Einige Kalküle der Logik

Thomas Pajor
thomas.pajor@logn.de

27. Februar 2006

Dieses kleine Paper soll eine Zusammenfassung einiger Kalküle aus der Aussagenlogik und Prädikatenlogik erster Stufe sein. Kalküle dienen dazu innerhalb eines formalen Systems, beispielsweise eines Logiksystems, Aussagen zu beweisen oder zu widerlegen. Da dies ein wichtiger Bestandteil der Vorlesung „Formale Systeme“ an der Universität Karlsruhe ist, erhoffe ich mir durch die Erstellung des Papiers einen kleinen Lerneffekt für mich. Aber vielleicht hilft es auch dem einen oder anderen beim Verständnis. Ich setze hier die Grundbegriffe der Logik wie „erfüllbar“, „allgemeingültig“ usw. voraus. Genauso sollte man wissen wie die Aussagenlogik und Prädikatenlogik syntaktisch definiert sind und was eine Interpretation bzw. ein Modell ist.

Bereits für die Prädikatenlogik zweiter Ordnung existieren keine vollständigen und korrekten Kalküle mehr. Das ist sehr schade, da mathematische Aussagen mit denen wir im Allgemeinen arbeiten oft durch PL2 Formeln beschrieben werden. Aber vielleicht hat das auch etwas Gutes, denn sonst könnte man jede Aussage von einem automatischen Beweisersystem beweisen lassen, und somit wären sehr viele Mathematiker arbeitslos.

Ich wünsche euch viel Spaß mit dem Papier, und falls ihr Anregungen oder Verbesserungsvorschläge habt, so schickt mir am besten eine E–Post.

Inhaltsverzeichnis

1 Abstrakte Kalküle

Bevor wir auf einige konkrete Kalküle in der Aussagen– sowie Prädikatenlogik eingehen wollen, betrachten wir erst einmal abstrakt was ein Kalkül ist. Dieses Kapitel ist etwas schwierig. Einerseits wollte ich es nicht zu formal machen, andererseits ist das ein Thema Formalismus ansich, und da ist es schwierig exakt zu bleiben und dabei gleichzeitig ohne zu viel Formalismus auszukommen.

Definition. Betrachte ein Alphabet Σ und L als eine entscheidbare Sprache über Σ. Da L immer über Σ definiert wird, fassen wir die Beiden als Tupel zusammen und nennen (Σ, L) einen *syntaktischen Formalismus*.

Beispiel. L kann in einem konkreten Fall zum Beispiel die Sprache aller gültigen Formeln der Aussagenlogik sein, dann wäre $\Sigma := \{0, 1, \neg, \wedge, \vee, \leftrightarrow, \rightarrow, (,)\} \cup P$ wobei P die Menge der aussagenlogischen Atome (oder Variablen ist). Die Sprache L der gültigen Formeln könnte man dann induktiv wie folgt definieren:

1. $1, 0 \in L$ und $P \subseteq L$
2. Mit A, B sind auch

   $$\neg A, (A \wedge B), (A \vee B), (A \leftrightarrow B), (A \rightarrow B)$$

   in L

Man sieht dass diese Sprache kontextfrei im Sinne der Chomsky–Hierarchie ist, das heißt sie ist insbesondere entscheidbar, und erfüllt damit unsere Kriterien. Siehe dazu auch [Sch05].

Definition. Ein *Kalkül* zu solch einem syntaktischen Formalismus (Σ, L) ist nun eine Menge von *Regeln* – formal eine Menge von Relationen über L. Betrachten wir nun eine solche n–stellige Regel $R \subseteq L^{n+1}$. Für eine *Instanz* $(u_1, \ldots, u_n, u_{n+1}) \in R$ der Regel R heißen

1. $u_1, \ldots, u_n$ die *Prämisse*
2. u_n die *Conclusio*

der Instanz. Instanzen von Nullstelligen Relationen heißen auch *Axiom*.

Wenn wir einen Kalkül definieren, definieren wir uns eine Regel häufig auf abstrakte Weise, da die Relationen (Regeln) im Allgmeinen unendlich viele Instanzen enthalten. Der Modus Ponens im aussagenlogischen Hilbertkalkül wird dann wie folgt definiert:

$$MP := \frac{A, A \rightarrow B}{B} \qquad A, B \in L$$

Diese schreibweise ist eher üblich als die normale Schreibweise zur Definition einer Relation:

$$MP := \{(A, A \rightarrow B, B) \mid A, B \in L\}$$

Man beachte dass in diesem Beispiel Fall $L := \mathrm{For0}$ gilt.

Nun stellt sich die Frage was uns so ein Kalkül bringt. Unser Ziel wird sein, später mit Hilfe eines konkret definierten Kalküls Beweise führen zu können. Das heißt wir haben, je nach Kalkül, eine Menge von Formeln oder auch nur eine Formel und möchten zum Beispiel deren Unerfüllbarkeit beweisen. Dies geschieht durch Anwendung mehrerer Regeln mit Hilfe derer sich aus einer Formelmenge neue Formeln ableiten lassen. Auf abstrakter Ebene sind solche „Formeln" nichts weiter als Wörter in der Sprache L und so können wir den Begriff der Ableitung wie folgt definieren.

Definition. Zu einem syntaktischen Formalismus (Σ, L) sei Kal ein Kalkül und $M \subseteq L$ eine Menge von Wörtern aus L. Dann ist eine *Ableitung* eine Folge

$$(u_1, \ldots, u_m)$$

von Wörtern in L, so dass stets einer der folgenden Fälle gilt

(1) u_i ist ein Axiom

(2) $u_i \in M$

(3) u_i geht geht durch Anwendung einer (höchstens i–stelligen) Regel aus den vorangegangen u_k hervor ($k < i$). Formal bedeutet das, dass es eine n–stellige Relation $R \in Kal$ geben muss mit $1 \leq n < i$ und eine Menge von Indizes $j_1, \ldots, j_n \in \{1, \ldots, i-1\}$ so dass $(u_{j_1}, \ldots, j_{j_n}, u_i) \in R$.

Für u_1 gilt insbesondere, dass u_1 aus einem Axiom hervorgeht oder selbst Element der Menge M ist.

Definition. Jetzt können wir auch definieren was es heißt, dass ein Wort w ableitbar ist aus einer Menge von Wörtern M. Es gilt

$$M \vdash_{Kal} w$$

also „w ist ableitbar aus M" genau dann wenn es eine Ableitungsfolge $(u_1, \ldots, u_m)$ in dem Kalkül Kal gibt, wobei $u_m = w$ ist.

Definition. Betrachten wir nun (allgemeine) Kalküle für Logiksysteme für die wir eine semantische Folgerung definiert haben (zum Beispiel Aussagenlogik, Prädikatenlogik, und so weiter). Dann nennen wir einen Kalkül Kal *korrekt* wenn für beliebige Formelmengen $L \subseteq M$ gilt

$$M \vdash_{Kal} A \quad \Rightarrow \quad M \models A \qquad A \in L$$

Analog nennen wir einen Kalkül Kal vollständig wenn gilt

$$M \models A \quad \Rightarrow \quad M \vdash_{Kal} A \qquad A \in L$$

Das Ziel ist es korrekte und vollständige Kalküle zu finden. So viel aber jetzt zur Theorie von Kalkülen. Im nächsten Abschnitt kommen einige wichtige Kalküle der Aussagenlogik konkret vorgestellt mit Beispielen.

2 Kalküle der Aussagenlogik

In den folgenden Kapiteln ist Σ nicht ein Alphabet sondern eine Signatur. Somit wäre $\mathrm{For0}_\Sigma$ die Menge aller aussagenlogischen Formeln über der Signatur Σ.

2.1 Resolutionskalkül

Der *Resolutionskalkül* der Aussagenlogik ist der einfachste Kalkül. Er operiert nicht direkt auf $\mathrm{For0}_\Sigma$, sondern auf einer Menge von Klauseln. Eine Klausel ist dabei ganz normal eine Disjunktion von Literalen über Σ. Um die Brücke zu den abstrakten Kalkülen zu schlagen, ist L diesmal die Menge aller Klauseln, also eine Einschränkung von $\mathrm{For0}_\Sigma$. Für die leere Klausel schreiben wir nicht $\emptyset$, sondern bezeichnen sie mit $\Box$.

Wir interpretieren eine Menge

$$M = \{C_1, \ldots, C_n\}$$

von n Klauseln als die Konjunktion ihrer Klauseln. Jede Menge M von Klauseln entspricht also einer Formel $\phi \in \mathrm{For0}_\Sigma$ in konjunktiver Normalform.

Wir können den Resolutionskalkül nun dazu benutzen um eine Formeln $\phi \in \mathrm{For0}_\Sigma$ zu widerlegen, also zu zeigen dass sie nicht erfüllbar ist. Dabei wird es unser Ziel sein aus der vorhandenen Menge M von Klauseln (die durch ϕ induziert wird) die leere Klausel $\Box$ abzuleiten. Dabei gibt es genau eine Regel:

Definition (Regeln im Resolutionskalkül)**.** Seien $P \in \Sigma$ und C_1, C_2 Klauseln über Σ. Dann existiert folgende Regel

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

Dabei nennt man auch den unteren Teil $C_1 \cup C_2$ die *Resolvente*.

Man bemerke, dass dieser Kalkül keine Axiome hat.

Um mit dem Kalkül zu beweisen dass eine Formel $\phi \in \mathrm{For0}_\Sigma$ allgemeingültig ist, reicht es natürlich zu zeigen dass $\neg\phi$ nicht erfüllbar ist. Kommen wir nun zu einem kleinen Beispiel, um das Ganze zu verdeutlichen.

Beispiel (Entnommen aus [Sch05])**.** Wir wollen zeigen, dass $\phi := (A \to B) \to ((B \to C) \to (A \to C))$ eine allgemeingültige Formel der Aussagenlogik ist. Dazu müssen wir die Formel $\neg\phi$ in konjunktive Normalform bringen, also

$$\begin{aligned}\neg\phi &= \neg\big((A \to B) \to ((B \to C) \to (A \to C))\big)\\ &\equiv \neg\big(\neg(\neg A \vee B) \vee (\neg(\neg B \vee C) \vee (\neg A \vee C))\big)\\ &\equiv (\neg A \vee B) \wedge (\neg B \vee C) \wedge (A) \wedge (\neg C)\end{aligned}$$

Übersetzt in eine Menge von Klauseln ist das also

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

Wir können nun folgende Folge von Ableitungen angeben, um die leere Klausel herzuleiten.

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

Zur Schreibweise: Die erste Spalte bezeichnet den Index in der Ableitungsfolge. Die zweite Spalte ist nur zur verdeutlichung aus welchen zwei Klauseln die jeweilige Klausel erzeugt wurde; Die verwendete Regel ist ja eindeutig. Und die letzte Spalte ist die jeweilig erzeugte neue Klausel.

Die ersten vier Ableitungsschritte ist der Fall (2) der abstrakten Definition der Ableitung ($\rightsquigarrow$ 1). Danach benutzen wir den Fall (3) in dem wir die Resolutionsregel tatsächlich auf vorangegangene Klauseln anwenden und somit neue Klauseln erzeugen.

Wir haben damit also bewiesen, dass $\neg\phi$ unerfüllbar, bzw. ϕ allgemeingültig ist.

2.2 Das DAVIS–PUTNAM–LOVELAND Verfahren

Das DAVIS–PUTNAM–LOVELAND Verfahren ist ein schneller Algorithmus der auf dem Resolutionskalkül beruht. Wir operieren also wieder auf eine Klauselmenge M und können den Algorithmus dazu benutzen eine Formel ϕ zu widerlegen. Leite dazu entsprechend dem Resolutionskalkül die konjunktive Normalform von ϕ her, und benutze als Input M die Klauselmenge, die durch ϕ_{KNF} induziert wird.

Algorithmus 1 beschreibt das Verfahren.

Beispiel. kommt noch :-)

2.3 Tableau Kalkül

Der Tableau Kalkül ist ein etwas komplizierterer Kalkül für die Aussagenlogik. Während wir beim Resolutionskalkül auf lineare Art und Weise vorgehen, ist es beim Tableau Kalkül Ziel einen Baum zu erstellen, dessen Wurzel die zu widerlegende Formel enthält, und jeder Pfad von der Wurzel zu einem der Blätter *geschlossen* werden muss. Mehr zu diesen Begriffen später, zunächst sind noch einige Definitionen nötig.

Algorithm 1 WIDERLEGE(M)

if $M = \emptyset$ **then**
  STOP. (S ist erfüllbar)
repeat
  if S enhält keine Einerklausel **then**
    Wähle beliebige Variable P
    WIDERLEGE($M \cup \{P\}$)
    WIDERLEGE($M \cup \{\neg P\}$)
  else
    Wähle Einerklausel $K \in M$
    Lösche alle Klauseln, die K als Teilklausel enthalten
    Lasse in allen übrigen Klauseln das zu K komplementäre Literal weg
until $\square \in M$
STOP. (S ist nicht erfüllbar)

Definition (Vorzeichenformel)**.** Eine Formel $\mathbf{0}A$ bzw. $\mathbf{1}A$ mit $A \in \mathrm{For0}_\Sigma$ wollen wir als *Vorzeichenformel* bezeichnen. Ein solches Vorzeichen tritt in jeder Formel nur einmal am Anfang auf.

Damit diese Vorzeichen auch einen Sinn machen, erweitern wir die Auswertungsfunktion auf Vorzeichenformeln:

$$\mathrm{val}_I(\mathbf{1}A) = \mathbf{W} \quad \Leftrightarrow \quad \mathrm{val}_I(A) = \mathbf{W}$$

und

$$\mathrm{val}_I(\mathbf{0}A) = \mathbf{W} \quad \Leftrightarrow \quad \mathrm{val}_I(\neg A) = \mathbf{W}$$

Salopp gesagt, können wir diese Formeln als „bejahte“ bzw. „beneinte“ Formeln auffassen.

Ist A aus der oberen Definition eine atomare Formel, also ist $A \in \Sigma$, so nennen wir $\mathbf{1}A$ oder $\mathbf{0}A$ auch eine *elementare Vorzeichenformel*. Es existieren jedoch noch folgende andere Typen von Formeln:

Definition. Folgende Typen von Vorzeichenformeln gibt es außerdem, vermöge $B, C \in \mathrm{For0}_\Sigma$:

Typ α	(konjunktiver Typ)	$\mathbf{0}\neg B$, $\mathbf{1}\neg B$, $\mathbf{1}(V \wedge C)$, $\mathbf{0}(B \vee C)$, $\mathbf{0}(B \rightarrow C)$
Typ β	(disjunktiver Typ)	$\mathbf{0}(B \wedge C)$, $\mathbf{1}(B \vee C)$, $\mathbf{1}(B \rightarrow C)$

Nun wird es langsam Zeit zu den Tableau Regeln zu gelangen. Wie schon erwähnt konstruiert man in einem Tableau–Beweis einen Baum beginnend bei der Wurzel. Dies führt uns zu der allgemeinen Definition von Tableau Regeln:

Definition (Regeln im Tableau Kalkül)**.** Eine allgemeine Tableau–Regel hat folgende Gestalt

$$\frac{\phi}{\begin{array}{ccc} \psi_{1,1} & \cdots & \psi_{n,1} \\ \vdots & \ddots & \vdots \\ \psi_{1,k_1} & \cdots & \psi_{n,k_n} \end{array}}$$

Dabei sind sowohl ϕ alsauch alle $\psi_{i,j}$ Vorzeichenformeln. Die Tableau Regel ist nur dann wohldefiniert, wenn für jede Interpretation I gilt

$$\mathrm{val}_I(\phi) = \mathbf{W} \quad \Leftrightarrow \quad \exists 1 \leq i \leq n : \forall 1 \leq j \leq k_i : \mathrm{val}_I(\psi_{i,j}) = \mathbf{W}$$

Es muss also mindestens eine Spalte existieren so dass in dieser Spalte alle Einträge zu wahr auswerten, und zwar genau dann wenn auch ϕ zu wahr auswertet. Man kann sich das vielleicht leichter vorstellen, wenn man fordert dass gelten muss

$$\emptyset \models \phi \leftrightarrow (\psi_{1,1} \wedge \ldots \wedge \psi_{1,k_1}) \vee \ldots \vee (\psi_{n,1} \wedge \ldots \wedge \psi_{n,k_n})$$

Unmittelbar aus dieser Definition können wir nun Tableau Regeln für unsere α und β Formeln herleiten.

Korollar. *Es gelten folgende Tableau Regeln für die oben definierten α Formeln:*

$$\frac{\mathbf{0}\neg B}{\mathbf{1}B} \qquad \frac{\mathbf{1}\neg B}{\mathbf{0}B} \qquad \frac{\mathbf{1}(B \wedge C)}{\begin{array}{c}\mathbf{1}B\\ \mathbf{1}C\end{array}} \qquad \frac{\mathbf{0}(B \vee C)}{\begin{array}{c}\mathbf{0}B\\ \mathbf{0}C\end{array}} \qquad \frac{\mathbf{0}(B \rightarrow C)}{\begin{array}{c}\mathbf{1}B\\ \mathbf{0}C\end{array}}$$

Für β Formeln gelten diese Regeln:

$$\frac{\mathbf{0}(B \wedge C)}{\mathbf{0}B \quad \mathbf{0}C} \qquad \frac{\mathbf{1}(B \vee C)}{\mathbf{1}B \quad \mathbf{1}C} \qquad \frac{\mathbf{1}(B \rightarrow C)}{\mathbf{0}B \quad \mathbf{1}C}$$

Analog lassen sich natürlich noch weitere Regeln konstruieren. Zum Beispiel für die Äquivalenz

$$\frac{\mathbf{1}(A \leftrightarrow B)}{\begin{array}{cc}\mathbf{1}A & \mathbf{0}A\\ \mathbf{1}B & \mathbf{0}B\end{array}} \qquad \frac{\mathbf{0}(A \leftrightarrow B)}{\begin{array}{cc}\mathbf{1}A & \mathbf{0}A\\ \mathbf{0}B & \mathbf{1}B\end{array}}$$

Kommen wir nun endlich zur Konstruktion des Tableaus. Der Tableau Kalkül ist wie der Resolutionskalkül ein *Widerlegungskalkül*, das heißt wir beweisen die Unerfüllbarkeit einer Formel ϕ. Wir schreiben zunächst $\mathbf{1}\phi$ in die Wurzel des Baumes. Wollen wir im Gegensatz zeigen, dass ϕ eine Tautologie ist, so ist das wegen der Erweiterung der val Funktion äquivalent dazu dass $\mathbf{0}\phi$ unerfüllbar ist, beginne in dem Fall also mit $\mathbf{0}\phi$ in der Wurzel.

Wir wenden nun sukzessive auf die Formel eine der Tableau Regeln von außen nach innen an. Bei einer α Regel entstehen dabei auf dem aktuellen Pfad zwei neue Knoten untereinander. Bei der Anwendung einer β Regel spaltet sich der Pfad auf. Siehe dazu auch Abbildung 1. Dabei entstehen nach und nach Pfade π von der Wurzel zu den jeweiligen Blattknoten. Die Anwendung

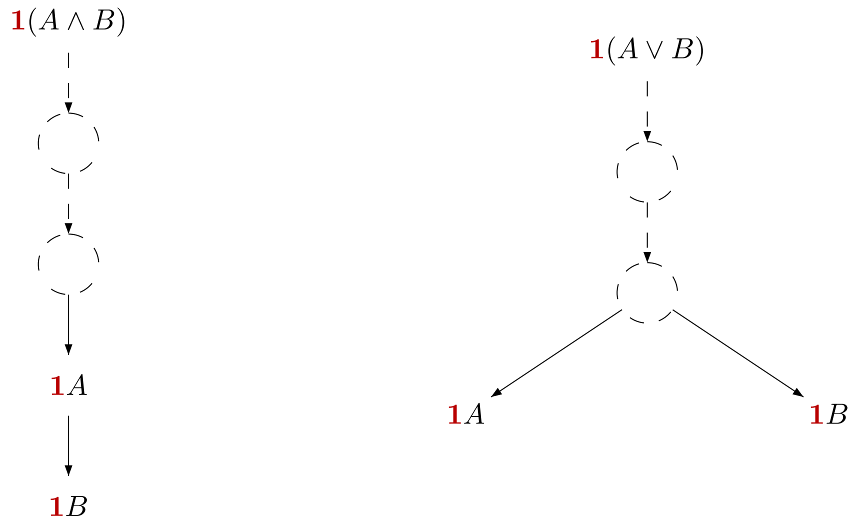

Abbildung 1: Anwendung einer α Regel (links) und einer β Regel (rechts) entlang eines Pfades π

einer Regel muss sich dabei nicht immer auf die Formel im untersten Knoten auf einem Pfad π beziehen, sondern kann auf einen beliebigen Knoten auf π angewendet werden.

Wir können einen Pfad π genau dann *schließen* wenn $P \in \Sigma$ eine beliebige atomare Formel ist, für die sowohl $\mathbf{1}P$ alsauch $\mathbf{0}P$ auf π vorkommen. Siehe dazu auch Abbildung 2. Lassen sich alle Pfade des Tableaus schließen, so konnten wir unsere ursprüngliche Formel ϕ (die in der Wurzel steht) widerlegen. Lässt sich ein Pfad nicht schließen, so liefert uns dieser Pfad eine erfüllende Interpretation I für die Formel ϕ, nämlich

$$I(P) := \begin{cases} \mathbf{W} & \text{falls } \mathbf{1}P \in \pi \\ \mathbf{F} & \text{falls } \mathbf{0}P \in \pi \\ & \text{beliebig sonst} \end{cases}$$

Kommen wir nun zu einem Beispiel.

Beispiel. Wir möchten Zeigen, dass die Formel

$$\phi := (P \vee (Q \wedge R)) \rightarrow ((P \vee Q) \wedge (P \vee R))$$

eine Tautologie ist. Das heißt wir möchten $\neg\phi$ mit dem Tableau Kalkül widerlegen. Dies liefert zum Beispiel das Tableau aus Abbildung 3.

$$
\begin{tikzcd}
\mathbf{1}\phi \arrow[d, dashed] \\
\mathbf{1}P \arrow[d, dashed] \\
\mathbf{0}P \arrow[d] \\
*
\end{tikzcd}
$$

Abbildung 2: Schließung eines Pfades π. Sowohl $\mathbf{1}P$ alsauch $\mathbf{0}P$ ($P \in \Sigma$) müssen irgendwo auf π vorkommen.

2.4 Sequenzenkalkül

Der Sequenzenkalkül ist der hochgradigste Kalkül für die Aussagenlogik in diesem Papier. Er ist außerdem kein Widerlegungskalkül, sondern man kann mit ihm direkt beweisen dass eine Formel allgemeingültig ist. Zunächst müssen wir uns jedoch den Begriff der Sequenz definieren:

Definition. Eine *Sequenz* ist eine Folge zweier endlicher Mengen. Wir bezeichnen die Mengen mit Γ und Δ und schreiben für die Sequenz

$$\Gamma \rightarrowtail \Delta$$

Der linke Teil, also Γ, wird auch *Antezedent* und der rechte Teil, also Δ, wird *Sukzedent* genannt.

Die Mengen werden später endliche Mengen von aussagenlogischen Formeln sein. Wie auch beim Tableau Kalkül müssen wir die Auswertungsfunktion val erweitern, und zwar durch

$$\mathrm{val}_I(\Gamma \rightarrowtail \Delta) = \mathrm{val}_I(\bigwedge \Gamma \to \bigvee \Delta)$$

Wir interpretieren die Menge Γ also als die Konjunktion ihrer Elemente, und Δ als die Disjunktion ihrer Elemente. Den Sequenzpfeil interpretieren wir als normale Implikation.

Wir definieren zunächst erst einmal die Regeln für den Sequenzenkalkül.

Definition (Regeln im Sequenzenkalkül)**.** Folgende Regeln werden definiert and–left: $\dfrac{\Gamma, F, G \rightarrowtail \Delta}{\Gamma, F \wedge G \rightarrowtail \Delta}$ and–right: $\dfrac{\Gamma \rightarrowtail F, \Delta \quad \Gamma \rightarrowtail G, \Delta}{\Gamma \rightarrowtail F \wedge G, \Delta}$

or–left: $\dfrac{\Gamma, F \rightarrowtail \Delta \quad \Gamma, G \rightarrowtail \Delta}{\Gamma, F \vee G \rightarrowtail \Delta}$ or–right: $\dfrac{\Gamma \rightarrowtail F, G, \Delta}{\Gamma \rightarrowtail F \vee G, \Delta}$

impl–left: $\dfrac{\Gamma \rightarrowtail F, \Delta \quad \Gamma, G \rightarrowtail \Delta}{\Gamma, F \rightarrow G \rightarrowtail \Delta}$ impl–right: $\dfrac{\Gamma, F \rightarrowtail G, \Delta}{\Gamma \rightarrowtail F \rightarrow G, \Delta}$

not–left: $\dfrac{\Gamma \rightarrowtail F, \Delta}{\Gamma, \neg F \rightarrowtail \Delta}$ not–right: $\dfrac{\Gamma, F \rightarrowtail \Delta}{\Gamma \rightarrowtail \neg F, \Delta}$

Außerdem gibt es noch ein *Axiom* wie folgt

$$\overline{\Gamma, F \rightarrowtail F, \Delta}$$

Die Regeln mögen jetzt wie vom Himmel gefallen erscheinen und keinen Sinn ergeben. Denkt man jedoch etwas darüber nach[1], so machen sie dennoch ein wenig Sinn. Die and–left Regel beispielsweise verknüpft einfach zwei Formeln, die auf der linken Seite einer Sequenz stehen durch eine Konjunktion. Dies macht unter dem Hintergrund der Auswertungsfunktion, die alle Elemente der linken Seite als Konjunktion auffasst, durchaus Sinn. Analoges gilt zum Beispiel für die or–right Regel — Hier wertet die val Funktion die rechte Seite ja sowieso als Disjunktion.

Der „natürliche" Weg mit diesem Kalkül einen Beweis für die Allgemeingültigkeit einer Formel ϕ zu führen besteht darin, mit Hilfe des Axioms diverse Formeln zu erzeugen, so dass diese durch Regelanwendung zu der Sequenz

$$\emptyset \rightarrowtail \phi$$

führen. Es ist wohl einsichtlich dass dieser Weg nicht gerade angenehm ist, da man die Sequenzen, die man mit Hilfe des Axioms erzeugen kann „raten" müsste, damit der Beweis aufgeht.

Wir bedienen uns also eines Tricks und lesen die Regeln rückwärts. Wir beginnen also mit der Sequenz

$$\emptyset \rightarrowtail \phi$$

und wenden die Regeln einfach rückwärts an (von unten nach oben lesen!). Dabei kann es passieren dass wir aus einer Sequenz zwei erzeugen, daher empfiehlt es sich das Ganze ähnlich zum Tableau–Kalkül wieder in Baumstruktur zu notieren. Jeder Pfad π in diesem Baum muss nun in den Blättern mit Hilfe eines Axioms „geschlossen" werden können, unser Ziel wird es also sein darauf hinzuarbeiten, auf jedem Pfad eine Sequenz zu finden, die eine Formel F sowohl auf der linken alsauch auf der rechten Seite enthält.

Zeit für ein Beispiel um den Kalkül zu verdeutlichen.

[1] Bitte nicht zu lange!

Beispiel. Wir möchten wieder die Allgemeingültigkeit der Formel

$$\phi := (P \vee (Q \wedge R)) \rightarrow ((P \vee Q) \wedge (P \vee R))$$

beweisen. Dabei gehen wir rückwärts im Sequenzenkalkül vor, und leiten aus

$$\emptyset \rightarrowtail (P \vee (Q \wedge R)) \rightarrow ((P \vee Q) \wedge (P \vee R))$$

Sequenzen her, auf die wir das Axiom anwenden können.

Ein Beispiel Ableitungsbaum ist in Abbildung 4 zu sehen.

2.5 Numerisches Verfahren

Dieses Verfahren stellt weniger einen Kalkül dar, als eine Art Reduktion auf ein anderes Problem. Statt eine Formel $\phi \in \mathrm{For0}_\Sigma$ zu widerlegen, versuchen wir für ein lineares Ungleichungssystem zu widerlegen dass es in den ganzen Zahlen lösbar ist. Es gilt dann

$$\phi \text{ ist erfüllbar} \quad \Leftrightarrow \quad U(\phi) \text{ ist in den ganzen Zahlen lösbar}$$

wobei U ein lineares Ungleichungssystem ist. Wie wir U definieren müssen, liefert die folgende Definition.

Definition. Sei $\phi = C_1 \wedge \ldots \wedge C_n$ eine Formel der Aussagenlogik in konjunktiver Normalform. Die C_i sind also Klauseln über Σ. Definiere zu jeder Klausel eine Summe U_i indem ein positives Literal P_j ersetzt wird durch X_j, und ein negiertes Literal $\neg P_j$ wird ersetzt durch $(1 - X_j)$. Die Disjunktionen werden durch Summen ersetzt. Definiere dazu folgende Ungleichungen

$$U_i \geq 1 \qquad i = 1, \ldots, n$$

Zudem wollen wir den Wertebereich der Variablen X_j einschränken. Definiere also für jedes Atom folgende Ungleichungen

$$0 \leq X_j \leq 1 \qquad j = 1, \ldots, |\Sigma|$$

Die Menge der so erzeugten Ungleichungen sei das Ungleichungssystem $U(\phi)$ zur Formel ϕ.

Man kann nun versuchen dieses Ungleichungssystem (auf naive Art) zu Lösen. Gelangt man dabei auf einen Widerspruch, also ist $U(\phi)$ nicht lösbar in den ganzen Zahlen, so war die Formel ϕ widersprüchlich.

Beispiel. Wir wollen zeigen, dass die Formel

$$\phi := (A \vee B \vee \neg C) \wedge (\neg A) \wedge (A \vee B \vee C) \wedge (A \vee \neg B)$$

unerfüllbar ist.

Anwendung des obigen Verfahrens zur Herstellung des Ungleichungssystems $U(\phi)$ liefert

$$a + b + (1 - c) \geq 1 \tag{1}$$
$$(1 - a) \geq 1 \tag{2}$$
$$a + b + c \geq 1 \tag{3}$$
$$a + (1 - b) \geq 1 \tag{4}$$

Außerdem die Gleichungen $0 \leq a \leq 1$, $0 \leq b \leq 1$, $0 \leq c \leq 1$, die die Wertebereiche für a, b und c einschränken.

Umformen der Gleichung (2) ergibt $a \leq 0$, woraus $a = 0$ folgt. Dies liefert die folgende Situation (in den Gleichungen wurde a durch 0 ersetzt.

$$b + (1 - c) \geq 1 \tag{5}$$
$$a = 0 \tag{6}$$
$$b + c \geq 1 \tag{7}$$
$$(1 - b) \geq 1 \tag{8}$$

Umformen der Gleichung (8) führt zu $b \leq 0$, also auch $b = 0$. Es folgt

$$(1 - c) \geq 1 \tag{9}$$
$$a = 0 \tag{10}$$
$$c \geq 1 \tag{11}$$
$$b = 0 \tag{12}$$

Die Gleichung (9) lässt sich umformen zu $c \leq 0$. Das ist offensichtlich ein Widerspruch zu Gleichung (11), in der $c \geq 1$ gefordert wird.

Das Gleichungssystem besitzt somit keine Lösung in $\mathbb{N}$, und daher ist ϕ unerfüllbar.

3 Kalküle der Prädikatenlogik erster Ordnung

Hier stelle ich einige Kalküle für die Prädikatenlogik erster Stufe vor. Die Syntax und Semantik der PL1 kann in [Sch05] ab Seite 85 nachgelesen werden.

3.1 Resolutionskalkül

Der Resolutionskalkül der PL1 ist sehr ähnlich zum Resolutionskalkül für die Aussagenlogik, jedoch ist hier mehr Vorarbeit nötig.

Wie auch in der Aussagenlogik operieren wir auch hier wieder auf einer Klauselmenge. Ebenso handelt es sich bei dem Resolutionskalkül um einen Widerlegungskalkül. Wir können mit dem Kalkül jedoch nicht nur eine einzelne Formel widerlegen, sondern wir können sogar für eine Menge von Formeln $M \subseteq \mathrm{For}_\Sigma$ und $A \in \mathrm{For}_\Sigma$ zeigen, dass gilt

$$M \models A$$

Zunächst müssen jedoch noch einige Begriffe eingeführt werden. Im Unterschied zur Aussagenlogik bezeichnet ein Literal hier nicht nur aussagenlogische Atome, sondern jedes Atom aus At_Σ, bzw. dessen Negat. Insbesondere ist für $p \in P_\Sigma$ zum Beispiel auch $\neg p(x, y)$ ein Literal. Dies führt zu dem Begriff der Variante:

Definition. Ist $A \in \mathrm{For}_\Sigma$ eine quantorenfreie Formel und σ eine Variablenumbenennung. Dann heißt $\sigma(A)$ eine *Variante* von A.

Wir führen außerdem noch eine Notation für „das komplementäre Literal“ ein. Ist also L ein Literal, dann ist $\sim L$ das zu L komplementäre Literal. Genauso ist zu einer Klausel C die Klausel $\sim C$ die Menge aller komplementären Literale von C.

Der Resolutionskalkül besteht nun wie auch schon in der Aussagenlogik aus nur einer einzigen Regel, jedoch ist diese auf Anhieb vielleicht ein wenig abschreckender. Also ganz langsam:

Definition (Resolution)**.** Seien C_1, C_2, K_1, K_2 Klauseln über eine Klauselmenge $\mathcal{K}$ wobei $K_1, K_2 \neq \square$. Außerdem haben $C_1 \cup K_1$ und $C_2 \cup K_2$ *keine* gemeinsamen Variablen, also

$$\mathrm{Var}(C_1 \cup K_1) \cap \mathrm{Var}(C_2 \cup K_2) = \emptyset$$

Weiterhin ist μ allgemeinster Unifikator von $K_1 \cup \sim K_2$, dann lässt sich folgende Regel anwenden

$$\frac{C_1 \cup K_1 \qquad C_2 \cup K_2}{\mu(C_1 \cup C_2)}$$

Obwohl der Kalkül es erlaubt auf einmal mehrere Literale zu resolvieren, empfiehlt es sich das Schrittweise mit einem Literal zu machen. Wir suchen also zwei Klauseln mit zwei komplementären Literalen. Wichtig ist, dass die Klauseln keine gemeinsamen Variablen haben, ansonsten bilden wir einfach eine Variante einer Klausel. Wir suchen dann einen allgemeinsten

Unifikator μ, vermöge K_1 und K_2 die komplementären Klauseln, der $K_1 \cup \sim K_2$ unifiziert. Dann können wir K_1 und K_2 aus unseren beiden Klauseln rausschmeißen, und erhalten so eine neue Klausel durch $\mu(C_1 \cup C_2)$.

Die große Frage, die noch aussteht ist: Wie kommt man überhaupt von einer allgemeinen Formel zu der Klauselmenge $\mathcal{K}$? Wie eingangs schon erwähnt ist dazu etwas mehr vorarbeit nötig. Am besten ist, man hält sich an folgendes Schema. Sei $M \subseteq \mathrm{For}_\Sigma$ und $A \in \mathrm{For}_\Sigma$. Wir möchten den Resolutionskalkül dazu benutzen zu beweisen dass $M \models A$ gilt. Für den Fall dass für eine Formel A die Allgemeingültigkeit zu zeigen ist, wähle einfach $M = \emptyset$.

(1) Definiere eine Menge $E := M \cup \{\neg A\}$.

(2) Bringe alle Elemente aus E in Negationsnormalform und bilde für freie Variablen den All–Abschluss. Dies führt zu der neuen Menge E'.

(3) Bringe nun jede Formel aus E' in Skolemnormalform (vermutlich über die Pränexnormalform). Bringe außerdem die Matrizen der Formeln in konjunktive Normalform. Dies liefert E''.

(4) Lasse die Allquantoren fallen (Existenzquantoren haben wir ja keine) und bilde aus den Matrizen der Formeln Klauselmengen die vereinigt dann gerade $\mathcal{K}$ ergeben.

Nun können wir den Resolutionskalkül auf $\mathcal{K}$ anwenden und versuchen über Variantenbildung und Resolution die leere Klausel herzuleiten.

Beispiel. Wir möchten für folgende Formeln ϕ und ψ beweisen, dass $\phi \models \psi$ gilt. Die Formeln sind dabei wie folgt definiert:

$$\begin{aligned}\phi := {} & \forall x(p(a,x)) \wedge \forall x(\forall y(\forall z(p(x,y) \wedge p(y,z) \to p(x,z)))) \wedge \\ & \forall x(\forall y(p(x,y) \to p(f(x),f(y))))\end{aligned}$$

und

$$\psi := p(a, f(f(a)))$$

Wir erzeugen also zunächst unsere Menge $E := \{\phi, \neg\psi\}$. Transformieren von ϕ in Skolemnormalform mit Matrix in KNF ergibt

$$\begin{aligned}\phi_{\text{snf}} = {} & \forall x_1 \forall x_2 \forall x_3 \forall y_1 \forall y_2 \forall z_1 p(a,x_1) \wedge (\neg p(x_2,y_1) \vee \neg p(y_1,z_1) \vee p(x_2,z_1)) \wedge \\ & (\neg p(x_3,y_2) \vee p(f(x_3),f(y_2)))\end{aligned}$$

Die Formel $\neg\psi$ ist bereits in geeigneter Form. Daraus können wir nun folgende Klauselmenge ableiten:

$$\begin{aligned}\mathcal{K} = \big\{ & \{p(a,x_1)\}, \\ & \{\neg p(x_2,y_1), \neg p(y_1,z_1), p(x_2,z_1)\}, \\ & \{\neg p(x_3,y_2), \neg p(f(x_3),f(y_2))\}, \\ & \{\neg p(a,f(f(a)))\}\big\}\end{aligned}$$

Folgender Resolutionsbeweis liefert dann zum Beispiel die leere Klausel

(1)			$\{p(a, x_1)\}$
(2)			$\{\neg p(x_2, y_1), \neg p(y_1, z_1), p(x_2, z_1)\}$
(3)			$\{\neg p(x_3, y_2), \neg p(f(x_3), f(y_2))\}$
(4)			$\{\neg p(a, f(f(a)))\}$
(5)	[1,4]	$\mu := \{x_1/f(f(a))\}$	$\square$

Zugegebenermaßen ist das Beispiel etwas pathologisch, aber es lässt sich dennoch gut erkennen wie die Resolution funktioniert. Im allgemeinen Fall sind natürlich mehr Schritte notwendig um die leere Klausel abzuleiten.

3.2 Tableau Kalkül

Der Tableau Kalkül für die Prädikatenlogik baut stark auf dem Tableau Kalkül der Aussagenlogik auf, daher sollte man sich zunächst den zu Gemüte führen bevor man die prädikatenlogische Variante studieren möchte. Wie auch in der Aussagenlogik bedienen wir uns *Vorzeichenformeln*. Somit sind $\mathbf{1}\phi$ und $\mathbf{0}\phi$ für $\phi \in \mathrm{For}_\Sigma$ gültige Vorzeichenformeln. Elementare Vorzeichenformeln sind diesmal Formeln $\mathbf{1}A$ und $\mathbf{0}A$ wobei A eine atomare Formel ist. Es existieren jetzt jedoch noch neue Typen von Formeln:

Definition. Folgende Typen von Vorzeichenformeln werden vereinbart:

Typ α	(konjunktiver Typ)	$\mathbf{0}\neg B$, $\mathbf{1}\neg B$, $\mathbf{1}(V \wedge C)$, $\mathbf{0}(B \vee C)$, $\mathbf{0}(B \rightarrow C)$
Typ β	(disjunktiver Typ)	$\mathbf{0}(B \wedge C)$, $\mathbf{1}(B \vee C)$, $\mathbf{1}(B \rightarrow C)$
Typ γ	(allquantifizierter Typ)	$\mathbf{1}\forall x B(x)$, $\mathbf{0}\exists x B(x)$
Typ δ	(existenzquantifizierter Typ)	$\mathbf{1}\exists x B(x)$, $\mathbf{0}\forall x B(x)$

wobei $B, C \in \mathrm{For}_\Sigma$.

Die allgemeinen Regeln sind nun genauso definiert wie im aussagenlogischen Tableau Kalkül. Neu hinzu kommen Regeln für γ und δ Formeln sowie für die Behandlung der Objektgleichheit, die ganz schön eklig werden kann in der Prädikatenlogik. Somit ergeben sich die folgenden Regeln (ich liste hier auch nochmal die Regeln für α und β Formeln.

Definition (Regeln im Tableau Kalkül)**.** Regeln zu α Formeln:

$$\frac{\mathbf{0}\neg B}{\mathbf{1}B} \qquad \frac{\mathbf{1}\neg B}{\mathbf{0}B} \qquad \frac{\mathbf{1}(B \wedge C)}{\begin{matrix}\mathbf{1}B\\ \mathbf{1}C\end{matrix}} \qquad \frac{\mathbf{0}(B \vee C)}{\begin{matrix}\mathbf{0}B\\ \mathbf{0}C\end{matrix}} \qquad \frac{\mathbf{0}(B \rightarrow C)}{\begin{matrix}\mathbf{1}B\\ \mathbf{0}C\end{matrix}}$$

Regeln zu β Formeln:

$$\frac{\mathbf{0}(B \wedge C)}{\mathbf{0}B \quad \mathbf{0}C} \qquad \frac{\mathbf{1}(B \vee C)}{\mathbf{1}B \quad \mathbf{1}C} \qquad \frac{\mathbf{1}(B \rightarrow C)}{\mathbf{0}B \quad \mathbf{1}C}$$

Regeln zu γ Formeln:

$$\frac{\mathbf{1}\forall x B(x)}{\mathbf{1}B(X)} \qquad \frac{\mathbf{0}\exists x B(x)}{\mathbf{0}B(X)}$$

Der Quantor wird also fallengelassen und die Variable x wird in der Formel B durch eine Variable X ersetzt, die auf dem aktuellen Pfad π noch nicht vorkommt.

Regeln zu δ Formeln:

$$\frac{\mathbf{1}\exists x B(x)}{\mathbf{1}B(f(x_1,\ldots,x_n))} \qquad \frac{\mathbf{0}\forall x B(x)}{\mathbf{0}B(f(x_1,\ldots,x_n))}$$

Hier wird der Quantor ebenso fallengelassen, jedoch wird die quantifizierte Variable x durch ein neues n stelliges Funktionssymbol f ersetzt wobei $x_1,\ldots,x_n$ die freien Variablen in B sind.

Regeln zur Gleichheit:

$$\frac{\mathbf{1}t\doteq s \quad B(t')}{\sigma(B(s))} \qquad \frac{\mathbf{1}t\doteq s \quad B(s')}{\sigma(B(t))}$$

wobei für die linke Regel gilt $\sigma(t') = \sigma(t)$ und für die rechte Regel $\sigma(s') = \sigma(s)$. Dabeie sind t' und s' *Teilterme* der Formel B. Wir dürfen also den Teilterm s' bzw. t' durch t bzw s ersetzen, wenn es eine Substitution mit den geforderten Eigenschaften gibt. Die Substitution muss auf das gesamte Tableau angewendet werden!

Wir stellen außerdem noch zwei Axiome auf, mit denen wir unser Tableau starten können:

$$\frac{}{\mathbf{0}\,\mathrm{CL}_\forall(A)} \qquad \frac{}{\mathbf{1}\,\mathrm{CL}_\forall(B)}$$

Es ist möglich eine Regel für eine γ Formel mehrfach anzuwenden um so mehrere „Instanzen“ von einer Formel zu erzeugen. Manchmal lässt sich ein Tableau nicht schließen ohne dieses Prozedere, also ist dieses Vorgehen für die Vollständigkeit des Kalküls essentiell.

Um einen Pfad π in unserem Tableau zu schließen müssen wir jetzt etwas komplizierter vorgehen. Existieren zwei „ähnliche“ Formeln $\mathbf{1}B$ und $\mathbf{0}C$ auf einem Pfad π, will heißen lässt sich eine Substition σ finden, für die gilt $\sigma(B) = \sigma(C)$, so lässt sich der Pfad π, unter Anwendung der Substition σ **auf das gesamte Tableau**, schließen. Ebenso lässt sich ein Pfad π schließen, falls es zwei Terme s und t und eine Substitution σ mit $\sigma(s) = \sigma(t)$ und die Formel $\mathbf{0}s\doteq t$ auf π gibt. Analog zur Aussagenlogik ist ein Tableau genau dann geschlossen, wenn alle Pfade auf dem Tableau geschlossen sind.

Zur Vorgehensweise: Will man Beweisen, dass für eine Menge $M \subseteq \mathrm{For}_\Sigma$ und eine Formel $A \in \mathrm{For}_\Sigma$ gilt $M \models A$, so muss man das Tableau mit den zwei Axiomen eröffnen. Für die zu beweisende Aussage A verwendet man das erste Axiom, für die Prämissen $B \in M$ verwendet man das zweite Axiom. Im Falle dass nur eine allgemeingültige Aussage zu beweisen ist, also $M = \emptyset$ gilt, startet man das Tableau analog zum aussagenlogischen Fall mit $\mathbf{0}A$ durch Anwendung des ersten Axioms.

Nun geht man ganz normal vor, wie wir es schon aus dem einfachen aussagenlogischen Tableau Kalkül gewohnt sind. Natürlich müssen die neuen Regeln beachtet werden. Auf eine Formel $\mathbf{1}\forall x(p(x) \wedge q(x))$ lässt sich zum Beispiel nicht eine α Regel anwenden bevor nicht der Allquantor durch eine γ Regel eliminiert wurde. Strategisch günstig ist es, nicht jeden Pfad π sofort zu schließen. Da wir das meist nicht ohne Anwendung einer Substitution machen können, müsste das komplette Tableau substituiert und ggf. neu aufgeschrieben werde. Besser man wartet bis man für alle Pfade Kandidaten hat um sie zu schließen. Jetzt lässt sich eine Substitution finden die, angwendet auf das Tableau, alle Pfade auf einmal schließt. So spart man sich eine Menge Schreibarbeit.

Höchste Zeit für zwei Beispiele:

Beispiel. Wir möchten für folgende Formel A zeigen, dass sie allgemeingültig ist:

$$\begin{aligned} A := &\Big(\forall x \forall y (p(x,y) \to q(x,y)) \land \forall x \forall y (r(x,y) \to s(x,y))\Big) \to \\ &\forall x \forall y \Big(\exists z (p(x,z) \land r(z,y)) \to \exists u (q(x,u) \land s(u,y))\Big) \end{aligned}$$

Wir beginnen also in der Wurzel mit der Anwendung des ersten Axioms, und schreiben $\mathbf{0}A$ in die Wurzel. Das vollständige (etwas größere) Tableau ist in Abbildung 5 zu sehen.

Beispiel. Noch ein zweites Beispiel um die Verwendung der Gleichheitsformeln zu illustrieren. Wir wollen zeigen dass folgende Formel ϕ allgemeingültig ist

$$\phi := \forall x (g(x) \doteq f(g(x))) \land g(g(a)) \doteq c \to f(g(f(g(a)))) \doteq c$$

Dies führt zu dem Tableau in Abbildung 6.

3.3 Sequenzenkalkül

Der Sequenzenkalkül in der Prädikatenlogik ist extrem ähnlich zu der Variante in der Aussagenlogik, daher braucht es nicht mehr viel neue Erklärungen. Es kommen lediglich ein paar neue Regeln hinzu, die Quantoren sowie Objektgleichheit behandeln. Um die Definition von Sequenz nachzulesen, schaue man bitte im Abschnitt über den aussagenlogischen Sequenzkalkül nach.

Die Erweiterung der Auswertungsfunktion ist wie folgt:

Definition. Sei $\mathcal{D} := (D, I)$ eine Interpretation und β eine Variablenbelegung, dann gilt

$$\mathrm{val}_{\mathcal{D},\beta}(\Gamma \rightarrowtail \Delta) = \mathrm{val}_{\mathcal{D},\beta}(\bigwedge \Gamma \to \bigvee \Delta)$$

Auch keine große Überraschung hier.

Wir können also gleich zu der Definition der Regeln fortschreiten.

Definition (Regeln im Sequenzenkalkül)**.** Regeln der Aussagenlogik:

and–left:
$$\frac{\Gamma, F, G \rightarrowtail \Delta}{\Gamma, F \wedge G \rightarrowtail \Delta}$$

and–right:
$$\frac{\Gamma \rightarrowtail F, \Delta \quad \Gamma \rightarrowtail G, \Delta}{\Gamma \rightarrowtail F \wedge G, \Delta}$$

or–left:
$$\frac{\Gamma, F \rightarrowtail \Delta \quad \Gamma, G \rightarrowtail \Delta}{\Gamma, F \vee G \rightarrowtail \Delta}$$

or–right:
$$\frac{\Gamma \rightarrowtail F, G, \Delta}{\Gamma \rightarrowtail F \vee G, \Delta}$$

impl–left:
$$\frac{\Gamma \rightarrowtail F, \Delta \quad \Gamma, G \rightarrowtail \Delta}{\Gamma, F \rightarrow G \rightarrowtail \Delta}$$

impl–right:
$$\frac{\Gamma, F \rightarrowtail G, \Delta}{\Gamma \rightarrowtail F \rightarrow G, \Delta}$$

not–left:
$$\frac{\Gamma \rightarrowtail F, \Delta}{\Gamma, \neg F \rightarrowtail \Delta}$$

not–right:
$$\frac{\Gamma, F \rightarrowtail \Delta}{\Gamma \rightarrowtail \neg F, \Delta}$$

Das Axiom wird ebenfalls übernommen:

$$\frac{}{\Gamma, F \rightarrowtail F, \Delta}$$

Regeln für quantifizierte Formeln:

all–left:
$$\frac{\Gamma, \forall x F(x), F(X) \rightarrowtail \Delta}{\Gamma, \forall x F(x) \rightarrowtail \Delta}$$

all–right:
$$\frac{\Gamma \rightarrowtail F(f(x_1, \ldots, x_n)), \Delta}{\Gamma \rightarrowtail \forall x F(x), \Delta}$$

ex–left:
$$\frac{\Gamma, F(f(x_1, \ldots, x_n)) \rightarrowtail \Delta}{\Gamma, \exists x F(x) \rightarrowtail \Delta}$$

ex–right:
$$\frac{\Gamma \rightarrowtail \exists x F(x), F(X), \Delta}{\Gamma \rightarrowtail \exists x F(x), \Delta}$$

Dabei ist X eine neue Variable und f ein neues n–stelliges Funktionssymbol, wobei $x_1, \ldots, x_n$ die freien Variablen in $F(x)$ sind.

Regeln für Objektgleichheit:

identity:
$$\frac{}{\Gamma \rightarrowtail s \doteq s, \Delta}$$

symmetry–left:
$$\frac{\Gamma, s \doteq t \rightarrowtail \Delta}{\Gamma, t \doteq s \rightarrowtail \Delta}$$

symmetry–right:
$$\frac{\Gamma \rightarrowtail s \doteq t, \Delta}{\Gamma \rightarrowtail t \doteq s, \Delta}$$

eq–subst–left:
$$\frac{\Gamma, F(t), s \doteq t \rightarrowtail \Delta}{\Gamma, F(s), s \doteq t \rightarrowtail \Delta}$$

eq–subst–right:
$$\frac{\Gamma, s \doteq t \rightarrowtail F(t), \Delta}{\Gamma, s \doteq t \rightarrowtail F(s), \Delta}$$

Das Vorgehen bei einem Beweis ist analog zum Sequenzenkalkül der Aussagenlogik. Wir werden also wieder „rückwärts“ vorgehen, und auf die Axiome schließen. Dabei entsteht ein Baum. Aus den Blättern des Baumes lässt sich möglicherweise nur das Axiom herleiten, indem man eine Substitution σ auf den kompletten Baum anwendet. Diese Substitution darf erst am Schluss durchgeführt werden. Einzige Ausnahme sind die Regeln „eq–subst–right“ und „eq–subst–left“ die auch noch nach dieser Substitution angewendet werden dürfen.

Beispiel. Wir möchten wieder für folgende Formel beweisen, dass sie allgemeingültig ist:

$$\phi := \forall x(g(x) \doteq f(g(x))) \land g(g(a)) \doteq c \to f(g(f(g(a)))) \doteq c$$

Dies führt zu dem Baum aus Abbildung 7.

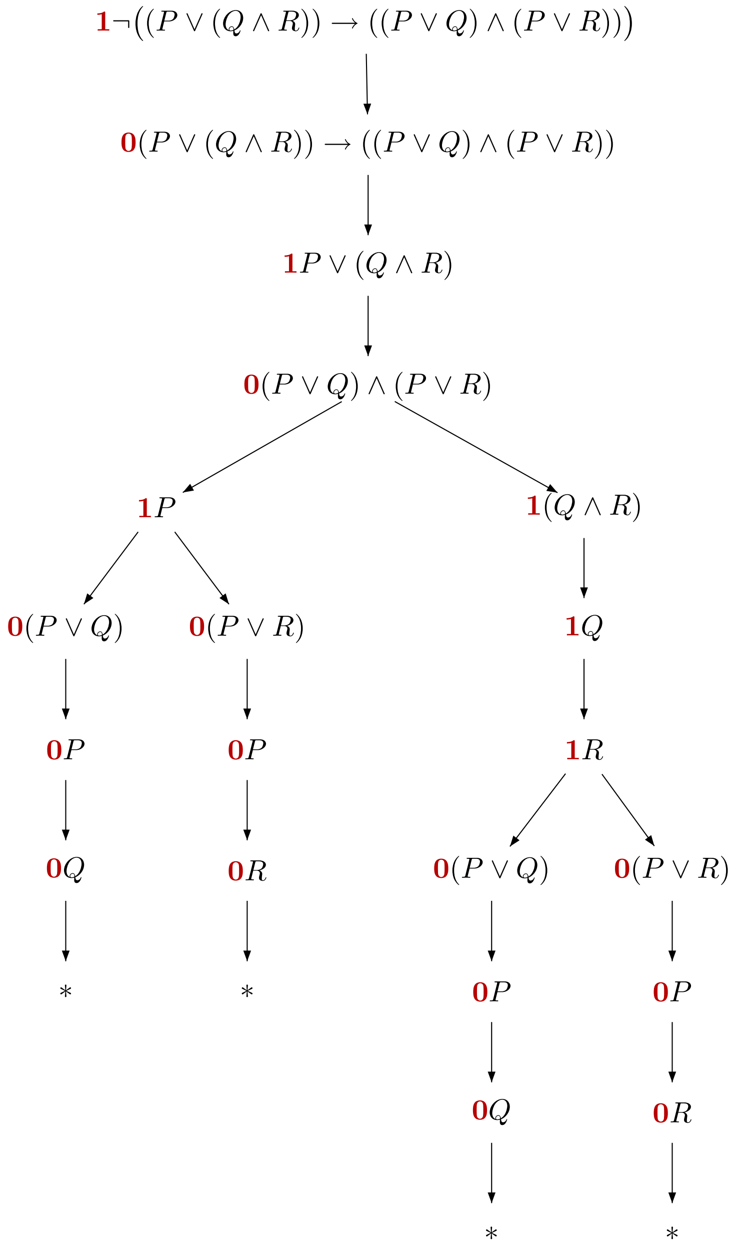

Abbildung 3: Beispiel für einen Tableau Beweis. Alle Pfade konnten geschlossen werden, das heißt die Formel in der Wurzel des Tableaus ist nicht erfüllbar.

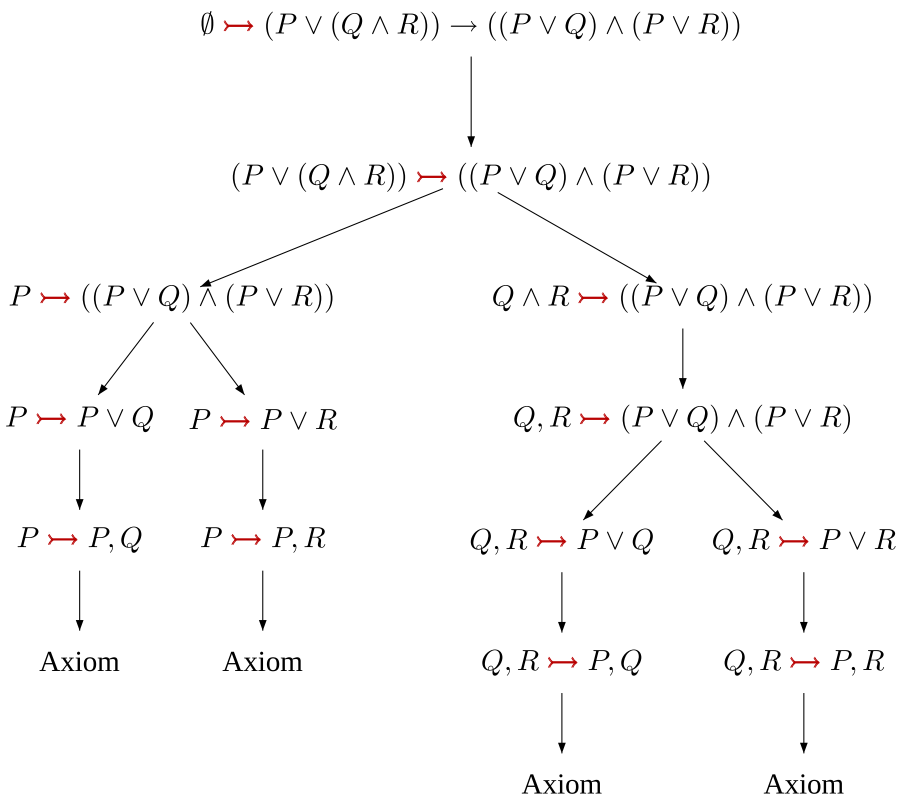

Abbildung 4: Beispiel für einen Beweis im Sequenzenkalkül. Die eigentliche Folge von Ableitungen ist von unten nach oben zu lesen.

[0,-] $\mathbf{0}A$

[1,0] $\mathbf{1}\forall x \forall y(p(x,y) \rightarrow q(x,y)) \wedge \forall x \forall y(r(x,y) \rightarrow s(x,y))$

[2,0] $\mathbf{0}\forall x \forall y(\exists z(p(x,z) \wedge r(z,y)) \rightarrow \exists u(q(x,u) \wedge s(u,y)))$

[3,1] $\mathbf{1}\forall x \forall y(p(x,y) \rightarrow q(x,y))$

[4,1] $\mathbf{1}\forall x \forall y(r(x,y) \rightarrow s(x,y))$

[5,2] $\mathbf{0}\exists z(p(a,z) \wedge r(z,b)) \rightarrow \exists u(q(a,u) \wedge s(u,b))$

[6,5] $\mathbf{1}\exists z(p(a,z) \wedge r(z,b))$

[7,5] $\mathbf{0}\exists u(q(a,u) \wedge s(u,b))$

[8,6] $\mathbf{1}p(a,c) \wedge r(c,b)$

[9,8] $\mathbf{1}p(a,c)$

[10,8] $\mathbf{1}r(c,b)$

[11,7] $\mathbf{0}q(a,U) \wedge s(U,b)$

[12,11] $\mathbf{0}q(a,U)$	[13,7] $\mathbf{0}s(U,b)$
[14,3] $\mathbf{1}p(X,Y) \rightarrow q(X,Y)$	[17,4] $\mathbf{1}r(X',Y') \rightarrow s(X',Y')$

[15,14] $\mathbf{0}p(X,Y)$	[16,14] $\mathbf{1}q(X,Y)$	[18,17] $\mathbf{0}r(X',Y')$	[19,17] $\mathbf{1}s(X',Y')$
$*$ [9,15]	$*$ [12,16]	$*$ [10,18]	$*$ [13,19]

Abbildung 5: Beispiel für einen Tableau Beweis in der Prädikatenlogik. Das Tableau schließt mit der Substitution $\sigma := \{X/a, Y/c, U/c, X'/c, Y'/c\}$. Für die Werte in den eckigen Klammern gilt: Die erste Zahl ist ein Zähler und die zweite Zahl gibt an aus welcher Formel sich die Formel abgeleitet hat. In den Blättern sind in der Klammer die beiden Formeln die den Pfad unter Anwendung von σ schließen.

[1,-] $\mathbf{0}\forall x(g(x)\doteq f(g(x))) \land g(g(a))\doteq c \rightarrow f(g(f(g(a))))\doteq c$

[2,1] $\mathbf{1}\forall x(g(x)\doteq f(g(x))) \land g(g(a))\doteq c$

[3,1] $\mathbf{0}f(g(f(g(a))))\doteq c$

[4,2] $\mathbf{1}\forall x(g(x)\doteq f(g(x)))$

[5,2] $\mathbf{1}g(g(a))\doteq c$

[6,4] $\mathbf{1}g(X)\doteq f(g(X))$

[7,3/6,$\sigma := \{X/f(g(a))\}$] $\mathbf{0}g(f(g(a)))\doteq c$

[7] $\mathbf{0}g(f(g(a)))\doteq c$

[8,4] $\mathbf{1}g(Y)\doteq f(g(Y))$

[9,7/8,$\sigma := \{Y/a\}$] $\mathbf{0}g(g(a))\doteq c$

$*$ [5,9]

Abbildung 6: Beispiel für einen Tableau Beweis mit Gleichungsregeln. Zur Herleitung der Formel 7 wird die rechte Gleichungsregel angewendet. Dazu werden die blau markierten Terme durch die Substitution σ unifiziert. Der Term $f(g(f(g(a))))$ in der Formel 3 ist dabei unser s' in der Definition. Das heißt wir dürfen in der Gleichung 3 dieses blaue s' durch die linke Seite der Gleichung 6 (t in der Definition der Regel) ersetzen. Schließlich müssen wir noch σ auf unsere neue Formel anwenden, und erhalten somit Gleichung 7. Zur Herleitung von Gleichung 9 wurde wieder die rechte Gleichungsregel benutzt. Dabei ist der blaue Teilterm in Gleichung 7 unser s' was durch σ mit der rechten Seite von 8 unifiziert wird. Also dürfen wir in 7 das s' durch die linke Seite von 8 ersetzen. Anwenden der Substitution darauf liefert dann die Gleichung 9. Da die Substitutionen jeweils auf das gesamte Tableau hätten angewendet werden müssen, sind die Gleichungen 6 und 8 so nicht mehr zulässig am Ende, daher sind sie grau gezeichnet.

$$\emptyset \rightarrowtail \forall x(g(x)\doteq f(g(x))) \land g(g(a))\doteq c \rightarrow f(g(f(g(a))))\doteq c$$

impl–right

$$\forall x(g(x)\doteq f(g(x))) \land g(g(a))\doteq c \rightarrowtail f(g(f(g(a))))\doteq c$$

and–left

$$\forall x(g(x)\doteq f(g(x))), g(g(a))\doteq c \rightarrowtail f(g(f(g(a))))\doteq c$$

all–left

$$\forall x(g(x)\doteq f(g(x))), g(g(a))\doteq c, g(X)\doteq f(g(X)) \rightarrowtail f(g(f(g(a))))\doteq c$$

all–left

$$\begin{array}{ll} \forall x(g(x)\doteq f(g(x))), g(g(a))\doteq c, g(X) = f(g(X)), & \rightarrowtail f(g(f(g(a))))\doteq c \\ g(Y)\doteq f(g(Y)) & \end{array}$$

$\sigma := \{X/f(g(a)), Y/a\}$

$$\begin{array}{ll} \forall x(g(x)\doteq f(g(x))), g(g(a))\doteq c, g(f(g(a)))\doteq f(g(f(g(a)))), & \rightarrowtail f(g(f(g(a))))\doteq c \\ g(a) = f(g(a)) & \end{array}$$

eq–subst–right

$$\begin{array}{ll} \forall x(g(x)\doteq f(g(x))), g(g(a))\doteq c, g(f(g(a)))\doteq f(g(f(g(a)))), & \rightarrowtail g(f(g(a)))\doteq c \\ g(a) = f(g(a)) & \end{array}$$

eq–subst–right

$$\begin{array}{ll} \forall x(g(x)\doteq f(g(x))), g(g(a))\doteq c, g(f(g(a)))\doteq f(g(f(g(a)))), & \rightarrowtail g(g(a))\doteq c \\ g(a) = f(g(a)) & \end{array}$$

Axiom

Abbildung 7: Beispiel eines Sequenzen Beweises in der Prädikatenlogik. Die all–left sowie die ex–right Regel dürfen mehrfach angewendet werden. Außerdem dürfen nach der Substitution noch Gleichungsregeln benutzt werden.

Literatur

[Sch05] Prof. Dr. P.H. Schmitt. Formale systeme. 2005.