

4. Prädikatenlogik

Die Sprache der *Prädikatenlogik* erster Stufe (engl. *First-order Predicate Logic, PL1*) wird gegenüber der Aussagenlogik so erweitert, dass gewisse Formen von Aussagen, die in der Aussagenlogik nicht möglich sind, ausgedrückt werden können, z.B. dass

- Objekte in einer gewissen Beziehung zueinander stehen,
- eine Eigenschaft für alle Objekte gilt,
- es ein Objekt mit einer bestimmten Eigenschaft gibt.

Syntax: Die im Vergleich zur Aussagenlogik neuen Sprachelemente sind

- Variable (für Individuen)
- Funktionssymbole
- Prädikatensymbole
- Quantoren: \forall heißt *Allquantor*, \exists *Existenzquantor*.

PL1: Syntax

Gegeben:

- eine Menge von Konstanten- und Funktionssymbolen f_j
- eine Menge von Prädikatensymbolen P_k
- eine Menge von Variablen x_i

Jedes Funktions- und Prädikatensymbol hat eine *Stelligkeit* (engl. *arity*), die die Anzahl der Argumente angibt.

Nullstellige Funktionssymbole und Konstanten werden gewöhnlich gleichgesetzt.

Terme: wie in Gleichungslogik

1. Jede Variable x_i ist ein Term.
2. Sind t_1, \dots, t_n Terme und f ein n -stelliges Funktionssymbol, dann ist auch $f(t_1, \dots, t_n)$ ein Term.

PL1: Syntax (2)

Atomare Formeln:

- Die Konstanten W und F sind atomare Formeln.
- Sind t_1, \dots, t_n Terme und P ein n -stelliges Prädikatensymbol, dann ist $P(t_1, \dots, t_n)$ eine atomare Formel.

Atomare Formeln spielen in der Prädikatenlogik eine ähnliche Rolle wie die Symbole der Aussagenlogik: ein *Literal* ist eine atomare Formel oder eine negierte atomare Formel.

Formeln (well-formed formulae, wff's):

1. Eine atomare Formel ist eine Formel.
2. Ist F eine Formel, dann auch $\neg F$.
3. Sind F und G Formeln, dann sind auch $F \wedge G$, $F \vee G$ usw. Formeln.
4. Ist x eine Variable und ist F eine Formel, dann sind $\exists x. F$ und $\forall x. F$ Formeln. Formeln dieser Art heißen *quantifizierte Formeln*.
5. Alle Formeln werden nach 1. – 4. gebildet.

Terminologie

Eine *Teilformel* ist eine Formel, die als Teil einer anderen Formel auftritt.

Ein Term, der keine Variablen enthält, heißt *Grundterm*; eine Formel, die keine Variablen enthält, heißt *Grundformel*.

Vorkommen von Variablen in Formeln können *frei* oder *gebunden* sein: Ein Vorkommen der Variable x in einer quantifizierten Formel der Form $\forall x. F$ oder $\exists x. F$ heißt *gebunden* und F der *Bindungsbereich* von x . Vorkommen von Variablen, die nicht in dieser Weise gebunden sind, sind freie Vorkommen.

Dieselbe Variable kann in einer Formel in verschiedenen Teilformeln sowohl frei wie auch gebunden vorkommen. Beispiel:

$$(\forall x. P(x)) \wedge (\forall y. Q(x, y))$$

Eine Formel ohne Vorkommen einer freien Variablen heißt *geschlossen* oder eine *Aussage*.

Eine Formel ohne Quantoren heißt *offen* (dies ist nur dann interessant, wenn die Formel nicht eine Grundformel ist, d.h. Vorkommen von freien Variablen enthält).

PL1: Semantik

Die Semantik von Termen und Formeln wird durch eine *Interpretation* der vorkommenden Symbole gegeben.

Die Grundlage einer Interpretation ist eine nichtleere Menge \mathcal{U} , die *Grundmenge*, *Individuenbereich* oder *Universum* genannt wird. Über dieser Menge werden Konstanten, Funktionen und Prädikate interpretiert: die Interpretationsabbildung I ordnet zu:

- jedem n -stelligen Funktionssymbol f eine n -stellige Funktion $I[f] : \mathcal{U}^n \rightarrow \mathcal{U}$;
- jedem n -stelligen Prädikatensymbol P eine n -stellige Relation $I[P] \subseteq \mathcal{U}^n$.

Die Abbildung I wird auf Grundterme fortgesetzt:

- Sind f ein n -stelliges Funktionssymbol und t_i Terme, so ist

$$I[f(t_1, \dots, t_n)] = I[f](I[t_1], \dots, I[t_n])$$

.

PL1: Semantik (2)

Eine *Variablenbelegung* (Variablenzuweisung) ordnet jeder Variablen einen Wert aus \mathcal{U} zu. Eine Interpretation zusammen mit einer Variablenbelegung erlaubt es, auch Nicht-Grundtermen Werte in \mathcal{U} zuzuordnen, d.h. sie zu interpretieren.

Interpretation von Formeln:

Formeln werden unter der Interpretation auf Wahrheitswerte abgebildet.

- Für eine atomare Formel $P(t_1, \dots, t_n)$ hat $I[P(t_1, \dots, t_n)]$ den Wert “wahr” genau dann, wenn (gdw.) $(I[t_1], \dots, I[t_n]) \in I[P]$ gilt.
- $I[\neg F]$ ist wahr gdw. $I[F]$ falsch ist.
- $I[F \vee G]$ ist wahr gdw. $I[F]$ wahr ist oder $I[G]$ wahr ist.
- $I[F \wedge G]$ ist wahr gdw. $I[F]$ wahr ist und $I[G]$ wahr ist.
- $I[\exists x. F]$ ist wahr gdw. es ein $u \in \mathcal{U}$ gibt, so daß $I[F]$ wahr ist, wenn x in F der Wert u zugewiesen wird.
- $I[\forall x. F]$ ist wahr gdw. für alle $u \in \mathcal{U}$ gilt, dass $I[F]$ wahr ist, wenn x in F der Wert u zugewiesen wird.

PL1: Semantik (3)

Eine offene Formel wird wie ihr universeller Abschluss interpretiert.

Für eine Formel oder eine Menge von Formeln kann es viele Interpretationen geben. Man ist i.a. nur an denjenigen Interpretationen interessiert, unter denen die vorgegebenen Formeln wahr sind.

Eine Interpretation I heißt *Modell* einer Formelmenge F , wenn jede Formel in F unter I wahr ist.

Eine Formelmenge, für die es Modelle gibt, heißt *erfüllbar*.

Eine Formelmenge, für die es keine Modelle gibt, heißt *unerfüllbar*.

PL1: Semantik (4)

Eine Formel, für die jede Interpretation ein Modell ist, heißt *allgemeingültig* oder *Tautologie*.

Es genügt, sich bei allgemeingültigen Formeln auf geschlossene Formeln zu beschränken (warum?).

Wie in der Aussagenlogik gilt in der Prädikatenlogik, dass eine Formel F eine Tautologie genau dann ist, wenn $\neg F$ *unerfüllbar* ist.

Schreibweise: Für eine Interpretation I und Formel(menge) F bedeutet $I \models F$, dass F in (oder unter) I gilt, oder dass I ein Modell für F ist.

Entsprechend bedeutet $\models F$, dass F allgemeingültig ist, also unter allen Interpretationen gilt.

Bemerkung: Diese Bedeutung des Symbols \models ist völlig analog zu der früher für Aussagenlogik angegebenen.

PI1: Entscheidbarkeitsfragen

Satz: Prädikatenlogik ist unentscheidbar:

- Es gibt kein Verfahren, mit dem für eine beliebige prädikatenlogische Formel entschieden werden kann, ob sie allgemeingültig ist.
- Es gibt kein Verfahren, mit dem für eine beliebige prädikatenlogische Formel entschieden werden kann, ob sie erfüllbar ist.

Satz: Prädikatenlogik ist semi-entscheidbar:

- Es gibt ein Verfahren, mit dem alle prädikatenlogischen Tautologien erzeugt (“aufgezählt”) werden können.

Spezialfälle der Prädikatenlogik

- Aussagenlogik kann als Spezialfall einer Prädikatenlogik angesehen werden, in der alle Prädikatensymbole nullstellig sind, d.h. Wahrheitswerte bezeichnen. Die Variablen, Quantoren usw. spielen dann keine Rolle mehr.
- Quantorenfreie Prädikatenlogik: Formeln können Variablen enthalten, die als implizit universell quantifiziert angesehen werden.
- Horn-Logik: Formeln haben die Gestalt

$$P_1 \wedge \dots \wedge P_n \Rightarrow P_{n+1}$$

(P_i atomare Formeln; ergibt eine eingeschränkte Form der quantorenfreie Prädikatenlogik)

- Gleichungslogik (wie bereits behandelt):
= als einziges Prädikatensymbol

PL1: Gesetze

Die folgenden logischen Äquivalenzen (“Gleichungen”) sind allgemeingültige prädikatenlogische Formeln.

- Die für Aussagenlogik angegebenen Gesetze
- Vertauschbarkeit von Quantoren:

$$(\forall x. \forall y. Q(x, y)) \Leftrightarrow (\forall y. \forall x. Q(x, y))$$

$$(\exists x. \exists y. Q(x, y)) \Leftrightarrow (\exists y. \exists x. Q(x, y))$$

- Umbenennung von gebundenen Variablen:

$$(\forall x. P(x)) \Leftrightarrow (\forall y. P(y))$$

$$(\exists x. P(x)) \Leftrightarrow (\exists y. P(y))$$

- Negierte quantifizierte Formeln:

$$\neg(\forall x. P(x)) \Leftrightarrow \exists x. \neg P(x)$$

$$\neg(\exists x. P(x)) \Leftrightarrow \forall x. \neg P(x)$$

PL1: Gesetze (2)

- Veränderung des Bindungsbereichs (engl. *scope*) von Quantoren:

$$(\forall x. P(x)) \wedge (\forall x. Q(x)) \Leftrightarrow (\forall x. P(x) \wedge Q(x))$$

$$(\exists x. P(x)) \vee (\exists x. Q(x)) \Leftrightarrow (\exists x. P(x) \vee Q(x))$$

Bem. Die entsprechenden Äquivalenzen mit \vee (für \forall) bzw. \wedge (für \exists) gelten im allgemeinen *nicht*.

Falls x nicht frei in R vorkommt:

$$(\forall x. P(x)) \wedge R \Leftrightarrow (\forall x. P(x) \wedge R)$$

$$(\forall x. P(x)) \vee R \Leftrightarrow (\forall x. P(x) \vee R)$$

$$(\exists x. P(x)) \wedge R \Leftrightarrow (\exists x. P(x) \wedge R)$$

$$(\exists x. P(x)) \vee R \Leftrightarrow (\exists x. P(x) \vee R)$$

Getypte Prädikatenlogik

Warum getypte Logik ?

- Es ist natürlich, von verschiedenen Wertebereichen auszugehen (z.B. natürliche Zahlen, Wahrheitswerte, Listen, . . .)
- “natürlich” für Informatiker – wie in getypten Programmiersprachen
- Syntaktische Typ-Überprüfung ist möglich, anstelle von Deduktion

Grundidee:

- Verschiedene Individuen können unterschiedlichen Typen von Werten angehören.
- Funktions- und Prädikatsymbolen wird nicht nur eine Stelligkeit, sondern eine *Signatur* zugeordnet, die die Typen der Argumente und des Resultats angibt.

Bemerkung: Die Terminologie ist nicht einheitlich. Es ist üblicher, von “mehrsortiger Logik” (*many-sorted logic*) als von “getypter Logik” zu sprechen; hier wird i.a. “Typ” benutzt, wo in der Literatur häufig von “Sorte” gesprochen wird. Die Begriffe “Typ” und “Sorte” werden hier also (im wesentlichen) als ynonyme benutzt.

Getypte Prädikatenlogik (2)

Syntax: (vgl. getypte Gleichungslogik)

\mathcal{S} bezeichne eine (endliche, oder abzählbare) Menge von Sortensymbolen; \mathcal{S} soll ein Symbol *Bool* (für den Typ der *Wahrheitswerte*) enthalten.

Σ *Signatur* über \mathcal{S} : \mathcal{S} -typisierte Funktionen, Prädikate und Konstanten (genauer: Funktionssymbole usw.)

$f^{(n)} : S_1 \times \dots \times S_n \rightarrow S_{n+1}$ für eine n-stellige Funktion

$p^{(n)} : S_1 \times \dots \times S_n \rightarrow Bool$ für ein n-stelliges Prädikat

$c : S_i$ für eine Konstante

Die Notation $t : S_i$ soll bedeuten “der Term t hat den Typ S_i ”

Variable werden i.a. als getypt angenommen:

$v_i \in V_S$ Variable vom Typ S , für jedes $S \in \mathcal{S}$

formal: S -indizierte Familie $\{V_S\}_{S \in \mathcal{S}}$ von Variablenmengen

Getypte Prädikatenlogik (3)

Term- und Formel-Bildung:

– Das Bilden von Termen und Formeln muss typ-korrekt sein:

$$\frac{f : S_1 \times \dots \times S_n \rightarrow S_{n+1} \quad t_i : S_i \text{ (für alle } i \in \{1, \dots, n\})}{f(t_1, \dots, t_n) : S_{n+1}}$$

– entsprechend für Prädikate

– Getypte Quantoren: für $P : S \rightarrow Bool$

$$\forall v : S. P(v) \quad \exists v : S. P(v)$$

Alle anderen Begriffe bleiben unverändert.

Getypte Prädikatenlogik Semantik

Modelle sind \mathcal{S} -indizierte Strukturen M :

- (1) Für jedes Typsymbol $S \in \mathcal{S}$ eine *nichtleere* Menge M_S (“Trägermenge”, *carrier*, *domain* von S); über die Beziehungen zwischen den verschiedenen M_S , insbesondere Disjunktheit, werden i.a. keine weitere Annahmen gemacht.

$$M_{Bool} := \{W, F\}$$

- (2) Für jedes Funktionssymbol $f : S_1 \times \dots \times S_n \rightarrow S_{n+1}$ eine Funktion

$$f_M : M_{S_1} \times \dots \times M_{S_n} \rightarrow M_{S_{n+1}}$$

- (3) Für jedes Prädikatsymbol $P : S_1 \times \dots \times S_n \rightarrow Bool$ ein Prädikat (Teilmenge)

$$P_M \subseteq M_{S_1} \times \dots \times M_{S_n}$$

bzw.

$$P_M : M_{S_1} \times \dots \times M_{S_n} \rightarrow M_{Bool}$$

Getypte Prädikatenlogik Semantik

(4) Für jedes Konstanten-Symbol $c : S_i$ eine Konstante $c_M : M_{S_i}$

Wertebelegung für Variable:

$$\mathcal{V}_S : V_S \rightarrow M_S \quad \text{für jedes } S \in \mathcal{S}$$

Bemerkungen:

Die Familie von Trägermengen tritt an die Stelle des *einen* Universums.

Nichtleere Mengen als Trägermengen sind notwendig, um Probleme mit Quantifikation zu vermeiden.

Ansonsten bleibt alles wie gehabt.

Reduktion auf (ein-sortige) Prädikatenlogik

Für jedes $S \in \mathcal{S}$ erforderlich: charakteristisches Prädikat D_S für den Wertebereich von S :

$$x : S \quad \rightsquigarrow \quad D_S(x) \text{ wahr}$$

Eine quantifizierte Formel P wird ersetzt durch ihre Relativisierung $\mathcal{R}[P]$:

$$\begin{aligned} \mathcal{R}[\forall x : S. P] &\rightsquigarrow \forall x. D_S(x) \Rightarrow \mathcal{R}[P] \\ \mathcal{R}[\exists x : S. P] &\rightsquigarrow \exists x. D_S(x) \wedge \mathcal{R}[P] \end{aligned}$$

Zusätzliche *Axiome* Ax_S :

- (1) $\exists x. D_S(x)$ für jeden nicht-leeren Typ $S \in \mathcal{S}$
- (2) $\forall x_1, \dots, x_n. D_{S_1}(x_1) \wedge \dots \wedge D_{S_n}(x_n) \Rightarrow D_{S_{n+1}}(f^{(n)}(x_1, \dots, x_n))$
für jedes $f^{(n)}$; entsprechend für $P^{(n)}$

Dann gilt für jede Formel A : $\vdash A$ gdw. $Ax_S \vdash \mathcal{R}[A]$

Modelle: Vereinigung der M_S mit entsprechender Interpretation der D_{S_i}

Sequenzen-Kalkül

Der Sequenzenkalkül ist einer der von dem Logiker G. Gentzen vorgeschlagenen Kalküle; daher auch die alternative Bezeichnung “Gentzen-Kalkül”.

- Das Grundelement des Kalküls ist die *Sequenz* (engl. *sequent*); der Kalkül beschreibt im wesentlichen, wie mit Sequenzen umgegangen wird.
- Eine Sequenz ist ein Ausdruck der Form $\Gamma \vdash \Delta$.
 Γ und Δ sind beliebige (endliche) Mengen (oder Multimengen) von Formeln (mit gewissen Einschränkungen; siehe unten).
 Γ heißt der *Antezedent*, Δ der *Sukzedent* der Sequenz.
- Antezedent und Sukzedent können jeweils auch leer sein.

Sequenzen – intuitiv

Intuitive Interpretation von Sequenzen:

Einer Sequenz $A_1, \dots, A_m \vdash B_1, \dots, B_n$ entspricht die Aussage, daß die Implikation

$$A_1 \wedge \dots \wedge A_m \Rightarrow B_1 \vee \dots \vee B_n$$

wahr ist

(bzw. daß $\neg A_1 \vee \dots \vee \neg A_m \vee B_1 \vee \dots \vee B_n$ wahr ist).

Bei atomaren A_i, B_j erkennt man unschwer die Nähe zur Klausel-Form.

Eine Sequenz $A_1, \dots, A_m \vdash B_1, \dots, B_n$ heißt *gültig*, wenn die Formel $A_1 \wedge \dots \wedge A_m \Rightarrow B_1 \vee \dots \vee B_n$ unter jeder Interpretation wahr ist.

Praktisches Ziel des Sequenzen-Kalküls ist die Ableitung eines Beweis für Formelmengende Δ unter der Annahme der Formeln in Γ .

Also: $\Gamma \vdash \Delta$.

Sequenzen-Kalkül: Aussagenlogik

Das einzige *Axiom* ist:

$$\Gamma, A \vdash A, \Delta$$

Jede Sequenz, in der im Antezedenten und Sukzedenten dieselbe Formel auftritt, ist eine Axiom.

Strukturelle Regeln:

$$\frac{A, A, \Gamma \vdash \Delta}{A, \Gamma \vdash \Delta} \text{contr:L} \qquad \frac{\Gamma \vdash \Delta, A, A}{\Gamma \vdash \Delta, A} \text{contr:R}$$

$$\frac{\Gamma \vdash \Delta, A \quad A, \Gamma \vdash \Delta}{\Gamma \vdash \Delta} \text{cut}$$

Sequenzen-Kalkül: Aussagenlogik (2)

Für die logischen Operatoren gibt es jeweils Paare von Regeln.

↷ Symmetrie zwischen linker und rechter Seite einer Sequenz

Logische Regeln:

$$\frac{\Gamma \vdash \Delta, A \quad B, \Gamma \vdash \Delta}{A \Rightarrow B, \Gamma \vdash \Delta} \Rightarrow L$$

$$\frac{A, \Gamma \vdash \Delta, B}{\Gamma \vdash \Delta, A \Rightarrow B} \Rightarrow R$$

$$\frac{A, B, \Gamma \vdash \Delta}{A \wedge B, \Gamma \vdash \Delta} \wedge L$$

$$\frac{\Gamma \vdash \Delta, A \quad \Gamma \vdash \Delta, B}{\Gamma \vdash \Delta, A \wedge B} \wedge R$$

$$\frac{A, \Gamma \vdash \Delta \quad B, \Gamma \vdash \Delta}{A \vee B, \Gamma \vdash \Delta} \vee L$$

$$\frac{\Gamma \vdash \Delta, A, B}{\Gamma \vdash \Delta, A \vee B} \vee R$$

$$\frac{\Gamma \vdash \Delta, A}{\neg A, \Gamma \vdash \Delta} \neg L$$

$$\frac{A, \Gamma \vdash \Delta}{\Gamma \vdash \Delta, \neg A} \neg R$$

Sequenzen-Kalkül: Aussagenlogik (3)

Bemerkungen:

- Antezedent und Sukzedent werden vollständig symmetrisch behandelt
- Daher jetzt immer Anwendbarkeitskriterium für die Regeln. Ausnahme: Die Schnittregel *cut*.
- Es ist nicht nötig, beim Rückwärtsbeweis eine Formel zu erraten.

Die Regeln haben folgende *Unterformel-Eigenschaft*: Tritt eine Formel in der Vorbedingung einer Regel auf, so auch in der Nachbedingung (zumindest als Unterformel).

Ausnahme: Schnittregel.

Sequenzen-Kalkül – Beispiele

Beispiel: Beweise $A \wedge B \vdash A \vee C$

1.

$$\frac{\frac{A, B \vdash A}{A \wedge B \vdash A} \wedge L}{A \wedge B \vdash A \vee C} \vee R$$

2. Es ist nicht notwendig, sich auf die Wahl von A oder C festzulegen:

$$\frac{\frac{A, B \vdash A, C}{A \wedge B \vdash A, C} \wedge L}{A \wedge B \vdash A \vee C} \vee R$$

3. Ein Beweis kann auch “Umwege” enthalten:

$$\frac{\frac{A, C \vdash C}{A \wedge C \vdash C} \wedge L \quad \frac{C \vdash C, B}{C \vdash C \vee B} \vee R}{A \wedge C \vdash C \vee B} cut$$

Wang-Algorithmus

Die Regeln des Sequenzenkalküls ergeben ein Beweisverfahren für Aussagenlogik, das als “Wang-Algorithmus” (H. Wang 1960) bekannt ist. Diese Beweismethode ist im allgemeinen schneller als das Aufstellen einer Wahrheitstafel.

Voraussetzung: Zu beweisende Aussage wird formuliert in Form einer Sequenz: Prämissen links vom Ableitungssymbols \vdash , die abzuleitende Schlußfolgerung rechts von \vdash . Beispiel:

$$P \Rightarrow Q, \quad Q \Rightarrow R, \quad \neg R \vdash \neg P$$

Der Wang-Algorithmus besteht im wesentlichen darin, die Ziel-Sequenz schrittweise durch “Rückwärts-Anwendung” der Sequenzenregeln so umzuformen, dass einfachere Sequenzen entstehen.

\rightsquigarrow ziel-orientierter Aufbau eines Beweises

Wang-Algorithmus (2)

1. Eine negierte Aussage $\neg X$ wird gestrichen und X auf der anderen Seite von \vdash hinzugefügt.
2. Für eine Konjunktion $X \wedge Y$ auf der linken Seite wird \wedge durch ein Komma ersetzt. Für eine Disjunktion $X \vee Y$ auf der rechten Seite wird \vee durch ein Komma ersetzt.
3. Wenn eine Aussage auf der linken Seite die Form $X \vee Y$ hat, wird die Sequenz durch zwei neue Sequenzen ersetzt, in denen die Disjunktion durch X bzw. Y ersetzt ist.
4. Wenn eine Aussage auf der rechten Seite die Form $X \wedge Y$ hat, wird die Sequenz durch zwei neue Sequenzen ersetzt, in denen die Konjunktion durch X bzw. Y ersetzt ist.

Wang-Algorithmus (3)

5. Eine Implikation $X \Rightarrow Y$ wird durch $\neg X \vee Y$ ersetzt, d.h. Implikationen werden eliminiert.
6. (Terminierungsregel 1): Eine Sequenz wird als bewiesen angesehen, wenn eine Aussage X sowohl auf der linken wie auf der rechten Seite von \vdash auftritt. (Man beachte, dass X auch zusammengesetzt sein kann, also kein Symbol sein muss.) Eine solche Sequenz wird *Axiom* genannt.
Die ursprüngliche Sequenz ist bewiesen, wenn alle aus ihr abgeleiteten Sequenzen bewiesen sind.
7. (Terminierungsregel 2): Eine Sequenz ist nicht als gültig nachgewiesen, wenn alle Formeln in ihr individuelle Aussagensymbole sind und kein Symbol sowohl auf der linken wie der rechten Seite von \vdash auftritt.
Wenn eine solche Sequenz gefunden worden ist, terminiert der Algorithmus, und die ursprüngliche Schlußfolgerung folgt nicht aus den Prämissen.

Wang-Algorithmus: Beispiel

Beispiel eines Beweises mit dem Wang-Algorithmus für die o.a. Sequenz:

$$(1) P \Rightarrow Q, Q \Rightarrow R, \neg R \vdash \neg P$$

$$(2) \neg P \vee Q, \neg Q \vee R, \neg R \vdash \neg P$$

$$(3) \neg P \vee Q, \neg Q \vee R \vdash \neg P, R$$

$$(4) \neg P, \neg Q \vee R \vdash \neg P, R$$

$$(5) Q, \neg Q \vee R \vdash \neg P, R$$

$$(6) Q, \neg Q \vdash \neg P, R$$

$$(7) Q, R \vdash \neg P, R$$

$$(8) Q \vdash \neg P, R, Q$$

Ausgangs-Sequenz

$2 \times \Rightarrow$ -Regel

\neg -Regel

aus (3) mit \vee -Regel

\Rightarrow (4) ist Axiom (\checkmark)

auch aus (3) mit \vee -Regel

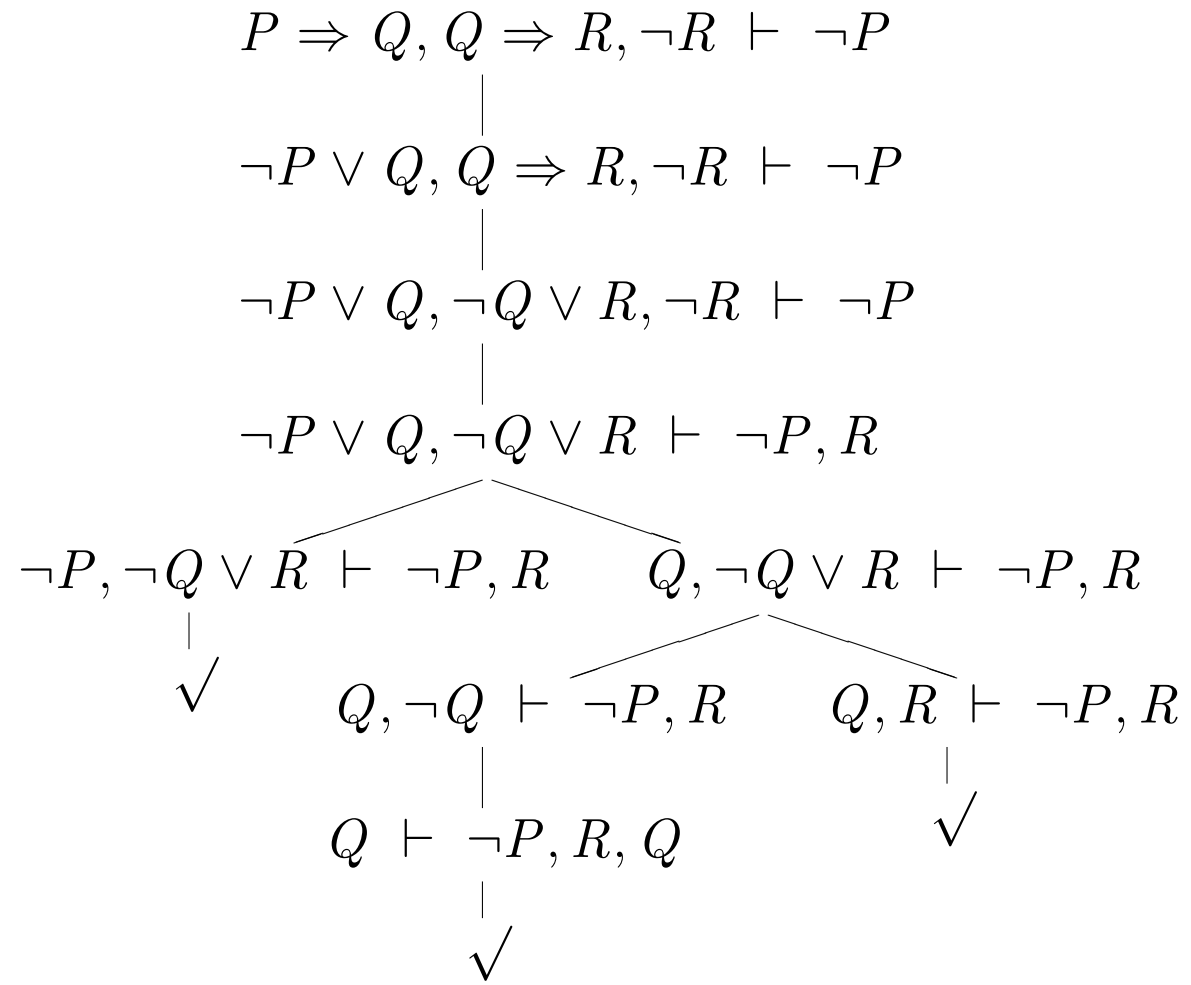
aus (5) mit \vee -Regel

auch aus (5) mit \vee -Regel: \checkmark

aus (6) mit \neg -Regel: \checkmark

alle Sequenzen auf Axiome reduziert: \rightsquigarrow fertig

Beispiel-Beweis als (Und-)Baum:



Wang-Algorithmus (4)

Korrektheit des Wang-Algorithmus?

Der Algorithmus terminiert immer mit einer eindeutigen Lösung:

In jedem Schritt wird eine Verknüpfungsoperation eliminiert, wodurch die Sequenz verkürzt wird, selbst wenn neue Sequenzen generiert werden.

Die Reihenfolge, in der die Regeln angewendet werden, hat keinen Einfluss auf die Terminierung und das Resultat, wohl aber auf die Länge der Ableitung.

Elimination der Schnittregel und Anwendungen

Die Schnittregel *cut* stört einige wünschenswerten Eigenschaften.

Man kann zeigen, daß die Schnittregel redundant ist und sich jeder Beweis durch Schnittelimination “normalisieren” läßt:

Schnitt-Eliminationssatz (Gentzen):

Jeder Beweis, der einen Schnitt enthält, läßt sich in einen Beweis ohne Schnitt überführen

Dabei kann (!) die Beweislänge über-exponentiell anwachsen.

Anwendungen:

Durch die Unterformel-Eigenschaft der verbleibenden Regelmengen enthält man Entscheidungsverfahren für die klassische Aussagenlogik.

Sequenzen-Kalkül für Prädikatenlogik

Zu den aussagenlogischen Regeln kommen die Regeln zur Behandlung quantifizierter Formeln hinzu:

$$\frac{A[x \leftarrow t], \Gamma \vdash \Delta}{\forall x.A, \Gamma \vdash \Delta} \forall L$$

$$\frac{\Gamma \vdash \Delta, A[x \leftarrow y]}{\Gamma \vdash \Delta, \forall x.A} \forall R$$

$$\frac{A[x \leftarrow y], \Gamma \vdash \Delta}{\exists x.A, \Gamma \vdash \Delta} \exists L$$

$$\frac{\Gamma \vdash \Delta, A[x \leftarrow t]}{\Gamma \vdash \Delta, \exists x.A} \exists R$$

Notation: $A[x \leftarrow t]$ bezeichnet Substitution von t für jedes freie Vorkommen der Variablen x in A .

y ist jeweils eine beliebige Variable und t ein beliebiger Term, die folgenden Bedingungen genügen:

- In den Regeln $\forall L$ und $\exists R$ darf keine Variable von t von einem Quantor in A gebunden werden (“*variable capture*”); t muss ‘frei für x in A ’ sein.

Gegebenenfalls sind gebundene Variablen umzubenennen.

- In $\forall R$ und $\exists L$ darf die *Eigenvariable* y nicht frei in der Schlußfolgerung der Regel vorkommen (“Eigenvariablen-Bedingung”)

Beispiele für prädikatenlogische Beweise

1. Beweis der Sequenz $(\forall x.P(x) \wedge Q(x)) \vdash (\forall x.P(x)) \wedge (\forall x.Q(x))$

$$\frac{\frac{\frac{Pv \wedge Qv \vdash Pv}{(\forall x.Px \wedge Qx) \vdash Pv} \forall L}{(\forall x.Px \wedge Qx) \vdash \forall x.Px} \forall R \quad \frac{\frac{Pv \wedge Qv \vdash Qv}{(\forall x.Px \wedge Qx) \vdash Qv} \forall L}{(\forall x.Px \wedge Qx) \vdash \forall x.Qx} \forall R}{(\forall x.Px \wedge Qx) \vdash (\forall x.Px) \wedge (\forall x.Qx)} \wedge R$$

2. Beweis der Sequenz $P(a), \forall x.(P(x) \Rightarrow P(f(x))) \vdash P(f(f(a)))$

$$\frac{\frac{\frac{Pa, Pa \Rightarrow Pfa, Pfa \Rightarrow Pffa \vdash Pffa}{Pa, Pa \Rightarrow Pfa, (\forall x.Px \Rightarrow Pfx) \vdash Pffa} \forall L}{Pa, (\forall x.Px \Rightarrow Pfx), (\forall x.Px \Rightarrow Pfx) \vdash Pffa} \forall L}{Pa, (\forall x.Px \Rightarrow Pfx) \vdash Pffa} \text{contr:L}$$

Es kann also nötig sein, mehrmals die Kontraktionsregel *contr* anzuwenden, um eine Sequenz zu beweisen.

3. Beweis für $(\exists x \forall y.P(x, y)) \vdash (\forall y \exists x.P(x, y))$

$$\frac{\frac{\frac{P(v, w) \vdash P(v, w)}{P(v, w) \vdash \exists x.P(x, w)} \exists R}{\forall y.P(v, y) \vdash \exists x.P(x, w)} \forall L}{\forall y.P(v, y) \vdash \forall y \exists x.P(x, y)} \forall R}{\exists x \forall y.P(x, y) \vdash \forall y \exists x.P(x, y)} \exists L$$

Negativ-Beispiele

Die folgenden Beispiele falscher Regelanwendungen motivieren die Bedingungen für die prädikatenlogischen Regeln.

Vermeidung des Bindens freier Variablen:

Man versuche zu beweisen: $\forall x.P(x, fx) \vdash \exists y \forall x.P(x, y)$

Dies ist eine ungültige Sequenz (interpretiere z.B. P als das Prädikat “ist Vorgänger von” auf den ganzen Zahlen, f als die Nachfolger-Funktion).

$$\frac{\forall x.P(x, fx) \vdash \forall x.P(x, fx)}{\forall x.P(x, fx) \vdash \exists y \forall x.P(x, y)} \exists R$$

Bei dem Rückwärtsschritt gerät die Variable x in dem Term fx unter den Einfluß des All-Quantors.

Negativ-Beispiele [2]

Eigenvariablenbedingung:

Versuche zu beweisen: $(\forall y \exists x.P(x, y)) \vdash (\exists x \forall y.P(x, y))$

Dies ist eine ungültige Sequenz (interpretiere z.B. P als das Prädikat \geq auf den ganzen Zahlen).

$$\frac{\frac{\frac{P(v, w) \vdash P(v, w)}{\exists x.P(x, w) \vdash P(v, w)} \exists L}{\exists x.P(x, w) \vdash \forall y.P(v, y)} \forall R}{\forall y \exists x.P(x, y) \vdash \forall y.P(v, y)} \forall L}{\forall y \exists x.P(x, y) \vdash \exists x \forall y.P(x, y)} \exists R$$

Bei diesem Beweis ist in dem Schritt $\forall R$ die Eigenvariablenbedingung verletzt (w ist frei in der Nachbedingung der Regel).