal, so sei $\neg L = A$.

2) $K_3 = (K_1 \setminus \{L\}) \cup (K_2 \setminus \{\neg L\})$

b) Ist K_3 eine Resolvente von K_1 und K_2 , so verwenden wir nebenstehende grafische Darstellung:



c) Dabei ist die leere Klausel $K_3 = \square$ zulässig. Sie ergibt sich z.B. als Resolvente von $K_1 = \{L\}$ und $K_2 = \{\neg L\}$.

Eine Klauselmengende, die \square enthält, wird als **unerfüllbar** bezeichnet.

Semantik:

Ist $I : \Sigma \rightarrow \{W, F\}$ eine Interpretation der Atome, so hat man als zugehörige Auswertung

$val_I : \text{Mengen von Klauseln} \rightarrow \{W, F\}$:

$$val_I(M) = \begin{cases} W & \text{falls für alle } C \in M \text{ gilt:} \\ & val_I(\{C\}) = W \\ F & \text{sonst} \end{cases}$$

$$val_I(\{C\}) = \begin{cases} W & \text{falls ein } L \in C \text{ existiert mit} \\ & I(L) = W \\ F & \text{sonst} \end{cases}$$

für Klauseln C ,

Es folgt also

$$val_I(\{\square\}) = F,$$

$$val_I(\emptyset) = W$$

(\emptyset ist die leere Klauselmengende.)

Die Regel in Mengenschreibweise:

Die *aussagenlogische Resolutionregel* ist die Regel

$$\frac{C_1 \cup \{P\}, C_2 \cup \{\neg P\}}{C_1 \cup C_2}$$

mit einer AL-Variablen P und Klauseln C_1, C_2

$C_1 \cup C_2$ heisst *Resolvente* von $C_1 \cup \{P\}, C_2 \cup \{\neg P\}$.

Der Resolutionskalkül enthält die Resolutionsregel als einzige Regel.

Ziel des Resolutionskalküls ist der Nachweis der Unerfüllbarkeit einer Klauselmenge K durch Ableitung von \square aus K . Wünschenswert sind dabei:

Korrektheit: \square ist nur dann ableitbar aus K , wenn K unerfüllbar ist.

Vollständigkeit: Wenn K unerfüllbar ist, so lässt sich \square aus K ableiten.

Beispiel:

Gegeben sei die Klauselmenge

$$M = \{\{P_1, P_2\}, \{P_1, \neg P_2\}, \{\neg P_1, P_2\}, \{\neg P_1, \neg P_2\}\}$$

$$\frac{\{P_1, P_2\}, \{P_1, \neg P_2\}}{\{P_1\}}$$

$$\frac{\{\neg P_1, P_2\}, \{\neg P_1, \neg P_2\}}{\{\neg P_1\}}$$

$$\frac{\{P_1\}, \{\neg P_1\}}{\square}$$

Insgesamt:

$$M \vdash_{Res} \square$$

Ein zweites Beispiel:

Es soll gezeigt werden, dass

$$(A \rightarrow B) \rightarrow ((B \rightarrow C) \rightarrow (A \rightarrow C))$$

eine Tautologie ist.

Dazu werden wir zeigen, dass die Negation dieser Formel

$$\neg((A \rightarrow B) \rightarrow ((B \rightarrow C) \rightarrow (A \rightarrow C)))$$

nicht erfüllbar ist.

Klauselnormalform:

$$M = \{\{\neg A, B\}, \{\neg B, C\}, \{A\}, \{\neg C\}\}$$

$$M = \{\{\neg A, B\}, \{\neg B, C\}, \{A\}, \{\neg C\}\}$$

Ableitung der leeren Klausel aus M :

- (1) \square $\{\neg A, B\}$
- (2) \square $\{\neg B, C\}$
- (3) \square $\{A\}$
- (4) \square $\{\neg C\}$
- (5) [1, 3] $\{B\}$
- (6) [2, 5] $\{C\}$
- (7) [4, 6] \square

Korrektheit und Vollständigkeit:

Theorem

Für eine Menge M von Klauseln gilt

$$M \text{ unerfüllbar} \Leftrightarrow M \vdash_{R0} \square.$$

Beweisidee für die Vollständigkeit (" \Rightarrow "):

- Aus $M \not\vdash_{R0} \square$ beweisen wir die Erfüllbarkeit von M .
- Beliebige aber feste Reihenfolge der Atome: P_0, \dots, P_n, \dots
- M_0 sei die Menge aller Klauseln C mit $M \vdash_{R0} C$.
Es gilt also $M \subseteq M_0$ und $\square \notin M_0$.
- Wir definieren (induktiv) eine Interpretation I so dass:
für alle $C \in M_0$ gilt $val_I(C) = W$.
Insbesondere ist dann M als erfüllbar nachgewiesen.

Umformulierung des Theorems:

• Definition:

a) Für jede Klauselmenge K setzen wir

$$\text{Res}^1(K) = K \cup \{R \mid R \text{ Resolvente zweier Klauseln in } K\}.$$

b) Für $n \geq 2$ sei $\text{Res}^n(K) = \text{Res}(\text{Res}^{n-1}(K))$.

Für $n = 0$ sei $\text{Res}^0(K) = K$. Wir nennen $\text{Res}^n(K)$ die Menge der Resolventen **n-ter Stufe** von K .

c) Schließlich setzen wir $\text{Res}^\infty(K) = \bigcup_{n \geq 0} \text{Res}^n(K)$.

• Satz: (Resolutionssatz der Aussagenlogik)

Eine Formel F ist genau dann unerfüllbar, wenn $\square \in \text{Res}^\infty(K(F))$ gilt.

- **Algorithmus:** Erfüllbarkeitstest für aussagenlogische Formeln

Gegeben sei eine aussagenlogische Formel F in KNF.

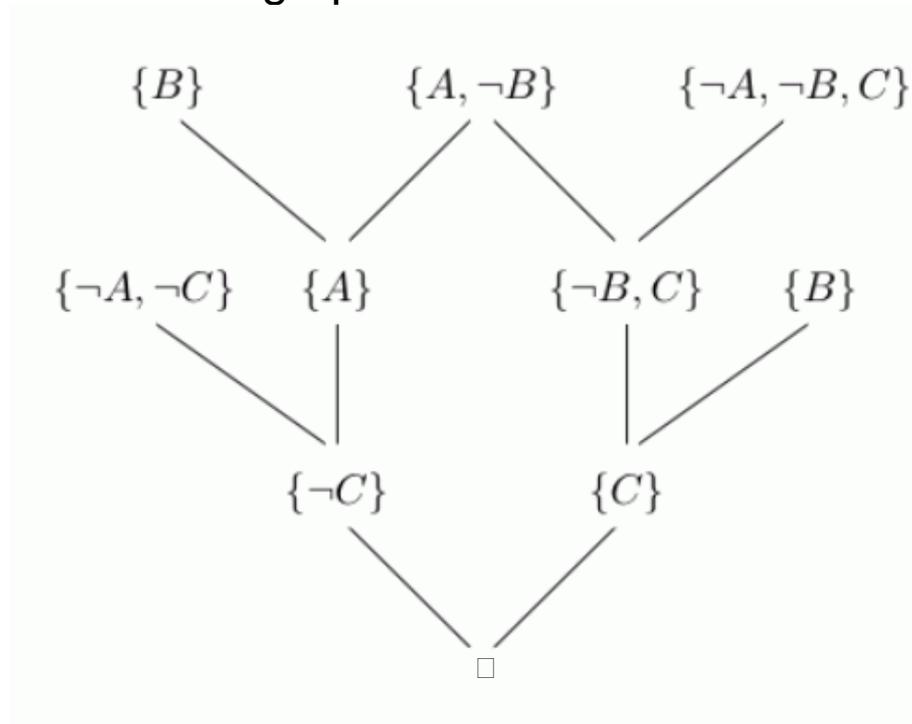
- 1) Bilde die Klauselmengemenge $K(F)$.
- 2) Für $n = 1, 2, 3, \dots$ berechne $\text{Res}^n(K(F))$ solange, bis
 - $\in \text{Res}^n(K(F))$ oder $\text{Res}^n(K(F)) = \text{Res}^{n-1}(K(F))$ gilt.
- 3) Im ersten Fall gib "F ist unerfüllbar" aus, im zweiten Fall gib "F ist erfüllbar" aus und stoppe.

- **Definition:**

Eine **Deduktion (Herleitung)** der leeren Klausel aus einer Klauselmengemenge F ist eine Folge K_1, \dots, K_m von Klauseln mit $K_m = \square$ und jedes K_i ($i = 1, \dots, m$) ist Element aus F oder kann aus Klauseln K_a, K_b mit $a, b < i$ resolviert werden.

- **Beispiel:** $\mathbf{K} = \{\{B\}, \{A, \neg B\}, \{\neg A, \neg B, C\}, \{\neg A, \neg C\}\}$

Resolutionsgraph:



Das Ableiten der leeren Klausel aus einer Menge von Klauseln kann als *Suchproblem* aufgefasst werden.

Es sind verschiedene Suchstrategien möglich,

z.B.:

- "linear": Die Resolvente ist jeweils wieder Elternklausel im nächsten Schritt
- "input resolution": eine der Elternklauseln ist immer eine Eingabeklausel
- "unit resolution": eine der Elternklauseln darf nur ein Literal enthalten
- "ordered resolution"
- SLD-Resolution (s.u.)

1-Resolution

Die 1-Resolutionsregel ist ein Spezialfall der allgemeinen Resolutionsregel:

$$\frac{\{P\}, C_2 \cup \{\neg P\}}{C_2} \qquad \frac{\{\neg P\}, C_2 \cup \{P\}}{C_2}$$

Der 1-Resolutionskalkül ist nicht vollständig.

Die Klauselmenge

$$E = \{\{P_1, P_2\}, \{P_1, \neg P_2\}, \{\neg P_1, P_2\}, \{\neg P_1, \neg P_2\}\}$$

ist nicht erfüllbar, aber mit 1-Resolution ist aus E nichts ableitbar, also auch nicht \square

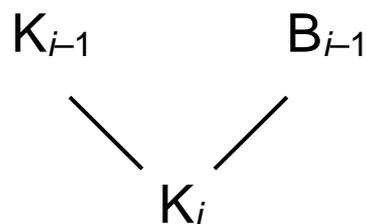
Lineare Resolution

• Definition:

Sei F eine aussagenlogische Formel in KNF.

a) Die leere Klausel \square ist aus $K(F)$ **linear resolvierbar**, falls es eine Klausel $K_0 \in K(F)$ gibt und eine Folge von Klauseln K_1, \dots, K_n mit folgenden Eigenschaften:

a1) für $i = 1, \dots, n$ gilt



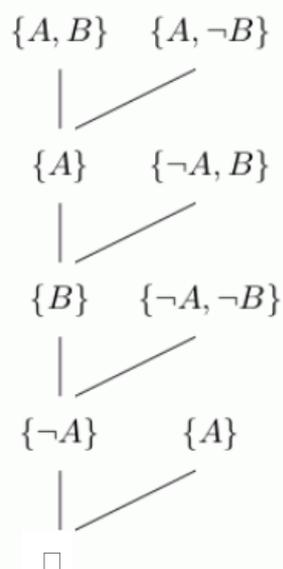
wobei die **Seitenklausel** B_{i-1} entweder ein Element von $K(F)$ ist oder $B_{i-1} = K_j$ mit $j < i - 1$.

a2) $K_n = \square$.

b) Eine Folge von Klauseln K_0, \dots, K_n wie in a) heißt eine **lineare Resolution** von F .

Beispiel: Sei $K = \{\{A, B\}, \{A, \neg B\}, \{\neg A, B\}, \{\neg A, \neg B\}\}$

Lineare Resolution



Input-Resolution

- **Definition:**

Sei F eine aussagenlogische Formel in KNF.

Eine **Input-Resolution** von F ist eine lineare Resolution von F , bei der in jedem Schritt als Seitenklausel eine der Klauseln aus $K(F)$ verwendet wird.

- **Bemerkung:**

Nicht jede unerfüllbare aussagenlogische Formel in KNF besitzt eine Input-Resolution;
u.a. die im letzten Beispiel.

SLD-Resolution

- **Definition:**

Sei F eine Hornformel.

Eine **SLD-Resolution** ("Linear resolution with Selection function restricted to Definite clauses") ist eine Input-Resolution der folgenden Form:

- a) K_0 ist eine negative Klausel, die sog. **Zielklausel**.
- b) Bei jedem Resolutionsschritt ist eine der Elternklauseln eine nicht-negative Hornklausel, also eine **Programmklause**l.

- **Anmerkungen:**

- Eine SLD-Resolution benötigt immer auch eine **Auswahlfunktion**, die bei jedem Schritt angibt, welches Literal als nächstes zu resolvieren ist.
Diese ist bei der Logik-Programmierung zu spezifizieren.

- **Satz:**

1. Die lineare Resolution ist vollständig, d.h. für jede unerfüllbare Klauselmeng K gibt es eine Klausel $k \in K$, so dass die leere Klausel durch lineare Resolution aus K , basierend auf k , herleitbar ist.
2. Die SLD-Resolution ist vollständig für Hornformeln.

Die SLD-Resolution ist die Grundlage des logischen Schießens in der Programmiersprache Prolog.

Exkurs: *Prolog-Syntax*

Tatsachenklauseln (atomare Formeln, "Fakten") werden in der Form $A :- .$ oder $A .$ eingegeben.

Prozedurklauseln (Klauseln mit genau einem positiven und mindestens einem negativen Literal) werden in der Form

$$A :- B_1, B_2, \dots, B_r$$

eingegeben. Dies bedeutet

$$A \vee \neg B_1 \vee \dots \vee \neg B_r$$

oder, gleichwertig:

$$(B_1 \wedge \dots \wedge B_r) \rightarrow A .$$

Weitere Kalküle

Der Tableauekalkül

Wesentliche Eigenschaften

- Widerlegungskalkül: Testet auf Unerfüllbarkeit

$$M \models A \quad \Leftrightarrow \quad M \cup \{\neg A\} \vdash_{\text{TO}} \mathbf{0}.$$

- Beweis durch Fallunterscheidung
- Top-down-Analyse der gegebenen Formeln

Vorteile

- Intuitiver als Resolution
- Formeln müssen nicht in Normalform sein
- Falls Formelmenge erfüllbar ist (Test schlägt fehl), wird ein Gegenbeispiel (eine erfüllende Interpretation) konstruiert

Nachteil

- Mehr als eine Regel

Der Tableauekalkül verwendet eine Erweiterung der aussagenlogischen Formeln als Hilfskonstruktion: "Vorzeichenformeln"

Definition (Vorzeichenformel)

Eine **Vorzeichenformel** ist eine Zeichenkette der Gestalt

$$0A \text{ oder } 1A \quad \text{mit } A \in \text{For}_0.$$

0, 1 sind neue Sonderzeichen (die Vorzeichen) im Alphabet der Objektsprache.

Definition

Wir setzen val_I fort auf die Menge aller Vorzeichenformeln durch

$$val_I(0A) = val_I(\neg A),$$

und

$$val_I(1A) = val_I(A).$$

2 Typen von Vorzeichenformeln:

Konjunktive Formeln: Typ α

- $1(A \wedge B)$
- $0(A \vee B)$
- $0(A \rightarrow B)$
- $0\neg A$
- $1\neg A$

Disjunktive Formeln: Typ β

- $0(A \wedge B)$
- $1(A \vee B)$
- $1(A \rightarrow B)$

Zuordnungsregeln Formeln / Unterformeln

α	α_1	α_2	β	β_1	β_2
$1(A \wedge B)$	$1A$	$1B$	$0(A \wedge B)$	$0A$	$0B$
$0(A \vee B)$	$0A$	$0B$	$1(A \vee B)$	$1A$	$1B$
$0(A \rightarrow B)$	$1A$	$0B$	$1(A \rightarrow B)$	$0A$	$1B$
$0\neg A$	$1A$	$1A$			
$1\neg A$	$0A$	$0A$			

Regeln des Tableauekalküls:

$\frac{\alpha}{\alpha_1 \alpha_2}$	konjunktiv	$1(p \wedge q)$ $\begin{array}{c} \\ 1p \\ \\ 1q \end{array}$
$\frac{\beta}{\beta_1 \mid \beta_2}$	disjunktiv	$1(p \vee q)$ $\begin{array}{c} \\ 1p \quad \backslash \\ \quad \quad \quad \\ \quad \quad \quad 1q \end{array}$
$\frac{1F}{0F} \quad \frac{01}{*} \quad \frac{10}{*}$	Widerspruch	$\begin{array}{ccc} 1F & 01 & 10 \\ & & \\ 0F & * & * \\ & & \\ * & & \end{array}$

Beachte: Disjunktionen führen zu Verzweigungen.

Instanzen der α - und β -Regel:

Instanzen der α -Regel				
$\frac{1(P \wedge Q)}{1P \quad 1Q}$	$\frac{0(P \vee Q)}{0P \quad 0Q}$	$\frac{0(P \rightarrow Q)}{1P \quad 0Q}$	$\frac{0\neg P}{1P}$	$\frac{1\neg P}{0P}$

Instanzen der β -Regel		
$\frac{1(P \vee Q)}{1P \mid 1Q}$	$\frac{0(P \wedge Q)}{0P \mid 0Q}$	$\frac{1(P \rightarrow Q)}{0P \mid 1Q}$

Sequenzkalkül

- geht zurück auf G. Gentzen 1935
- hat nichts mit "Folgen" zu tun
(engl.: *sequent*, nicht *sequence*)

Definition:

Eine *Sequenz* ist ein Paar endlicher Mengen von Formeln und wird notiert in der Form

$$\Gamma \Rightarrow \Delta.$$

Γ wird Antezedent und Δ Sukzedent genannt.

Sowohl links wie rechts vom Sequenzenpfeil \Rightarrow kann auch die leere Menge stehen. Wir schreiben dann

$$\Rightarrow \Delta \quad \text{bzw.} \quad \Gamma \Rightarrow \quad \text{bzw.} \quad \Rightarrow .$$

Ist $\Gamma = \{A_1, \dots, A_n\}$ und B eine Formel, so schreiben wir

$$\Gamma, B \quad \text{für die Menge} \quad \{A_1, \dots, A_n, B\},$$

Entsprechend werden

$$B, \Gamma \quad \Gamma, \Delta \quad \Gamma, B, \Delta$$

benutzt.

Semantik von Sequenzen:

Sei I eine Interpretation.

Definition: Auswertung von Sequenzen

$$\begin{aligned} \text{val}_I(\Gamma \Rightarrow \Delta) &= \text{val}_I(\bigwedge \Gamma \rightarrow \bigvee \Delta) \quad \text{für } \Gamma, \Delta \neq \emptyset \\ \text{val}_I(\Gamma \Rightarrow) &= \text{val}_I(\neg \bigwedge \Gamma) \\ \text{val}_I(\Rightarrow \Delta) &= \text{val}_I(\bigvee \Delta) \\ \text{val}_I(\Rightarrow) &= \mathbf{F} \end{aligned}$$

$$\begin{aligned} \bigwedge \Gamma &= \text{Konjunktion aller Formeln in } \Gamma \\ \bigvee \Gamma &= \text{Disjunktion aller Formeln in } \Gamma \end{aligned}$$

Die Konjunktion über die leere Folge wird demnach zu **W** und die Disjunktion über die leere Folge zu **F** ausgewertet.

Axiome und Regeln des Kalküls:

Axiome:

$$\Gamma, A, \Gamma' \Rightarrow \Delta, A, \Delta'.$$

Die Regeln sind

$$(\neg \text{links}) \frac{\Gamma, \Gamma' \Rightarrow A, \Delta}{\Gamma, \neg A, \Gamma' \Rightarrow \Delta}$$

$$(\wedge \text{links}) \frac{\Gamma, A, B, \Gamma' \Rightarrow \Delta}{\Gamma, (A \wedge B), \Gamma' \Rightarrow \Delta}$$

$$(\vee \text{links}) \frac{\Gamma, A, \Gamma' \Rightarrow \Delta \quad \Gamma, B, \Gamma' \Rightarrow \Delta}{\Gamma, (A \vee B), \Gamma' \Rightarrow \Delta}$$

$$(\rightarrow \text{links}) \frac{\Gamma, \Gamma' \Rightarrow A, \Delta \quad B, \Gamma, \Gamma' \Rightarrow \Delta}{\Gamma, (A \rightarrow B), \Gamma' \Rightarrow \Delta}$$

(Fortsetzung:)

$$(\neg\text{rechts}) \frac{A, \Gamma \Rightarrow \Delta, \Delta'}{\Gamma \Rightarrow \Delta, \neg A, \Delta'}$$

$$(\wedge\text{rechts}) \frac{\Gamma \Rightarrow \Delta, A, \Delta' \quad \Gamma \Rightarrow \Delta, B, \Delta'}{\Gamma \Rightarrow \Delta, (A \wedge B), \Delta'}$$

$$(\vee\text{rechts}) \frac{\Gamma \Rightarrow \Delta, A, B, \Delta'}{\Gamma \Rightarrow \Delta, (A \vee B), \Delta'}$$

$$(\rightarrow\text{rechts}) \frac{A, \Gamma \Rightarrow B, \Delta, \Delta'}{\Gamma \Rightarrow \Delta, (A \rightarrow B), \Delta'}$$

Zusätzliche Regeln für eine Menge M von Voraussetzungen

$$(\text{einfügen}_M) \frac{B, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta}$$

für $B \in M$

Def. Ableitbarkeit:

Für $A \in \text{For}_0$, $M \subseteq \text{For}_0$ sagen wir, dass A aus M ableitbar in $\mathbf{S0}$ ist,

$$M \vdash_{\mathbf{S0}} A,$$

gdw.

eine Ableitung von $\Rightarrow A$ aus M in $\mathbf{S0}$ existiert.

Satz (Korrektheit und Vollständigkeit):

$\mathbf{S0}$ ist korrekt und vollständig:

es gilt also

$$M \models A \iff M \vdash_{\mathbf{S0}} A.$$

Ausschnitte entnommen aus
Beckert (2010), Görz (2010), Otto (2010) und Schmitt (2008)
(genaue Quellenangaben siehe http://www.uni-forst.gwdg.de/~wkurth/fs10_lit.htm)