

b) Formel $F: \forall x \forall y \exists z$

$$\rightarrow (P(x, y) \rightarrow P(x, z) \wedge P(z, y))$$

In der Struktur $A = (\mathbb{R}; <)$ wird dies
zu:

$$\forall x \forall y \exists z ((x < y) \rightarrow ((x < z) \wedge (z < y)))$$

oder 'Für alle reellen Zahlen x, y
gilt: Es gibt eine Zahl z , so daß:

Wenn $x < y$ so folgt $x < z < y$.'

Aber: Für die Struktur $A' = (\mathbb{N}, <)$ gilt
die Formel nicht.

Man schreibt: $A \models F$ (bzw. $A \not\models F$).

'A erfüllt F' oder 'A ist ein Modell für F'

Def. (Erfüllbarkeit): Eine Formel heißt
erfüllbar, wenn sie (mindestens) ein
Modell besitzt

Def. (Tautologie): Eine Formel ist allgemei-
ngültig, wenn sie von sich (zu Signatur
passenden) Strukturen erfüllt wird.

Tautologien sind die Regeln der PL
($\hat{=}$ Ableitungsregeln bei Termersetzungssystemen)

Beispiele für Tautologien

64

$$a) (P(x) \rightarrow Q(x)) \wedge (Q(x) \rightarrow R(x)) \equiv P(x) \rightarrow R(x)$$

(Transitivität)

$$b) \neg \forall x P(x) \equiv \exists x (\neg P(x))$$

De Morgan

$$\neg \exists x P(x) \equiv \forall x (\neg P(x))$$

$$c) \forall x (P(x) \wedge Q(x)) \equiv (\forall x P(x)) \wedge (\forall x Q(x))$$
$$\exists x (P(x) \vee Q(x)) \equiv (\exists x P(x)) \vee (\exists x Q(x))$$

$$d) (\exists x \forall y P(x, y)) \rightarrow (\forall y \exists x P(x, y))$$

Aber: die Umkehrung

$$(\forall y \exists x P(x, y)) \rightarrow (\exists x \forall y P(x, y))$$

ist keine Tautologie, allerdings erfüllbar

(siehe Heimstudium)

e) ... weitere Tautologien siehe Literatur
(sehr empfehlenswert!)

Bezeichnung: Damit einer Formel ein eindeutiger Wahrheitswert zugeordnet werden kann, darf sie keine freien Variablen enthalten. Alle Variablen müssen durch ein Quantor gebunden sein (sich in seinem Einflussbereich befinden).

Gegenbeispiel: Sei $F = (\forall x G) \wedge H$ und kommt x in H vor, so ist F nicht Tautologie.

Resolutionskalkül ist Korrekt :

Wenn die leere Klausel ableitbar ist,
dann ist die Ausgangsformel unerfüllbar

65

Resolutionskalkül ist vollständig :

Wenn die Ausgangsformel unerfüllbar ist,
dann läßt sich die leere Klausel
ableiten

Bemerkung : In der Programmiersprache
PROLOG werden Beweise mit Hilfe
des Resolutionskalküls gefunden. Programme
werden dabei als Klauseln formuliert
(als KNF-Formeln). Prolog verwendet
dabei sog. Horn-Klauseln, das sind
Klauseln mit höchstens einer nicht neg.
Variable, also der Form $(x \vee \bar{y}_1 \vee \bar{y}_2 \vee \dots \vee \bar{y}_n)$.
Diese Klauseln können mit Hilfe von de
Morgan und der Äquivalenz $(a \rightarrow b) \Leftrightarrow (\neg a \vee b)$
umgeformt werden zu :

$$x \leftarrow (y_1 \wedge y_2 \wedge \dots \wedge y_n),$$

was der Prolog-Syntax

$$x :- y_1, y_2, y_3 \text{ entspricht.}$$

Beispiele wie ist

Wichtige Technik zum Beweis der Erfüllbarkeit
von PL-Formeln: Resolution

66

Motivation: wäre es wert, daß von jeder
unerfüllbaren Formel bewiesen werden kann,
daß sie unerfüllbar ist und zwar in endlich
viele Schritten. Resolution versucht, Beweise
durch Widerlegung zu finden. Also: die
Negation einer Aussage stellt in Widerspr-
spruch zu bekannten Aussagen, daß also
die wegige Aussage unerfüllbar ist

Zunächst: Resolution in der Aussagenlogik
hier Verwendung der Res., um ebenfalls
Tautologie bzw. Erfüllbarkeits eigenschaften
nach zu weisen. Dazu Anwendung von Regeln
wie Distributivgesetz, De Morgan, ... solange,
bis ableitbar, daß Formel z.B. Tautologie ist.

Def.: (Kalkül): Systematische Aufstellung der
Umformungsregeln mit Angabe ihrer Art
und Reihenfolge der Anwendung heißt
(Logik-) Kalkül.

Resolutionskalkül besteht nun aus
 einer Regel, die anwendbar ist auf zwei
 geg. Formeln F und G , einer ~~Regel~~
 Formelmengen und daraus die neue
 Formel H erzeugt.

F und G müssen Disjunktiv sein,
 also aus Teilformeln f_i und g_i bestehen,
 die disjunktiv verknüpft sind:

$$F = f_1 \vee f_2 \vee \dots \vee f_n \quad G = g_1 \vee g_2 \vee \dots \vee g_m$$

Um Resolutionsregel anwenden zu können,
 muß es eine Teilformel f_i geben, die
 die Negation einer Teilformel aus G ist,

also $f_i = \bar{g}_j$. Vermöge der Beziehung
 $(x \vee \bar{x}) \wedge (x \vee x) = x$ wird die resultierende
 Formel H , die Resolvente gebildet zu:

$$H = f_1 \vee \dots \vee f_{i-1} \vee f_{i+1} \vee \dots \vee f_n \vee g_1 \vee \dots \vee g_{j-1} \vee g_{j+1} \vee \dots \vee g_m$$

Also: Alle Teilformeln werden 'verodert',
 f_i und g_j werden weggelassen und evtl.
 Duplikate werden weggelassen.

Die eigentliche Ableitung (Beweis) besteht aus
 einer Folge von Formeln, die entweder die

ursprünglichen sind oder paarweise aus Resolution entstanden sind.

Am Ende der Folge: 'Leere Klausel'; besteht aus keiner Teilformel mehr und heißt: Ausgangsformel unerfüllbar

Satz (Resolution): Sei $F_1 \wedge F_2 \wedge \dots \wedge F_n$ die Ausgangsformel in KNF (Konjunktive Normalform), d.h. F_i ist eine Disjunktion von negierten oder nichtnegierten Variablen $(x_1 \vee \bar{x}_2 \vee \dots \vee x_m \vee \dots \vee x_n)$ und wird auch Klausel genannt.

Dann ist $F_1 \wedge F_2 \wedge \dots \wedge F_n$ genau dann unerfüllbar, wenn es einen Resolutionsbeweis für die leere Klausel gibt, welcher mit den Ausgangsklauseln beginnt.

Beispiel:

$$(x \vee \bar{z}) \wedge (x \vee y \vee z) \wedge (\bar{x} \vee \bar{y} \vee \bar{z}) \wedge (\bar{x} \vee y) \wedge (\bar{y} \vee z)$$

