Kapitel 10

Prädikatenlogik

In diesem Kapitel behandeln wir die Familie der prädikatenlogischen Sprachen erster Ordnung. Mit ASSN haben wir bereits eine solche Sprache kennengelernt. Eine prädikatenlogische Sprache erster Ordnung ist durch eine Menge U (das sogenannte Universum, bei ASSN $\mathbb{Z}$) und durch eine Menge von Funktionen des Typs $U^n \to U$ oder $U^n \to \mathbb{B}$ gegeben. Die Funktionen des zweiten Typs nennt man Prädikate (ASSN hat das Prädikat $\leq$).

Lesematerial

[Schöning, Kapitel 2]

10.1 Syntax

Abbildung 10.1 definiert die Syntax von prädikatenlogischen Sprachen. Ausgangspunkt sind die drei Mengen *Var*, *Fun* und *Pre*, die die benötigten Symbole zur Verfügung stellen. Jedem Funktions- und Prädikatensymbol ist durch die Stelligkeitsfunktion eine feste Anzahl von Argumenten zuordnet. Nullstellige Funktionssymbole bezeichnet man auch als *Konstanten*. Aufbauend auf den Symbolen werden Terme und Formeln definiert. Die Funktion dieser syntaktischen Objekte ist aus ASSN bekannt.

Die Konstruktion der Symbole sorgt dafür, dass es unendlich viele Variablen und für jede Stelligkeit unendlich viele Funktions- und Prädikatensymbole gibt. Ausserdem sind die Mengen *Ter* und *For* gödelisierbar und entscheidbar.

Eine *atomare Formel* ist eine Formel der Form $p(t_1, \ldots, t_n)$.

Eine *Signatur* Σ ist eine nichtleere und entscheidbare Teilmenge von $Fun \cup Pre$.

x, y	$\in$	$\mathit{Var} = \mathbb{N}$	*Variable*
f, g	$\in$	$\mathit{Fun} = \{0\} \times \mathbb{N} \times \mathbb{N}$	*Funktionssymbol*
p, q	$\in$	$\mathit{Pre} = \{1\} \times \mathbb{N} \times \mathbb{N}$	*Prädikatensymbol*

$$|\cdot| \in \mathit{Fun} \cup \mathit{Pre} \to \mathbb{N} \qquad \textit{Stelligkeit}$$
$$|(k, m, n)| = n$$

$$s, t \in \mathit{Ter} = x \mid f(t_1, \ldots, t_{|f|}) \qquad \textit{Term}$$

$A, B \in \mathit{For} =$ *Formel*

$p(t_1, \ldots, t_{\|p\|})$	*Atom*
$\mid \neg A$	*Negation*
$\mid A_1 \wedge A_2$	*Konjunktion*
$\mid \exists x A$	*existenzielle Quantifizierung*

Abbildung 10.1: Prädikatenlogische Syntax

Ein Σ-*Term* ist ein Term, in dem neben Variablen nur Funktions- und Prädikatensymbole aus Σ vorkommen. Eine Σ-*Formel* ist eine Formel, in der neben Variablen nur Funktions- und Prädikatensymbole aus Σ vorkommen. Die Mengen aller Σ-Terme und Σ-Formeln bezeichnen wir mit Ter_Σ und For_Σ. Wenn M eine Menge syntaktischer Objekte ist, bezeichen wir mit M_Σ die größte Teilmenge von M, deren Objekte neben Variablen nur Funktions- und Prädikatensymbole aus Σ enthalten.

Die Syntax einer konkreten prädikatenlogischen Sprache wird durch eine Signatur festgelegt. Beispielsweise enthält eine Signatur für ASSN die folgenden Symbole:

1. Für jede ganze Zahl eine Konstante (also ein nullstelliges Funktionssymbol).
2. Für $+$ und $*$ je ein zweistelliges Funktionssymbol.
3. Für $\leq$ ein zweistelliges Prädikatensymbol.

Welche Symbole dabei genau gewählt werden spielt keine Rolle.

Beachten Sie, dass Abbildung 10.1 die abstrakte Syntax von prädikatenlogischen Sprachen definiert. Für die konkrete Syntax verwendet man je nach Anwendung

$$\begin{array}{rcl} A_1 \vee A_2 & \mapsto & \neg(\neg A_1 \wedge \neg A_2) \\ A_1 \Rightarrow A_2 & \mapsto & \neg A_1 \vee A_2 \\ A_1 \Leftrightarrow A_2 & \mapsto & (A_1 \Rightarrow A_2) \wedge (A_2 \Rightarrow A_1) \\ \forall x\, A_1 & \mapsto & \neg\exists x \neg A \end{array}$$

Abbildung 10.2: Abkürzungen für prädikatenlogische Sprachen

unterschiedliche Notationen. Eine Möglichkeit haben wir am Beispiel von ASSN gesehen.

Wenn s ein nullstelliges Funktions- oder Prädikatensymbol ist, werden wir im Folgenden stets s statt $s()$ schreiben. Für nullstellige Funktionssymbole verwenden wir in der Regel die Metavariablen a und b. Ausserdem verwenden wir die Abkürzungen in Abbildung 10.2. Damit wir nicht so viele Klammern schreiben müssen, priorisieren wir die logischen Operatoren wie folgt: $\wedge$, $\vee$, $\Rightarrow$, $\Leftrightarrow$. Damit können wir beispielsweise

$$(((\neg A \wedge B) \vee A) \Rightarrow (A \wedge B))) \Leftrightarrow (A \vee B)$$

ohne Klammern schreiben:

$$\neg A \wedge B \vee A \Rightarrow A \wedge B \Leftrightarrow A \vee B$$

Die *freien Variablen* $FV(A)$ einer Formel A sind analog zu ASSN definiert. Eine Formel heißt *geschlossen* genau dann, wenn sie keine freien Variablen hat.

Sei $A \in For$ eine Formel mit $FV(A) = \{x_1, \dots, x_n\}$ und $x_1 < \dots < x_n$.[1] Wir definieren:

$$\forall A \overset{\text{def}}{=} \forall x_1 \dots \forall x_n A \qquad \textit{Allabschluss von } A$$

$$\exists A \overset{\text{def}}{=} \exists x_1 \dots \exists x_n A \qquad \textit{Existenzabschluss von } A$$

10.2 Semantik

Die Semantik einer konkreten prädikatenlogische Sprache mit der Signatur Σ gibt man durch eine nichtleere Menge U (das sogenannte *Universum*) und durch Interpretationen für die Funktions- und Prädikatensymbole in Σ vor. Dabei muss ein n-stelliges Funktionssymbol durch eine Funktion in $U^n \to U$ und ein n-stelliges

[1] Die Variablen $x_1, \dots, x_n$ sind natürliche Zahlen.

Prädikatensymbol durch eine Funktion in $U^n \to \mathbb{B}$ interpretiert werden. Die Menge U zusammen mit den Interpretationen für die Symbole in Σ bezeichnet man als eine *Struktur*. Der Ordnung halber wollen wir Strukturen noch explizit definieren.

Eine *Struktur* ist ist ein Paar $\langle U, I \rangle$ wie folgt:

- U ist eine nichtleere Menge.
- I ist eine Funktion.
- $\mathit{Dom}\ I$ ist eine Signatur.
- Für alle $f \in \mathit{Dom}\ I \cap \mathit{Fun}$ ist $I(f) \in U^{|f|} \to U$.
- Für alle $p \in \mathit{Dom}\ I \cap \mathit{Pre}$ ist $I(p) \in U^{|p|} \to \mathbb{B}$.

Gegeben eine Struktur $\mathcal{A}$, verwendet wir die folgenden Notationen:

- $U_{\mathcal{A}}$ bezeichnet das Universum von $\mathcal{A}$.
- $\Sigma_{\mathcal{A}}$ bezeichnet die Signatur von $\mathcal{A}$ (also die Menge aller Funktions- und Prädikatensymbole, für die $\mathcal{A}$ Interpretationen vorgibt).
- $s_{\mathcal{A}}$ bezeichnet die Interpretation, die $\mathcal{A}$ dem Funktions- oder Prädikatensymbol $s \in \Sigma_{\mathcal{A}}$ zuordnet.
- $\mathit{Val}_{\mathcal{A}}$ bezeichnet die Menge $\mathit{Var} \to U_{\mathcal{A}}$. Die Elemente dieser Menge bezeichnet man als *Valuationen* oder *Belegungen*.

Eine Σ-*Struktur* ist ein Struktur $\mathcal{A}$ mit $\Sigma_{\mathcal{A}} = \Sigma$.

Jede Struktur legt also genau eine prädikatenlogische Sprache fest.

Abbildung 10.3 definiert zu einer Struktur die Denotationsfunktionen für Terme und Formeln. Statt $\mathcal{A}^{\mathrm{T}}[\![t]\!]$ und $\mathcal{A}^{\mathrm{F}}[\![A]\!]$ schreiben wir kürzer $\mathcal{A}[\![t]\!]$ und $\mathcal{A}[\![A]\!]$.

Eine Struktur *passt* zu einem Term oder einer Formel genau dann, wenn sie alle Funktions- und Prädikatensymbole des Terms oder der Formel interpretiert.

Substitution

Am Beispiel von ASSN haben wir gezeigt, wie man eine Substitutionsfunktion für prädikatenlogische Formeln definiert (Abschnitt 7.5). Das war nicht ganz einfach. Glücklicherweise lässt sich die Konstruktion leicht auf beliebige prädikatenlogische Sprachen übertragen. Also gehen wir im Folgenden davon aus, dass wir eine Substitutionsfunktion

$$\mathit{For} \times \mathit{Ter} \times \mathit{Var} \to \mathit{For}$$

$$\mathcal{A}^{\mathrm{T}} \in Ter_{\Sigma_{\mathcal{A}}} \to Val_{\mathcal{A}} \to U_{\mathcal{A}}$$

$$\begin{aligned} \mathcal{A}^{\mathrm{T}}[\![x]\!]\sigma &= \sigma x \\ \mathcal{A}^{\mathrm{T}}[\![f(t_1, \ldots, t_n)]\!]\sigma &= f_{\mathcal{A}}(\mathcal{A}^{\mathrm{T}}[\![t_1]\!]\sigma, \ldots, \mathcal{A}^{\mathrm{T}}[\![t_n]\!]\sigma) \end{aligned}$$

$$\mathcal{A}^{\mathrm{F}} \in For_{\Sigma_{\mathcal{A}}} \to Val_{\mathcal{A}} \to \mathbb{B}$$

$$\begin{aligned} \mathcal{A}^{\mathrm{F}}[\![p(t_1, \ldots, t_n)]\!]\sigma &= p_{\mathcal{A}}(\mathcal{A}^{\mathrm{T}}[\![t_1]\!]\sigma, \ldots, \mathcal{A}^{\mathrm{T}}[\![t_n]\!]\sigma) \\ \mathcal{A}^{\mathrm{F}}[\![\neg A]\!]\sigma &= 1 - \mathcal{A}^{\mathrm{F}}[\![A]\!]\sigma \\ \mathcal{A}^{\mathrm{F}}[\![A_1 \wedge A_2]\!]\sigma &= \min\{\mathcal{A}^{\mathrm{F}}[\![A_1]\!]\sigma,\ \mathcal{A}^{\mathrm{F}}[\![A_2]\!]\sigma\} \\ \mathcal{A}^{\mathrm{F}}[\![\exists x A]\!]\sigma &= \max\{\, \mathcal{A}^{\mathrm{F}}[\![A]\!](\sigma[u/x]) \mid u \in U_{\mathcal{A}} \,\} \end{aligned}$$

Abbildung 10.3: Die Denotationsfunktionen einer Struktur $\mathcal{A}$

mit den folgen Eigenschaften haben:

1. $\forall A \in For_\Sigma\ \forall t \in Ter_\Sigma\ \forall x \in Var\colon\ A[t/x] \in For_\Sigma$.
2. Für jede Σ-Struktur $\mathcal{A}$:

$$\begin{aligned} &\forall A \in For_\Sigma\ \forall t \in Ter_\Sigma\ \forall x \in Var\ \forall \sigma \in Val_{\mathcal{A}}: \\ &\qquad \mathcal{A}[\![A[t/s]]\!]\sigma = \mathcal{A}[\![A]\!](\sigma[\mathcal{A}[\![t]\!]\sigma/x]) \end{aligned}$$

Äquivalenz

Zwei Formel A und B heißen *äquivalent* (wir schreiben $A \models\!\!\!\!\dashv B$) genau dann, wenn $\mathcal{A}[\![A]\!] = \mathcal{A}[\![B]\!]$ für jede Struktur $\mathcal{A}$ gilt, die zu A und B passt. Selbstverständlich ist Äquivalenz von Formeln eine Äquivalenzrelation, die die folgende Ersetzungseigenschaft hat:

Proposition 10.2.1 (Ersetzung) *Sei $A \models\!\!\!\!\dashv A'$ und sei B' aus B durch das Ersetzen eines Auftretens der Teilformel A durch A' erhaltbar. Dann $B' \models\!\!\!\!\dashv B$.*

Abbildung 10.4 zeigt einige hilfreiche prädikatenlogischen Äquivalenzen.

Pränex- und Skolem-Form

Eine Formel ist in *Pränexform* genau dann, wenn sie die Form

$$Q_1 x_1 \ldots Q_n x_n\ A$$

$$
\begin{array}{rcll}
\neg\exists x\,A & \models\!\mid & \forall x\neg A & \\
\neg\forall x\,A & \models\!\mid & \exists x\neg A & \\[1ex]
\neg\exists A & \models\!\mid & \forall\neg A & \\
\neg\forall A & \models\!\mid & \exists\neg A & \\[1ex]
\exists x\,A & \models\!\mid & \neg\forall x\neg A & \\
\forall x\,A & \models\!\mid & \neg\exists x\neg A & \\[1ex]
\exists x\,A & \models\!\mid & A & \text{falls } x \notin FV(A) \\
\forall x\,A & \models\!\mid & A & \text{falls } x \notin FV(A) \\[1ex]
\exists x\,A & \models\!\mid & \exists y(A[y/x]) & \text{falls } y \notin FV(A) \\
\forall x\,A & \models\!\mid & \forall y(A[y/x]) & \text{falls } y \notin FV(A) \\[1ex]
(\exists x\,A) \wedge B & \models\!\mid & \exists x(A \wedge B) & \text{falls } x \notin FV(B) \\
(\forall x\,A) \vee B & \models\!\mid & \forall x(A \vee B) & \text{falls } x \notin FV(B) \\[1ex]
(\exists x\,A) \vee B & \models\!\mid & \exists x(A \vee B) & \text{falls } x \notin FV(B) \\
(\forall x\,A) \wedge B & \models\!\mid & \forall x(A \wedge B) & \text{falls } x \notin FV(B) \\[1ex]
(\exists x\,A) \vee (\exists x\,B) & \models\!\mid & \exists x(A \vee B) & \\
(\forall x\,A) \wedge (\forall x\,B) & \models\!\mid & \forall x(A \wedge B) & \\[1ex]
\forall x\forall y\,A & \models\!\mid & \forall y\forall x\,A & \\
\exists x\exists y\,A & \models\!\mid & \exists y\exists x\,A &
\end{array}
$$

Abbildung 10.4: Einige prädikatenlogische Äquivalenzen hat, wobei $n \geq 0$, jedes Q_i für $\exists$ oder $\forall$ steht, und A eine quantorenfreie Formel ist. Eine Formel ist in *Skolem-Form* genau dann, wenn sie die Form

$$\forall x_1 \dots \forall x_n \; A$$

hat, wobei $n \geq 0$ und A eine quantorenfreie Formel ist.

Proposition 10.2.2 *Zu jeder Formel in* For_Σ *existiert eine äquivalente Formel in* For_Σ *in Pränexform.*

Beweis Folgt mit den Äquivalenzen in Abbildung 10.4. □

Modelle, Allgemeingültigkeit und Erfüllbarkeit

Eine Struktur $\mathcal{A}$ heißt *Modell* einer Formel A genau dann, wenn $\mathcal{A}$ zu A passt und $\mathcal{A}[\![A]\!]\sigma = 1$ für jede Belegung $\sigma \in Val_{\mathcal{A}}$ gilt. Eine Formel A ist in einer Struktur $\mathcal{A}$ *gültig* genau dann, wenn $\mathcal{A}$ ein Modell von A ist.

Eine Formel A heißt *allgemeingültig* genau dann, wenn jede zu A passende Struktur ein Modell von A ist.

Proposition 10.2.3 $\forall A, B \in For\colon \;\; A \models\!\!\!\mid B \;\;\Longleftrightarrow\;\; (A \Leftrightarrow B)$ *allgemeingültig.*

Eine Formel A heißt *erfüllbar* genau dann, wenn es eine zu A passende Struktur $\mathcal{A}$ und eine Belegung $\sigma \in Val_{\mathcal{A}}$ gibt, so dass $\mathcal{A}[\![A]\!]\sigma = 1$. Eine Formel heißt *unerfüllbar* genau dann, wenn sie nicht erfüllbar ist.

Eine Struktur $\mathcal{A}$ heißt *Modell einer Formelmenge* M genau dann, wenn $\mathcal{A}$ ein Modell jeder Formel in M ist. Eine Formelmenge heißt *erfüllbar* genau dann, wenn sie ein Modell hat. Eine Formelmenge heißt *unerfüllbar* genau dann, wenn sie nicht erfüllbar ist.

Proposition 10.2.4 *Für jede Formel* A *sind die folgenden Aussagen äquivalent:*

1. A *allgemeingültig.*
2. $\forall A$ *allgemeingültig.*
3. $\neg A$ *unerfüllbar.*
4. $\exists \neg A$ *unerfüllbar.*
5. $\{\exists \neg A\}$ *unerfüllbar.*

Wir schreiben $M \models_\Sigma A$ genau dann, wenn $M \cup \{A\} \subseteq For_\Sigma$ und jedes Σ-Modell von M ein Modell von A ist. Wenn $M \models_\Sigma A$ gilt, bezeichnet man A als eine *logische Konsequenz* von M.

Proposition 10.2.5 *Sei* $M \cup \{A\} \subseteq For_\Sigma$. *Dann:*

$$M \models_\Sigma A \iff M \cup \{\exists\neg A\} \text{ unerfüllbar}$$

In diesem Kapitel werden wir zeigen, dass die Menge der allgemeingültigen Formeln testbar und unentscheidbar ist (nach Gödelisierung). Die Testbarkeit von Allgemeingültigkeit ist keineswegs offensichtlich, da Allgemeingültigkeit mit einer Quantifizierung über alle Strukturen definiert ist.

10.3 Variablenfreie Formeln

Eine Term oder eine Formel heißen *variablenfrei* genau dann, wenn sie keine Variablen enthalten. Variablenfreie Terme und Formeln nennt man auch *Grundterme* und *Grundformeln*. Wir definieren:

$$GTer \stackrel{\text{def}}{=} \{\, t \in Ter \mid t \text{ variablenfrei} \,\}$$

$$GFor \stackrel{\text{def}}{=} \{\, A \in For \mid A \text{ variablenfrei} \,\}$$

$$AGFor \stackrel{\text{def}}{=} \{\, A \in GFor \mid A \text{ atomar} \,\}$$

Wir werden zeigen, dass variablenfreie prädikatenlogische Formeln auch aussagenlogische Formeln sind, und dass für solche Formeln prädikatenlogische Erfüllbarkeit mit aussagenlogischer Erfüllbarkeit identisch ist.

Wir haben die aussagenlogische Sprache AL so definiert, dass wir die Menge der Variablen frei wählen können. Für den Kompaktheitssatz haben wir einschränkend verlangt, dass die Menge der Variablen abzählbar ist. Damit haben wir die Möglichkeit, die Menge *AGFor* der variablenfreien atomaren prädikatenlogischen Formeln als die Menge der aussagenlogische Variablen zu wählen. Das werden wir ab sofort tun. Wenn wir jetzt die Definition der prädikatenlogischen Syntax (Abbildung 10.1) mit der Definition der aussagenlogischen Syntax (Abbildung 4.4) vergleichen, stellen wir fest, dass die Menge der aussagenlogischen Formeln genau die Menge Menge *GFor* der variablenfreien prädikatenlogischen Formeln ist. Wir werden zeigen, dass eine aussagenlogische Formel genau dann gültig [erfüllbar] ist, wenn sie als prädikatenlogische Formel gültig [erfüllbar] ist.

Sei $\mathcal{A}$ eine Σ-Struktur, $\sigma \in Val_{\mathcal{A}}$, $t \in GTer_\Sigma$ und $A \in GFor_\Sigma$. Da $\mathcal{A}[\![t]\!]\sigma$ und $\mathcal{A}[\![A]\!]\sigma$ nicht von σ abhängt, definieren wir:

$$\mathcal{A}^{\mathrm{G}}[\![t]\!] \stackrel{\text{def}}{=} \mathcal{A}[\![t]\!]\sigma$$

$$\mathcal{A}^{\mathrm{G}}[\![A]\!] \stackrel{\text{def}}{=} \mathcal{A}[\![A]\!]\sigma$$

Sei $\mathcal{A}$ eine Σ-Struktur. Dann ist

$$\sigma_{\mathcal{A}} \stackrel{\text{def}}{=} \lambda\, A \in \mathit{AGFor}\,\textbf{.}\ \text{if } A \in \mathit{AGFor}_\Sigma \text{ then } \mathcal{A}^{\mathrm{G}}[\![A]\!] \text{ else } 0$$

eine aussagenlogische Belegung.

Lemma 10.3.1 *Für jede Σ-Struktur $\mathcal{A}$ gilt: $\forall A \in \mathit{GFor}_\Sigma\colon\ \mathcal{F}[\![A]\!]\sigma_{\mathcal{A}} = \mathcal{A}^{G}[\![A]\!]$.*

Beweis Beweis durch strukturelle Induktion über $A \in \mathit{GFor}_\Sigma$. □

Eine *Termstruktur* ist eine Struktur $\mathcal{A}$ wie folgt:

- $U_{\mathcal{A}} = \mathit{GTer}_{\Sigma_{\mathcal{A}}}$.
- $\forall t \in \mathit{GTer}_{\Sigma_{\mathcal{A}}}\colon\ \mathcal{A}^{\mathrm{G}}[\![t]\!] = t$.

Termstrukturen nennt man auch *Herbrand-Strukturen*. Ein *Termmodell* für eine Formel A ist eine Termstruktur, die ein Modell von A ist.

Lemma 10.3.2 *Sei Σ eine Signatur mit mindestens einem Konstantensymbol. Dann existiert für jede aussagenlogische Belegung $\sigma \in \mathit{AGFor} \to \mathbb{B}$ eine Termstruktur $\mathcal{A}$ mit $\Sigma_{\mathcal{A}} = \Sigma$, so dass σ und $\sigma_{\mathcal{A}}$ auf AGFor_Σ übereinstimmen.*

Beweis Sei $\sigma \in \mathit{AGFor} \to \mathbb{B}$. Sei $\mathcal{A}$ die Termstruktur mit $\Sigma_{\mathcal{A}} = \Sigma$ und

$$\forall p \in \Sigma \cap \mathit{Pre}\colon\ p_{\mathcal{A}} = \lambda\,(t_1, \ldots, t_{|p|}) \in \mathit{GTer}_\Sigma^{|p|}\,\textbf{.}\ \sigma(p(t_1, \ldots, t_{|p|}))$$

Dann stimmen σ und $\sigma_{\mathcal{A}}$ auf $\mathit{AGFor}_{\Sigma_{\mathcal{A}}}$ überein. □

Satz 10.3.3 (Transfer) *Sei $A \in \mathit{GFor}$. Dann:*

1. *A ist erfüllbar genau dann, wenn A aussagenlogisch erfüllbar ist.*
2. *A ist allgemeingültig genau dann, wenn A aussagenlogisch gültig ist.*

Beweis Sei A erfüllbar. Dann existiert eine zu A passende Struktur $\mathcal{A}$ mit $\mathcal{A}^{\mathrm{G}}[\![A]\!] = 1$. Also $\mathcal{F}[\![A]\!]\sigma_{\mathcal{A}} = 1$ mit Lemma 10.3.1. Also ist A aussagenlogisch erfüllbar.

Sei A aussagenlogisch erfüllbar. Dann existiert ein $\sigma \in \mathit{AGFor} \to \mathbb{B}$ mit $\mathcal{F}[\![A]\!]\sigma = 1$. Wir wählen eine Signatur Σ mit mindestens einem Konstantensymbol, so dass $A \in \mathit{GFor}_\Sigma$. Nach Lemma 10.3.2 existiert eine Termstruktur $\mathcal{A}$ mit $\Sigma_{\mathcal{A}} = \Sigma$, so dass σ und $\sigma_{\mathcal{A}}$ auf AGFor_Σ übereinstimmen. Also $\mathcal{F}[\![A]\!]\sigma_{\mathcal{A}} = 1$. Also $\mathcal{A}^{\mathrm{G}}[\![A]\!] = 1$ mit Lemma 10.3.1. Also ist A erfüllbar.

Die zweite Behauptung folgt aus der ersten:

$$\begin{aligned} A \text{ allgemeingültig} &\iff \neg A \text{ unerfüllbar} \\ &\iff \neg A \text{ aussagenlogisch unerfüllbar} \\ &\iff A \text{ aussagenlogisch gültig} \end{aligned}$$ □

Korollar 10.3.4 *Die Allgemeingültigkeit und Erfüllbarkeit von variablenfreien prädikatenlogischen Formeln ist entscheidbar.*

10.4 Aussagenlogisch gültige Formeln

Auch auf Formeln mit Variablen lassen sich aussagenlogische Ergebnisse übertragen. Sei A eine allgemeingültige variablenfreie Formel. Sei A' aus A dadurch erhalten, dass man atomare variablenfreie Teilformeln konsistent durch irgendwelche Formeln ersetzt. Dann ist A' allgemeingültig. Aus dieser Behauptung folgt beispielsweise, dass die Formel

$$(A_1 \wedge A_2) \vee B \Leftrightarrow (A_1 \vee B) \wedge (A_2 \vee B)$$

für beliebige Formeln A_1, A_2 und B allgemeingültig ist.

Um diese Behauptung beweisen zu können, müssen wir präzise sagen, was wir unter konsistenter Ersetzung meinen. Jede Funktion

$$\gamma \in \mathit{AGFor} \to \mathit{For}$$

lässt sich wie folgt auf *GFor* erweitern:

$$\gamma^{\mathrm{F}} \in \mathit{GFor} \to \mathit{For}$$

$$\begin{aligned}
\gamma^{\mathrm{F}}(p(t_1, \ldots, t_n)) &= \gamma(p(t_1, \ldots, t_n)) \\
\gamma^{\mathrm{F}}(\neg A) &= \neg(\gamma^{\mathrm{F}}(A)) \\
\gamma^{\mathrm{F}}(A_1 \wedge A_2) &= \gamma^{\mathrm{F}}(A_1) \wedge \gamma^{\mathrm{F}}(A_2)
\end{aligned}$$

Lemma 10.4.1 (Substitution) *Sei $\gamma \in \mathit{AGFor} \to \mathit{For}$ und $\mathcal{A}$ eine Struktur, die zu jeder Formel passt. Dann:*

$$\begin{aligned}
&\forall A \in \mathit{GFor}\ \forall \sigma \in \mathit{Val}_{\mathcal{A}}: \\
&\qquad \mathcal{A}[\![\gamma^F(A)]\!]\sigma = \mathcal{F}[\![A]\!](\lambda B \in \mathit{AGFor}.\ \mathcal{A}[\![\gamma(B)]\!]\sigma)
\end{aligned}$$

Beweis Durch strukturelle Induktion über $A \in \mathit{GFor}$. □

Eine Struktur $\mathcal{A}'$ heißt *konservative Erweiterung* einer Struktur $\mathcal{A}$ genau dann, wenn $U_{\mathcal{A}} = U_{\mathcal{A}'}$, $\Sigma_{\mathcal{A}} \subseteq \Sigma_{\mathcal{A}'}$ und $\mathcal{A}$ und $\mathcal{A}'$ alle Symbole in $\Sigma_{\mathcal{A}}$ gleich interpretieren.

Satz 10.4.2 (Transfer) *Sei $A \in \mathit{GFor}$ allgemeingültig und $\gamma \in \mathit{AGFor} \to \mathit{For}$. Dann ist $\gamma^F(A)$ allgemeingültig.*

Beweis Sei $\mathcal{A}$ eine Struktur, die zu $\gamma^{\mathrm{F}}(A)$ passt. Sei $\mathcal{A}'$ eine konservative Erweiterung von $\mathcal{A}$, die zu allen Formeln passt. Dann folgt mit Lemma 10.4.1, dem ersten Transfersatz 10.3.3 und der Allgemeingültigkeit von A, dass $\mathcal{A}'$ ein Modell von $\gamma^{\mathrm{F}}(A)$ ist. Also ist $\mathcal{A}$ ein Modell von $\gamma^{\mathrm{F}}(A)$. □

Damit gelten wegen Proposition 10.2.3 alle aussagenlogischen Äquivalenzen auch für prädikatenlogische Sprachen. Insbesondere gilt dies für die Äquivalenzen in Abbildung 4.2.

10.5 Berechnung von Skolemformen

Wir zeigen jetzt, dass man zu jeder Formel A eine Formel B in Skolemform berechnen kann, so dass A genau dann erfüllbar ist, wenn B erfüllbar ist. Dafür benötigen wir eine Technik zur Elimination von Existenzquantoren. Das in diesem Abschnitt entwickelte Verfahren zur Elimination von Existenzquantoren ist der erste Teil eines Verfahrens zum Testen von Allgemeingültigkeit.

Zwei Strukturen $\mathcal{A}$ und $\mathcal{B}$ *unterscheiden sich nur auf* Σ genau dann, wenn $U_{\mathcal{A}} = U_{\mathcal{B}}$, $\Sigma_{\mathcal{A}} - \Sigma = \Sigma_{\mathcal{B}} - \Sigma$, und wenn $\mathcal{A}$ und $\mathcal{B}$ alle Symbole in $\Sigma_{\mathcal{A}} - \Sigma$ gleich interpretieren.

Lemma 10.5.1 (Skolem) *Sei A eine Formel mit $FV(\exists x\, A) = \{x_1, \dots, x_n\}$ und sei f ein n-stelliges Funktionssymbol, das nicht in A vorkommt. Dann:*

1. *Für jede Struktur, die zu $A[f(x_1, \dots, x_n)/x]$ passt, gilt:*

$$\forall \sigma \in Val_{\mathcal{A}}\colon\ \mathcal{A}[\![A[f(x_1, \dots, x_n)/x]]\!]\sigma = 1 \;\Rightarrow\; \mathcal{A}[\![\exists x\, A]\!]\sigma = 1$$

2. *Für jede Struktur $\mathcal{A}$, die zu $\exists x\, A$ passt, existiert eine Struktur $\mathcal{B}$, die zu $A[f(x_1, \dots, x_n)/x]$ passt, wie folgt:*

   a) *$\mathcal{A}$ und $\mathcal{B}$ unterscheiden sich nur auf $\{f\}$.*

   b) $\forall \sigma \in Val_{\mathcal{A}}\colon\ \mathcal{A}[\![\exists x\, A]\!]\sigma = 1 \;\Rightarrow\; \mathcal{B}[\![A[f(x_1, \dots, x_n)/x]]\!]\sigma = 1.$

Beweis (1) Sei $\mathcal{A}$ eine Struktur, die zu $A[f(x_1, \dots, x_n)/x]$ passt. Sei $\sigma \in Val_{\mathcal{A}}$ mit $\mathcal{A}[\![A[f(x_1, \dots, x_n)/x]]\!]\sigma = 1$. Dann:

$$\begin{aligned}
\mathcal{A}[\![\exists x\, A]\!]\sigma &= \max\{\, \mathcal{A}[\![A]\!](\sigma[u/x]) \mid u \in U_{\mathcal{A}} \,\} \\
&\geq \mathcal{A}[\![A]\!](\sigma[\mathcal{A}[\![f(x_1, \dots, x_n)]\!]\sigma/x]) \\
&= \mathcal{A}[\![A[f(x_1, \dots, x_n)/x]]\!]\sigma \\
&= 1
\end{aligned}$$

(2) Sei $\mathcal{A}$ eine Struktur, die zu $\exists x\, A$ passt. Dann existiert für jedes $\sigma \in Val_{\mathcal{A}}$ mit $\mathcal{A}[\![\exists x\, A]\!]\sigma = 1$ ein $u \in U_{\mathcal{A}}$ mit $\mathcal{A}[\![A]\!](\sigma[u/x]) = 1$. Dabei hängt u nur von $\sigma(x_1), \ldots, \sigma(x_n)$ ab. Also gibt es eine Funktion $\phi \in U_{\mathcal{A}}^n \to U_{\mathcal{A}}$ (Auswahlaxiom) mit:

$$\forall \sigma \in Val_{\mathcal{A}}:\ \mathcal{A}[\![\exists x\, A]\!]\sigma = 1 \ \Rightarrow\ \mathcal{A}[\![A]\!](\sigma[\phi(\sigma(x_1), \ldots, \sigma(x_n))/x]) = 1$$

Sei $\mathcal{B}$ die Struktur mit $f_{\mathcal{B}} = \phi$, die sich nur auf $\{f\}$ von $\mathcal{A}$ unterscheidet. Sei $\sigma \in Val_{\mathcal{A}}$ mit $\mathcal{A}[\![\exists x\, A]\!]\sigma = 1$. Da f in A nicht vorkommt, gilt:

$$\begin{aligned}\mathcal{B}[\![A[f(x_1, \ldots, x_n)/x]]\!]\sigma &= \mathcal{B}[\![A]\!](\sigma[\mathcal{B}[\![f(x_1, \ldots, x_n)]\!]\sigma/x]) \\ &= \mathcal{A}[\![A]\!](\sigma[\phi(\sigma(x_1), \ldots, \sigma(x_n))/x]) \\ &= 1\end{aligned}$$

□

Die für die Elimination des Existenzquantors eingeführte Funktion nennt man eine *Skolem-Funktion*.

Satz 10.5.2 (Skolem) *Zu jeder Formel A kann man eine Formel B in Skolemform berechnen, so dass gilt:*

1. *A ist genau dann erfüllbar, wenn B erfüllbar ist.*
2. *Jedes Modell von B ist ein Modell von A.*
3. *Zu jedem Modell $\mathcal{A}$ von A existiert ein Modell $\mathcal{B}$ von B, so dass $\mathcal{A}$ und $\mathcal{B}$ sich nur auf Funktionssymbolen unterscheiden, die in A nicht vorkommen.*

Beweis Zuerst bringt man A in Pränexform. Dann eliminiert man den äußersten Existenzquantor von A gemäß Lemma 10.5.1. Dabei sind eventuelle Allquantoren über dem Existenzquantor kein Problem, da Modelle freie Variablen implizit allquantifizieren. Dies wiederholt man solange, bis alle Existenzquantoren eliminiert sind. □

10.6 Quantorenfreie Klauseldarstellung

Wir geben jetzt ein detailliertes Verfahren zur Berechnung von Skolem-Formen an, dass mit einer prädikatenlogischen Klauseldarstellung arbeitet. Insgesamt bekommen wir ein Verfahren, mit dem Klauselmengen in quantorenfreie Klauselmengen überführt werden können. Dabei ist die Ausgangsmenge genau dann erfüllbar, wenn die quantorenfreie Ergebnismenge erfüllbar ist.

Eine *Klausel* ist eine endliche Menge von Formeln. Wir vereinbaren die folgende Notation:

$$C, D \in Cla \stackrel{\text{def}}{=} \mathcal{P}_{fin}(For) \qquad \textit{Klausel}$$

$$S \in \mathcal{P}_{fin}(For_\Sigma) \qquad \textit{Klauselmenge}$$

$$FV(C) \stackrel{\text{def}}{=} \bigcup_{A \in C} FV(A) \qquad \textit{Freie Variablen}$$

Wir werden mit der konjunktiven Interpretation von Klauselmengen arbeiten. Zusätzlich definieren wir die Semantik von Klauselmengen so, dass eine Klausel die Variablen, die in ihren Formeln frei vorkommen, allquantifiziert. Eine Klausel

$$\{A_1, \dots, A_n\}$$

wird also wie die Formel

$$\forall x_1 \ \dots \ \forall x_m \ (A_1 \vee \cdots \vee A_n)$$

interpretiert, wobei $\{x_1, \dots, x_m\} = FV(A_1) \cup \cdots \cup FV(A_n)$.

Eine Struktur $\mathcal{A}$ heißt *Modell* einer Klausel C genau dann, wenn $\mathcal{A}$ zu jeder Formel in C passt und

$$\forall \sigma \in Val_{\mathcal{A}} \ \exists A \in C\colon \ \mathcal{A}[\![A]\!]\sigma = 1$$

gilt. Eine Struktur $\mathcal{A}$ heißt *Modell* einer Klauselmenge S genau dann, wenn $\mathcal{A}$ ein Model jeder Klausel in S ist.

Eine Klauselmenge heißt *erfüllbar* genau dann, wenn sie mindestens ein Modell hat. Eine Klauselmenge heißt *unerfüllbar* genau dann, wenn sie kein Modell hat.

Proposition 10.6.1 *Zu jeder endlichen Klauselmenge $S \subseteq Cla_\Sigma$ existiert eine geschlossene Formel $A \in For_\Sigma$, die dieselben Modelle wie S hat.*

Proposition 10.6.2 *Sei M eine Formelmenge. Dann haben M und die Klauselmenge $\{\, \{A\} \mid A \in M \,\}$ dieselben Modelle.*

Proposition 10.6.3 *Jede Klauselmenge, die die leere Klausel enthält, ist unerfüllbar.*

Eine Klausel C heißt *trivial* genau dann, wenn es eine Formel A gibt, so dass $A \in C$ und $\neg A \in C$.

$$
\begin{array}{lrcl}
(1) & S,\ (C,\ \neg\neg A) & \xrightarrow{s} & S,\ (C,\ A) \\
(2) & S,\ (C,\ \neg(A_1 \wedge A_2)) & \xrightarrow{s} & S,\ (C,\ \neg A_1,\ \neg A_2) \\
(3) & S,\ (C,\ A_1 \wedge A_2) & \xrightarrow{s} & S,\ (C,\ A_1),\ (C,\ A_2) \\
(4) & S,\ (C,\ \neg\exists x\, A) & \xrightarrow{s} & S,\ (C,\ \neg A[y/x]) \\
 & & & \text{falls } y \notin FV(C) \ \wedge\ (x = y \vee y \notin FV(A)) \\
(5) & S,\ (C,\ \exists x\, A) & \xrightarrow{s} & S,\ (C,\ A[f(x_1, \dots, x_n)/x]) \\
 & & & \text{falls } FV(\exists x\, A) = \{x_1, \dots, x_n\} \text{ und } f \text{ ein } n\text{-stelliges} \\
 & & & \text{Funktionssymbol ist, das in } S, C \text{ und } A \text{ nicht vorkommt}
\end{array}
$$

Abbildung 10.5: Vereinfachungsregeln für prädikatenlogische Klauselmengen

Proposition 10.6.4 (Triviale Klauseln) *Sei S eine Klauselmenge und C eine triviale Klausel. Dann haben $S \cup \{C\}$ und S dieselben Modelle.*

Ein *Literal* ist entweder eine atomare Formel $p(t_1, \dots, t_n)$ oder eine negierte atomare Formel $\neg p(t_1, \dots, t_n)$.

Eine Klausel heißt *literal* genau dann, wenn sie nur Literale enthält. Eine Klauselmenge heißt *literal* genau dann, wenn sie nur literale Klauseln enthält.

Abbildung 10.5 zeigt die Vereinfachungsregeln für prädikatenlogische Klauselmengen. Die ersten drei Regeln entsprechen den konjunktiven Vereinfachungsregeln für aussagenlogische Klauselmengen. Wir verzichten diesmal auf die Eliminationsregel für triviale Klauseln, damit bei der Vereinfachung keine Funktions- und Prädikatensymbole verloren gehen. Die vierte Regel vereinfacht interne Allquantifizierung zu externer Allquantifizierung. Die fünfte Regel eliminiert interne Existenzquantoren durch die Einführung von Skolemfunktionen.

Wir schreiben $S \xrightarrow{s} S'$ genau dann, wenn die Klauselmenge S' aus der Klauselmenge S durch die Anwendung einer Vereinfachungsregel erhalten werden kann.

Wir schreiben $S \models\!\!\Rightarrow S'$ genau dann, wenn die folgenden Aussagen gelten:

1. Jedes Modell von S' ist ein Modell von S.
2. Zu jedem Modell $\mathcal{A}$ von S existiert ein Modell $\mathcal{B}$ von S', so dass $\mathcal{A}$ und $\mathcal{B}$ sich nur auf Funktionssymbolen unterscheiden, die in A nicht vorkommen.

Die wesentlichen Eigenschaften der Vereinfachungsregeln werden durch den folgenden Satz formuliert.

Satz 10.6.5 (Vereinfachung) *Sei S eine Klauselmenge. Dann:*

1. *Wenn $S \xrightarrow{s} S'$, dann $S \models\!\!\!\Rightarrow S'$.*
2. *S ist literal genau dann, wenn es kein S' mit $S \xrightarrow{s} S'$ gibt.*
3. *Wenn S endlich ist, dann gibt es keine unendliche Kette*

$$S \xrightarrow{s} S_1 \xrightarrow{s} S_2 \xrightarrow{s} S_3 \xrightarrow{s} \cdots$$

Beweis Behauptung (1) folgt mit dem Skolem-Lemma 10.5.1. Behauptung (2) gilt offensichtlich. Behauptung (3) folgt mit ähnlichen Argumenten wie bei den aussagenlogischen Vereinfachungsregeln. □

Eine Klauselmenge S heißt *sparsam* genau dann, wenn zu jedem $n \in \mathbb{N}$ unendlich viele n-stellige Funktionssymbole existieren, die nicht in S vorkommen.

Lemma 10.6.6 *Zu jeder sparsamen Klauselmenge S existiert eine literale Klauselmenge S' mit $S \models\!\!\!\Rightarrow S'$. Zudem gibt es eine berechenbare Funktion, die zu einem Tester für S einen Tester für S' berechnet.*

Beweis Da die Menge aller Formeln abzählbar ist, hat jede Klauselmenge nur abzählbar viele Klauseln. Damit kann man die Klauseln einer Klauselmenge Schritt für Schritt vereinfachen. Diese eventuell unendliche Berechnung konvergiert gegen eine literale Klauselmenge. □

Eine Formelmenge heißt M *sparsam* genau dann, wenn die M entsprechende Klauselmenge $\{\,\{A\} \mid A \in M\,\}$ sparsam ist. Wir schreiben $M \models\!\!\!\Rightarrow S$ genau dann, wenn $\{\,\{A\} \mid A \in M\,\} \models\!\!\!\Rightarrow S$ gilt.

Satz 10.6.7 (Elimination) *Zu jeder sparsamen Formelmenge M existiert eine literale Klauselmenge S mit $M \models\!\!\!\Rightarrow S$. Zudem gibt es eine berechenbare Funktion, die zu einem Tester für M einen Tester für S berechnet.*

Beweis Folgt aus Proposition 10.6.2 und Lemma 10.6.6. □

Der obige Satz heißt Eliminationssatz, da er die Elimination aller Quantoren der internen Syntax erlaubt.

Proposition 10.6.8 *Mithilfe einer konsistenten Umbenennung*

$$\lambda\,(0, m, n) \in \mathit{Fun}.\ (0, 2m, n)$$

der Funktionssymbole kann man jede Formelmenge M in eine sparsame Formelmenge überführen, die bis auf diese Umbenennung dieselben Modelle wie M hat.

10.7 Der Expansionssatz

Sei S eine quantorenfreie Klauselmenge. Die Menge aller Grundinstanzen der Klauseln in S heißt die Expansion von S und ist eine aussagenlogische Klauselmenge. Wir zeigen, dass eine quantorenfreie Klauselmenge genau dann unerfüllbar ist, wenn ihre Expansion unerfüllbar ist. Mit diesem Ergebnis lassen sich die Kompaktheits- und Testbarkeitseigenschaften der Aussagenlogik auf die Prädikatenlogik übertragen. Insbesondere erhalten wir ein Testverfahren für die Allgemeingültigkeit von Formeln.

Proposition 10.7.1 *Sei $S \subseteq Cla_\Sigma$ eine erfüllbare Klauselmenge. Dann existiert ein Σ-Modell $\mathcal{A}$ für S.*

Beweis Da S erfüllbar ist, existiert ein Modell $\mathcal{A}$ für S. Wenn in $\Sigma_{\mathcal{A}}$ Symbole aus Σ fehlen, erweitern wir $\mathcal{A}$ entsprechend. Die Denotationen der neuen Symbole können wir beliebig wählen, da sie nicht in A vorkommen. Wir haben jetzt ein Modell $\mathcal{A}'$ für S mit $\Sigma \subseteq \Sigma_{\mathcal{A}'}$. Durch Eliminieren der Interpretationen für die überflüssigen Symbole bekommen wir ein Modell $\mathcal{A}''$ für S mit $\Sigma_{\mathcal{A}''} = \Sigma$ (da S nur Symbole in Σ enthält). $\square$

Satz 10.7.2 *Sei $S \subseteq Cla_\Sigma$ eine variablenfreie und nichtleere Klauselmenge. Dann sind die folgenden Aussagen äquivalent:*

1. *S erfüllbar.*
2. *S hat ein Σ-Termmodell $\mathcal{A}$.*
3. *$\mathcal{K}[\![S]\!] \neq \emptyset$.*

Beweis Σ enthält mindestens ein Konstantensymbol, da S variablenfrei und nichtleer ist.

(1) $\Rightarrow$ (3). Sei S erfüllbar. Dann hat S ein Σ-Modell $\mathcal{A}$. (Proposition 10.7.1). Damit gilt

$$\forall C \in S\ \exists A \in C: \ \mathcal{A}^{\mathrm{G}}[\![A]\!] = 1$$

Mit Lemma 10.3.1 folgt

$$\forall C \in S\ \exists A \in C: \ \mathcal{F}[\![A]\!]\sigma_{\mathcal{A}} = 1$$

Also $\sigma_{\mathcal{A}} \in \mathcal{K}[\![S]\!]$.

(3) $\Rightarrow$ (2). Sei $\mathcal{K}[\![S]\!] \neq \emptyset$. Dann gibt es ein $\sigma \in AGFor \to \mathbb{B}$ mit

$$\forall C \in S\ \exists A \in C: \ \mathcal{F}[\![A]\!]\sigma = 1$$

Lemma 10.3.2 liefert uns zu σ eine Σ-Termstruktur $\mathcal{A}$ und der Eigenschaft, dass $\sigma_{\mathcal{A}}$ und σ auf $AGFor_\Sigma$ übereinstimmen. Also gilt

$$\forall C \in S\ \exists A \in C\colon\ \mathcal{F}[\![A]\!]\sigma_{\mathcal{A}} = 1$$

Mit Lemma 10.3.1 folgt

$$\forall C \in S\ \exists A \in C\colon\ \mathcal{A}^{\mathrm{G}}[\![A]\!] = 1$$

Also ist $\mathcal{A}$ ein Termmodell von S.

(2) $\Rightarrow$ (1). Trivial. □

Wir definieren:

$$QFor \stackrel{\text{def}}{=} \{\, A \in For \mid A \text{ quantorenfrei} \,\}$$

Für jede Funktion $\sigma \in Var \to GTer_\Sigma$ definieren wir zwei Funktionen:

$$\sigma^{\mathrm{T}} \in Ter_\Sigma \to GTer_\Sigma$$

$$\begin{aligned} \sigma^{\mathrm{T}}(x) &= \sigma x \\ \sigma^{\mathrm{T}}(f(t_1, \dots, t_n)) &= f(\sigma^{\mathrm{T}}(t_1), \dots, \sigma^{\mathrm{T}}(t_n)) \end{aligned}$$

$$\sigma^{\mathrm{F}} \in QFor_\Sigma \to GFor_\Sigma$$

$$\begin{aligned} \sigma^{\mathrm{F}}(p(t_1, \dots, t_n)) &= p(\sigma^{\mathrm{T}}(t_1), \dots, \sigma^{\mathrm{T}}(t_n)) \\ \sigma^{\mathrm{F}}(\neg A) &= \neg(\sigma^{\mathrm{F}}(A)) \\ \sigma^{\mathrm{F}}(A_1 \wedge A_2) &= \sigma^{\mathrm{F}}(A_1) \wedge \sigma^{\mathrm{F}}(A_2) \end{aligned}$$

Lemma 10.7.3 (Substitution) *Sei $\mathcal{A}$ eine Σ-Struktur und $\sigma \in Var \to GTer_\Sigma$. Dann:*

1. $\forall t \in Ter_\Sigma\colon\ \mathcal{A}^G[\![\sigma^T(t)]\!] = \mathcal{A}[\![t]\!](\lambda x \in Var .\ \mathcal{A}^G[\![\sigma(x)]\!])$
2. $\forall A \in QFor_\Sigma\colon\ \mathcal{A}^G[\![\sigma^F(A)]\!] = \mathcal{A}[\![A]\!](\lambda x \in Var .\ \mathcal{A}^G[\![\sigma(x)]\!])$.

Beweis Durch strukturelle Induktion über $t \in Ter_\Sigma$ und $A \in QFor_\Sigma$. □

Lemma 10.7.4 (Substitution) *Sei $\mathcal{A}$ eine Σ-Termstruktur und $\sigma \in Val_{\mathcal{A}}$. Dann:*

1. $\forall t \in Ter_\Sigma\colon\ \mathcal{A}[\![t]\!]\sigma = \sigma^T(t)$.
2. $\forall A \in QFor_\Sigma\colon\ \mathcal{A}[\![A]\!]\sigma = \mathcal{A}^G[\![\sigma^F(A)]\!]$.

Beweis Folgt unmittelbar aus dem allgemeineren Lemma 10.7.3. □

Eine Klauselmenge heißt *quantorenfrei* genau dann, wenn alle ihre Klauseln nur quantorenfreie Formeln enthalten.

Die Σ-*Expansion* einer quantorenfreien Klauselmenge $S \subseteq Cla_\Sigma$ ist wie folgt definiert:

$$Exp_\Sigma(S) \stackrel{\text{def}}{=} \{\, \{\, \sigma^{\mathrm{F}}(A) \mid A \in C \,\} \mid C \in S \text{ und } \sigma \in Var \to GTer_\Sigma \,\}$$

Offensichtlich ist $Exp_\Sigma(S)$ eine variablenfreie Teilmenge von Cla_Σ. Die Klauseln in $Exp_\Sigma(S)$ nennt man auch die Σ-*Grundinstanzen* der Klauseln in S.

Lemma 10.7.5 *Sei $S \subseteq Cla_\Sigma$ eine quantorenfreie Klauselmenge und $\mathcal{A}$ ein Σ-Termmodell von $Exp_\Sigma(S)$. Dann ist $\mathcal{A}$ ein Modell von S.*

Beweis Sei $C \in S$ und $\sigma \in Val_\mathcal{A}$. Es genügt zu zeigen, dass ein $A \in C$ existiert mit $\mathcal{A}[\![A]\!]\sigma = 1$.

Offensichtlich ist $\{\, \sigma^{\mathrm{F}}(A) \mid A \in C \,\} \in Exp_\Sigma(S)$. Also ist $\mathcal{A}$ ein Modell dieser variablenfreien Klausel. Also existiert ein $A \in C$ mit $\mathcal{A}^{\mathrm{G}}[\![\sigma^{\mathrm{F}}(A)]\!] = 1$. Mit dem Substitutionslemma 10.7.4 folgt $\mathcal{A}[\![A]\!]\sigma = 1$. □

Proposition 10.7.6 *Sei $S \subseteq Cla_\Sigma$ eine erfüllbare quantorenfreie Klauselmenge. Dann ist $Exp_\Sigma(S)$ erfüllbar.*

Beweis Wenn Σ kein Konstantensymbol enthält, dann ist $Exp_\Sigma(S)$ leer und folglich erfüllbar.

Enthalte Σ mindestens ein Konstantensymbol und sei $\mathcal{A}$ ein Modell von S. Wegen Proposition 10.7.1 können wir annehmen, dass $\Sigma_\mathcal{A} = \Sigma$. Sei $C \in S$ und $\sigma \in Var \to GTer_\Sigma$. Es genügt zu zeigen, dass ein $A \in C$ existiert mit $\mathcal{A}^{\mathrm{G}}[\![\sigma^{\mathrm{F}}(A)]\!] = 1$.

Da $\mathcal{A}$ ein Modell von S ist, gibt es ein $A \in C$ mit

$$\mathcal{A}[\![A]\!](\lambda x \in Var\,.\, \mathcal{A}^{\mathrm{G}}[\![\sigma(x)]\!])$$

Also gilt $\mathcal{A}^{\mathrm{G}}[\![\sigma^{\mathrm{F}}(A)]\!] = 1$ mit Lemma 10.7.3. □

Satz 10.7.7 (Expansion) *Sei $S \subseteq Cla_\Sigma$ eine quantorenfreie Klauselmenge und enthalte Σ mindestens ein Konstantensymbol. Dann sind die folgenden Aussagen äquivalent:*

1. *S erfüllbar.*
2. *S hat ein Σ-Termmodell.*
3. *$\mathcal{K}[\![Exp(S)_\Sigma]\!] \neq \emptyset$.*

Beweis (1) $\Rightarrow$ (3). Sei S erfüllbar. Dann ist $Exp(S)_\Sigma$ erfüllbar (Proposition 10.7.6). Also $\mathcal{K}[\![Exp_\Sigma(S)]\!] \neq \emptyset$ mit Satz 10.7.2.

(3) $\Rightarrow$ (2). Sei $\mathcal{K}[\![Exp_\Sigma(S)]\!] \neq \emptyset$. Dann liefert Satz 10.7.2 ein Σ-Termodell $\mathcal{A}$ für $Exp_\Sigma(S)$. Mit Lemma 10.7.5 folgt, dass $\mathcal{A}$ ein Modell von S ist.

(2) $\Rightarrow$ (1). Trivial. $\square$

Satz 10.7.8 (Herbrand) *Sei $S \subseteq Cla_\Sigma$ eine quantorenfreie Klauselmenge und enthalte Σ mindestens ein Konstantensymbol. Dann ist S unerfüllbar genau dann, wenn es eine endliche Teilmenge $S' \subseteq Exp_\Sigma(S)$ gibt mit $\mathcal{K}[\![S']\!] = \emptyset$.*

Beweis Folgt aus dem Expansionssatz 10.7.7 und dem aussagenlogischen Kompaktheitssatz 4.18.2. $\square$

Eine Struktur heißt *abzählbar* genau dann, wenn ihr Universum abzählbar ist. Da die Menge aller Terme abzählbar ist, ist jedes Termmodell abzählbar.

Satz 10.7.9 (Löwenheim-Skolem) *Jede erfüllbare Formelmenge hat ein abzählbares Modell.*

Beweis Folgt aus dem Eliminationssatz 10.6.7, Proposition 10.6.8 und dem Expansionssatz 10.6.7, da jede Termstruktur abzählbar ist. $\square$

Satz 10.7.10 (Kompaktheit) *Eine Formelmenge ist erfüllbar genau dann, wenn jede ihrer endlichen Teilmengen erfüllbar ist.*

Beweis Folgt aus dem Eliminationssatz 10.6.7, Proposition 10.6.8 und dem Satz von Herbrand (10.7.8). $\square$

Satz 10.7.11 (Testbarkeit) *Die Unerfüllbarkeit von testbaren Formelmengen ist testbar.*

Beweis Folgt aus dem Eliminationssatz 10.6.7, Proposition 10.6.8, dem Expansionssatz 10.7.7 und der Tatsache, dass aussagenlogische Erfüllbarkeit testbar ist. $\square$

Korollar 10.7.12 *Allgemeingültigkeit von Formeln ist testbar.*

10.8 Unifikation

Unifikation ist eine Operation, die allgemeinste Lösungen für Termgleichungen berechnet. Wir benötigen Unifikation als Baustein für einen praktisch brauchbaren prädikatenlogischen Resolutionsbeweiser. Darüber hinaus ist Unifikation ein wichtiger Baustein für einen Typableiter für Standard ML.

Im Folgenden legen wir eine Signatur Σ zugrunde, die mindestens ein Konstantensymbol enthält. Wir werden ausschließlich mit syntaktischen Objekten arbeiten, in denen nur Funktions- und Prädikatensymbole aus Σ vorkommen. Um unsere Notation übersichtlich zu halten lassen wir bei Ter_Σ, For_Σ und so weiter den Index Σ weg und schreiben einfach *Ter*, *For* und so weiter.

Eine *Substitution* ist eine endliche Funktion $\theta \in \mathit{Var} \xrightarrow{\mathit{fin}} \mathit{Ter}$, die die folgende Bedingung erfüllt: $\forall x \in \mathit{Dom}\,\theta\colon\ \theta x \neq x$. Diese Bedingung sorgt dafür, dass beispielsweise $\{x \mapsto x\}$ keine Substitution ist. Sie ist notwendig, damit Proposition 10.8.1 gilt. Die Menge aller Substitutionen bezeichnen wir mit *Sub*.

Die Erweiterungen einer Substitution θ auf Terme und quantorenfreie Formeln und Klauseln ist wie folgt definiert:

$\theta^{\mathrm{T}} \in \mathit{Ter} \to \mathit{Ter}$

$$\begin{aligned} \theta^{\mathrm{T}}(x) &= \text{if } x \in \mathit{Dom}\,\theta \text{ then } \theta x \text{ else } x \\ \theta^{\mathrm{T}}(f(t_1,\ldots,t_n)) &= f(\theta^{\mathrm{T}}(t_1),\ldots,\theta^{\mathrm{T}}(t_n)) \end{aligned}$$

$\theta^{\mathrm{F}} \in \mathit{QFor} \to \mathit{QFor}$

$$\begin{aligned} \theta^{\mathrm{F}}(p(t_1,\ldots,t_n)) &= p(\theta^{\mathrm{T}}(t_1),\ldots,\theta^{\mathrm{T}}(t_n)) \\ \theta^{\mathrm{F}}(\neg A) &= \neg(\theta^{\mathrm{F}}(A)) \\ \theta^{\mathrm{F}}(A_1 \wedge A_2) &= \theta^{\mathrm{F}}(A_1) \wedge \theta^{\mathrm{F}}(A_2) \end{aligned}$$

$\theta^{\mathrm{C}} \in \mathit{QCla} \to \mathit{QCla}$

$$\theta^{\mathrm{C}}(C) = \{\,\theta^{\mathrm{F}}(A) \mid A \in C\,\}$$

Da θ immer eine Substitution bezeichnen wird, schreiben wir statt $\theta^{\mathrm{T}}(t)$, $\theta^{\mathrm{F}}(A)$ und $\theta^{\mathrm{C}}(C)$ kürzer θt, θA und θC.

Proposition 10.8.1 *Seien θ_1 und θ_2 Substitutionen. Dann gilt:*

$$\theta_1 = \theta_2 \iff \forall t \in \mathit{Ter}\colon\ \theta_1 t = \theta_2 t$$

Proposition 10.8.2 *Seien $\theta_1, \theta_2 \in \mathit{Sub}$. Dann existiert genau ein $\theta \in \mathit{Sub}$ mit $\theta^T = \lambda t \in \mathit{Ter}.\ \theta_1(\theta_2 t)$. Die Substitution θ heißt die* Komposition θ_1 nach θ_2 *und wird mit $\theta_1 \circ \theta_2$ bezeichnet.*

Sei $S \subseteq \mathit{Sub}$. Dann heißt $\theta \in S$ ein *allgemeinstes Element* von S genau dann, wenn $\forall \psi \in S\ \exists \phi \in \mathit{Sub}\colon\ \psi = \phi \circ \theta$.

Eine *Umbenennung* ist eine Substitution θ mit $\emptyset = \theta \circ \theta$. Wenn θ eine Umbenennung ist, dann ist θ^{T} eine bijektive Funktion $\mathit{Ter} \to \mathit{Ter}$ die sich selbst als Umkehrfunktion hat. Für eine Umbenennung θ gilt also $\forall t \in \mathit{Ter}\colon\ t = \theta(\theta t)$.

Lemma 10.8.3 (Substitution) *Für jede Σ-Struktur $\mathcal{A}$ gilt:*

(1) $\forall t \in \mathit{Ter}\ \forall\theta \in \mathit{Sub}\ \forall\sigma \in \mathit{Val}_{\mathcal{A}}:$
$$\mathcal{A}[\![\theta t]\!]\sigma = \mathcal{A}[\![t]\!](\lambda x \in \mathit{Var}.\ \mathcal{A}[\![\theta x]\!]\sigma)$$

(2) $\forall A \in \mathit{QFor}\ \forall\theta \in \mathit{Sub}\ \forall\sigma \in \mathit{Val}_{\mathcal{A}}:$
$$\mathcal{A}[\![\theta A]\!]\sigma = \mathcal{A}[\![A]\!](\lambda x \in \mathit{Var}.\ \mathcal{A}[\![\theta x]\!]\sigma)$$

Beweis Durch strukturelle Induktion über $t \in \mathit{For}$ und $A \in \mathit{For}$. □

Satz 10.8.4 (Instanzen) *Sei $\mathcal{A}$ ein Σ-Modell einer Klauselmenge $S \subseteq \mathit{Cla}$. Dann: $\forall C \in S\ \forall\theta \in \mathit{Sub}$: $\mathcal{A}$ ist Modell von θC.*

Beweis Sei $C \in S$, $\theta \in \mathit{Sub}$ und $\sigma \in \mathit{Val}_{\mathcal{A}}$. Da $\mathcal{A}$ Modell von S ist, existiert ein $A \in C$ mit $\mathcal{A}[\![A]\!](\lambda x \in \mathit{Var}.\ \mathcal{A}[\![\theta x]\!]\sigma) = 1$. Mit Lemma 10.8.3 folgt $\mathcal{A}[\![\theta A]\!]\sigma = 1$. Also ist $\mathcal{A}$ eine Modell von θC. □

Eine *Gleichungsmenge* ist eine endliche Menge von Paaren (s, t), die aus zwei Termen bestehen. Wir vereinbaren die folgende Notation:

$$E \in GM \stackrel{\text{def}}{=} \mathcal{P}_{\mathit{fin}}(\mathit{Ter} \times \mathit{Ter}) \qquad \textit{Gleichungsmenge}$$
$$s \doteq t \in E \qquad \textit{Gleichung}$$

Die *Unifikatoren* einer Gleichungsmenge sind wie folgt definiert:

$$\mathcal{U} \in GM \to \mathit{Sub}$$
$$\mathcal{U}[\![E]\!] = \{\,\theta \in \mathit{Sub} \mid \forall s \doteq t \in E:\ \theta s = \theta t\,\}$$

Die Elemente von $\mathcal{U}[\![E]\!]$ heißen die *Unifikatoren* von E. Eine Gleichungsmenge E heißt *unifizierbar* genau dann, wenn $\mathcal{U}[\![E]\!] \neq \emptyset$.

Sei E eine Gleichungsmenge. Eine Substitution $\theta \in \mathcal{U}[\![E]\!]$ heißt *allgemeinster Unifikator* von E genau dann, wenn θ ein allgemeinstes Element von $\mathcal{U}[\![E]\!]$ ist. Eine Substitution $\theta \in \mathcal{U}[\![E]\!]$ heißt *prinzipaler Unifikator* von E genau dann, wenn die folgenden Bedingungen gelten:

1. θ ist ein allgemeinster Unifikator von E.
2. $\theta = \theta \circ \theta$ (Idempotenz).
3. In $\mathit{Dom}\ \theta$ und $\mathit{Ran}\ \theta$ kommen nur Variablen vor, die auch in E vorkommen.

Eine Gleichung heißt *widerlegt* genau dann, wenn sie eine der folgenden zwei Formen hat:

1. $f(s_1, \ldots, s_m) \doteq g(t_1, \ldots, t_n)$ mit $f \neq g$.

$$
\begin{array}{lrcl}
(1) & E,\ t \doteq t & \xrightarrow{\text{u}} & E \\
(2) & E,\ f(s_1,\dots,s_n) \doteq f(t_1,\dots,t_n) & \xrightarrow{\text{u}} & E,\ s_1 \doteq t_1,\ \dots,\ s_n \doteq t_n \\
(3) & E,\ x \doteq t & \xrightarrow{\text{u}} & E[t/x],\ x \doteq t \\
 & & & \text{falls } x \text{ in } E \text{ vorkommt und in } t \text{ nicht vorkommt} \\
(4) & E,\ f(t_1,\dots,t_n) \doteq x & \xrightarrow{\text{u}} & E,\ x \doteq f(t_1,\dots,t_n)
\end{array}
$$

Abbildung 10.6: Unifikationsregeln

2. $x = f(t_1, \dots, t_n)$ mit x kommt in einem der Terme $t_1, \dots, t_n$. vor.

Eine Gleichungsmenge E heißt *widerlegt* genau dann, wenn sie eine widerlegte Gleichung enthält.

Proposition 10.8.5 *Widerlegte Gleichungsmengen sind nicht unifizierbar.*

Eine Gleichungsmenge E heißt *gelöst* genau dann, wenn sie die Form

$$E = \{x_1 \doteq t_1, \dots, x_n \doteq t_n\}$$

hat mit

1. $x_1, \dots, x_n$ paarweise verschieden.
2. $\forall i, j \in \{1, \dots, n\}$: x_i kommt nicht t_j vor.

Proposition 10.8.6 *Sei $E = \{x_1 \doteq t_1, \dots, x_n \doteq t_n\}$ eine gelöste Gleichungsmenge. Dann ist*

$$\{x_1 \mapsto t_1, \dots, x_n \mapsto t_n\}$$

ein prinzipaler Unifikator von E.

Abbildung 10.6 zeigt Vereinfachungsregeln für Gleichungsmengen, die man als *Unifikationsregeln* bezeichnet. Wir schreiben $E \xrightarrow{\text{u}} E'$ genau dann, wenn die Gleichungsmenge E' aus der Gleichungsmenge E durch die Anwendung einer Vereinfachungsregel erhalten werden kann.

Satz 10.8.7 (Unifikation) *Sei E eine Gleichungsmenge. Dann:*

1. *Wenn $E \xrightarrow{\mathrm{u}} E'$, dann $\mathcal{U}[\![E]\!] = \mathcal{U}[\![E']\!]$.*
2. *E ist widerlegt oder gelöst genau dann, wenn es kein E' mit $E \xrightarrow{\mathrm{u}} E'$ gibt.*
3. *Es gibt keine unendliche Kette $E \xrightarrow{\mathrm{u}} E_1 \xrightarrow{\mathrm{u}} E_2 \xrightarrow{\mathrm{u}} \cdots$.*

Gegeben eine Gleichungsmenge E, kann man mit den Unifikationsregeln entscheiden, ob E unifizierbar ist. Dazu wendet man die Unifikationsregel solange auf E an, bis man eine Gleichungsmenge E' bekommt, die nicht weiter vereinfacht werden kann. Dann hat E' genau dieselben Unifikatoren wie E. Außerdem ist E' entweder widerlegt oder gelöst. Wenn E' widerlegt ist, dann ist E nicht unifizierbar. Wenn E' gelöst ist, dann liefert E' gemäß Proposition 10.8.6 einen prinzipalen Unifikator für E.

Es kann passieren, dass das vereinfachte Gleichungssystem exponentiell größer ist als das Ausgangssystem (wegen Regel (3)). Die Aufblähung kann man in der Praxis leicht vermeiden, indem man die Terme durch die Knoten eines Graphens repräsentiert. Man kann zeigen, dass man die Unifizierbarkeit einer Gleichungsmenge mit linearer Zeit entscheiden kann.

10.9 Prädikatenlogische Resolution

Wir verallgemeinern jetzt die aussagenlogische Resolventenbildung so, dass man damit die Unerfüllbarkeit von prädikatenlogischen Klauselmengen zeigen kann.

Wie im Abschnitt über Unifikation lassen wir den Signaturindex Σ der Übersichtlichkeit halber weg und nehmen an, dass alle syntaktischen Objekte gemäß einer Signatur Σ gebildet sind, die wenigstens eine Konstante enthält.

Äquivalenz von zwei Klauselmengen $S, S' \in \mathit{Cla}$ definieren wir wie folgt:

$$S \models\!\mid S' \overset{\text{def}}{\iff} S \text{ und } S' \text{ haben dieselben } \Sigma\text{-Modelle}$$

Die Menge der quantorenfreien Klauseln bezeichnen wir mit *QCla*.

Abbildung 10.7 definiert mithilfe zweier Inferenzregeln den *naiven prädikatenlogischen Resolutionsabschluss $\mathit{Res}(S)$* einer quantorenfreien Klauselmenge $S \subseteq \mathit{QCla}$. Die durch die zweite Inferenzregel eingeführten Klauseln heißen wie im aussagenlogischen Fall *Resolventen*. Für variablenfreie Klauselmengen ist der naive prädikatenlogische Resolutionsabschluss identisch mit dem aussagenlogischen Resolutionsabschluss.

$$\frac{C \in S}{C \in Res(S)}$$

$$\frac{A \in C_1 \in Res(S) \qquad \neg B \in C_2 \in Res(S) \qquad \theta_1 A = \theta_2 B \qquad \theta_1, \theta_2 \in Sub}{(\theta_1 C_1 - \{\theta_1 A\}) \cup (\theta_2 C_2 - \{\neg\theta_2 B\}) \in Res(S)}$$

Abbildung 10.7: Naiver prädikatenlogischer Resolutionsabschluss

Proposition 10.9.1 *Sei S eine quantorenfreie Klauselmenge. Dann $Res(S) \models\!\!\!\!\!\!= S$.*

Beweis Folgt aus Satz 10.8.4 und der Tatsache, dass

$$(A \vee B_1) \;\wedge\; (\neg A \vee B_2) \;\Rightarrow\; B_1 \vee B_2$$

aussagenlogisch gültig ist. □

Satz 10.9.2 (Resolution) *Eine literale Klauselmenge S ist unerfüllbar genau dann, wenn $\emptyset \in Res(S)$.*

Beweis Folgt aus Proposition 10.9.1, dem Expansionssatz 10.7.7, der Gleichung $Res(Exp_\Sigma(S)) = Res(S) \cup Exp_\Sigma(S)$ und dem aussagenlogischen Resolutionssatz 4.15.2. □

Abbildung 10.8 zeigt einen prädikatenlogischen Resolutionsgraphen. Der Resolutionssatz garantiert uns, dass die Unerfüllbarkeit jeder literalen Klauselmenge durch einen endlichen Resolutionsgraphen gezeigt werden kann.

Ein *Beweiser* ist ein Programm, das versucht, zu einer literalen Klauselmenge einen Resolutionsgraphen zu konstruieren, der zeigt, dass die Klauselmenge uenerfüllbar ist. Der Resolutionssatz sagt, dass ein solcher Graph genau dann existiert, wenn die Klauselmenge unerfüllbar ist. Die Resolutionsregel in Abbildung 10.7 hat in Bezug auf eine praktikable Realisierung des Beweisers jedoch das Problem, dass sie zu zwei Klauseln im Allgemeinen unendlich viele Resolventen generiert. Der Informatiker J.A. Robinson hat 1964 entdeckt, dass man die zu betrachtenden Resolventen auf endlich viele allgemeinste Resolventen einschränken kann, die man mithilfe von Unifikation aus den beiden Elternklauseln berechnen kann. Aufbauend auf dieser Grundlage kann man mit weiteren Techniken praktikable Beweiser realisieren.

Mit *GCla* bezeichnen wir die Menge aller variablenfreien Klauseln. Die Funktion

$$GR \in GCla \times GCla \to \mathcal{P}(GCla)$$
$$GR(C_1, C_2) = \{\, (C_1 - \{A\}) \cup (C_2 - \{\neg A\}) \mid A \in C_1,\ \neg A \in C_2 \,\}$$

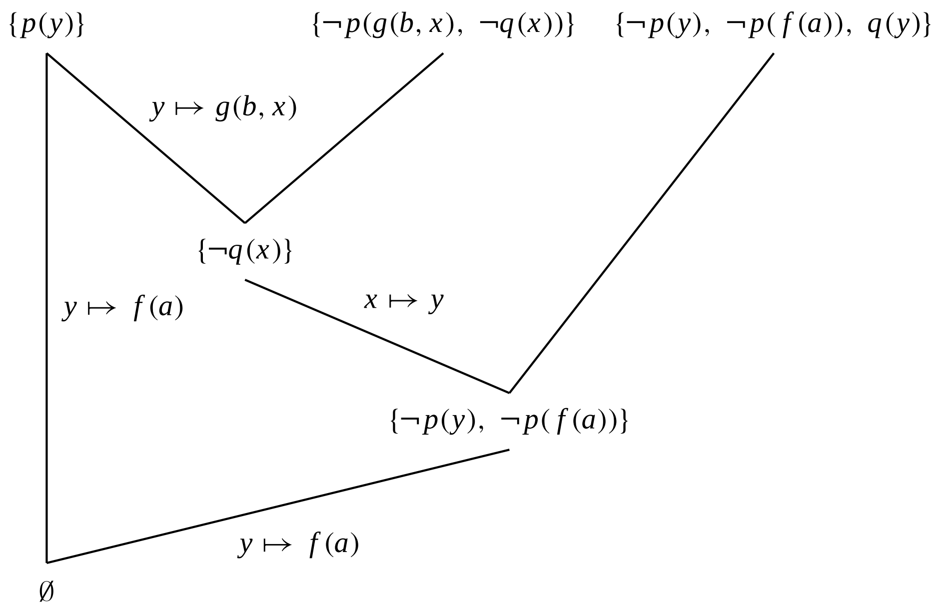

Abbildung 10.8: Ein prädikatenlogischer Resolutionsgraph

liefert zu zwei variablenfreien Klauseln die Menge aller ihrer Resolventen.

Mit *LCla* bezeichnen wir die Menge aller literalen Klauseln. Ein *Resolvierer* ist eine Funktion

$$r \in LCla \times LCla \to \mathcal{P}(LCla)$$

die zu zwei literalen Klauseln eine Menge von literalen Klauseln wie folgt liefert:

$$Exp(r(C_1, C_2)) = \bigcup \{\, GR(C_1', C_2') \mid C_1' \in Exp\{C_1\},\ C_2' \in Exp\{C_2\} \,\}$$

Abbildung 10.9 definiert zu einem Resolvierer r und einer literalen Klauselmenge S eine literale Klauselmenge $Res(r, S)$.

Der *triviale Resolvierer*

$$\begin{aligned} r_0(C_1, C_2) = \{\, (\theta_1 C_1 - \{\theta_1 A\}) \cup (\theta_2 C_2 - \{\neg\theta_2 B\}) \mid \\ A \in C_1 \in Res(S),\ \neg B \in C_2 \in Res(S), \\ \theta_1 A = \theta_2 B,\ \theta_1 \in Sub,\ \theta_2 \in Sub \,\} \end{aligned}$$

entspricht der Resolutionsregel in Abbildung 10.7. Offensichtlich gilt:

$$\forall S \subseteq LCla\colon\ Res(S) = Res(r_0, S)$$

$$\frac{C \in S}{C \in Res(r, S)} \qquad \frac{C_1 \in Res(r, S) \quad C_2 \in Res(r, S) \quad C \in r(C_1, C_2)}{C \in Res(r, S)}$$

Abbildung 10.9: Resolvierer-basierter Resolutionsabschluss

Satz 10.9.3 (Resolution) *Sei r ein Resolvierer. Dann ist eine literale Klauselmenge S genau dann unerfüllbar, wenn $\emptyset \in Res(r, S)$.*

Eine Substitution θ heißt *Unifikator* einer literalen Klausel C genau dann, wenn θC aus genau einem Literal besteht. Ein Unifikator θ einer literalen Klausel C heißt *allgemeinster Unifikator* von C genau dann, wenn für jeden Unifikator ψ von C eine Substitution ϕ mit $\psi = \phi \circ \theta$ existiert. Offensichtlich kann man das Unifikationsproblem für Klauseln auf das Unifikationsproblem für Gleichungsmengen zurückführen. Wir wissen also, wie man die Unifizierbarkeit einer Klauselmenge entscheidet und wie man gegebenfalls einen allgemeinsten Unifikator berechnet.

Wir wählen ein Objekt $\perp \notin Sub$. Ein *Unifizierer* ist eine Funktion (*mgu* steht für most general unifier)

$$mgu \in LCla \to Sub \cup \{\perp\}$$

so dass für jede literale Klausel C gilt:

1. C unifizierbar genau dann, wenn $mgu(C) \neq \perp$
2. Wenn C unifizierbar ist, dann ist $mgu(C)$ ein allgemeinster Unifikator von C.

Ein Klausel $C \in QCla$ heißt *Variante* einer Klausel $D \in QCla$ genau dann, wenn es eine Umbenennung θ gibt mit $C = \theta D$.

Zwei Klausel heißen *variablendisjunkt* genau dann, wenn es keine Variable gibt, die in beiden Klauseln vorkommt. Ein *Umbenenner* ist eine Funktion

$$ren \in LCla \times LCla \to LCla$$

so dass für alle literalen Klauseln C_1 und C_2 die Klausel $ren(C_1, C_2)$ eine Variante von C_1 ist, die variablendisjunkt zu C_2 ist.

Satz 10.9.4 (Resolution) *Sei mgu ein Unifizierer und ren ein Umbenenner. Dann* *ist*

$$\begin{aligned} r(C_1, C_2) = \{\, (\theta C_1 - \theta C_1') \cup (\theta C_2 - \theta C_2') \mid \\ \emptyset \neq C_1' \subseteq ren(C_1, C_2), \\ \emptyset \neq C_2' = \{\neg B_1, \ldots, \neg B_n\} \subseteq C_2 \\ \theta = mgu(C_1' \cup \{B_1, \ldots, B_n\}) \neq \bot \,\} \end{aligned}$$

ein Resolvierer.

Der in diesem Satz angegebene Resolvierer hat die für die Realisierung eines Beweisers wichtige Eigenschaft, dass er zu zwei Klausel nur endlich viele Resolventen liefert.

10.10 Allgemeingültigkeit ist unentscheidbar

Satz 10.10.1 (Unentscheidbarkeit) *Sei Σ eine Signatur mit mindestens einer Konstante, zwei einstelligen Funktionssymbolen, und einem zweistelligen Prädikatensymbol. Dann ist die Menge $\{\, A \in For_\Sigma \mid A$ allgemeingültig $\}$ unentscheidbar.*

Die Satz wurde um 1936 erstmals von Alonzo Church gezeigt. Er gilt auch für Signaturen ohne Funktionssymbole, die genügend viele Prädikatensymbole haben.

Um den Unentscheidbarkeitssatz zu zeigen, benötigen wir ein unentscheidbares, aber testbares Problem, dass wir auf das Allgemeingültigkeitsproblem reduzieren können (das Allgemeingültigkeitsproblem ist testbar, siehe Satz 10.7.11). Dafür eignet sich beispielsweise das Halteproblem für IMP (Satz 9.5.3). Wir können aber viel Arbeit sparen, wenn wir ein anderes als unentscheidbar aber testbar bekanntes Problem wählen, das sich einfacher mit prädikatenlogischen Formeln codieren lässt. Zum Beispiel ist das Postsche Korrespondenzproblem dafür gut geeignet. Einen entsprechenden Beweis finden Sie in [Schöning, Kapitel 2.3].

Ein *Beweissystem* für eine Menge U ist ein Paar (B, β) wie folgt:

1. B ist eine in polynomialer Zeit entscheidbare Menge.
2. $\beta \in B \to U$ ist eine in polynomialer Zeit berechenbare surjektive Funktion.

Dabei heißt $b \in B$ *Beweis für* $u \in U$ genau dann, wenn $\beta(b) = u$. Offensichtlich ist jede Menge, zu der ein Beweissystem existiert, testbar.

Für die Menge der allgemeingültigen Formeln können wir ein Beweissystem wie folgt konstruieren. Als Beweise (die Elemente der Menge B) wählen wir Tripel wie folgt:

1. Eine Formel A.
2. Eine Ableitung $\{\{\exists\neg A\}\} \xrightarrow{s} \cdots \xrightarrow{s} S$, so dass S eine literale Klauselmenge ist.
3. Ein Resolutionsgraph, der aus S die leere Klausel ableitet.

10.11 Theorien und Axiomatisierung

In diesem Abschnitt arbeiten wir wieder mit mehr als einer Signatur.

Eine Σ-*Theorie* ist eine Formelmenge $T \subseteq For_\Sigma$, so dass

$$T = \{\, A \in For_\Sigma \mid T \models_\Sigma A \,\}$$

gilt. Eine Theorie ist also eine unter logischer Konsequenz abgeschlossene Formelmenge. Eine Σ-Theorie T heißt *vollständig* genau dann, wenn für jede geschlossene Formel $A \in For_\Sigma$ gilt: $A \in T$ oder $\neg A \in T$.

Sei $\mathcal{A}$ eine Σ-Struktur. Dann bezeichnen wir die Formelmenge

$$Th(\mathcal{A}) \overset{\text{def}}{=} \{\, A \in For_\Sigma \mid A \text{ gültig in } \mathcal{A} \,\}$$

als die *Theorie von* $\mathcal{A}$. Offensichtlich ist $Th(\mathcal{A})$ eine vollständige Σ-Theorie.

Sei $M \subseteq For_\Sigma$. Dann bezeichnen wir

$$Th_\Sigma(M) \overset{\text{def}}{=} \{\, A \in For_\Sigma \mid M \models_\Sigma A \,\}$$

als die *von M erzeugte Theorie*.

Proposition 10.11.1 *Sei $M \subseteq For_\Sigma$. Dann $M \subseteq Th_\Sigma(M)$. Außerdem ist M genau dann eine Σ-Theorie, wenn $M = Th_\Sigma(M)$ gilt.*

Proposition 10.11.2 *Sei $M \subseteq For_\Sigma$ testbar. Dann ist $Th_\Sigma(M)$ testbar.*

Beweis Folgt aus Satz 10.7.11. □

Proposition 10.11.3 *Eine vollständige Theorie ist testbar genau dann, wenn sie entscheidbar ist.*

Beweis Sei T eine testbare und vollständige Σ-Theorie. Sei $A \in For_\Sigma$ eine geschlossene Formel. Wir können einen steuerbaren Tester für T abwechselnd auf A und $\neg A$ anwenden. Da T vollständig ist, wissen wir nach endlich vielen Schritten, ob A oder $\neg A$ in T ist. □

Eine *Axiomatisierung* einer Σ-Theorie T ist eine entscheidbare Formelmenge $M \subseteq For_\Sigma$ mit $T = Th_\Sigma(M)$.

Proposition 10.11.4 *Sei M eine Axiomatisierung einer Σ-Theorie T. Dann ist T testbar. Wenn T vollständig ist, dann ist T sogar entscheidbar.*

Beweis Folgt aus Proposition 10.11.2 und 10.11.3. □

Von besonderem Interesse sind *minimale Axiomatisierungen*. Dabei heißt eine Formelmege $M \subseteq For_\Sigma$ *minimal* genau dann, wenn

$$\forall A \in M\colon\ Th_\Sigma(M - \{A\}) \neq Th_\Sigma(M)$$

Um 1900 glaubten viele Logiker (zum Beispiel David Hilbert), dass sich die üblichen mathematischen Theorien in Prädikatenlogik erster Stufe axiomatisieren lassen. Dieser Glaube war etwas verwegen, da man das Konzept der Berechenbarkeit noch gar nicht formal definieren konnte (das gelang erst Church und Turing um 1937). Gödel zeigte um 1933 mit seinem Unvollständigkeitssatz 9.6.1, dass die Theorie der ganzen Zahlen mit der Signatur $\{0, 1, -1, +, *, =\}$ nicht axiomatisierbar ist.

Es gibt aber durchaus interessante Strukturen, deren Theorie axiomatisierbar ist. Beispielweise ist die Theorie der natürlichen Zahlen mit der Signatur $\{0, 1, +, =\}$ entscheidbar (sogenannte Presburger Arithmetik). Dieses Ergebnis wurde von Presburger, einem Diplomanden Tarskis, mit der Methode der Quantorenelimination gezeigt.

Auch die Theorie der reellen Zahlen mit der Signatur $\{0, 1, -1, +, *, =\}$ ist entscheidbar. Dieses Ergebnis wurde von Tarski ebenfalls mit der Methode der Quantorenelimination gezeigt. Der Satz von Löwenheim-Skolem 10.7.9 liefert uns das auf den ersten Blick überraschende Ergebnis, dass die Theorie der reellen Zahlen auch ein abzählbares Modell hat.

Die gerade aufgeführten Resulte für die Theorie der reellen Zahlen sind nur auf den ersten Blick verblüffend. Sie sagen im Wesentlichen, dass die Expressivität (Aussdrucksstärke) von prädikatenlogischen Sprachen erster Stufe begrenzt ist in dem Sinne, dass ihre Formeln wesentliche Eigenschaften der reellen Zahlen nicht ausdrücken können. Diese Schwierigkeit verschwindet, wenn man mit prädikatenlogischen Sprachen höherer Ordnung arbeitet, die auch über Funktionen und Prädikate quantifizieren können.

Wir können also einerseits feststellen, dass die Expressivität der Prädikatenlogik erster Stufe eingeschränkt ist. Andererseits ist sie aus der Sicht der Berechenbarkeit nicht gerade bescheiden: die Theorie der ganzen Zahlen ist nicht testbar und Allgemeingültigkeit ist immerhin unentscheidbar. Wenn man zu expressiveren Sprachen übergeht, bezahlt man damit, dass Allgemeingültigkeit nicht mehr testbar ist.

Bisher haben wir nur die vollständigen Theorien konkreter Strukturen wie $\mathbb{Z}$ und $\mathbb{R}$ betrachtet. Beim mathematischen Arbeiten verwendet man jedoch oft Abstraktionen wie Gruppen oder partielle Ordnungen. Diese Abstraktionen entsprechen unvollständigen Theorien und lassen sich oft mit prädikatenlogischen Sprachen erster Stufe axiomatisieren. Dabei verwendet man typischerweise prädikatenlogische Sprachen mit Gleichheit. Diese stellen atomare Formeln $s = t$ bereit, die als Gleichheitsaussagen interpretiert werden. Man kann zeigen, dass man das eingebaute Gleichheitsprädikat mithilfe eines zweistelligen Prädikatensymbols axiomatisieren kann, ohne dass sich der Begriff der Gültigkeit ändert.

