3. Prädikatenlogik

Gegenüber der Aussagenlogik wird die Sprache der *Prädikatenlogik* (*Predicate Logic*) 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

Syntax (Forts.)

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

Syntax (Forts. 2)

Atomare Formeln:

- Die Konstanten W und F sind atomare Formeln.
- Sind $t_1, \ldots, t_n$ Terme und P ein n-stelliges Prädikatensymbol, dann ist $P(t_1, \ldots, t_n)$ 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*; $\forall$ heißt *Allquantor*, $\exists$ *Existenzquantor*.
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).

Prädikatenlogik: 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 $I[f] : \mathcal{U}^n \to \mathcal{U}$;
- 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

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

.

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 $P(t_1, \ldots, t_n)$ hat $I[P(t_1, \ldots, t_n)]$ den Wert "wahr" genau dann, wenn (gdw.) $(I[t_1], \ldots, I[t_n]) \in I[P]$ 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.

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*.

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 konsistent mit der früher für Aussagenlogik angegebenen.

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 Tautologien generiert 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

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

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

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

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:

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

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

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:

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

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

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

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

Mehrsortige Prädikatenlogik

(*many-sorted predicate logic*)

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.

Vgl. Typ-Begriff in streng-getypten Programmiersprachen wie Modula-2, Pascal usw.

Bemerkung: Die Terminologie ist nicht einheitlich. Es ist üblicher, von "mehrsortiger Logik" als von "getypter Logik" zu sprechen; wir werden hier i.a. "Typ" benutzen, wo in der Literatur häufig von "Sorte" gesprochen wird. Die Begriffe Typ und Sorte sollen hier (im wesentlichen) dasselbe bedeuten.

Mehrsortige Prädikatenlogik (2)

Syntax: (vgl. mehrsortige 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: Funktions*symbole* usw.)

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

$p^{(n)} : S_1 \times \ldots \times S_n \to Bool$ 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

Mehrsortige Prädikatenlogik (3)

Term- und Formel-Bildung:

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

$$\frac{f : S_1 \times \ldots \times S_n \to S_{n+1} \qquad t_i : S_i \;\; (\text{für alle } i \in \{1, \ldots n\})}{f(t_1, \ldots, t_n) : S_{n+1}}$$

– entsprechend für Prädikate

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

$\forall\ v : S.\ P(v) \qquad \exists\, v : S.\ P(v)$

Alle anderen Begriffe bleiben unverändert.

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 M_S, insbesondere Disjunktheit, werden i.a. keine weitere Annahmen gemacht.

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

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

$$f_M : M_{S_1} \times \ldots \times M_{S_n} \to M_{S_{n+1}}$$

(3) Für jedes Prädikatensymbol $P : S_1 \times \ldots \times S_n \to Bool$ ein Prädikat (Teilmenge)

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

bzw.

$$P_M : M_{S_1} \times \ldots \times M_{S_n} \to M_{Bool}$$

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

Wertebelegung für Variable:

$$\mathcal{V}_S : V_S \to M_S \qquad \text{für jedes } S \in \mathcal{S}$$

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:

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

Quantifizierte Formeln werden modifiziert durch Relativisierung ($\mathcal{R}[P]$ bezeichnet die relativisierte Form von P):

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

Zusätzliche *Axiome* $Ax_{\mathcal{S}}$:

(1) $\exists\, x.\ D_S(x)$ für jedes $S \in \mathcal{S}$ (nicht-leere Typen)

(2) $\forall\ x_1, \ldots, x_n.$
$$D_{S_1}(x_1) \wedge \ldots \wedge D_{S_n}(x_n) \Rightarrow D_{S_{n+1}}(f^{(n)}(x_1, \ldots, x_n))$$
für jedes $f^{(n)}$; entsprechend für $P^{(n)}$

Dann gilt für jede Formel A:

$$\vdash A \quad \text{gdw.} \quad Ax_{\mathcal{S}} \vdash \mathcal{R}[A]$$

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

Warum mehrsortige 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

Mehrsortige prädikatenlogische Theorien

Logische Beschreibung mehrsortiger Algebren

Beispiel:

$$\mathcal{S} = \{Bool, Int, Liste\}$$

$$\begin{array}{rcl} W, F & : & Bool \\ \wedge, \vee, \Rightarrow, \Leftrightarrow & : & Bool \times Bool \rightarrow Bool \\ \neg & : & Bool \rightarrow Bool \end{array}$$

$$\begin{array}{rcl} \mathsf{0} & : & Int \\ \mathsf{1} & : & Int \\ +, - & : & Int \times Int \rightarrow Int \\ =_I & : & Int \times Int \rightarrow Bool \end{array}$$

$$
\begin{array}{rcl}
Kopf & : & Liste \rightarrow Int \\
Rumpf & : & Liste \rightarrow Liste \\
leer & : & Liste \\
ist\_leer & : & Liste \rightarrow Bool \\
\_ \cdot \_ & : & Int \times Liste \rightarrow Liste \\
=_L & : & Liste \times Liste \rightarrow Bool
\end{array}
$$

Typische Ausdrücke:

$$(3+5)-(2+1)$$

$$1 \cdot ((x+y) \cdot leer)$$

$$\neg(leer =_L 2 \cdot leer)$$

(Nichtlogische) Axiome:

Axiome für Operationen auf $Bool$:

$$W \wedge W \Leftrightarrow W$$
$$\forall\ b : Bool.\ \ b \wedge F \Leftrightarrow F$$
$$\forall\ b_1 : Bool.\ \forall\ b_2 : Bool.\ \ b_1 \wedge b_2 \Leftrightarrow b_2 \wedge b_1$$
$$\ldots$$

Axiome für Operationen auf Int:

$$\forall\ n : Int.\, 0 + n =_I n$$
$$\forall\ n : Int.\ \forall\ m : Int.\ \forall\ p : Int.$$
$$\quad n + (m + p) =_I (m + m) + p$$
$$\ldots$$

Axiome für Operationen auf $Liste$:

$$\forall\ n : Int.\ \forall\ l : Liste.\ \ Kopf(n \cdot l) =_I n$$
$$\forall\ n : Int.\ \forall\ l : Liste.\ \ Rumpf(n \cdot l) =_L l$$
$$\forall\ n : Int.\ \forall\ l : Liste.\ \ ist\_leer(leer) \Leftrightarrow W$$
$$\forall\ n : Int.\ \forall\ l : Liste.\ \ ist\_leer(n \cdot l) \Leftrightarrow F$$
$$\ldots$$

Im weiteren werden wir boolsche Ausdrücke mit (atomaren) Formeln identifizieren und einfach schreiben $P(x)$ statt $P(x) \Leftrightarrow W$;
entsprechend $\neg P(x)$ für $P(x) \Leftrightarrow F$.

- Axiomatisierung mit Hilfe von Gleichungen: “Gleichungsspezifikation”

- Modelle sind mehrsortige Algebren.

- Beispiel kann als schichtenweise Erweiterung der Theorie für $Bool$ aufgefaßt werden.

Axiomatisches System für Prädikatenlogik

“Hilbert-Kalkül” für Prädikatenlogik:
erweitert das für Aussagenlogik bereits besprochene System

Axiomen-Schemata: für beliebige Formeln P, Q, R

$$P \Rightarrow (Q \Rightarrow P)$$

$$(P \Rightarrow (Q \Rightarrow R)) \Rightarrow ((P \Rightarrow Q) \Rightarrow (P \Rightarrow R))$$

$$(\neg P \Rightarrow \neg Q) \Rightarrow (Q \Rightarrow P)$$

$$(\forall\ x.\ P) \Rightarrow P[x \leftarrow t]$$

sofern die Substitution zulässig ist (keine freien Variablen in t werden durch das Einsetzen gebunden).

$$(\forall\ x.\ P \Rightarrow Q) \Rightarrow (P \Rightarrow (\forall\ x.\ Q)$$

sofern x nicht frei in P vorkommt.

Schlußregeln:

modus ponens

$$\frac{X \qquad X \Rightarrow Y}{Y}$$

Generalisierung:

$$\frac{P}{\forall\, x.\, P}$$

Der Kalkül definiert eine Ableitungsrelation $\vdash_{PL1}$ für PL1.

Axiomatisches System für Prädikatenlogik (2)

Satz:

- Jede Instanz der Axiomenschemata ist eine allgemeingültige Formel.
- Die Schlußregeln erhalten Gültigkeit.

Satz: Der Kalkül ist korrekt:
Für jede Formelmenge M und Formel P gilt
$M \vdash P$ impliziert $M \models P$

Satz: (Deduktionstheorem für Prädikatenlogik)
Für jede Formelmenge M, geschlossene Formel P und Formel Q gilt
$M \cup \{P\} \vdash Q$ genau dann, wenn $M \vdash (P \Rightarrow Q)$

Dieser Kalkül ist primär von theoretischem Interesse, d.h. geeignet für metatheoretische Ableitungen, aber nicht als Grundlage für Beweisverfahren.

Normalformen

Konjunktive und *disjunktive Normalform* sind wie in der Aussagenlogik definiert, aber bezüglich des erweiterten Begriffs von Literalen.
Diese Normalformen sind nur für quantorenfreie Formeln interessant.

Eine Formel F heißt *bereinigt*, wenn keine Variable sowohl frei als auch gebunden in F vorkommt und alle hinter Quantoren auftretenden Variablen verschieden sind (d.h. wenn dieselbe Variable nicht an verschiedenen Stellen gebunden wird).
Durch Umbenennung von gebundenen Variablen (s.o.) kann jede Formel in eine äquivalente bereinigte Formel umgewandelt werden.

Pränex-Normalform (PNF): Eine Formel ist in PNF, wenn sie die Form

$$Q_1 x_1. \ \dots Q_n x_n. F$$

hat, wobei jedes Q_i für einen Quantor ($\forall$ oder $\exists$) steht und die Formel F keinen Quantor enthält, d.h. quantorenfrei ist.
F wird auch *Matrix* der Formel in PNF genannt.

Satz: Jede Formel kann in eine äquivalente Formel in Pränex-Normalform umgewandelt werden.

Beweis: durch Induktion über den Aufbau von Formeln und Ausnutzung z.B. der oben angegebenen Äquivalenzen.

Ab sofort wird für Formeln in PNF angenommen, daß sie auch bereinigt und universell abgeschlossen sind.

Resolution

Resolution zunächst als Beweisverfahren für Aussagenlogik

Theorem: Eine Formel Y folgt semantisch aus einer Formelmenge $\{X_1, \ldots, X_n\}$ genau dann, wenn

$$X_1 \wedge \ldots \wedge X_n \Rightarrow Y$$

gültig ist.

Bemerkung: Eine Formel X ist (algemein)gültig genau dann, wenn ihre Negation $\neg X$ unerfüllbar, also widersprüchlich ist.

Beweis durch Widerspruch: Um zu zeigen, daß aus einer Formel X die Formel Y folgt, d.h. daß $X \Rightarrow Y$ gültig ist, weist man nach, daß die Negation $\neg(X \Rightarrow Y)$, also $X \wedge \neg Y$, widersprüchlich ist.

Resolventenregel:

$$\frac{\{A, L_1, \ldots, L_m\} \qquad \{\neg A, K_1, \ldots, K_n\}}{\{L_1, \ldots, L_m, K_1, \ldots, K_n\}}$$

Die Schlußfolgerung der Regel heißt *Resolvente* der Prämissen.

Resolution (2)

Bemerkung: Die Resolventenregel kann als Verallgemeinerung von *Modus ponens* aufgefaßt werden:

$$\frac{\{A\} \qquad \{\neg A, B\}}{\{B\}}$$

Resolution ist ein Beweisverfahren mit der Resolventenregel als (einziger) Schlußregel. Es ist ein Widerlegungsverfahren, d.h. ein Verfahren für Beweise durch Widerspruch.

Resolutionssatz: Eine Klauselmenge ist widersprüchlich genau dann, wenn sich aus ihr mit der Resolventenregel die leere Klausel, bezeichnet durch $\Box$, ableiten läßt.

Resolution (3)

Um mit Resolution zu zeigen, daß eine Formel Y aus einer Menge von Formeln X_1, . . . X_n folgt, geht man folgendermaßen vor:

- Die Formel $X_1 \wedge \ldots X_n \wedge \neg Y$ wird in konjunktive Normalform gebracht, so daß sie als Menge von Klauseln dargestellt werden kann.
- Auf geeignete Paare von Klauseln aus der Klauselmenge wird die Resolventenregel angewendet; die Resolvente wird der Klauselmenge hinzugefügt.
- Wenn die leere Klausel abgeleitet wird, ist der Widerspruch gezeigt.

  Andernfalls bricht das Verfahren ab, wenn keine neuen Klauseln mehr mit der Resolventenregel abgeleitet werden können.

Skolem-Form von Formeln

Skolemisierung: Ersatz von existentiell quantifizierten Variablen durch Aufrufe von *Skolem-Funktionen* nach folgendem Verfahren:

Sei $F = \forall\ x_1.\ \ldots \forall\ x_n.\ \exists\ y.\ G$, also y sei die am weitesten links auftretende existentiell quantifizierte Variable, und G sei eine Formel in PNF (mit den angegebenen Einschränkungen).

- Wähle ein neues n-stelliges Funktionssymbol f.
- Streiche $\exists\ y.$ in F.
- Ersetze jedes Vorkommen von y in G durch den Term $f(x_1, \ldots, x_n)$.

Auf diese Weise werden in F sukzessive alle $\exists$-Quantoren ersetzt. Das Resultat dieser Transformation heißt *Skolemform* von F.

Satz: Eine Formel (in PNF) F ist unerfüllbar (widersprüchlich) genau dann, wenn ihre Skolemform unerfüllbar ist.

Konversion von Formeln in Klausel-Form

Um Resolution auf prädikatenlogische Formeln anwenden zu können, müssen sie in Klausel-Form gebracht werden.
Mit den bereits eingeführten Transformationen läßt sich dies erreichen:

- Implikationen eliminieren.
- Negationen nach innen schieben, so daß sie nur in negativen Literalen vorkommen.

- Variablen umbenennen, soweit notwendig.
- Quantoren nach außen schieben: Erstellen einer PNF.
- Existentielle Quantoren eliminieren: Skolemisierung.
- Verbleibende Matrix in KNF bringen.
- Verbleibende All-Quantoren streichen.

Skolemisierung kann auch vor Erstellen einer PNF erfolgen.

Resolution in Prädikatenlogik erfordert im allgemeinen *Unifikation*.

Resolution für Prädikatenlogik

Binäre Resolution zweier Klauseln K_1 und K_2, die keine Variablen gemeinsam haben: Literale L_1 in K_1 und L_2 in K_2, so daß $\sigma = \mathsf{mgu}(L_1, \neg L_2)$

$$\frac{K_1 \qquad\quad K_2}{\sigma(K_1 - L_1) \cup \sigma(K_2 - L_2)}$$

Beispiele für Unifikation bei Resolutionsschritten:

- Für die Klauseln $\{P(f(a), x), Q(x)\}$ und $\{\neg P(g(a), x), R(x)\}$ gibt es keinen Unifikator für einen Resolutionsschritt.

- In den Klauseln $\{P(f(a), x), Q(x)\}$ und $\{\neg P(y, g(b)), R(g(b))\}$ sind die betreffenden Literale unifizierbar mit der Substitution $\{x \leftarrow g(b), y \leftarrow f(a)\}$.
  Resolvente: $\{Q(g(b)), R(g(b))\}$.

Faktor-Regel: Für $K_1 \subset K$, $\sigma = \mathsf{mgu}(K_1)$,

$$\frac{K}{\sigma(K)}$$

Beispiel für Faktor:

$$\begin{array}{ll} K & = \{P(x), Q(x), Q(f(a))\} \\ \sigma & = \{x \leftarrow f(a)\} \\ \sigma(K) & = \{P(f(a)), Q(f(a))\} \end{array}$$

Resolutionsprinzip: Eine Resolvente von zwei Klauseln K_1 und K_2 ist eine binäre Resolvente von K_1 oder einem Faktor von K_1, und K_2 oder einem Faktor von K_2.

Vor Anwendung der Resolutionsregel kann es notwendig sein, Variablen in den Klauseln so umzubenennen, daß die Mengen der freien Variablen in den Klauseln disjunkt sind.

Beispiel für einen Resolutionsschritt mit Variablen-Umbenennung:

$$\{P(f(x)), \neg Q(z), P(z)\} \qquad \{\neg P(x), R(g(x), a)\}$$

$$\sigma_1 = \{\} \qquad \sigma_2 = \{x \leftarrow u\}$$

$$\{\neg P(u), R(g(u), a)\}$$

$$\sigma = \{z \leftarrow f(x), u \leftarrow f(x)\}$$

$$\{\neg Q(f(x)), R(g(f(x)), a)\}$$

Ableitung einer Klausel K aus einer Menge S von Klauseln ist eine endliche Folge von Resolutionsschritten, wobei die Eingabe-Klauseln entweder Elemente der Menge S oder vorhergehende Resolventen sind, und K die letzte Resolvente ist.

Ein einfaches Beispiel:

Prädikate:

$Auf(x, y)$	Objekt x liegt *auf* Objekt y
$Ueb(x, y)$	Objekt x liegt *über* Objekt y

Formeln, die die Prädikate charakterisieren:

(i) $\forall\ u, v.\ Auf(u, v) \Rightarrow Ueb(u, v)$
(ii) $\forall\ x, y, z.\ Ueb(x, y) \wedge Ueb(y, z) \Rightarrow Ueb(x, z)$

Beschreibung der Situation:

(a) $Auf(A, Tisch)$
(b) $Auf(B, A)$

zu zeigen: $Ueb(B, Tisch)$

Klauseln:

(1)	$\neg Auf(u, v),\ Ueb(u, v)$	(i) als Klausel
(2)	$\neg Ueb(x, y),\ \neg Ueb(y, z),\ Ueb(x, z)$	(ii) als Klausel
(3)	$Auf(B, A)$	(a)

(4)	$Auf(A, Tisch)$	(b)	
(5)	$\neg Ueb(B, Tisch)$	negierte Zielformel	

Resolventen:

(6)	$\neg Ueb(B, y),\ \neg Ueb(y, Tisch)$	(2),(5)	$\{x \leftarrow B, z \leftarrow Tisch\}$
(7)	$\neg Ueb(B, y),\ \neg Auf(y, Tisch)$	(6),(1)	$\{u \leftarrow y, v \leftarrow Tisch\}$
(8)	$\neg Auf(B, y),\ \neg Auf(y, Tisch)$	(7),(1)	$\{u \leftarrow B, v \leftarrow y\}$
(9)	$\neg Auf(A, Tisch)$	(8),(3)	$\{y \leftarrow A\}$
(10)	$\square$	(9),(4)	$\{\}$

Herbrand-Theorie

Im folgenden sei G eine geschlossene Formel in Skolemform.

Herbrand-Universum $D(G)$ von G: Menge aller variablenfreien Terme, die aus Bestandteilen von G gebildet werden können:

- Alle in G vorkommenden Konstanten sind in $D(G)$; falls keine Konstanten in G vorkommen, wird eine beliebige Konstante $a \in D(G)$ gewählt.
- $D(G)$ ist unter Termbildung mit allen in G vorkommenden Funktionssymbolen f abgeschlossen, d.h. für $t_1, \ldots, t_n \in D(G)$ ist $f(t_1, \ldots, t_n) \in D(G)$.

Beispiel für G: $\forall\, x.\ \forall\, y.\ \forall\, z.\ P(x, f(y), g(z, x))$

$$D(G) = \{a, f(a), g(a, a), g(a, f(a)), f(f(a)), f(g(a, a)), \ldots\}$$

Herbrand-Interpretation für eine Formel G:

- Universum ist $D(G)$.
- Interpretation eines Funktionssymbols ist die syntaktische Termbildung mit dem Symbol (Grundterme werden “durch sich selbst” interpretiert).
- Interpretation der Prädikatensymbole ist nicht festgelegt.

Herbrand-Theorie (2)

Ein *Herbrand-Modell* einer Formelmenge ist ein Modell auf der Grundlage einer Herbrand-Interpretation.

Satz: Wenn eine Formelmenge überhaupt ein Modell hat, dann hat ihre skolemisierte Form auch ein Herbrand-Modell.

Herbrand-Theorem: Eine Formel G in KNF (bzw. eine Menge von Klauseln) ist unerfüllbar genau dann, wenn es eine unerfüllbare endliche Menge von Grundinstanzen der Klauseln in G gibt.

Satz: Vollständigkeit der Resolution (in Prädikatenlogik):
Eine Menge S von Klauseln ist unerfüllbar genau dann, wenn die leere Klausel aus S abgeleitet werden kann (Widerlegungsvollständigkeit).

Semantische Tableaux

Semantische Tableaux geben ein Beweisverfahren, mit dem ähnlich wie mit Resolution eine Formel dadurch bewiesen wird, daß ihre Negation als widersprüchlich abgeleitet wird (*proof by refutation*).

Semantische Tableaux basieren in gewisser Weeise auf einer Darstellung von Formeln in disjunktiver Normalform (Resolution: konjunktive Normalform).
Es wird ein Baum konstruiert, in dem jeder Knoten mit einer Formel markiert ist. Ein Pfad von der Wurzel zu einem Blatt stellt die Konjunktion aller Formeln der Knoten entlang des Pfads dar; eine Verzweigung stellt eine Disjunktion dar.

Der Baum wird aufgebaut durch Anwendung einer der Tableau-Erweiterungsregeln.

Ein Pfad in einem Tableau ist *abgeschlossen*, wenn entlang des Pfads sowohl X wie $\neg X$ für eine Formel X auftreten, oder wenn F auftritt.
(X muß nicht atomar sein.)
Ein Tableau heißt abgeschlossen, wenn alle seine Pfade abgeschlossen sind.

Ein Tableau-Beweis für eine Formel X ist ein abgeschlossenes Tableau für $\neg X$.

Auswahl der Regeln bei der Erweiterung eines Tableau ist nichtdeterministisch. Für aussagenlogische Tableaux kann die Auswahl etwas eingeschränkt werden:

Ein Tableau heißt *strikt*, wenn keine Formel entlang eines Pfads mehr als einmal mit einer Regel erweitert wurde.

Tableau-Erweiterungsregeln

zunächst für Aussagenlogik:

$$\frac{\neg\neg X}{X} \qquad \frac{\neg W}{F} \qquad \frac{\neg F}{W}$$

für konjunktive Formeln („α-Regeln"):

$$\frac{\alpha}{\begin{array}{c}\alpha_1\\ \alpha_2\end{array}} \qquad \frac{X \wedge Y}{\begin{array}{c}X\\ Y\end{array}} \qquad \frac{\neg(X \vee Y)}{\begin{array}{c}\neg X\\ \neg Y\end{array}} \qquad \frac{\neg(X \Rightarrow Y)}{\begin{array}{c}X\\ \neg Y\end{array}}$$

für disjunktive Formeln („β-Regeln"):

$$\frac{\beta}{\beta_1 \mid \beta_2} \qquad \frac{X \vee Y}{X \mid Y} \qquad \frac{\neg(X \wedge Y)}{\neg X \mid \neg Y} \qquad \frac{X \Rightarrow Y}{\neg X \mid Y}$$

Tableau-Beweis: Beispiel

Beispiel eines aussagenlogischen Tableau-Beweises für

$(P \Rightarrow (Q \Rightarrow R)) \Rightarrow ((P \Rightarrow Q) \Rightarrow (P \Rightarrow R))$

(1)	$\neg[(P \Rightarrow (Q \Rightarrow R)) \Rightarrow ((P \Rightarrow Q) \Rightarrow (P \Rightarrow R))]$	
(2)	$(P \Rightarrow (Q \Rightarrow R))$	(α aus 1)
(3)	$\neg[(P \Rightarrow Q) \Rightarrow (P \Rightarrow R)]$	(α aus 1)
(4)	$(P \Rightarrow Q)$	(α aus 3)
(5)	$\neg(P \Rightarrow R)$	(α aus 3)
(6)	P	(α aus 5)
(7)	$\neg R$	(α aus 5)
(8)	$\neg P$ | (9) $(Q \Rightarrow R)$	(β aus 2)
	(10) $\neg Q$ | (11) R	(β aus 9)
	(12) $\neg P$ | (13) Q	(β aus 4)

Tableau-Erweiterungsregeln für Prädikatenlogik

- Erweiterungsregeln wie für Aussagenlogik - in den Regeln stehen X und Y dann für beliebige (prädikatenlogische) Formeln
- Zusätzlich die folgenden Regeln für die Behandlung quantifizierter Formeln:

$$\frac{\gamma}{\gamma[t]} \qquad \frac{\delta}{\delta[c]}$$

γ ist eine universell quantifizierte Formel, δ eine existentiell quantifizierte Formel.

$\gamma[t]$ und $\delta[c]$ ergeben sich aus den folgenden Tabellen:

γ	$\gamma[t]$
$\forall\, x.\, \Phi$	$\Phi[x \leftarrow t]$
$\neg \exists\, x.\, \Phi$	$\neg\Phi[x \leftarrow t]$

δ	$\delta[c]$
$\exists\, x.\, \Phi$	$\Phi[x \leftarrow c]$
$\neg \forall\, x.\, \Phi$	$\neg\Phi[x \leftarrow c]$

Hierbei sind t ein Grundterm (oder allgemeiner: ein Term, der keine Variablen enthält, die in Φ gebunden sind) und c eine “neue” Konstante.

Tableau-Beweis in PL1: Beispiel

Beispiel: Tableau-Beweis für

$$(\forall\ x.\ P(x) \lor Q(x)) \Rightarrow (\exists\ x.\ P(x)) \lor (\forall\ x.\ Q(x))$$

(1)	$\neg[(\forall\ x.\ P(x) \lor Q(x)) \Rightarrow (\exists\ x.\ P(x)) \lor (\forall\ x.\ Q(x))]$	
(2)	$\forall\ x.\ P(x) \lor Q(x)$	(α aus 1)
(3)	$\neg[(\exists\ x.\ P(x)) \lor (\forall\ x.\ Q(x))]$	(α aus 1)
(4)	$\neg\exists\ x.\ P(x)$	(α aus 3)
(5)	$\neg\forall\ x.\ Q(x)$	(α aus 3)
(6)	$\neg Q(c)$	(δ aus 5)
(7)	$\neg P(c)$	(γ aus 4)
(8)	$P(c) \lor Q(c)$	(γ aus 2)
(9)	$P(c)$ | (10) $Q(c)$	(β aus 6)

Tableau-Regeln mit Variablen

Statt geeignete Instanzen der γ- und δ-Regeln zu “erraten”, können auch freie Variablen und Skolem-Funktionen in der Erweiterung benutzt werden.

Die folgenden Regeln ersetzen die vorher für Tableau-Beweise angegebenen γ- und δ-Regeln.

Tableau-Erweiterungsregeln mit Skolem-Funktionen und freien Variablen

$$\frac{\gamma}{\gamma[v]} \qquad\qquad \frac{\delta}{\delta[f(v_1, .., v_n)]}$$

Hierbei sind v eine neue Variable, f eine neue Skolem-Funktion, $v_1, \ldots, v_n$ alle bisher in dem betreffenden Zweig des Tableaus eingeführten freien Variablen.

Tableau-Substitutionsregel:

T sei ein Tableau für eine Menge S von geschlossenen Formeln, σ eine Substitution, die frei für T (d.h. frei für alle Formeln in T) ist; dann ist $\sigma(T)$ (d.h. jede Formel X in T durch $\sigma(X)$ ersetzt) auch eine Tableau für S.

Eine elementare Substitution $\{x \leftarrow t\}$ ist *frei für eine Formel* P, wenn x in P nicht im Bindungsbereich einer Variablen y vorkommt, die auch in t vorkommt.

Eine Substitution σ ist frei für eine Formel P, when jede ihrer Elementar-Substitutionen frei für P ist.

Tableau-Beweis in PL1: Beispiel (modifiziert)

Das erste Beispiel mit Benutzung von freien Variablen:

...				
(6)	$\neg Q(c)$			(δ aus 5)
(7)	$\neg P(v_1)$			(γ aus 4)
(8)	$P(v_2) \vee Q(v_2)$			(γ aus 2)
(9)	$P(v_2)$	|	(10) $Q(v_2)$	(β aus 6)

Mit der Substitution $\sigma = \{v_1 \leftarrow c,\ v_2 \leftarrow c\}$ kann das Tableau geschlossen werden

Beispiel eines Tableau-Beweises mit **freien Variablen** und **Skolem-Funktionen**:

Zu beweisen:

$$(\exists\ w.\ \forall\ x.\ R(x, w, g(x, w))) \Rightarrow (\exists\ w.\ \forall\ x.\ \exists\ y.\ R(x, w, y))$$

(1)	$\neg[(\exists\ w.\ \forall\ x.\ R(x, w, g(x, w)))$ $\Rightarrow (\exists\ w.\ \forall\ x.\ \exists\ y.\ R(x, w, y))]$	
(2)	$\exists\ w.\ \forall\ x.\ R(x, w, g(x, w))$	(α aus 1)
(3)	$\neg[\exists\ w.\ \forall\ x.\ \exists\ y.\ R(x, w, y)]$	(α aus 1)
(4)	$\forall\ x.\ R(x, a, g(x, a))$	(δ aus 2)
(5)	$\neg\forall\ x.\ \exists\ y.\ R(x, v_1, y)$	(γ aus 3)
(6)	$\neg\exists\ y.\ R(b(v_1), v_1, y)$	(δ aus 5)
(7)	$R(v_2, a, g(v_2, a))$	(γ aus 4)
(8)	$\neg R(b(v_1), v_1, v_3)$	(γ aus 6)

$\sigma = \{v_1 \leftarrow a, v_2 \leftarrow b(a), v_3 \leftarrow g(b(a), a)\}$ ist frei für alle Formeln im Tableau (keine rechte Seite enthält eine Variable)

daher: Die Tableau-Substitutionsregel kann mit σ angewendet werden;
unter σ sind die Zeilen (7) und (8) widersprüchlich.

Tableau-Verfahren: Eigenschaften

Sätze über das Tableau-Verfahren:

1. Tableau-Regeln erhalten *Erfüllbarkeit*: wenn die Wurzel-Formel eines Tableaus erfüllbar ist, dann gibt es mindestens einen Pfad im Tableau, der erfüllbar ist.
2. Eine Tableau-Formel G ist unerfüllbar genau dann, wenn ein geschlossenes Tableau für G existiert.
   (Korrektheit und Widerlegungsvollständigkeit des Verfahrens)

Das Verfahren kann auf unerfüllbare *Mengen* von Formeln erweitert werden.

Tableaux-Verfahren werden zunehmend auch für maschinelles Beweisen herangezogen, als Alternative zu Resolution. Wegen der relativ einfachen Modifierbarkeit der Dekompositionsregeln werden Tableau-Verfahren insbesondere auch für andere Logiken (wie intuitionistische Logik, Modal-Logiken) entwickelt.

Literatur zum Tableau-Verfahren: insbesondere
R. M. Smullyan, *First Order Logic*, Springer-Verlag, 1968.

Tableau-Verfahren und Resolution

Gemeinsamkeiten:

- Beweis durch Widerlegung
- ausgehend von der zu beweisenden (bzw. zu widerlegenden) Formel
- Steuerung durch Auswahl-Strategien

Unterschiede:

- Resolution setzt Transformation der Formeln in Klauselform voraus; Tableau-Verfahren nimmt in gewisser Weise eine Transformation während der Konstruktion des Tableau vor, d.h. Tranformation ist Teil der Regeln.
- Resolution erfordert Reduktion auf leere Klausel für erfolgreichen Beweis; Tableau-Verfahren kann (im Prinzip) erfolgreich stoppen, ohne daß die Formel vollständig auf Atome reduziert ist.
- Es ist vermutlich einfacher, aus einem nicht weiter reduzierbaren Tableau ein Modell (und damit ein Gegenbeispiel) abzulesen, als aus einem fehlgeschlagenen Resolutionsbeweis.