Logik für Informatiker

Prof. Dr. Dietmar Seipel
Universität Würzburg

Wintersemester 2018/19
20. Dezember 2018

2 Prädikatenlogik

Die Prädikatenlogik erweitert die Aussagenlogik syntaktisch um Symbole für *Variablen*, Konstanten und Funktionen und durch *Quantoren*.

Die Semantik der Prädikatenlogik und der Begriff der logischen Folgerung werden über Strukturen definiert. In der Praxis reichen für uns die einfacheren Herbrand–Stukturen aus.

Wir können Formeln durch Verschieben der Quantoren in Normalform bringen: Pränexform, Skolemform, Klauselform.

Die zentrale Inferenzmethode zur Ableitung neuer Formeln aus gegebenen ist wieder die *Resolution*. Wir werden auch zwei Speziallfälle behandeln: die Hyperresolution für Deduktive Datenbanken (bottom–up) und die SLD–Resolution der Logikprogrammierung (top–down).

In der Prädikatenlogik kann man die in der Aussagenlogik bereits betrachteten atomaren Aussagen eleganter und allgemeiner formulieren:

A = „Bayern München ist deutscher Fußballmeister“:
$deutscher\_fussballmeister('Bayern\ Muenchen', 2018)$.

B = „eine Stadt mit mindestens 100000 Einwohnern ist groß“:
$grosse\_stadt(Stadt) \leftarrow einwohner(Stadt, X) \wedge X \geq 100000$.

C = „Würzburg hat über 100.000 Einwohner“:
$einwohner('Wuerzburg', 100000)$.

Dann kann man *generische Aussagen* machen. Z.B. ist die Regel B auf alle Städte mit mindestens 100000 Einwohnern anwendbar. Insbesondere könnte die Regel B für $Stadt = Wuerzburg$ aus dem Fakt C schließen, daß diese eine $grosse\_stadt$ ist.

2.1 Grundbegriffe der Prädikatenlogik

Erweiterung der Aussagenlogik um

Variablen–	Symbole	U, V, W, X, Y, Z
Funktions–	Symbole	f, g, h a, b, c
Prädikaten–	Symbole	p, q, r

und Quantoren:

- Existenzquantor: $\exists$,
- Allquantor: $\forall$.

Die bekannten Junktoren $\wedge$, $\vee$, $\neg$ und $\rightarrow$ sind weiterhin erlaubt.

Wir nehmen im Folgenden meist an, daß Variablensymbole mit Großbuchstaben beginnen, und daß Funktions– und Prädikatensymbole mit Kleinbuchstaben beginnen oder in Hochkommata eingeschlossen sind.

Beispiel (Prädikatenlogiche Formeln)

1. Die Formel

$$(\exists\, Jahr\ \ weltmeister(brasilien, Jahr))\ \wedge$$
$$(\forall\, Jahr\ \ \neg weltmeister(portugal, Jahr))$$

   besagt, daß es ein Jahr gibt, in dem Brasilien Fußballweltmeister war, und daß Portugal in allen Jahren nicht Fußballweltmeister war.

2. Die Formel mit Implikation

$$\forall\, Stadt\ \forall X\ (\ grosse\_stadt(Stadt) \leftarrow$$
$$einwohner(Stadt, X) \wedge X \geq 100000\ )$$

   mit zwei Allquantoren ist äquivalent zu

$$\forall\, Stadt\ (\ grosse\_stadt(Stadt) \leftarrow$$
$$\exists X\ (\ einwohner(Stadt, X) \wedge X \geq 100000\ )\ ).$$

   In der Logikprogrammierung werden die Quantoren weggelassen; man nimmt implizit Allquantoren an – vgl. obige Formel B.

3. Die Formel zur Konvergenz (Grenzwert) $\lim_{n \to \infty} f(n) = a$ ist in der Analysis definiert als

$$\forall\, \epsilon > 0 \;\exists\, n_0 \;\forall\, n \geq n_0 \;\; |f(n) - a| < \epsilon.$$

Hier sind – ausnahmsweise – ϵ, n_0 und n Variablensymbole (nur über diese kann man quantifizieren), 0, a, f, $-$ und $||$ sind Funktionssymbole, und $>$, $\geq$ und $<$ sind Prädikatensymbole.

Funktionssymbole ohne Argumente – wie 0 und a – nennt man auch Konstantensymbole; ihre Stelligkeit ist 0. f und $||$ sind 1–stellig, $-$ ist 2–stellig. Hier sind auch alle Prädikatensymbole 2–stellig.

Für $f(n) = 1/n$ gilt z.B. $\lim_{n \to \infty} f(n) = \lim_{n \to \infty} 1/n = 0 = a$.
Für jede vorgegebene reelle Zahl $\epsilon > 0$ existiert ein $n_0 \in I\!N$, etwa $n_0 = \lceil 1/\epsilon \rceil + 1$, so daß für alle $n \geq n_0$ gilt $|1/n - 0| < \epsilon$.

4. Die unendliche Formelmenge $M = \{ F_n \,|\, n \in I\!N_0 \}$ aus Kapitel 1 mit

$$\begin{aligned} F_0 &= A_1 \vee A_2 \vee \ldots = \textstyle\bigvee_{n=1}^{\infty} A_n, \\ F_n &= \neg A_n, \text{ für } n \in I\!N_+, \end{aligned}$$

mit der unendlichen – in der Aussagenlogik eigentlich nicht erlaubten – Disjunktion F_0 kann man für $A_n = p(n)$ in der Prädikatenlogik als eine Formel $G_1 \wedge G_2$ ausdrücken mit

$$\begin{aligned} G_1 &= \exists n \in I\!N_+ \;\; p(n), \\ G_2 &= \forall n \in I\!N_+ \;\; \neg p(n), \end{aligned}$$

denn die unendliche Disjunktion F_0 ist dann äquivalent zu G_1 und die restliche Formelmenge $\{ F_n \,|\, n \in I\!N_+ \}$ ist äquivalent zu G_2. Auch hier ist – ausnahmsweise – n ein Variablensymbol.

Die unendliche Formelmenge $M' = \{ G_1 \} \cup \{ \neg p(n) \,|\, n \in I\!N_+ \}$ zeigt, daß der Kompaktheitssatz in der Prädikatenlogik nicht gilt.

Syntax

Wir setzen folgende abzählbaren Mengen voraus:

- Variablensymbole: $\mathcal{V} = \{ X_1, X_2, \ldots \}$,
- Funktionssymbole: $\mathcal{F} = \{ f_1, f_2, \ldots \}$,
- Prädikatensymbole: $\mathcal{P} = \{ p_1, p_2, \ldots \}$.

Jedes Funktions– bzw. Prädikatensymbol hat eine *Stelligkeit* $k \in \mathbb{N}_0$.
Schöning schreibt f_i^k, p_i^k, wir schreiben f_i/k, p_i/k.

Allerdings kann dasselbe Funktions– bzw. Prädikatensymbol in einer Formel mit unterschiedlichen Stelligkeiten auftreten. In

$$f(f(a), f(a, a, a))$$

ist das erste Auftreten von f zweistellig, das zweite einstellig, und das dritte dreistellig. Eigentlich handelt es sich dabei um unterschiediche Funktionssymbole $f/2$, $f/1$ und $f/3$ mit demselben Namen f.

Terme und Formeln werden induktiv definiert.

Terme

1. Jedes Variablensymbol $X \in \mathcal{V}$ ist ein Term.
2. Ist f ein Funktionssymbol der Stelligkeit k, und sind $t_1, \ldots, t_k$ (ebenso viele) Terme, so ist auch $f(t_1, \ldots, t_k)$ ein Term.
3. Ist $k = 0$, so schreibt man kurz f ausstelle von $f()$, und man nennt f dann eine *Konstante*.

Beispiel (Terme)

- $f_1(X_2)$,
- $f_3(f_2, f_1(X_3))$.

Stelligkeiten:

f_1	f_2	f_3
1	0	2

Formeln

1. Für ein k–stelliges Prädikatensymbol p und Terme $t_1, \ldots t_k$ ist $p(t_1, \ldots , t_k)$ eine *atomare Formel*.
   Falls $k = 0$ ist, so schreiben wir kurz p anstelle von $p()$.
2. Für jede Formel F ist auch $\neg F$ eine Formel.
3. Sind F_1 und F_2 Formeln, so auch $(F_1 \wedge F_2)$ und $(F_1 \vee F_2)$.
4. Falls X ein Variablensymbol ist und F eine Formel, so sind auch $\exists X\ F$ und $\forall X\ F$ Formeln.

Man kann wieder redundante Klammerpaare weglassen, und Teilformeln werden analog zur Aussagenlogik definiert.

Man könnte weitere Formeln mittels anderer Junktoren bilden, z.B. die Implikation $F_1 \rightarrow F_2$. Andererseits könnte man diese Formel auch wie $\neg F_1 \vee F_2$ auffassen.

Beispiel (Formeln)

1. In der Formel

   $$F \;=\; (\exists X_1\, p_1(X_1, f_1(X_2))) \;\vee\; (\forall X_2\, p_2(X_2, f_3(f_2, f_1(X_3))))$$

   haben die Prädikatensymbole die folgenden Stelligkeiten:

p_1	p_2
2	2

2. Man kann auch über Variablensymbole quantifizieren, die nicht in der betroffenen Formel vorkommen: $F \;=\; \forall X\; p(Y)$.

3. In der folgenden Implikation bezeichnen die beiden Vorkommen von X dasselbe: $F \;=\; \forall X\; (\, \mathit{student}(X) \rightarrow \mathit{person}(X)\,)$.

4. In $F \;=\; \forall X\; (\, p(X) \rightarrow \exists X\; q(X)\,)$ bezieht sich die Allquantifizierung $\forall X$ auf $p(X)$, die Existenzquantifizierung $\exists X$ auf $q(X)$. Es gibt keinen Zusammenhang zwischen den beiden Vorkommen von X.

Klammerungsregeln

1. Wir nehmen an, daß die Quantoren am stärksten binden. Für einen Quantor $Q \in \{\forall, \exists\}$ und einen Junktor $\otimes \in \{\wedge, \vee, \rightarrow\}$ gilt also:

   $QX\ F \otimes G$ bedeutet $(QX\ F) \otimes G$.

   Z.B. bedeutet

   $$\forall X\, \forall Y\, (\, \exists Z\, (p(X,Z) \wedge p(Z,Y)) \rightarrow p(X,Y)\,)$$

   dasselbe wie

   $$\forall X\, \forall Y\, (\, (\exists Z\, (p(X,Z) \wedge p(Z,Y))) \rightarrow p(X,Y)\,)$$

2. Wir nehmen die Bindungsreihenfolge $\neg, \wedge, \vee, \rightarrow$ an; Negation bindet also stärker als Konjunktion, Disjunktion und Implikation, Konjunktion bindet stärker bindet als Disjunktion (Punkt vor Strich) und Implikation, und Disjunktion bindet stärker als Implikation. D.h.

   $F \vee \neg\, G \wedge G' \rightarrow H$ bedeutet $(F \vee ((\neg\, G) \wedge G')) \rightarrow H$.

Schreibweisen

Die mathematische Formel

$$\forall\, \epsilon > 0 \;\; \exists\, n_0 \;\; \forall\, n \geq n_0 \;\; |f(n) - a| < \epsilon$$

müßte man eigentlich ausführlicher wie folgt schreiben:

$$\begin{array}{l} \forall \epsilon \, (\, \epsilon \in I\!R \;\wedge\; \epsilon > 0 \;\rightarrow \\ \qquad \exists n_0 \, (\, n_0 \in I\!N \;\wedge \\ \qquad\qquad \forall n \, (\, n \in I\!N \;\wedge\; n \geq n_0 \;\rightarrow\; |f(n) - a| < \epsilon \,)\;)\;). \end{array}$$

In der Kurzschreibweise wurde implizit angenommen, daß ϵ eine reelle Zahl ist, und daß n und n_0 natürliche Zahlen sind.

Daneben wurde die vereinfachende Infix– bzw. Zirkumfix–Notation für Funktions– und Prädikatensymbole verwendet.

Manchmal schreibt man anstelle von $\forall X\; F$ auch $\forall X : F$; analog für $\exists$.

Infix– und Zirkumfix–Notation

Zur leichteren Lesbarkeit kann man gewisse binäre Funktions– und Prädikatensymbole in Infix–Notation schreiben.

Anstelle der Präfix–Notation $\odot(t_1, t_2)$ schreiben wir dann die Infix–Notation $t_1 \odot t_2$.

Die atomare Formel

$$|f(n) - a| \; < \; \epsilon$$

mit dem binären Prädikatensymbol $<$ und dem binären Funktionssymbol $-$ würde in Präfix–Notation wie folgt aussehen:

$$< (\; || \, (-(f(n), a)), \; \epsilon \;)$$

Einen Term $||(t)$ mit dem unären Funktionssymbol $||$ (Betrag) können wir in Zirkumfix–Notation $|t|$ schreiben.

Freie und gebundene Variablen

1. Ein Vorkommen einer Variable X in einer Formel F heißt *gebunden*, falls X quantifiziert in einer Teilformel von F der Form $\exists X\ G$ oder $\forall X\ G$ vorkommt (d.h. im Geltungsbereich eines Quantors über X).
2. Andernfalls heißt das Vorkommen von X *frei*.
3. Eine Formel F ohne Vorkommen von freien Variablen heißt *geschlossen*, oder eine Aussage. Andernfalls heißt F *offen*.

Beispiel: Die folgende Formel F ist offen:

$$F \ = \ (\exists X_1\, p_1(\underline{X_1}, f_1(X_2))) \ \vee \ (\forall X_2\, p_2(\underline{X_2}, f_3(f_2, f_1(X_3)))).$$

- Die beiden unterstrichenen Vorkommen der Variablen X_1 und X_2 sind gebunden.
- Alle anderen Vorkommen der Variablen sind frei: d.h. das linke Vorkommen von X_2 und das (einzige) Vorkommen von X_3 sind frei.

Kartesisches Produkt

Für eine Menge M und eine natürliche Zahl $k \in I\!N_+$ ist M^k das k–fache kartesische Produkt, d.h. die Menge aller k–Tupel mit Komponenten aus M.

Für die Menge $M = \{ 1, 2 \}$ erhalten wir z.B.

$$
\begin{aligned}
M^1 &= \{ (1), (2) \}, \\
M^2 &= \{ (1,1), (1,2), (2,1), (2,2) \}, \\
&\vdots \\
M^k &= \{ (t_1, \ldots, t_k) \mid t_1, \ldots, t_k \in M \}.
\end{aligned}
$$

Für $|M| = n$ ist $|M^k| = n^k$. Wenn M unendlich groß ist, dann ist natürlich auch M^k unendlich groß.

Definition (Semantik der Prädikatenlogik)

Eine *Struktur* $\mathcal{A} = (\mathcal{U}_\mathcal{A}, \mathcal{I}_\mathcal{A})$ besteht aus einem Universum $\mathcal{U}_\mathcal{A}$ und einer Interpretation $\mathcal{I}_\mathcal{A}$:

1. Das Universum $\mathcal{U}_\mathcal{A}$ ist eine beliebige, nicht–leere Menge, auch Grundmenge, Grundbereich oder Individuenbereich genannt.

2. Die Interpretation $\mathcal{I}_\mathcal{A}$ ist eine partielle Abbildung mit dem Definitionsbereich $\mathit{def}\,(\mathcal{I}_\mathcal{A})$:

   $$\mathit{def}\,(\mathcal{I}_\mathcal{A}) \subseteq \mathcal{V} \cup \mathcal{F} \cup \mathcal{P}.$$

   $\mathit{def}\,(\mathcal{I}_\mathcal{A})$ ist meist eine echte Teilmenge von $\mathcal{V} \cup \mathcal{F} \cup \mathcal{P}$. Man setzt

   $$\mathcal{I}_\mathcal{A}(\alpha) \;=\; \alpha^\mathcal{A}, \text{ für alle } \alpha \in \mathcal{V} \cup \mathcal{F} \cup \mathcal{P}.$$

   Falls α außerhalb des Definitionsbereichs liegt, so ist $\alpha^\mathcal{A}$ undefiniert.

3. Die Interpretation $\mathcal{I}_\mathcal{A}$ verwendet die kartesischen Produkte $\mathcal{U}_\mathcal{A}^k$ des Universiums:
   - jedem Variablensymbol $X \in \mathcal{V} \cap def(\mathcal{I}_\mathcal{A})$ ist ein Individuum $X^\mathcal{A} \in \mathcal{U}_\mathcal{A}$ zugeordnet.
   - jedem k–stelligen Funktionssymbol $f \in \mathcal{F} \cap def(\mathcal{I}_\mathcal{A})$ ist eine k–stellige Funktion $f^\mathcal{A} : \mathcal{U}_\mathcal{A}^k \to \mathcal{U}_\mathcal{A}$ zugeordnet.
   - jedem k–stelligen Prädikatensymbol $p \in \mathcal{P} \cap def(\mathcal{I}_\mathcal{A})$ ist eine k–stellige Relation $p^\mathcal{A} \subseteq \mathcal{U}_\mathcal{A}^k$ zugeordnet (Prädikat genannt).
4. Die Struktur $\mathcal{A} = (\mathcal{U}_\mathcal{A}, \mathcal{I}_\mathcal{A})$ *paßt* zu einer Formel F, falls $\mathcal{I}_\mathcal{A}$ für alle in F vorkommenden Prädikaten–, Funktions– und Variablensymbole für freie Variablen definiert ist.

Beispiel (Struktur)

Die folgende Struktur $\mathcal{A}$ paßt zur Formel

$$F \;=\; \forall X\; p(X, f(X)) \;\wedge\; q(g(a, Z)).$$

Die Grundmenge von $\mathcal{A}$ und die Interpretation der Prädikaten– und Funktionssymbole seien wie folgt:

$$\begin{aligned}
\mathcal{U}_{\mathcal{A}} &= I\!N_0 \;=\; \{\, 0, 1, 2, \ldots \,\}, \\
p^{\mathcal{A}} &= \{\, (m, n) \in I\!N_0^2 \mid m < n \,\}, \\
q^{\mathcal{A}} &= \{\, (n) \in I\!N_0^1 \mid n \text{ ist eine Primzahl} \,\}, \\
f^{\mathcal{A}}(n) &= n + 1, \;\text{ d.h. } f^{\mathcal{A}} \text{ ist die Nachfolgerfunktion auf } I\!N_0, \\
g^{\mathcal{A}}(n, m) &= n + m, \;\text{ d.h. } g^{\mathcal{A}} \text{ ist die Additionsfunktion auf } I\!N_0.
\end{aligned}$$

Das 0–stellige Funktionssymbol a werde als $a^{\mathcal{A}} = 2$ interpretiert, und die Variablensymbole als $X^{\mathcal{A}} = 23$ und $Z^{\mathcal{A}} = 3$.

In dieser Struktur $\mathcal{A}$ gilt F (die entsprechenden Definitionen der Semantik folgen noch), denn

- die Nachfolgerfunktion $f^{\mathcal{A}}$ paßt zur Kleiner–Relation $p^{\mathcal{A}}$: $p(X, f(X))$ entspricht in $\mathcal{A}$ dem Vergleich $X < X + 1$; und
- $g^{\mathcal{A}}(a^{\mathcal{A}}, Z^{\mathcal{A}}) = 2 + 3 = 5$ ist in der Primzahl–Relation $q^{\mathcal{A}}$.

Ändert man dagegen das Prädikat $q^{\mathcal{A}}$ zum folgenden Prädikat $q^{\mathcal{A}'}$ und läßt den Rest unverändert, so gilt F in der veränderten Struktur $\mathcal{A}'$ nicht, da 5 keine gerade Zahl ist:

$$q^{\mathcal{A}'} \;=\; \{\, (n) \in I\!N_0^1 \mid n \text{ ist eine gerade Zahl} \,\}.$$

Wenn man eine fest vorgegebene Interpretation eines Funktions– oder Prädikatensymbols betrachten will, dann muß man diese durch Hinzunahme geeigneter Axiome zu F erzwingen.

Definition (Semantik der Prädikatenlogik)

Wir betrachten im folgenden Formeln F und zu F passende Strukturen $\mathcal{A}$.

Der Wert $\mathcal{A}(t)$ eines Terms t wird induktiv definiert als:

1. $\mathcal{A}(t) = t^{\mathcal{A}}$, falls $t \in \mathcal{V}$ eine Variable ist.
2. $\mathcal{A}(t) = f^{\mathcal{A}}(\mathcal{A}(t_1), \ldots, \mathcal{A}(t_k))$, falls $t = f(t_1, \ldots, t_k)$ ein komplexer Term ist.

Für $X \in \mathcal{V}$ und $u \in \mathcal{U}_{\mathcal{A}}$ sei $\mathcal{A}_{[X\,|\,u]}$ diejenige Struktur $\mathcal{B}$, welche man aus $\mathcal{A}$ erhält, indem man $X^{\mathcal{A}}$ auf $X^{\mathcal{B}} = u$ ändert und $\mathcal{A}$ ansonsten unverändert läßt:

- für alle $\alpha \in (\mathcal{V} \cup \mathcal{F} \cup \mathcal{P}) \setminus \{X\}$ gilt $\alpha^{\mathcal{B}} = \alpha^{\mathcal{A}}$,
- $X^{\mathcal{B}} = u$ (unabhängig von $X^{\mathcal{A}}$).

Der Wahrheitswert $\mathcal{A}(F)$ einer Formel F wird ebenfalls induktiv definiert:

$$\begin{aligned}
\mathcal{A}(p(t_1,\dots,t_k)) &= \begin{cases} 1, & \text{falls } (\mathcal{A}(t_1),\dots,\mathcal{A}(t_k)) \in p^{\mathcal{A}} \\ 0, & \text{sonst} \end{cases} \\
\mathcal{A}(\neg G) &= \neg \mathcal{A}(G), \\
\mathcal{A}(F_1 \otimes F_2) &= \mathcal{A}(F_1) \otimes \mathcal{A}(F_2), \text{ für } \otimes \in \{\wedge, \vee\}, \\
\mathcal{A}(\forall X G) &= \begin{cases} 1, & \text{falls für alle } u \in \mathcal{U}_{\mathcal{A}} \text{ gilt } \mathcal{A}_{[X|u]}(G) = 1 \\ 0, & \text{sonst} \end{cases} \\
\mathcal{A}(\exists X G) &= \begin{cases} 1, & \text{falls es ein } u \in \mathcal{U}_{\mathcal{A}} \text{ gibt mit } \mathcal{A}_{[X|u]}(G) = 1, \\ 0, & \text{sonst} \end{cases}
\end{aligned}$$

Damit gilt offensichtlich auch $\mathcal{A}(F \to G) = \mathcal{A}(F) \to \mathcal{A}(G)$.

Die Definitionen für die booleschen Junktoren sind exakt analog zur Aussagenlogik. Die Fallunterscheidungen könnte man wie folgt abkürzen:

$$\begin{array}{lcl} \mathcal{A}(p(t_1, \dots, t_k)) & = & ((\mathcal{A}(t_1), \dots, \mathcal{A}(t_k)) \in p^{\mathcal{A}}), \\ \mathcal{A}(\forall X G) & = & \wedge_{u \in \mathcal{U}_{\mathcal{A}}} \mathcal{A}_{[X\,|\,u]}(G), \\ \mathcal{A}(\exists X G) & = & \vee_{u \in \mathcal{U}_{\mathcal{A}}} \mathcal{A}_{[X\,|\,u]}(G). \end{array}$$

1. Ersteres gilt, da der Test $e \in M$ auf Elementschaft in einer Menge einen der Wahrheitswerte 1 oder 0 als Resultat hat.

2. Die – potentiell unendliche – Konjunktion im zweiten Teil ist genau dann 1, wenn für alle $u \in \mathcal{U}_{\mathcal{A}}$ gilt $\mathcal{A}_{[X\,|\,u]}(G) = 1$.

3. Die – potentiell unendliche – Disjunktion im dritten Teil ist genau dann 1, wenn es (mindestens) ein $u \in \mathcal{U}_{\mathcal{A}}$ gibt mit $\mathcal{A}_{[X\,|\,u]}(G) = 1$.

Die Prädikatenlogik verallgemeinert die Aussagenlogik.

Die atomaren Formeln p der Aussagenlogik entsprechen 0–stelligen Prädikatensymbolen. Wegen $\mathcal{U}_{\mathcal{A}}^0 = \{()\}$ gibt es zwei mögliche Relationen $p^{\mathcal{A}} \subseteq \mathcal{U}_{\mathcal{A}}^0$:

- Falls $p^{\mathcal{A}} = \{()\}$ nur das 0–stellige Tupel $()$ enthält, so ist $\mathcal{A}(p()) = 1$.
- Falls $p^{\mathcal{A}} = \{\}$ leer ist, so ist $\mathcal{A}(p()) = 0$.

In der Aussagenlogik gibt es keine höherstelligen Prädikatensymbole, und deswegen braucht man auch keine Variablen– bzw. Funktionssymbole um Terme zu bilden.

Auch die Quantifizierung ist in der Aussagenlogik sinnlos, da es dort keine Variablensymbole gibt.

Definition (Modelle)

1. Eine Struktur $\mathcal{A}$ *erfüllt* eine Formel F, falls $\mathcal{A}$ zu F paßt und $\mathcal{A}(F) = 1$ ist.
2. Dann sagen wir auch, daß die Formel F in $\mathcal{A}$ *gilt*.
3. Dann ist $\mathcal{A}$ ein *Modell* von F, und wir schreiben auch $\mathcal{A} \models F$.

Für allgemeine Formeln der hier vorgestellten Prädikatenlogik der ersten Stufe (PL1) ist die Definition der Modelle als Strukturen $\mathcal{A}$ entscheident.

Im folgenden reicht uns aber eine einfachere Form von Strukturen. Herbrand–Strukturen können alleine durch die Interpretation der Prädikatensymbole charakterisiert werden, und man kann sie als Mengen $I_{\mathcal{A}}$ von Atomen repräsentieren.

Herbrand–Strukturen

In der Praxis kann man bei geschlossenen Formeln F mit Allquantoren Herbrand–Strukturen $\mathcal{A} = (\mathcal{U}_\mathcal{A}, \mathcal{I}_\mathcal{A})$ betrachten:

1. Ihre Grundmenge $\mathcal{U}_\mathcal{A}$ besteht aus den Termen über den Funktionssymbolen von F.

2. Terme t ohne Variablensymbole werden nicht interpretiert; eine Herbrand–Struktur bildet sie auf sich selbst ab: $\mathcal{A}(t) = t$.
   Bekanntlich sind Terme ohne Variablensymbole von der Form
   $t = f(t_1, \dots, t_k)$.

3. Die Variablensymbole werden aufgrund der Quantifizierung durch Terme ohne Variablensymbole ersetzt.

$\mathcal{A}$ entspricht einer Menge $I_\mathcal{A}$ von Atomen ohne Variablensymbole.

Beispiel (Grundmenge $\mathcal{U}_\mathcal{A}$ einer Herbrand–Struktur)

- In der Formel

  $$F = married(a, b) \wedge \forall X\,(student(X) \rightarrow person(X))$$

  gibt es nur 0–stellige Funktionssymbole (Konstanten); diese bilden die Grundmenge $\mathcal{U}_\mathcal{A} = \{\, a, b\,\}$.

- Die Formel

  $$F = p(f(a), g(b))$$

  enthält zwei Konstanten (0–stellige Funktionssymbole) a und b und zwei einstellige Funktionssymbole f und g. Hier ist die Grundmenge $\mathcal{U}_\mathcal{A}$ unendlich groß, denn sie enthält alle Terme, die man daraus bilden kann: $\mathcal{U}_\mathcal{A} = \{\, a, b, f(a), f(b), g(a), g(b), f(f(a)), f(g(a)), \ldots\,\}$.

Terme als Datenstrukturen

In PROLOG kann man auch aussagenlogische Formeln als Terme repräsentieren. Im Aufruf

$$G = boolean\_normalize\_knf(-((a, (b; c))), F)$$

repräsentiert der Term $-((a, (b; c)))$ im ersten Argument z.B. die prädikatenlogische Formel $\neg(a \wedge (b \vee c))$. Der Funktor $-$ repräsentiert die unäre Negation.

- Die *Klammer* um den Term $t = (a, (b; c))$ ist erforderlich, wenn man den Funktor $-$ davor schreibt; man muß also $-(t)$ schreiben. Ohne Klammer würde $-$ als binärer Funktor aufgefaßt, d.h. $-t = -(a, (b; c))$ als Präfix–Notation zur Infix–Notation $a - (b \vee c)$. Dann wäre das Komma der Trenner zwischen den beiden Argumenten von $-$, und es würde nicht als die Konjunktion $\wedge$ aufgefaßt.

- Die Formel G enthält drei Konstanten (0–stellige Funktionssymbole) a, b und c, ein einstelliges Funktionssymbol ”–”, und zwei zweistellige Funktionssymbole ”,” und ”;”.

  Auch hier ist die Grundmenge $\mathcal{U}_\mathcal{A}$ unendlich groß, denn sie enthält alle Terme, die man daraus bilden kann:

  $$\mathcal{U}_\mathcal{A} = \{\, a, b, c, -a, \ldots\, , (a, b), (a; b), -(a, (b; c)), \ldots \}.$$

  Diese Terme bilden *Datenstrukturen* zur Repräsentation von aussagenlogischen Formeln. Die entsprechenden Regeln zum Prädikatensymbol $\mathit{boolean\_normalize\_knf}\,/\,2$ werten diese Terme nicht aus; sie operieren nur symbolisch darauf.

  Ähnlich kann man auch Listen und Bäume als Terme repräsentieren.

Variablenbelegungen

- Für geschlossene Formeln F ist die Belegung der Variablen $X \in \mathcal{V}$ mit Werten $X^{\mathcal{A}} \in \mathcal{U}_{\mathcal{A}}$ irrelevant.
- Für eine quantifizierte Variable X werden sowieso alle Strukturen $\mathcal{A}_{[X\,|\,u]}$ für $u \in \mathcal{U}_{\mathcal{A}}$ untersucht – ohne den Wert $X^{\mathcal{A}}$ zu beachten.
- Der Wert $X^{\mathcal{A}}$ einer Variable X ist nur relevant, falls X mindestens einmal frei in F vorkommt.

  Für die Formel $F = (\forall X\, p(X)\,) \wedge q(X)$ mit je einem gebundenen und einem freien Vorkommen des Variablensymbols X, ist $X^{\mathcal{A}}$ nur für das zweite Vorkommen von X relevant. Es gilt $p^{\mathcal{A}_{[X\,|\,u]}} = p^{\mathcal{A}}$ und

$$\begin{aligned}\mathcal{A}(F) &= \wedge_{u \in \mathcal{U}_{\mathcal{A}}} \mathcal{A}_{[X\,|\,u]}(p(X)) \wedge \mathcal{A}(q(X)) \\ &= \wedge_{u \in \mathcal{U}_{\mathcal{A}}} (u) \in p^{\mathcal{A}} \wedge (X^{\mathcal{A}}) \in q^{\mathcal{A}}.\end{aligned}$$

Beispiel (Herbrand–Struktur, Modell)

Wir betrachten die folgende geschlossene Formel $F = f_1 \wedge f_2 \wedge \forall X\, r$, die besagt, daß a und b Studierdende sind, und daß jeder Studierende eine Person ist:

$$\begin{aligned} f_1 &= student(a), \\ f_2 &= student(b), \\ r &= student(X) \rightarrow person(X). \end{aligned}$$

Wir werden später sehen, daß man das gebundene Variablensymbol X in $\forall X\, r$ umbenennen könnte, z.B. in Y. Dann würde man die Teilformel

$$\forall\, Y \;(\; student(Y) \rightarrow person(Y)\;)$$

erhalten, die – entsprechend unserer Anschauung – zu $\forall X\, r$ äquivalent ist, da man für X und Y sowieso alle Werte der Grundmenge einsetzen kann.

- Wir betrachten eine Herbrand–Struktur $\mathcal{A}$. Diese bildet alle Funktionssymbole auf sich selbst ab. Hier gibt es nur 0–stellige Funktionssymbole (Konstanten); diese bilden die Grundmenge $\mathcal{U}_\mathcal{A}$.

$$\mathcal{U}_\mathcal{A} \; = \; \{ \, a, b \, \} \text{ und } \alpha^\mathcal{A} \; = \; \alpha, \text{ für alle Konstanten } \alpha \in \mathcal{U}_\mathcal{A}.$$

- $\mathcal{A}$ interpretiere die Prädikatensymbole wie folgt:

$$\begin{aligned} \mathit{student}^\mathcal{A} &= \{ \, (a), (b) \, \}, \\ \mathit{person}^\mathcal{A} &= \{ \, (a) \, \}. \end{aligned}$$

Diese Interpretationen könnte man durch eine einzige Menge $I_\mathcal{A}$ von Atomen repräsentieren:

$$I_\mathcal{A} \; = \; \{ \, \mathit{student}(a), \; \mathit{student}(b), \; \mathit{person}(a) \, \}.$$

- Die Struktur $\mathcal{A}$ paßt zu F auch ohne eine Interpretation $X^\mathcal{A}$ des Variablensymbols X anzugeben, da über X quantifiziert wird.

- Dann gilt

$$\begin{array}{lcl}
\mathcal{A}(f_1) & = & ((a) \in student^{\mathcal{A}}) \;=\; 1, \\
\mathcal{A}(f_2) & = & ((b) \in student^{\mathcal{A}}) \;=\; 1, \\
\mathcal{A}(\forall X\; r) & = & \wedge_{u \in \mathcal{U}_{\mathcal{A}}} \left(\, (u) \in student^{\mathcal{A}} \to (u) \in person^{\mathcal{A}} \, \right) \\
 & = & \left(\, (a) \in student^{\mathcal{A}} \to (a) \in person^{\mathcal{A}} \, \right) \wedge \\
 & & \left(\, (b) \in student^{\mathcal{A}} \to (b) \in person^{\mathcal{A}} \, \right) \;=\; 0,
\end{array}$$

  da $(b) \in student^{\mathcal{A}}$ und $(b) \notin person^{\mathcal{A}}$. Da $student$ und $person$ 1–stellige Prädikatensymbole sind, werden in r Tupel $(u) \in \mathcal{U}_{\mathcal{A}}^1$ mit nur einer Komponente untersucht.

- Also ist die untersuchte Struktur $\mathcal{A}$ zwar ein Modell von $f_1 \wedge f_2$, aber kein Modell von $\forall X\; r$, und somit auch kein Modell von F.

- Man würde ein Modell $\mathcal{A}'$ für F erhalten, wenn man $I_{\mathcal{A}}$ erweitert zu

$$I_{\mathcal{A}'} \;=\; I_{\mathcal{A}} \cup \{ \, person(b) \, \}.$$

Beispiel (Herbrand–Struktur, Modell)

Wir betrachten die folgende geschlossene Formel F:

$$
\begin{aligned}
F &= f_1 \wedge f_2 \wedge f_3 \wedge \forall X\, \forall Y\, \forall Z\, r, \\
f_1 &= \mathit{father}(a, b), \\
f_2 &= \mathit{brother}(b, c), \\
f_3 &= \mathit{brother}(b, d), \\
r &= \mathit{father}(X, Y) \wedge \mathit{brother}(Y, Z) \rightarrow \mathit{uncle}(X, Z).
\end{aligned}
$$

Ein Onkel Z von X ist ein Bruder des Vaters Y. Man könnte father und $\mathit{brother}$ auch mittels zweier Prädikate parent und male definieren.

- Wir betrachten wieder eine Herbrand–Struktur $\mathcal{A}$, die die Konstanten (hier die einzigen Funktionssymbole) in F auf sich selbst abbildet:

  $$\mathcal{U}_{\mathcal{A}} = \{\, a, b, c, d \,\} \text{ und } \alpha^{\mathcal{A}} = \alpha, \text{ für alle Konstanten } \alpha \in \mathcal{U}_{\mathcal{A}}.$$

- $\mathcal{A}$ interpretiere die Prädikatensymbole wie folgt:

$$\begin{aligned} \mathit{father}^{\mathcal{A}} &= \{\, (a, b) \,\}, \\ \mathit{brother}^{\mathcal{A}} &= \{\, (b, c), (b, d) \,\}, \\ \mathit{uncle}^{\mathcal{A}} &= \{\, (a, c) \,\}. \end{aligned}$$

  Die zugehörige Menge von Atomen ist

$$I_{\mathcal{A}} = \{\, \mathit{father}(a, b),\ \mathit{brother}(b, c),\ \mathit{brother}(b, d),\ \mathit{uncle}(a, c) \,\}.$$

- Für die Variablenbelegung

$$X^{\mathcal{A}} = a,\ \ Y^{\mathcal{A}} = b,\ \ Z^{\mathcal{A}} = c,$$

  ist $\mathcal{A}$ ein Modell für die Fakten f_1, f_2, f_3 und die Regel r:

$$\begin{aligned} \mathcal{A}(r) &= \mathcal{A}(\mathit{father}(X, Y)) \wedge \mathcal{A}(\mathit{brother}(Y, Z)) \rightarrow \mathcal{A}(\mathit{uncle}(X, Z)) \\ &= (a, b) \in \mathit{father}^{\mathcal{A}} \wedge (b, c) \in \mathit{brother}^{\mathcal{A}} \rightarrow (a, c) \in \mathit{uncle}^{\mathcal{A}} \\ &= 1 \wedge 1 \rightarrow 1 \ = \ 1. \end{aligned}$$

- Die Struktur $\mathcal{A}_{[Z\,|\,d]}$ interpretiert die Prädikatensymbole wie $\mathcal{A}$. Die zugehörige Variablenbelegung interpretiert Z nun als d – während $Z^{\mathcal{A}} \;=\; c$ war:

$$X^{\mathcal{A}_{[Z\,|\,d]}} \;=\; a,\;\; Y^{\mathcal{A}_{[Z\,|\,d]}} \;=\; b,\;\; Z^{\mathcal{A}_{[Z\,|\,d]}} \;=\; d.$$

$\mathcal{A}_{[Z\,|\,d]}$ ist ebenfalls ein Modell für die Fakten f_1, f_2, f_3. Da $(a, d) \notin \mathit{uncle}^{\mathcal{A}}$, ist $\mathcal{A}_{[Z\,|\,d]}$ aber kein Modell für die Regel r:

$$\begin{aligned}
\mathcal{A}_{[Z\,|\,d]}(r) &= \mathcal{A}_{[Z\,|\,d]}(\mathit{father}(X, Y) \wedge \mathit{brother}(Y, Z) \rightarrow \mathit{uncle}(X, Z)) \\
&= (a, b) \in \mathit{father}^{\mathcal{A}} \wedge (b, d) \in \mathit{brother}^{\mathcal{A}} \rightarrow (a, d) \in \mathit{uncle}^{\mathcal{A}} \\
&= 1 \wedge 1 \rightarrow 0 \;=\; 0.
\end{aligned}$$

- Also ist $\mathcal{A}$ auch kein Modell für die quantifizierte Regel

  $$\forall X\ \forall Y\ \forall Z\ r,$$

  denn es gibt eine Variablenbelegung, für die diese nicht gilt. Somit ist $\mathcal{A}$ auch kein Modell für die komplette Formel

  $$F \;=\; f_1 \wedge f_2 \wedge f_3 \wedge \forall X\ \forall Y\ \forall Z\ r.$$

- Wir erhalten ein Modell $\mathcal{A}'$ von F, wenn wir $I_{\mathcal{A}}$ um $uncle(a, d)$ erweitern bzw. die Interpretation von $uncle$ zu

  $$uncle^{\mathcal{A}'} \;=\; uncle^{\mathcal{A}} \cup \{\, (a, d)\, \}$$

  und sonst alles wie in $\mathcal{A}$ belassen.

Logische Folgerung, Erfüllbarkeit, Gültigkeit

Seien F und G prädikatenlogische Formeln.

1. G *folgt logisch* aus F, falls jedes Modell $\mathcal{A}$ von F auch ein Modell von G ist. Dann schreiben wir $F \models G$.

2. F ist *erfüllbar*, falls es ein Modell von F gibt.

3. F ist *gültig*, falls jede zu F passende Struktur ein Modell von F ist.

Die logische Folgerung $F \models G$ ist äquivalent zur Unerfüllbarkeit von $F \wedge \neg G$ bzw. zur Gültigkeit von $\neg (F \wedge \neg G)$.

Man kann die bekannten Begriffe auf Formelmengen F ausdehnen.
Eine endliche Formelmenge $F = \{ f_1, \ldots , f_n \}$ ist dabei äquivalent zu einer Konjunktion $f_1 \wedge \ldots \wedge f_n$.

2.2 Normalformen

Wir werden im folgenden sehen, daß man eine prädikatenlogische Formel *äquivalent* umformen kann, so daß alle Quantoren am Anfang der Formel stehen (*Pränexform*).

Man kann sogar noch – mittels Skolemfunktionen – alle Existenzquantoren entfernen (*Skolemform*). Die Skolemform ist genau dann erfüllbar, wenn die Ausgangsformel erfüllbar ist.

Die Skolemform kann man dann – wie in der Aussagenlogik – auf *Klauselform* bringen (KNF). Darauf können wir dann eine verallgemeinerte Resolutionsmethode anwenden.

Definition (Äquivalenz)

Zwei prädikatenlogische Formeln F und G heißen *äquivalent*,
falls für alle sowohl zu F als auch zu G passenden Strukturen $\mathcal{A}$ gilt:

$$\mathcal{A}(F) \;=\; \mathcal{A}(G).$$

Dann schreiben wir $F \equiv G$.

Satz (Äquivalenz)

1. Negation vertauscht die Quantoren $\forall$ und $\exists$:

$$\begin{aligned} \neg\,\forall X\, F \quad &\equiv \quad \exists X\, \neg\, F, \\ \neg\,\exists X\, F \quad &\equiv \quad \forall X\, \neg\, F. \end{aligned}$$

   Nicht alle Strukturen passen zu F ist z.B. äquivalent zu
   es gibt Strukturen, die nicht zu F passen.

2. Falls X in G nicht frei vorkommt:

$$(\forall X\, F \otimes G) \;\equiv\; \forall X\, (F \otimes G),$$
$$(\exists X\, F \otimes G) \;\equiv\; \exists X\, (F \otimes G),$$

für einen Junktor $\otimes \in \{\, \wedge, \vee \,\}$.

3. Ausklammern von Quantoren:

$$(\forall X\, F \;\wedge\; \forall X\, G) \;\equiv\; \forall X\, (F \wedge G),$$
$$(\exists X\, F \;\vee\; \exists X\, G) \;\equiv\; \exists X\, (F \vee G).$$

4. Vertauschen gleichartiger Quantoren:

$$\forall X\, \forall Y\, F \;\equiv\; \forall Y\, \forall X\, F,$$
$$\exists X\, \exists Y\, F \;\equiv\; \exists Y\, \exists X\, F.$$

Das Ausklammern von Quantoren funktioniert nur für die beiden Paare

- $\forall$ und $\wedge$ bzw.
- $\exists$ und $\vee$.

Dagegen sind die folgenden Formeln im allgemeinen nicht äquivalent:

$$
\begin{aligned}
(\forall X F \;\vee\; \forall X G) &\not\equiv \forall X\, (F \vee G),\\
(\exists X F \;\wedge\; \exists X G) &\not\equiv \exists X\, (F \wedge G).
\end{aligned}
$$

Beispiel (Äquivalenz)

1. Nach 1. gilt $\neg \forall X\, r(X, Y) \;\equiv\; \exists X \neg r(X, Y)$.

2. Da X und Y nicht frei in $\neg\, q(Z)$ vorkommen, kann man nach 2. die Quantoren $\forall X$ und $\exists Y$ sich auf beide Formeln erstrecken lassen:

$$\begin{aligned} &\forall X\, \exists Y\, p(X, g(Y, f(X))) \vee \neg\, q(Z) \\ &\quad \equiv \forall X\, \exists Y\, (p(X, g(Y, f(X))) \vee \neg\, q(Z)). \end{aligned}$$

   Da X frei in $q(X)$ vorkommt, kann man den Quantor $\forall X$ nicht sich auf beide Formeln erstrecken lassen:

$$\forall X\, p(X) \wedge q(X) \;\not\equiv\; \forall X\, (\, p(X) \wedge q(X)\,).$$

3. Für Konjunktionen kann man nach 3. universelle Quantoren ausklammern:

$$\forall X\, p(X) \wedge \forall X\, q(X) \;\equiv\; \forall X\, (\, p(X) \wedge q(X)\,).$$

4. Dagegen kann man unterschiedliche Quantoren im Allgemeinen nicht vertauschen:

$$\forall X \, \exists Y \, F \quad \not\equiv \quad \exists Y \, \forall X \, F.$$

Für $F = Y > X$ (Infix–Notation) gilt: wenn man $>$ als die Größer–Relation auf den natürlichen Zahlen interpretiert, d.h.

$$\begin{aligned} \mathcal{U}_{\mathcal{A}} &= \mathbb{N}_0 = \{\, 0, 1, 2, \ldots \,\}, \\ >^{\mathcal{A}} &= \{\, (m, n) \in \mathbb{N}_0^2 \mid m > n \,\}, \end{aligned}$$

dann erfüllt $\mathcal{A}$ die Formel $\forall X \, \exists Y \, F$, da es für alle $X \in \mathbb{N}_0$ eine größere Zahl $Y \in \mathbb{N}_0$ gibt, während $\exists Y \, \forall X \, F$ nicht erfüllt ist, da es kein $Y \in \mathbb{N}_0$ gibt, das größer ist als alle Zahlen $X \in \mathbb{N}_0$:

$$\mathcal{A}(\forall X \, \exists Y \, F) \; = \; 1 \; \neq \; 0 \; = \; \mathcal{A}(\exists Y \, \forall X \, F).$$

Äquivalenz vs. Implikation

1. $F \equiv G$ gilt genau dann,
   wenn $F \rightarrow G$ und $G \rightarrow F$ gültig (Tautologien) sind.

2. Aus $F \equiv G$ folgt, daß $F \rightarrow G$ und $G \rightarrow F$ gültig sind.

Für die nicht geltenden Äquivalenzen

$$
\begin{array}{rcl}
(\forall X\, F \vee \forall X\, G) & \not\equiv & \forall X\, (F \vee G), \\
(\exists X\, F \wedge \exists X\, G) & \not\equiv & \exists X\, (F \wedge G).
\end{array}
$$

sind zumindest die folgenden Implikationen gültig:

$$
\begin{array}{rcl}
(\forall X\, F \vee \forall X\, G) & \rightarrow & \forall X\, (F \vee G), \\
(\exists X\, F \wedge \exists X\, G) & \leftarrow & \exists X\, (F \wedge G).
\end{array}
$$

Definition (Substitution, Ersetzung von Variablen)

1. Für eine Variable X und einen Term t wird $[X|t]$ als *Substitution* bezeichnet. Eine Folge

   $$\theta = [X_1|t_1] \ldots [X_n|t_n]$$

   von Substitutionen wird auch als Substitution bezeichnet.

2. Man kann eine Substitution auf eine Formel F anwenden. Dann bezeichnet $F[X|t]$ die Formel, die man aus F erhält, indem man jedes freie Vorkommen von X durch t ersetzt.

3. Für $\theta = [X_1|t_1][X_2|t_2] \ldots [X_n|t_n]$ bezeichnet $F\theta$ die Formel, die man durch sukzessive Anwendung der Substitutionen $[X_i|t_i]$ erhält:

   $$F\theta \;=\; (((F\,[X_1|t_1])[X_2|t_2])\ldots)[X_n|t_n],$$

Beispiel (Substitution)

Wir betrachten die folgende Formel F und die Substitution θ:

$$\begin{aligned} F &= (\forall X\; p(X, f(X), g(Y)) \vee q(X)), \\ \theta &= [X|h(Z)]\,[Y|U]. \end{aligned}$$

In der Formel sind die ersten beiden Vorkommen von X gebunden und werden nicht durch $h(Z)$ ersetzt.

Man erhält $F\theta$ durch sukzessive Anwendung der Substitutionen $[X|h(Z)]$ und $[Y|U]$:

$$\begin{aligned} F[X|h(Z)] &= (\forall X\; p(X, f(X), g(Y)) \vee q(h(Z))), \\ F[X|h(Z)]\,[Y|U] &= (\forall X p(X, f(X), g(U)) \vee q(h(Z))). \end{aligned}$$

Gebundene Umbenennung

Ist $F \; = \; QX \; G$ eine Formel mit einem Quantor $Q \in \{ \exists, \forall \}$ und X' eine Variable, die in G nicht vorkommt, dann gilt

$$F \; = \; QX \; G \; \equiv \; QX' \, G \, [X|X'].$$

Beispiel (Gebundene Umbenennung)

Für $G = p(X, g(Y, f(X)))$ gilt

$$\exists Y \; G \quad \equiv \quad \exists Y' \, G \, [Y|Y'] \; = \; \exists Y' \, p(X, g(Y', f(X))).$$

Beispiel (Gebundene Umbenennung)

1. Die folgenden offenen Formeln sind nicht äquivalent:

$$F = p(X),$$
$$G = p(Y).$$

Für die Struktur $\mathcal{A}$ mit der Grundmenge $\mathcal{U}_\mathcal{A} = \{ 1, 2 \}$ und $p^\mathcal{A} = \{ (1) \}$ und der Variablenbelegung $X^\mathcal{A} = 1,\ Y^\mathcal{A} = 2$ gilt

$$\mathcal{A}(F) = (1) \in p^\mathcal{A} = 1 \neq 0 = (2) \in p^\mathcal{A} = \mathcal{A}(G).$$

2. Die folgenden quantifizierten Formeln sind dagegen äquivalent:

$$F = \forall X\, p(X),$$
$$G = \forall Y\, p(Y).$$

Es gilt $G = \forall Y\, p(X)\,[X|Y]$.

Für gebundene Vorkommen kann man das Variablensymbol umbenennen.

Beispiel (Ausklammern von Quantoren)

1. Die folgenden Formeln sind bekanntermaßen im allgemeinen nicht äquivalent:

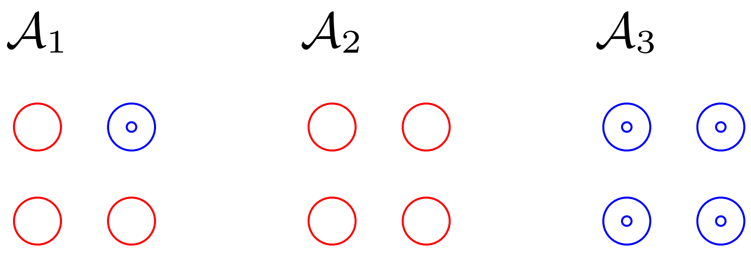
$$(\forall X F \;\vee\; \forall X G) \quad \not\equiv \quad \forall X\,(F \vee G),$$
$$(\exists X F \;\wedge\; \exists X G) \quad \not\equiv \quad \exists X\,(F \wedge G).$$

2. Sei $F = rot(X)$ und $G = blau(X)$.
   - Dann besagt $(\forall X F \;\vee\; \forall X G)$: entweder alle Kugeln sind rot, oder alle Kugeln sind blau.
   - Dagegen besagt $\forall X\,(F \vee G)$, daß jede Kugel entweder rot oder blau ist. Aber es kann sowohl rote als auch blaue Kugeln geben.

In den folgenden Strukturen $\mathcal{A}_i$ mit einfarbigen Kugeln unterscheiden sich die Formeln F_j und G_k:

$$F_1 = (\forall X\, rot(X) \vee \forall X\, blau(X)) \quad \not\equiv \quad F_2 = \forall X\, (rot(X) \vee blau(X)),$$
$$G_1 = (\exists X\, rot(X) \wedge \exists X\, blau(X)) \quad \not\equiv \quad G_2 = \exists X\, (rot(X) \wedge blau(X)).$$

Die Strukturen $\mathcal{A}_i$ erfüllen G_2 nicht, da es keine Kugeln gibt, die gleichzeitig rot und blau sind; alle drei Strukturen erfüllen F_2.



$\mathcal{A}_1$ erfüllt G_1, aber nicht F_1, denn jede Kugel ist entweder rot oder blau (mit Punkt in der Mitte), und es gibt rote und blaue Kugeln.

$\mathcal{A}_2$ und $\mathcal{A}_3$ erfüllen F_1, aber nicht G_1, denn in beiden Strukturen haben alle Kugeln dieselbe Farbe, rot bzw. blau.

Beispiel (Ausklammern nach gebundener Umbenennung)

1. Für ein frisches Variablensymbol X', das nicht in $(\forall X F \vee \forall X G)$ vorkommt, gilt:

$$(\forall X F \vee \forall X G) \equiv (\forall X' F[X|X'] \vee \forall X G)$$
$$\equiv \forall X' (F[X|X'] \vee \forall X G) \equiv \forall X' \forall X (F[X|X'] \vee G).$$

   Also gilt z.B.

$$(\forall X\, rot(X) \vee \forall X\, blau(X)) \equiv \forall X' \forall X\, (rot(X') \vee blau(X)).$$

   Prinzipiell können Kugeln mehrfarbig sein. Wenn es aber eine rote Kugel r gäbe, die nicht auch blau ist, und eine blaue Kugel b, die nicht auch rot ist, dann wäre $\mathcal{A}_{[X'|b][X|r]}(rot(X') \vee blau(X)) = 0$.

2. Für ein frisches Variablensymbol X', das nicht in $(\exists X F \wedge \exists X G)$ vorkommt, gilt analog auch:

$$(\exists X F \wedge \exists X G) \equiv \exists X' \exists X\, (F[X|X'] \wedge G).$$

3. Auch aus einer Implikation kann man nach gebundener Umbenennung ausklammern:

$$\begin{aligned} F &= \forall X \, (\, p(X) \to \exists X \, q(X) \,) \\ &\equiv \forall X \, (\, p(X) \to \exists X' \, q(X') \,) \\ &\equiv \forall X \, \exists X' \, (\, p(X) \to q(X') \,). \end{aligned}$$

   Hier mußte X gebunden in X' umbenannt werden, da das Vorkommen von X in $p(X)$ durch $\forall X$ und das Vorkommen von X in $q(X)$ durch $\exists X$ gebunden waren.

   Nach dem Vorziehen des Quantors $\exists X'$ erstrecken sich nun beide Quantoren auf $p(X) \to q(X')$, also auf beide Teilformeln.

Definition (BPF, bereinigte Pränexform)

1. Eine Formel F heißt *bereinigt*,
   - wenn es keine Variable X gibt, die in F sowohl gebunden als auch frei vorkommt, und
   - wenn hinter allen vorkommenden Quantoren verschiedene Variablen stehen.
2. Eine Formel F heißt Pränex oder in *Pränexform*, falls sie von der Form

   $$Q_1 X_1 \; Q_2 X_2 \; \ldots Q_n X_n \; G$$

   ist, mit Quantoren $Q_i \in \{\, \exists, \forall \,\}$ und Variablen X_i, und falls ferner G quantorenfrei ist.
3. Eine bereinigte Formel in Pränexform ist in BPF.

Satz (BPF)

Für jede Formel F gibt es eine äquivalente Formel G in BPF.

Beispiel (BPF)

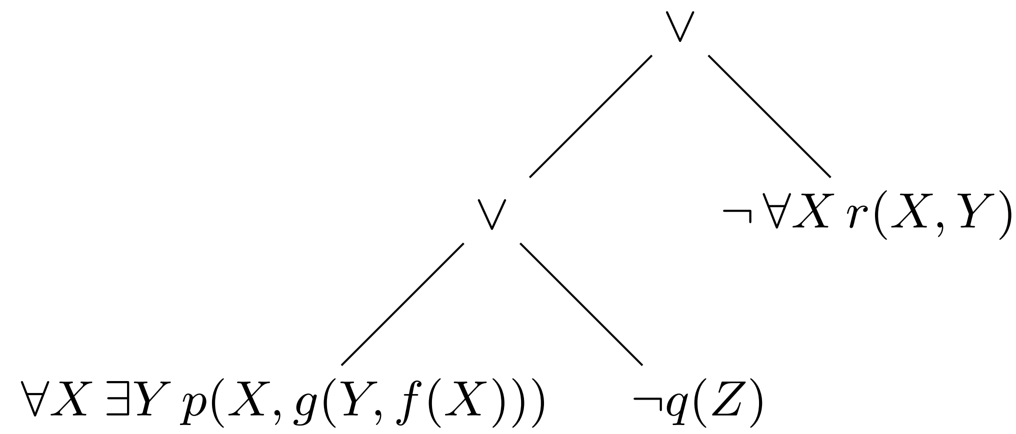
Die Formel F, eine Disjunktion zweier Teilformeln,

$$F \quad = \quad (\forall X\, \exists Y\; p(X, g(Y, f(X))) \;\vee\; \neg\, q(Z)) \;\vee\; \neg \forall X\, r(X, Y)$$

ist äquivalent zur folgenden BPF:

$$F \quad \equiv \quad \forall X\, \exists Y'\, \exists X'\, (\, p(X, g(Y', f(X))) \;\vee\; \neg\, q(Z) \;\vee\; \neg\, r(X', Y)\,).$$

Im folgenden zeigen wir wie man die beiden Disjunktionsglieder von F umformen kann.



Aus

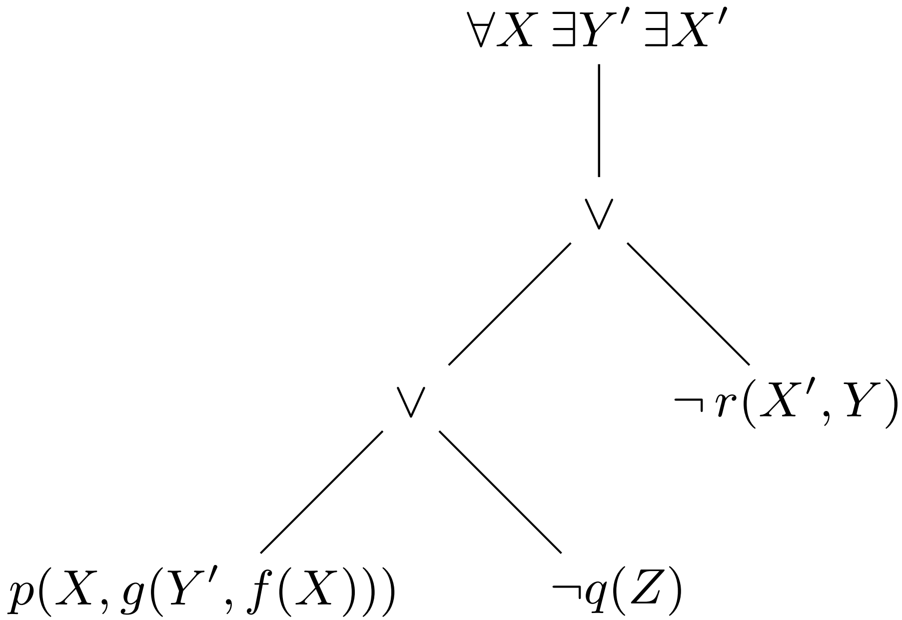
$$\begin{aligned} &\neg\,\forall X\, r(X,Y) \\ &\qquad \equiv\ \exists X\, \neg\, r(X,Y), \\ &\forall X\, \exists Y\, p(X, g(Y, f(X)))\ \vee\ \neg\, q(Z) \\ &\qquad \equiv\ \forall X\, \exists Y\, (p(X, g(Y, f(X)))\ \vee\ \neg\, q(Z)), \end{aligned}$$

folgt:

$$F \ \equiv\ \forall X\, \exists Y'\, \exists X'\, (\, p(X, g(Y', f(X)))\ \vee\ \neg\, q(Z)\ \vee\ \neg\, r(X', Y)\,).$$

Die Quantoren $\forall X$ und $\exists Y$ beziehen sich nicht auf $\neg\,\forall X\, r(X,Y)$.
Deswegen benennen wir Y vorne in Y' gebunden um und X hinten in X'.
Danach können alle Quantoren ganz nach vorne gezogen werden.

Die enstandene Formel in BPF hat immer noch 2 freie Vorkommen von Variablensymbolen, nämlich Y und Z.



Die gebundenen Umbenennungen waren erforderlich,

- da X einmal universell und einmal existentiell quantifiziert war, und
- da sich der Quantor $\exists Y'$ nicht auf $\neg \, r(X', Y)$ beziehen soll.

Beispiel (BPF)

In der Implikation

$$F \; = \; (\exists\, Y \; \underbrace{(father(X,Y) \land brother(Y,Z))}_{F_1}\,) \to \underbrace{uncle(X,Z)}_{F_2}$$

kommen der *Existenzquantor* und das quantifizierte Variablensymbol Y nur in der *Vorbedingung* $(\exists Y \; F_1)$ vor:

$$\begin{aligned} F \;&\equiv\; \neg\,(\exists\, Y \; F_1) \;\lor\; F_2 \;\equiv\; \forall\, Y \,\neg\, F_1 \;\lor\; F_2 \\ &\equiv\; \forall\, Y \,(\neg\, F_1 \;\lor\; F_2) \;\equiv\; \forall Y \,(F_1 \to F_2) \end{aligned}$$

Da Y nicht in der Folgerung F_2 vorkommt, kann man in der zweiten Zeile so umformen, daß sich der Allquantor auf beide Disjunktionsglieder erstreckt – und dann auf die *komplette Implikation* in BPF:

$$F \;\equiv\; \forall Y \,(father(X,Y) \land brother(Y,Z) \to uncle(X,Z)).$$

Exkurs: Existenzquantor im Regelkopf

Ein Existenzquantor $\exists\, Y$ im Kopf einer BPF–Regel,

$$F \;=\; \forall X\, (\, kugel(X) \rightarrow \exists\, Y\, farbe(X, Y)\,),$$

entspricht dagegen einem vor der Regel stehenden Existenzquantor $\exists\, Y$:

$$F \;\equiv\; \forall X\, \exists Y\, (\, kugel(X) \rightarrow farbe(X, Y)\,).$$

Für eine vorgegebene Struktur $\mathcal{A}$ ergibt er eine Disjunktion im Regelkopf:

$$\mathcal{A}(F) \;=\; \wedge_{u \in \mathcal{U}_{\mathcal{A}}}(\,(u) \in kugel^{\mathcal{A}} \rightarrow \vee_{v \in \mathcal{U}_{\mathcal{A}}}(u, v) \in farbe^{\mathcal{A}}\,).$$

Da die Grundmenge $\mathcal{U} = \mathcal{U}_{\mathcal{A}}$ für Herbrand–Strukturen unabhängig von $\mathcal{A}$ ist, könnte man F praktisch wie eine Regel mit einem disjunktiven Kopf auffassen:

$$F \;\hat{=}\; \forall X\, (\, kugel(X) \rightarrow \vee_{v \in \mathcal{U}}\, farbe(X, v)\,).$$

In der Praxis möchte man allerdings für $v \in \mathcal{U}$ nur Farben zulassen, etwa rot, $grün$, $blau$. Dann meint man eigentlich die folgende Formel:

$$\forall X\ (\ kugel(X)\ \rightarrow$$
$$farbe(X, rot) \vee farbe(X, grün) \vee farbe(X, blau)\ ).$$

Diese könnte man wieder mit einem Existenzquantor $\exists\, Y$ im Regelkopf ausdrücken,

$$G\ =\ \forall X\ (\ kugel(X) \rightarrow \exists\ Y\ (farbe(\,Y\,) \wedge farbe(X\,,\ Y))\ ).$$

Man könnte dazu ein 1–stelliges Prädikat $farbe$, das die zugelassenen Farben angibt, mit Hilfe geeigneter Formeln definieren.

Definition (Skolemform)

Zu einer Formel F in BPF erhält man die Skolemform durch Elimination der Existenzquantoren nach folgendem Schema:

1. Wir betrachten den ersten Existenzquantor in F:

   $$F = \forall X_1 \forall X_2 \ldots \forall X_n \exists X \, G.$$

2. Sei f ein neues bisher in F nicht vorkommendes n–stelliges Funktionssymbol. Dann setzen wir

   $$F' = \forall X_1 \forall X_2 \ldots \forall X_n \, G \, [X | f(X_1, \ldots, X_n)].$$

3. Ebenso eliminieren wir schrittweise nach demselben Schema alle weiteren Existenzquantoren in F'.

4. Sobald die so erhaltene Formel F'' keine Existenzquantoren mehr enthält, ist sie in *Skolemform*: $F'' = \forall X_1 \forall X_2 \ldots \forall X_k \, F^*$, mit quantorenfreier Formel F^*, die wir Matrix nennen.

Beispiel (Skolemform)

Wir betrachten die folgende Formel F in BPF:

$$\begin{array}{lcllllll}
F & = & \exists X & \forall Y & \exists Z & \forall V & \exists W & \neg p(X, Y, Z, V, W) \\
F' & = & & \forall Y & \exists Z & \forall V & \exists W & \neg p(a, Y, Z, V, W) \\
F'' & = & & \forall Y & & \forall V & \exists W & \neg p(a, Y, f(Y), V, W) \\
F''' & = & & \forall Y & & \forall V & & \neg p(a, Y, f(Y), V, g(Y, V)).
\end{array}$$

Wir ersetzen die existentiell quantifizierten Variablen durch Terme:

- Zuerst ersetzen wir X durch eine frische Konstante a.
- Dann ersetzen wir Z im Scope von Y durch einen Term $f(Y)$.
- Schließlich ersetzen wir W im Scope von Y und V durch einen Term $g(Y, V)$.

Beispiel (Skolemform)

Das 3–stellige Prädikatensymbol p steht für eine Gruppenoperation; wir schreiben $p(X, Y, Z)$ anstelle von $X \circ Y = Z$ (Produkt).

Die folgende Formel beschreibt die Gruppenaxiome zur Existenz eines links–neutralen Elements (X) und zur Existenz von Links–Inversen (Z):

$$F \;=\; \exists X\, (\, \forall Y\, p(X, Y, Y) \;\wedge\; \forall Y\, \exists Z\, p(Z, Y, X)\,).$$

Wir bringen F zuerst in BPF, und dann mittels zweier Skolemfunktionen e (0–stellig) und i (1–stellig) in Skolemform G:

$$\begin{aligned} F &\equiv \exists X\, \forall Y\, \exists Z\, (\, p(X, Y, Y) \;\wedge\; p(Z, Y, X)\,), \\ G &= \forall Y\, (\, p(e, Y, Y) \;\wedge\; p(i(Y), Y, e)\,). \end{aligned}$$

Bei der Erzeugung der BPF wurde der bekannte Satz zur Verschiebung der Quantoren $\forall Y$ und $\exists Z$ angewendet.

Es können auch Formeln mit freien Variablen skolemisiert werden.
Diese freien Variablen bleiben dann auch in der Skolemform frei.

Beispiel (Skolemform)

Eine Skolemform der Formel

$$F \;=\; \forall X \; \exists Y \; p(X, Y, Z)$$

mit der freien Variable Z ist

$$G \;=\; \forall X \; p(X, i(X), Z).$$

Die Skolemform G enthält ebenfalls die freie Variable Z.

Definition (Erfüllbarkeitsäquivalenz)

Zwei prädikatenlogische Formeln F und G heißen *erfüllbarkeitsäquivalent*, wenn gilt: F ist erfüllbar, genau dann wenn G erfüllbar ist.

Die Formeln F und G müssen nicht dieselben Modelle haben. Die Erfüllbarkeitsäquivalenz bedeutet aber, daß F und G entweder beide ein Modell besitzen, oder daß keine der beiden Formeln ein Modell besitzt.

Sowohl die Skolemisierung als auch die existenzielle Quantifizierung freier Variablen liefert erfüllbarkeitsäquivalente Formeln.

Beispiel (Erfüllbarkeitsäquivalenz)

Indem man die freie Variable X in F existentiell quantifiziert, erhält man eine erfüllbarkeitsäquivalente Formel G:

$$
\begin{aligned}
F &= \forall Y\, p(X, Y, Y) \wedge \forall Y\, \exists Z\, p(Z, Y, X), \\
G &= \exists X\, (\, \forall Y\, p(X, Y, Y) \wedge \forall Y\, \exists Z\, p(Z, Y, X)\,).
\end{aligned}
$$

Satz (Skolemform)

Eine Formel F in BPF ist genau dann erfüllbar, wenn ihre Skolemform erfüllbar ist.

Der folgende Beweis zeigt, daß die Eliminationsschritte zu den Existenzquantoren nichts an der Erfüllbarkeit ändern.

Beweis:

Wir betrachten den Eliminationsschritt $F \mapsto F'$ für den ersten Existenzquantor:

$$F \;=\; \forall X_1 \, \forall X_2 \, \ldots \forall X_n \, \exists X \, G.$$

1. Annahme: F' ist erfüllbar mit einer passenden Struktur $\mathcal{A}'$, d.h. $\mathcal{A}'(F') = 1$. Dann paßt $\mathcal{A}'$ auch zu F und es gilt

   $$\forall u_1, \ldots, u_n \in \mathcal{U}_{\mathcal{A}'} : \\ \mathcal{A}'_{[X_1|u_1] \ldots [X_n|u_n]}(G[X|f(X_1, \ldots, X_n)]) = 1.$$

Daras folgt

$$\forall u_1, \ldots, u_n \in \mathcal{U}_{\mathcal{A}'} :$$
$$\mathcal{A}'_{[X_1|u_1]\ldots[X_n|u_n][X|f^{\mathcal{A}'}(u_1,\ldots,u_n)]}(G) = 1.$$

Daraus folgt

$$\forall u_1, \ldots, u_n \in \mathcal{U}_{\mathcal{A}'} : \exists u = f^{\mathcal{A}'}(u_1, \ldots, u_n) \in \mathcal{U}_{\mathcal{A}'} :$$
$$\mathcal{A}'_{[X_1|u_1]\ldots[X_n|u_n][X|u]}(G) = 1.$$

Daraus folgt

$$\mathcal{A}'(\forall X_1 \ \ldots \ \forall X_n \ \exists X \ G) = 1.$$

Also ist $\mathcal{A}'$ auch ein Modell für F.

2. Annahme: F ist erfüllbar mit einer passenden Struktur $\mathcal{A}$,
   d.h. $\mathcal{A}(F) = 1$.

   Wir können annehmen, daß $I_{\mathcal{A}}$ auf keinen anderen als den in F vorkommenden Funktionssymbolen, Prädikatensymbolen und freien Variablen definiert ist.

   Dann gilt:

   $$\forall u_1, \dots, u_n \in \mathcal{U}_{\mathcal{A}} : \; \exists u \in \mathcal{U}_{\mathcal{A}} : \\ \mathcal{A}_{[X_1|u_1]\dots[X_n|u_n][X|u]}(G) = 1.$$

   Wir erweitern nun die Struktur $\mathcal{A}$ zu einer neuen Struktur $\mathcal{A}'$ durch Definition einer Funktion $f^{\mathcal{A}'}$ mit

   $$f^{\mathcal{A}'}(u_1, \dots, u_n) = u$$

   durch Verwendung des *Auswahlaxioms*. Es gilt $\mathcal{U}_{\mathcal{A}'} = \mathcal{U}_{\mathcal{A}}$.

Daraus folgt

$$\forall u_1, \ldots, u_n \in \mathcal{U}_{\mathcal{A}'} : \\ \mathcal{A}'_{[X_1|u_1]\ldots[X_n|u_n][X|f^{\mathcal{A}'}(u_1,\ldots,u_n)]}(G) = 1.$$

Daraus folgt

$$\forall u_1, \ldots, u_n \in \mathcal{U}_{\mathcal{A}'} : \\ \mathcal{A}'_{[X_1|u_1]\ldots[X_n|u_n]}(G[X|f(X_1, \ldots, X_n)]) = 1.$$

Daraus folgt

$$\mathcal{A}'(\forall X_1 \ \ldots \ \forall X_n \, G[X|f(X_1, \ldots, X_n)]) = 1.$$

Also ist $\mathcal{A}'$ ein Modell für F. $\square$

Zusammenfassung (BPF, Skolemform)

1. Für jede Formel F gibt es eine äquivalente Formel G in BPF.
2. F und G haben genau dieselben Modelle.
3. Für jede Formel F gibt es eine Formel H in Skolemform.
   - F und H müssen nicht dieselben Modelle haben.
   - Insbesondere unterscheiden sich die Modelle in der Regel aufgrund der Skolemfunktionen in H.
   - Die Skolemform H wird durch Elimination der Existenzquantoren aus der BPF G konstruiert.
4. Eine Formel F ist genau dann erfüllbar, wenn ihre Skolemform H erfüllbar ist.

Klauselform

Aus einer Formel F kann man über die Skolemform eine Klauselmenge M gewinnen. Eine Klausel ist eine Disjunktion prädikatenlogischer Literale.

1. Durch systematisches Umbenennen der gebundenen Variablen in F bilden wir eine neue, bereinigte Formel F_1, die zu F äquivalent ist.

2. Seien $Y_1, \ldots, Y_n$ die in F_1 vorkommenden freien Variablen. Dann enthält $F_2 = \exists Y_1 \ldots \exists Y_n \, F_1$ keine ungebundenen Variablen mehr, und F_2 ist erfüllbarkeitsäquivalent zu F_1.

3. Sei $F_3 = \forall X_1 \ldots \forall X_m \, G_3$ eine Skolemform zu F_2 mit der (quantorenfreien) Matrix G_3. Dann sind alle Variablen in F_3 universell quantifiziert, und F_3 ist erfüllbarkeitsäquivalent zu F_2.

4. Man kann die Matrix G_3 in eine KNF $G_4 = \wedge_{i=1}^{k} \beta_i$ mit Klauseln β_i umformen. Dann ist $F_4 = \forall X_1 \ldots \forall X_m \, G_4$ zu F_3 äquivalent.

Insgesamt gilt:

- F_4 ist erfüllbarkeitsäquivalent zu F, und
- alle Variablen in F_4 sind universell quantifiziert.

Die Klauselmenge $M = \{ \beta_1, \ldots, \beta_k \}$ zu F_4 spielt später bei Inferenzmethoden eine entscheidende Rolle. Man kann mittels der *Resolutionsmethode* genau dann die leere Klausel aus M ableiten, wenn F unerfüllbar ist.

Die Klauselmenge M enthält keine Quantoren mehr; man nimmt implizit an, daß alle Variablen universell quantifiziert sind. Deswegen ist hier die Existenzquantifizierung der freien Variablen in Schritt 2 vor der Skolemisierung in Schritt 3 erforderlich.

Beispiel (Klauselform)

Wir können die Formel $\forall Y\, p(X, Y, Y) \wedge \forall Y\, \exists Z\, p(Z, Y, X)$ zunächst durch Ausklammern des Allquantors umformen in

$$F \;=\; \forall Y\, (\, p(X, Y, Y) \wedge \exists Z\, p(Z, Y, X)\,).$$

Dann bestimmen wir die Klauselform zu F. $F_1 = F$ ist bereits eine bereinigte Formel.

- Durch Hinzufügen eines Existenzquantors für X erhalten wir die geschlossene Formel $F_2 = \exists X\, \forall Y\, (\, p(X, Y, Y) \wedge \exists Z\, p(Z, Y, X)\,)$.
- Durch Vorziehen des Existenzquantors für Z berechnen wir die Pränexform $\exists X\, \forall Y\, \exists Z\, (\, p(X, Y, Y) \wedge p(Z, Y, X)\,)$ mit drei Quantoren und daraus die Skolemform F_3, deren Matrix bereits eine KNF ist:

  $$F_3 = \forall Y\, (\, p(e, Y, Y) \wedge p(i(Y), Y, e)\,).$$

Die freie Variable X aus F wurde existenzquantifiziert und skolemisert; letztendlich wurde sie in F_3 durch die Skolemkonstante e ersetzt.

Wir erhalten $F_4 = F_3$ und die entsprechende Klauselmenge

$$M \ = \ \{\, p(e, Y, Y),\ p(i(Y), Y, e)\, \}.$$

Wenn wir den Allquantor am Anfang nicht ausklammern, dann erhalten wir die andere Pränexform $\exists X\, \forall Y\, \forall Y'\, \exists Z\, (\, p(X, Y, Y) \ \wedge\ p(Z, Y', X)\,)$ mit vier Quantoren und daraus die Skolemform

$$F_3' = \forall Y\, \forall Y'\, (\, p(e, Y, Y) \ \wedge\ p(i(Y, Y'), Y', e)\,),$$

deren Matrix ebenfalls in KNF ist. Die entsprechende Klauselmenge ist

$$M' \ = \ \{\, p(e, Y, Y),\ p(i(Y, Y'), Y', e)\, \}.$$

2.3 Unentscheidbarkeit

Es gibt prädikatenlogische Formeln F, die zwar erfüllbar sind, jedoch nur Modelle $\mathcal{A} = (\mathcal{U}_\mathcal{A}, \mathcal{I}_\mathcal{A})$ mit unendlicher Grundmenge $\mathcal{U}_\mathcal{A}$ besitzen.

Die folgende Formel F besitzt kein Modell mit endlicher Grundmenge:

$$\begin{aligned} F \;=\; & \forall X\, p(X, f(X)) \;\wedge\; \forall X\, \neg p(X, X) \;\wedge \\ & \forall X\, \forall Y\, \forall Z\, (\, (p(X,Y) \wedge p(Y,Z)) \rightarrow p(X,Z)\,) \end{aligned}$$

Aber es gibt das folgende Modell $\mathcal{A}$ mit der unendlichen Grundmenge $\mathcal{U}_\mathcal{A} \;=\; I\!N_0 \;=\; \{\, 0, 1, 2, \ldots \,\}$:

$$\begin{aligned} p^\mathcal{A} \;&=\; \{\, (m,n) \in I\!N_0^2 \,|\, m < n \,\}, \\ f^\mathcal{A}(n) \;&=\; n + 1. \end{aligned}$$

Angenommen $\mathcal{B} = (\mathcal{U}_\mathcal{B}, \mathcal{I}_\mathcal{B})$ ist ein Modell mit endlicher Grundmenge $\mathcal{U}_\mathcal{B}$.

Für ein beliebiges Element $m_0 \in \mathcal{U}_\mathcal{B}$ betrachten wir die Folge

$$m_0, m_1, m_2, \ldots \in \mathcal{U}_\mathcal{B}, \text{ mit } m_{i+1} = f^\mathcal{B}(m_i).$$

Wegen des ersten Konjunktionsgliedes von F gilt

$$(m_0, m_1), (m_1, m_2), \ldots, (m_i, m_{i+1}) \in p^\mathcal{B}, \text{ für alle } i \in I\!N_0.$$

Wegen des dritten Konjunktionsgliedes von F ist $p^\mathcal{B}$ transitiv, d.h.

$$(m_i, m_j) \in p^\mathcal{B}, \text{ für alle } i, j \in I\!N_0 \text{ mit } i < j.$$

Da $U_\mathcal{B}$ endlich ist, muß es zwei Indizes i und j mit $i < j$ geben, mit $m_i = m_j = m$. Also gibt es $(m, m) \in p^\mathcal{B}$, im Widerspruch zum zweiten Konjunktionsglied von F (Irreflexivität).

Obiges Beispiel zeigt, daß sich die Wahrheitstafelmethode nicht in die Prädikatenlogik übertragen läßt.

Entscheidbarkeit

Ein (ja/nein–) Problem heißt *entscheidbar* oder *rekursiv*, falls es ein Rechenverfahren gibt (z.B. formuliert als Programm in Java), das für alle Eingaben immer nach endlicher Zeit stoppt, und dann korrekt “ja” oder “nein” ausgibt. Anderenfalls heißt ein Problem *unentscheidbar*.

Gültigkeitsproblem der Prädikatenlogik:
Ist eine gegebene prädikatenlogische Formel F eine gültige Formel ?

Satz (Church)

Das Gültigkeitsproblem der Prädikatenlogik ist unentscheidbar.

Beweis durch Zurückführung auf das unentscheidbare *Postsche Korrespondenzproblem*:

- Gegeben eine endliche Folge $(x_1, y_1), \ldots, (x_n, y_n)$ von Paaren $x_i, y_i \in \{0, 1\}^+$ von nicht–leeren Strings über $\{0, 1\}$.
- Frage: Gibt es eine Folge von Indizes $i_1, \ldots, i_m \in \{ 1, \ldots, n \}$, mit $m \geq 1$, so daß die Konkatenationen der entsprechenden Strings übereinstimmen ?

$$x_{i_1} x_{i_2} \ldots x_{i_m} = y_{i_1} y_{i_2} \ldots y_{i_m}.$$

Beispiel (Postsches Korrespondenzproblem)

Sei $K = ((x_1, y_1), (x_2, y_2), (x_3, y_3))$ mit

$$x_1 = 1 \text{ und } y_1 = 101,$$
$$x_2 = 10 \text{ und } y_2 = 00,$$
$$x_3 = 011 \text{ und } y_3 = 11.$$

Lösung: $i_1 = 1, i_2 = 3, i_3 = 2, i_4 = 3$

$$x_1\, x_3\, x_2\, x_3 \;=\; \underbrace{\overbrace{1}^{x_1}\;\overbrace{0\;1}\;}_{y_1}\!\!{}^{\!\!\!\!\!\!\!\!x_3}\ldots$$

$$x_1\, x_3\, x_2\, x_3 = \overbrace{1}^{x_1}\,\overbrace{0\,1\,1}^{x_3}\,\overbrace{1\,0}^{x_2}\,\overbrace{0\,1\,1}^{x_3} = \underbrace{1\,0\,1}_{y_1}\,\underbrace{1\,1}_{y_3}\,\underbrace{0\,0}_{y_2}\,\underbrace{1\,1}_{y_3} = y_1\, y_3\, y_2\, y_3.$$

Es folgt, daß das *Erfüllbarkeitsproblem* der Prädikatenlogik

Ist eine gegebene prädikatenlogische Formel F erfüllbar ?

ebenfalls unentscheidbar ist.

2.4 Herbrand–Theorie

Jacques Herbrand, Kurt Gödel, Thoralf Skolem

Definition (Herbrand–Universum)

Das Herbrand–Universum HU_F einer Formel F ist die Menge aller variablenfreien Terme, die aus Bestandteilen von F gebildet werden können.

Falls F keine Konstante (d.h., kein 0–stelliges Funktionssymbol) enthält, so nehmen wir eine beliebige Konstante a zur Konstruktion von HU_F hinzu.

Das Herbrand–Universum HU_F ist meist unendlich groß, aber man kann es systematisch konstruieren.

Bestimmung des Herbrand–Universums

1. Alle in F vorkommenden Konstanten sind in HU_F.
2. Falls F keine Konstanten enthält, so nimmt man eine beliebige Konstante a zu HU_F hinzu.
3. Auch der Term $f(t_1, \ldots, t_n)$ ist in HU_F,
   - für jedes in F vorkommende n–stellige Funktionssymbol f und
   - Terme $t_1, \ldots, t_n \in HU_F$.

Für ein 1–stelliges Funktionssymbol f und eine Konstante a setzen wir

- $f^0(a) = a$ und
- $f^{k+1}(a) = f(f^k(a))$, für alle $k \in I\!N_0$.

Z.B. entsteht $f^2(a) = f(f(a))$ durch 2–malige Anwendung von f auf a.

Beispiel (Herbrand–Universum)

Die Formel $F = p(f(a), g(b))$ enthält zwei Konstanten (0–stellige Funktionssymbole) a und b und zwei einstellige Funktionssymbole f und g. Daraus setzen sich die Terme des Herbrand–Universums zusammen.

1. a und b sind in HU_F. Da F bereits Konstanten enthält, ist keine zusätzliche Konstante erforderlich.

2. Mit den Funktionssymbolen f und g kann man weitere Terme bilden. Die Tatsache, daß a in F nur als Argument von f vorkommt und b nur als Argument von g ist dabei unerheblich; auch die "gekreuzten" Terme $f(b)$ und $g(a)$ sind in HU_F.

3. Falls $h_1, \ldots, h_n$ eine beliebige Folge von Funktionssymbolen ist, mit $h_i \in \{f, g\}$, so sind auch $h_1(\ldots(h_n(a)))$ und $h_1(\ldots(h_n(b)))$ in HU_F.

4. $HU_F = \{\, a, b, f(a), f(b), g(a), g(b), f(f(a)), f(g(a)), \ldots \,\}$.

HU_F ist genau dann unendlich groß, wenn F ein n–stelliges Funktionssymbol mit $n \geq 1$ enthält.

Für ein 1–stelliges Funktionssymbol f in F und eine Konstante $a \in HU_F$ sind z.B. auch alle Terme $f^k(a)$ in HU_F.

Beispiel (Herbrand–Universum)

Die Formel F in BPF enthält ein einstelliges Funktionssymbol f:

$$\begin{aligned} F &= \exists X\, p(f(X)), \\ HU_F &= \{\, a, f(a), f(f(a)), \dots \,\} = \{\, f^k(a) \mid k \in \mathbb{N}_0 \,\}. \end{aligned}$$

Da F keine Konstanten enthält, nehmen wir die Konstante a zu HU_F hinzu.

Dasselbe Herbrand–Universum würden wir auch für die variablenfreie Skolemform $F' = p(f(a))$ von F erhalten.

Abzählbarkeit des Herbrand–Universums HU_F

Das Herbrand–Universum HU_F ist *abzählbar*, denn man kann es wie folgt systematisch als unendliche Vereinigung endlicher Mengen konstruieren:

- HU_1 sei die Menge aller in F vorkommenden Konstanten, bzw. $HU_1 = \{ a \}$, falls F keine Konstanten enthält.
- HU'_{k+1} sei die Menge aller Terme $f(t_1, \ldots, t_n)$ aus einem in F vorkommenden n–stelligen Funktionssymbol f und Termen $t_1, \ldots, t_n \in HU_k$, und $HU_{k+1} = HU_k \cup HU'_{k+1}$.
- Dann sind alle Mengen HU_k endlich, die Folge $(HU_k)_{k \in I\!N_+}$ ist aufsteigend, und $HU_F = \cup_{k=1}^{\infty} HU_k$.

Man erhält eine injektive Abbildung $HU_F \to I\!N$, indem man sukzessive die endlichen Mengen $HU_{k+1} \setminus HU_k$ fortlaufend durchnummeriert.

Beispiel (Herbrand–Universum)

Die folgende Formel F enthält vier Funktionssymbole:

$$F \;=\; \forall X \, \forall Y \; p(a, f(X), g(Y, b)).$$

a und b sind 0–stellig, und somit Konstanten. f ist 1–stellig, g ist 2–stellig.

$$\begin{aligned} HU_1 &= \{\, a, b \,\}, \\ HU_2 &= HU_1 \cup \{\, f(a), f(b), g(a, a), g(a, b), g(b, a), g(b, b) \,\}, \\ HU_{k+1} &= HU_k \cup \{\, f(t) \mid t \in HU_k \,\} \cup \{\, g(t_1, t_2) \mid t_1, t_2 \in HU_k \,\}. \end{aligned}$$

Sobald eine Formel F mindestens ein n–stelliges Funktionssymbol f, mit $n \geq 1$, enthält, ist das Herbrand–Universum HU_F unendlich groß: HU_F enthält immer eine Konstante, und man kann aus jedem Term $t \in HU_{k+1} \setminus HU_k$ einen neuen Term $f(t, \ldots, t) \in HU_{k+2} \setminus HU_{k+1}$ bilden.

Definition (Herbrand–Strukturen)

Sei F eine Formel und $\mathcal{A} = (\mathcal{U}_{\mathcal{A}}, \mathcal{I}_{\mathcal{A}})$ eine zu F passende Struktur.

Terme werden von Herbrand–Strukturen als sie selbst interpretiert. $\mathcal{A}$ ist eine Herbrand–Struktur für F, falls die Grundmenge $\mathcal{U}_{\mathcal{A}} = HU_F$ ist und falls für alle Terme $t = f(t_1, \ldots, t_n) \in HU_F$ gilt:

$$\mathcal{A}(t) \ = \ f^{\mathcal{A}}(\mathcal{A}(t_1), \ldots, \mathcal{A}(t_n)) \ = \ f(t_1, \ldots, t_n) \ = \ t.$$

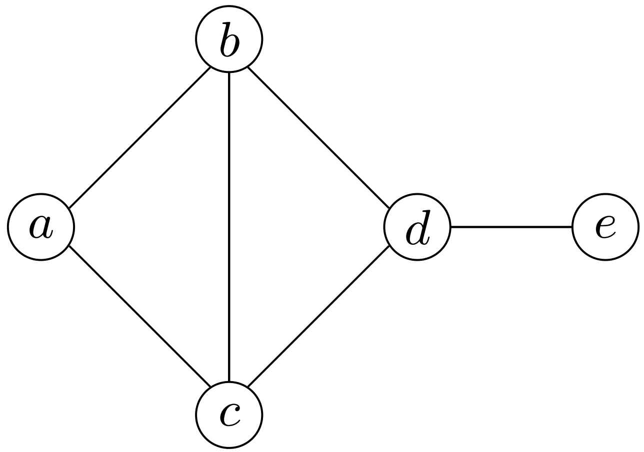
Herbrand–Strukturen für geschlossene Formeln sind vollständig durch die *Interpretation* $p^{\mathcal{A}}$ der Prädikatensymbole charakterisiert.
Man nennt die entsprechende Menge I eine Herbrand–Interpretation:

$$I \ = \ \{\, p(t_1, \ldots, t_n) \mid t_1, \ldots, t_n \in HU_F \wedge (t_1, \ldots, t_n) \in p^{\mathcal{A}} \,\}$$

Eine Herbrand–Struktur, welche ein Modell für eine Formel F ist, heißt *Herbrand–Modell* für F.

Beispiel (3–Färbbarkeit)

Gesucht ist eine Färbung der Knoten eines Graphen $G = \langle V, E \rangle$ mit drei Farben (rot, grün, blau), so daß adjazente Knoten unterschiedliche Farben haben:



Wir repräsentieren die Knoten und Kanten des Graphen als Fakten:

$$f_1 = node(a), \; f_2 = node(b), \; \dots, \; f_5 = node(e),$$
$$e_1 = edge(a, b), \; e_2 = edge(a, c), \; \dots, \; e_6 = edge(d, e).$$

$$
\begin{aligned}
g &= color(X, red) \lor color(X, green) \lor color(X, blue) \leftarrow node(X) \\
c_0 &= \leftarrow edge(X, Y) \land color(X, C) \land color(Y, C), \\
c_1 &= \leftarrow color(X, red) \land color(X, green), \\
c_2 &= \leftarrow color(X, red) \land color(X, blue), \\
c_3 &= \leftarrow color(X, green) \land color(X, blue).
\end{aligned}
$$

Die disjunktive Regel g wählt für jeden Knoten eine Farbe aus.

Die Integritätsbedingung c_0 verhindert, daß adjazente Knoten die gleiche Farbe bekommen. Die Integritätsbedingungen c_1, c_2, c_3 verhindern, daß ein Knoten zwei Farben bekommt.

Nebenbemerkung: Jeder planare Graph ist 4–färbbar.
Für allgemeine Graphen ist der Test auf 3–Färbbarkeit $\mathcal{NP}$–vollständig.

Wir betrachten dann die Formel

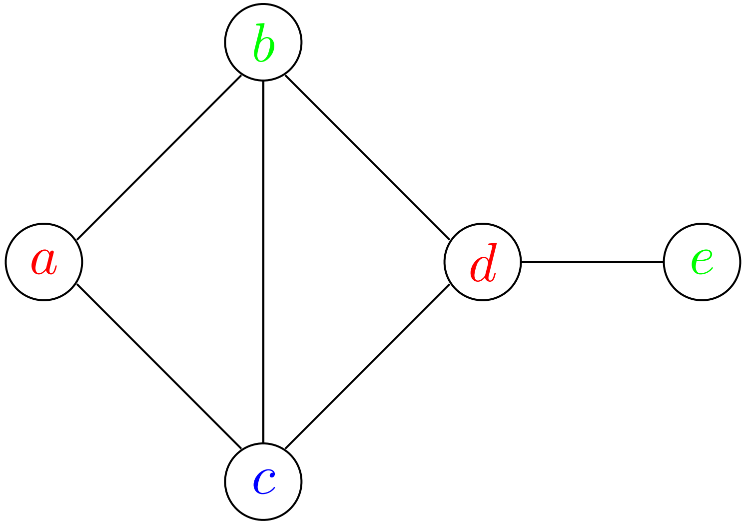
$$\begin{aligned} F \; = \; & f_1 \wedge \ldots \wedge f_5 \wedge e_1 \wedge \ldots \wedge e_6 \wedge \\ & \forall X g \wedge \\ & \forall X \forall Y \forall C\, c_0 \wedge \\ & \forall X c_1 \wedge \ldots \wedge \forall X c_3. \end{aligned}$$

F ist äquivalent zur folgenden Formel F' in Skolemform:

$$\begin{aligned} F' \; = \; & \forall X \, \forall Y \, \forall C \\ & \quad (f_1 \wedge \ldots \wedge f_5 \wedge e_1 \wedge \ldots \wedge e_6 \wedge g \wedge c_0 \wedge \ldots \wedge c_3). \end{aligned}$$

Das Herbrand–Universum ist

$$HU_F \; = \; \{\, a, b, c, d, e, \mathit{red}, \mathit{green}, \mathit{blue} \,\}.$$



Die folgende Herbrand–Interpretation I repräsentiert ein Herbrand–Modell für die Formel F:

$$\begin{aligned} I \;=\; \{\, & node(a),\, \dots,\, node(e),\; edge(a,b),\, \dots,\, edge(d,e), \\ & color(a, red),\; color(b, green),\; color(c, blue), \\ & color(d, red),\; color(e, green) \,\}. \end{aligned}$$

Die zugehörigen Interpretationen $p^{\mathcal{A}}$ der Prädikatensymbole sind folgende:

$$\begin{aligned} node^{\mathcal{A}} &= \{ (a), \dots, (e) \}, \\ edge^{\mathcal{A}} &= \{ (a,b), \dots, (d,e) \}, \\ color^{\mathcal{A}} &= \{ (a,red), (b,green), (c,blue), \\ &\qquad (d,red), (e,green) \}. \end{aligned}$$

Die Herbrand–Interpretation I faßt diese Mengen $p^{\mathcal{A}}$ zusammen.

Damit man die Tupel $(t_1, \dots, t_n) \in p^{\mathcal{A}}$ korrekt zuordnen kann, werden sie in I als Atome $p(t_1, \dots, t_n)$ mit Prädikatensymbol notiert:

$$I = \{ p(t_1, \dots, t_n) \mid t_1, \dots, t_n \in HU_F \wedge (t_1, \dots, t_n) \in p^{\mathcal{A}} \}.$$

Würde man einfach die Vereinigung der Mengen $p^{\mathcal{A}}$ benutzen, so könnte man z.B. nicht wissen, ob ein Paar (t_1, t_2) zu $edge^{\mathcal{A}}$ oder zu $color^{\mathcal{A}}$ gehört.

Satz (Herbrand–Modell)

Eine Aussage F in *Skolemform* ist genau dann erfüllbar, wenn F ein Herbrand–Modell besitzt.

Natürlich ist F erfüllbar, wenn F ein Herbrand–Modell besitzt.

Um die Rückrichtung zu zeigen, konstruiert man aus einem beliebigen Modell $\mathcal{A}$ einer erfüllbaren Aussage F in Skolemform ein Herbrand–Modell.

Beweis:

1. $\Leftarrow$: F ist erfüllbar, wenn F ein Herbrand–Modell besitzt.

2. $\Rightarrow$: Sei $\mathcal{A} = (\mathcal{U}_\mathcal{A}, \mathcal{I}_\mathcal{A})$ ein beliebiges Modell für F.

   Falls F keine Konstanten enthält und a als zusätzliche Konstante gewählt wurde, so setzen wir $a^\mathcal{A} = a'$, für ein beliebiges Element $a' \in \mathcal{U}_\mathcal{A}$. $\mathcal{A}$ ist danach immer noch ein Modell für F.

   Wir konstruieren nun aus $\mathcal{A}$ eine Herbrand–Struktur $\mathcal{B} = (\mathcal{U}_\mathcal{B}, \mathcal{I}_\mathcal{B})$ für F, d.h. mit der Grundmenge $\mathcal{U}_\mathcal{B} = HU_F$.

   Dazu müssen wir die Prädikatensymbole p aus F interpretieren.

Sei p ein n–stelliges Prädikatensymbol, und seien $t_1, \ldots, t_n \in HU_F$:

$$(t_1, \ldots, t_n) \in p^{\mathcal{B}} \text{ g.d.w. } (\mathcal{A}(t_1), \ldots, \mathcal{A}(t_n)) \in p^{\mathcal{A}}$$

Wir zeigen nun, daß $\mathcal{B}$ ein Modell für F ist. Mehr noch:
Für jede Aussage G in Skolemform, die aus den Bestandteilen von F aufgebaut ist, gilt:

$$\mathcal{A} \models G \;\Rightarrow\; \mathcal{B} \models G.$$

Wir führen eine Induktion über die Anzahl n der Allquantoren von G.

$n = 0$: G enthält keine Quantoren.
Dann haben alle Atome von G unter $\mathcal{A}$ und $\mathcal{B}$ denselben Wahrheitswert, und deshalb gilt $\mathcal{A}(G) = \mathcal{B}(G)$.

$n \to n + 1$: Sei G eine Formel mit $n + 1$ Allquantoren, und zwar von der Form $G = \forall X \; H$, wobei H nur n Allquantoren enthält.

Wegen $\mathcal{A} \models G$ gilt:

$$\forall\, u \in \mathcal{U}_{\mathcal{A}} : \; \mathcal{A}_{[X\,|\,u]}(H) = 1.$$

Insbesondere gilt für alle $u = \mathcal{A}(t) \in \mathcal{U}_{\mathcal{A}}$ mit $t \in HU_G$:

$$\mathcal{A}_{[X\,|\,\mathcal{A}(t)]}(H) = 1.$$

Deshalb gilt:

$$\forall\, t \in HU_G : \; \mathcal{A}(H[X|t]) = \mathcal{A}_{[X\,|\,\mathcal{A}(t)]}(H) = 1.$$

Nach Induktionsvoraussetzung gilt nun:

$$\forall\, t \in HU_G : \; \mathcal{B}(H[X|t]) = 1.$$

Deshalb gilt:

$$\forall\, t \in HU_G : \; \mathcal{B}_{[X\,|\,\mathcal{B}(t)]}(H) = \mathcal{B}(H[X|t]) = 1.$$

Also gilt $\mathcal{B}(G) = \mathcal{B}(\forall X\, H) = 1$. $\square$

Der obige Satz über Herbrand–Modelle wäre falsch, wenn wir nicht eine frische Konstante a zu HU_F hinzunehmen würden, falls F keine Konstanten enthält – denn dann wäre die Grundmenge $HU_F = \emptyset$.

Beispiel (Herbrand–Modell)

Für die Formel $F = \exists X\, p(X)$ ist das Herbrand–Universum $HU_F = \{\, a \,\}$, und das einzige Herbrand–Modell $\mathcal{A}$ ist gegeben durch $p^{\mathcal{A}} = \{\, (a) \,\}$.

Ohne die frische Konstante a könnte man kein Herbrand–Modell bilden, da F fordert, daß $p^{\mathcal{A}}$ mindestens ein Tupel (t), mit $t \in HU_F$, enthalten muß.

a entspricht dem 0–stelligen Funktionssymbol aus der Skolemform $p(a)$, das zur Elimination des Existenzquantors aus F hinzugenommen wird.

Satz (Löwenheim und Skolem)

Jede erfüllbare Formel F der Prädikatenlogik besitzt ein abzählbares Modell $\mathcal{B} = (\mathcal{U}_\mathcal{B}, \mathcal{I}_\mathcal{B})$, d.h. mit einer abzählbaren Grundmenge $\mathcal{U}_\mathcal{B}$.

Man kann für $\mathcal{B}$ ein *Herbrand–Modell* der Skolemform von F wählen.

Definition (Herbrand–Expansion)

Sei $F = \forall X_1 \forall X_2 \ldots \forall X_n\, G$ eine Formel in Skolemform mit der (quantorenfreien) Matrix G.

Dann ist die Herbrand–Expansion von F definiert als

$$\mathcal{E}(F) = \{\, G[X_1|t_1] \ldots [X_n|t_n] \mid t_1, \ldots, t_n \in HU_F \,\}.$$

Falls F Variablensymbole und Funktionssymbole der Stelligkeit $n \geq 1$ enthält, so sind HU_F und $\mathcal{E}(F)$ unendlich groß.

Beispiel (Herbrand–Expansion)

Für die folgende Formel F in Skolemform

$$F \;=\; \forall X \; (\underbrace{married(a,b)}_{f} \land \underbrace{(student(X) \rightarrow person(X))}_{r} \;),$$

ist $HU_F = \{ \, a, b \, \}$, und die Herbrand–Expansion

$$\begin{aligned} \mathcal{E}(F) \;&=\; \{ \; (f \land r)[X|a], \; (f \land r)[X|b] \; \} \\ &=\; \{ \; married(a,b) \land (student(a) \rightarrow person(a)), \\ &\qquad\quad married(a,b) \land (student(b) \rightarrow person(b)) \; \} \end{aligned}$$

hat 2 Elemente, da man das Variablensymbol X mit den zwei Konstanten aus HU_F belegen kann.

Beispiel (Herbrand–Expansion)

Für die Formel $F = \forall X_1 \forall X_2 \forall X_3\, G$, mit $G = F_1 \wedge F_2 \wedge F_3 \wedge F_4 \wedge F_5$ und

$$
\begin{aligned}
F_1 &= ancestor(X_1, X_2) \leftarrow parent(X_1, X_2),\\
F_1 &= ancestor(X_1, X_2) \leftarrow parent(X_1, X_3) \wedge ancestor(X_3, X_2),\\
F_3 &= parent(a, b),\\
F_4 &= parent(b, c),\\
F_5 &= parent(c, d),
\end{aligned}
$$

git $HU_F = \{\, a, b, c, d \,\}$, und $\mathcal{E}(F)$ hat $4^3 = 64$ Elemente. Eines davon ist z.B. $G\theta = F_1\theta \wedge F_2\theta \wedge F_3 \wedge F_4 \wedge F_5$, für $\theta = [X_1|a]\,[X_2|c]\,[X_3|b]$:

$$
\begin{aligned}
F_1\theta &= ancestor(a, c) \leftarrow parent(a, c),\\
F_2\theta &= ancestor(a, c) \leftarrow parent(a, b) \wedge ancestor(b, c).
\end{aligned}
$$

Beispiel (Herbrand–Expansion)

Für die Formel

$$F \;=\; \forall X_1 \forall X_2 \, p(X_1, f(X_2), g(X_1, X_2))$$

in Skolemform erhalten wir die folgende Herbrand–Expansion:

$$\begin{aligned}
HU_F \;&=\; \{\, a,\ f(a),\ g(a,a),\ f(f(a)),\ f(g(a,a)), \\
&\qquad g(a, f(a)),\ g(a, g(a,a)), \\
&\qquad g(f(a), a),\ g(f(a), f(a)),\ g(f(a), g(a,a)), \\
&\qquad g(g(a,a), a),\ g(g(a,a), f(a)),\ g(g(a,a), g(a,a)),\ \ldots \,\}, \\
\mathcal{E}(F) \;&=\; \{\, p(t_1, f(t_2), g(t_1, t_2)) \mid t_1, t_2 \in HU_F \,\}.
\end{aligned}$$

Da F keine Konstante enthält, wurde a zu HU_F hinzugenommen.

Die beiden Mengen HU_F und $\mathcal{E}(F)$ sind hier unendlich groß.

Man kann $\mathcal{E}(F)$ als eine – meist unendliche – aussagenlogische Formelmenge angesehen werden, wenn man die in $\mathcal{E}(F)$ vorkommenden Grundatome als atomare Formeln auffaßt.

Satz (Gödel, Herbrand und Skolem)

Eine Aussage F in Skolemform ist genau dann *unerfüllbar*, wenn es eine endliche Teilmenge von $\mathcal{E}(F)$ gibt, die (im aussagenlogischen Sinne) unerfüllbar ist.

Beweis:

Mittels des Satzes von Gödel, Herbrand und Skolem und des Endlichkeitssatzes der Aussagenlogik. $\square$

Semi–Entscheidungsverfahren für die Prädikatenlogik

Für jede prädikatenlogische Formel F ist die Herbrand–Expansion $\mathcal{E}(G) = \{ G_1, G_2, \ldots \}$ der Skolemform G von F abzählbar, da auch das Herbrand–Universum HU_F abzählbar ist.

F ist genau dann *erfüllbar*, wenn alle Konjunktionen $G_1 \wedge G_2 \wedge \ldots \wedge G_n$, mit $n \in I\!N_+ = \{ 1, 2, \ldots \}$, erfüllbar sind. Durch sukzessives Testen dieser Konjunktionen auf Unerfüllbarkeit erhält man also ein Verfahren, welches für unerfüllbare Formeln F nach endlicher Zeit stoppt (Gilmore).

Für erfüllbare Formeln F terminiert das Verfahren offensichtlich nicht immer. $\rightarrow$ *Semi–Entscheidbarkeit* des *Unerfüllbarkeitsproblems*.

Durch Anwendung des Testverfahrens auf $\neg F$ kann man auf Gültigkeit testen. $\rightarrow$ *Semi–Entscheidbarkeit* des *Gültigkeitsproblems*.

2.5 Resolution

Wir betrachten eine prädikatenlogische Formel F in Skolemform ohne freie Variablen:

$$F \;=\; \forall X_1 \forall X_2 \ldots \forall X_m \, G$$

Die Matrix G ist in KNF, quantorenfrei und von der Form $G = \wedge_{i=1}^{k} \beta_i$. Wir untersuchen nun die zugehörige Klauselmenge

$$M \;=\; \{\, \beta_1, \ldots, \beta_k \,\}.$$

Eine Struktur $\mathcal{A}$ heißt Modell für M, falls $\mathcal{A}$ ein Modell für F ist. Wir betrachten eine Klauselmenge M also wie eine KNF zusammen mit Allquantoren für alle vorkommenden Variablen.

Eine Klausel $\beta = L_1 \vee \ldots \vee L_n$ wird auch oft als Menge $\{\, L_1, \ldots, L_n \,\}$ von Literalen repräsentiert.

Definition (Resolutionsableitung)

Sei M eine Klauselmenge.

Eine Folge $K_1, K_2, \ldots, K_n$ von Klauseln nennt man *Resolutionsableitung*, falls für alle $1 \leq i \leq n$ gilt:

1. $K_i = K\theta$ ist eine Grundinstanz einer Klausel $K \in M$,
   d.h. $\theta = [X_1|t_1] \ldots [X_k|t_k]$, mit $t_1, \ldots, t_k \in HU_F$, oder
2. K_i ist eine (aussagenlogische) Resolvente
   zweier Klauseln K_a und K_b mit $1 \leq a, b \leq i - 1$.

Unerfüllbarkeit

Eine Klauselmenge M – bzw. die entsprechende prädikatenlogische Formel F – ist genau dann *unerfüllbar*, wenn es eine Resolutionsableitung der leeren Klausel $K_n = \square$ gibt.

Beispiel (Resolution)

Wir betrachten die Klauselmenge $M = \{ F_1, F_2, F_3 \}$ mit

$$\begin{aligned} F_1 &= \neg p(X) \vee \neg p(f(a)) \vee q(Y), \\ F_2 &= p(Y), \\ F_3 &= \neg p(g(b, X)) \vee \neg q(b). \end{aligned}$$

Dann kann man

- $F_1[X|f(a)]\,[Y|b]$ und $F_2[Y|f(a)]$ zu $K = \{ q(b) \}$ resolvieren, und
- $F_2[Y|g(b, a)]$ und $F_3[X|a]$ zu $K' = \{ \neg q(b) \}$.

Aus K und K' erhält man schließlich die leere Klausel $\Box$.

Also sind die Klauselmenge M und die zugehörige Formel F unerfüllbar:

$$F = \forall X \, \forall Y \, (F_1 \wedge F_2 \wedge F_3).$$

Baumdarstellung:

$$
\begin{tikzcd}
\{\, \neg p(X), \neg p(f(a)), q(Y) \,\} \arrow[d, swap, "{[X\,|\,f(a)]\ \color{red}[Y\,|\,b]}"] & & & \{\, p(Y) \,\} \arrow[dl, swap, "{[Y\,|\,f(a)]}"] \arrow[dr, "{[Y\,|\,g(b,a)]}"] & & & \{\, \neg p(g(b,X)), \neg q(b) \,\} \arrow[d, "{\color{red}[X\,|\,a]}"] \\
\{\, \neg p(f(a)), q(b) \,\} \arrow[dr] & & \{\, p(f(a)) \,\} \arrow[dl] & & \{\, p(g(b,a)) \,\} \arrow[dr] & & \{\, \neg p(g(b,a)), \neg q(b) \,\} \arrow[dl] \\
& \{\, q(b) \,\} \arrow[drr] & & & & \{\, \neg q(b) \,\} \arrow[dll] & \\
& & & \Box & & &
\end{tikzcd}
$$

Die Klausel $\{\, p(Y) \,\}$ wurde in der Resolutionsableitung zweimal benutzt.

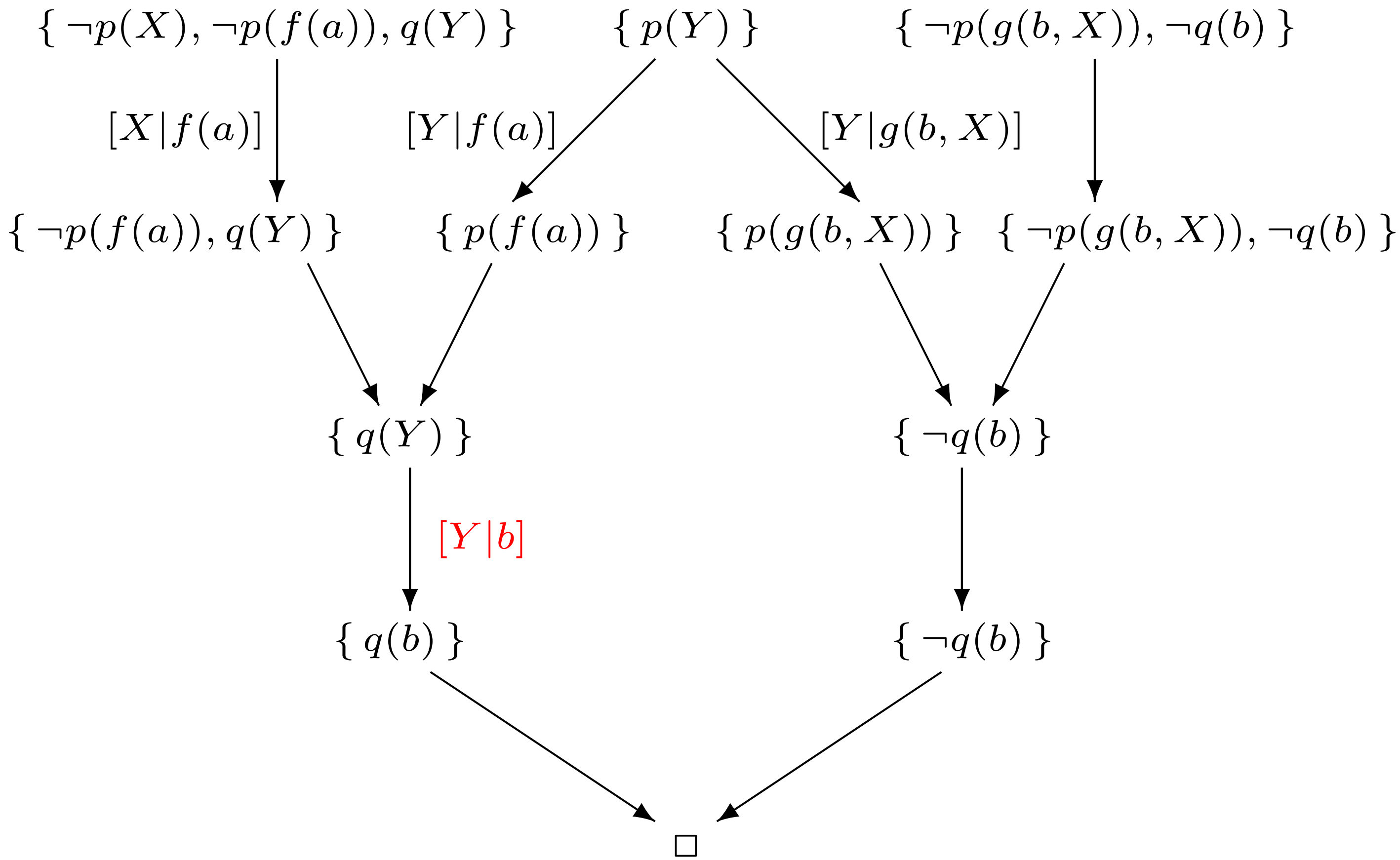
$$
\begin{tikzcd}[column sep=small]
\{\, \neg p(X), \neg p(f(a)), q(Y) \,\} \arrow[d, "{[X\,|\,f(a)]\;[Y\,|\,b]}"'] & & & \{\, p(Y) \,\} \arrow[dl, "{[Y\,|\,f(a)]}"'] \arrow[dr, "{[Y\,|\,g(b,a)]}"] & & & \{\, \neg p(g(b,X)), \neg q(b) \,\} \arrow[d, "{[X\,|\,a]}"] \\
K_1 = \{\, \neg A, B \,\} \arrow[dr] & & K_2 = \{\, A \,\} \arrow[dl] & & K_3 = \{\, C \,\} \arrow[dr] & & K_4 = \{\, \neg C, \neg B \,\} \arrow[dl] \\
& K_5 = \{\, B \,\} \arrow[drr] & & & & K_6 = \{\, \neg B \,\} \arrow[dll] & \\
& & & K_7 = \Box & & &
\end{tikzcd}
$$

Durch die vollständige Grundinstanziierung auf

$$A = p(f(a)),\; B = q(b),\; C = p(g(b,a)),$$

wird das Problem auf die Aussagenlogik reduziert.

Aus *Effizienzgründen* versucht man Substitutionen aber nur auszuführen, wenn es für den direkt nachfolgenden Resolutionsschritt erforderlich ist.



Definition (Unifikator, allgemeinster Unifikator)

Sei $\theta = [X_1|t_1] \dots [X_n|t_n]$ eine Substitution.

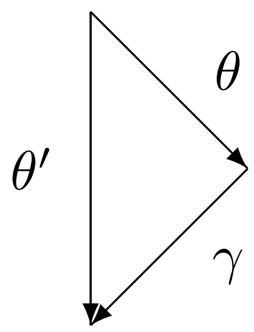
1. Für eine endliche Literalmenge

   $$\mathcal{L} = \{ L_1, \dots, L_k \}$$

   setzen wir $\mathcal{L}\theta = \{ L_1\theta, \dots, L_k\theta \}$.

2. θ ist ein *Unifikator* von $\mathcal{L}$, falls

   $$L_1\theta = L_2\theta = \dots = L_k\theta.$$



3. Ein Unifikator θ von $\mathcal{L}$ heißt *allgemeinster Unifikator* von $\mathcal{L}$, falls für jeden Unifikator θ' von $\mathcal{L}$ eine Substitution γ existiert mit

   $$\theta' = \theta\gamma,$$

   d.h. $F\theta' = (F\theta)\gamma$ für alle Formeln F.

Beispiel (Allgemeinster Unifikator)

1. θ ist ein allgemeinster Unifikator für die Literalmenge

   $$\mathcal{L} = \{\, p(X), p(f(a)) \,\}.$$

   θ' ist ein weiterer Unifikator, und es gilt $\theta' = \theta\gamma$.

$$\begin{tikzcd}
{} \arrow[dr, "\theta = [X|f(a)]"] \arrow[dd, "\theta' = [X|f(a)]\,[Y|b]"'] & \\
& {} \arrow[dl, "\gamma = [Y|b]"] \\
{} &
\end{tikzcd}$$

2. θ ist ein allgemeinster Unifikator für die Literalmenge

   $$\mathcal{L} = \{\, p(Y), p(g(b, X)) \,\}.$$

   θ' ist ein weiterer Unifikator, und es gilt $\theta' = \theta\gamma$.

$$
\begin{tikzcd}
{} \arrow[dr, "{\theta = [Y|g(b, X)]}"] \arrow[dd, "{\theta' = [Y|g(b, a)]\,[X|a]}"'] & \\
& {} \arrow[dl, "{\gamma = [X|a]}"] \\
{} &
\end{tikzcd}
$$

Allgemeinste Unifikatoren sind i.a. nicht eindeutig:

$$\mathcal{L} = \{\, p(X), p(Y)\, \}$$

hat z.B. die beiden folgenden allgemeinsten Unifikatoren:

$$\begin{aligned} \theta_1 &= [X|Y], \\ \theta_2 &= [Y|X]. \end{aligned}$$

Es gilt $\theta_1\theta_2 \;=\; \theta_2$ und $\theta_2\theta_1 \;=\; \theta_1$.

Der folgende Unifikator θ_3 von $\mathcal{L}$ ist kein allgemeinster Unifikator:

$$\theta_3 \;=\; [X|Z][Y|Z].$$

Es gilt $\theta_1[Y|Z] \;=\; \theta_3$ und $\theta_2[X|Z] \;=\; \theta_3$. Es existiert aber keine Substitution γ mit $\theta_1 = \theta_3\gamma$.

Die Substitutionen bilden keine Gruppe. Es gibt ein neutrales Element $[\,]$. Aber für

$$\theta_3 \;=\; [X|Z][Y|Z]$$

gibt es z.B. kein Rechts–Inverses γ_3 mit $\theta_3\gamma_3 = [\,]$. Man kann die Anwendung von θ_3 nicht rückgängig machen, da man für ein Z nicht weiß, ob es von einem X oder einem Y herrührt. Ebenso gibt es für

$$\theta_4 \;=\; [X|a]$$

kein Rechts–Inverses, da man die Konstante a nicht auf eine Variable X abbilden kann. Auch für θ_1 und θ_2 gibt es keine Rechts–Inverse.

Deswegen sind $\theta_1\theta_2 = \theta_2$ und $\theta_2\theta_1 = \theta_1$ möglich. Interessanterweise gilt hier außerdem $\theta_1\theta_1 = \theta_1$ und $\theta_2\theta_2 = \theta_2$ (Idempotenz).

Unifikationssatz (J.A. Robinson)

Jede unifizierbare Menge von Literalen besitzt auch einen *allgemeinsten* Unifikator.

Wir interessieren uns hauptsächlich für denjenigen allgemeinsten Unifikator, der durch den folgenden Algorithmus bestimmt wird.

Unifikationsalgorithmus

Eingabe: eine endliche, nicht–leere Literalmenge (Literalfolge) $\mathcal{L}$

$\theta = [\,]$; ($[\,]$ ist die leere Substitution, d.h. die identische Abbildung)

while $|\mathcal{L}\theta| > 1$ *do begin*

durchsuche die Literale in $\mathcal{L}\theta$ von links nach rechts, bis die erste Position gefunden ist, wo sich mindestens zwei Literale L_1 und L_2 unterscheiden (Prädikaten– oder Argument–Position);

if keines der beiden Zeichen an den gefundenen Positionen
ist eine Variable
then stoppe mit der Ausgabe “nicht unifizierbar”
else begin

sei X die an der einen Position gefundene Variable
und t der Term an der anderen Position;
if X kommt in t vor (“Occurs Check”)
then stoppe mit der Ausgabe “nicht unifizierbar”
else $\theta = \theta\,[X|t]$;

end;

end;

Ausgabe: Falls $|\mathcal{L}\theta| = 1$, so ist θ ein *allgemeinster Unifikator* von $\mathcal{L}$; sonst ist $\mathcal{L}$ nicht unifizierbar.

Beispiel (Unifikationsalgorithmus)

1. Für die Literalmenge $\mathcal{L} = \{ \, p(X), \, p(f(a)), \, p(Y) \, \}$ sind die Prädikatenpositionen alle gleich p.

$$\mathcal{L} = \{ \, p(\underbrace{X}), \, p(\underbrace{f(a)}), \, p(\underbrace{Y}) \, \}$$

Die ersten Unterschiede liegen in den Argumenten. Also setzen wir $\theta_1 = [X|f(a)]$ und erhalten $\mathcal{L}\theta_1 = \{ \, p(f(a)), \, p(Y) \, \}$, da die ersten beiden Atome zusammenfallen.

Mittels $\theta_2 = [Y|f(a)]$ erhalten wir $\theta = \theta_1\theta_2 = [X|f(a)][Y|f(a)]$ und $\mathcal{L}\theta = \{ \, p(f(a)) \, \}$. Wegen $|\mathcal{L}\theta| = 1$ ist θ ein allgemeinster Unifikator.

2. Für die Literalmenge $\mathcal{L} = \{ L_1, L_2 \}$, mit

$$\begin{aligned} L_1 &= p(f(Z, g(a, Y)), h(Z)), \\ L_2 &= p(f(f(U, V), W), h(f(a, b))), \end{aligned}$$

sind die ersten unterschiedlichen Positionen $X = Z$ und $t = f(U, V)$.
Somit erhalten wir $\theta_1 = [Z|f(U, V)]$.
Für $\mathcal{L}\theta_1 = \{ L_1\theta_1, L_2\theta_1 \}$, mit

$$\begin{aligned} L_1\theta_1 &= p(f(f(U, V), g(a, Y)), h(f(U, V))), \\ L_2\theta_1 &= L_2, \end{aligned}$$

sind die ersten unterschiedlichen Positionen $X = W$ und $t = g(a, Y)$.
Somit erhalten wir $\theta_2 = [W|g(a, Y)]$.

Für $\mathcal{L}\theta_1\theta_2 = \{ L_1\theta_1\theta_2, L_2\theta_1\theta_2 \}$, mit

$$\begin{aligned} L_1\theta_1\theta_2 &= p(f(f(U,V), g(a,Y)), h(f(U,V))) \text{ und} \\ L_2\theta_1\theta_2 &= p(f(f(U,V), g(a,Y)), h(f(a,b))), \end{aligned}$$

sind die ersten unterschiedlichen Positionen $X = U$ und $t = a$. Somit erhalten wir $\theta_3 = [U|a]$. Abschließend erhalten wir noch $\theta_4 = [V|b]$.

Insgesamt erhalten wir die Substitution

$$\begin{aligned} \theta &= \theta_1\theta_2\theta_3\theta_4 \\ &= [Z|f(U,V)][W|g(a,Y)][U|a][V|b] \\ &= [Z|f(a,b)][W|g(a,Y)][U|a][V|b], \end{aligned}$$

und

$$L_1\theta = p(f(f(a,b), g(a,Y)), h(f(a,b))) = L_2\theta.$$

3. $\mathcal{L} = \{ p, q \}$ ist nicht unifizierbar,
   da die Prädikatensymbole nicht übereinstimmen.

4. $\mathcal{L} = \{ p(a), p(b) \}$ ist nicht unifizierbar,
   da die Argumente unterschiedliche Konstanten sind.

5. $\mathcal{L} = \{ p(f(X)), p(g(X)) \}$ ist nicht unifizierbar,
   da die Funktionssymbole f und g nicht übereinstimmen.

6. $\mathcal{L} = \{ p(X), p(f(X)) \}$ ist ebenfalls nicht unifizierbar, denn der Occurs Check stellt fest, daß der Term $t = f(X)$ die Variable X enthält.

   Wenn man X mittels $\theta = [X|t']$ durch einen Term t' ersetzt, etwa durch $t' = t = f(X)$, dann enthält $p(X)\theta = p(t')$ immer ein Auftreten des Funktionssymbols f weniger als $p(f(X))\theta = p(f(t'))$.

Occurs Check

In PROLOG wird der Occurs Check aus Effizienzgründen meist weggelassen, da der auf X zu testende Term t sehr groß sein kann.

```
?- X = f(X).
X = f(**).
```

- Das Ergebnis der Unifikation von X und $f(X)$ ist in SWI–PROLOG die Substitution $[X|f(\star\star)]$.
- Wie bei Dauerschleifen in herkömmlichen Programmiersprachen wie JAVA, geht man aber davon aus, daß ein Programmierer in der Praxis diese fehlerhaften Unifikationen vermeidet.
- Aus rein theoretischer Sicht könnte man über unendlichen Termstrukturen als Ergebnis die Substitution $[X|f(f(f(\ldots)))]$ angeben, die X durch einen unendlich tief verschachtelten Term ersetzt.

Komplemente

1. Für ein Atom A setzen wir $\overline{A} = \neg A$ und $\overline{\neg A} = A$.
2. Für eine Literalmenge $\mathcal{L}$ setzen wird $\overline{\mathcal{L}} \ = \ \{ \, \overline{L} \, | \, L \in \mathcal{L} \, \}$.

Dann gilt offensichtlich $\overline{\overline{A}} = A$ und $\overline{\overline{\mathcal{L}}} = \mathcal{L}$.

Beispiel (Komplemente)

1. Das Komplement eines Atoms $A = p(f(X))$ ist seine Negation $\overline{A} = \neg p(f(X))$, für ein negiertes Atom $L = \neg p(f(X))$ erhalten wir das entsprechende positive Atom $\overline{L} = p(f(X))$.
2. Für die Literalmenge $\mathcal{L}$ erhalten wir die Komplementmenge $\overline{\mathcal{L}}$:

$$\begin{aligned} \mathcal{L} \ &= \ \{ \, p(f(X)), \neg q(Z), p(Z) \, \}, \\ \overline{\mathcal{L}} \ &= \ \{ \, \neg p(f(X)), q(Z), \neg p(Z) \, \}. \end{aligned}$$

Definition (Prädikatenlogische Resolution)

Seien K_1 und K_2 prädikatenlogische Klauseln.

1. Seien θ_1 und θ_2 Substitutionen, welche die Variablen von K_1 und K_2 so umbenenennen, daß $K_1\theta_1$ und $K_2\theta_2$ keine gemeinsamen Variablen mehr enthalten – wir nennen sie dann *variablendisjunkt*.

2. Es gebe nun nicht–leere Literalmengen $\mathcal{L}_1 \subseteq K_1\theta_1$ und $\mathcal{L}_2 \subseteq K_2\theta_2$, so daß $\mathcal{L} = \mathcal{L}_1 \cup \overline{\mathcal{L}_2}$ unifizierbar ist mit einem allgemeinsten Unifikator θ.

3. Dann ist die folgende Klausel K eine prädikatenlogische *Resolvente* von K_1 und K_2:

$$K \;=\; ((K_1\theta_1 \setminus \mathcal{L}_1) \cup (K_2\theta_2 \setminus \mathcal{L}_2))\theta$$

$$\begin{tikzcd} K_1 \arrow[dr] & & K_2 \arrow[dl] \\ & K & \end{tikzcd}$$

Beispiel (Prädikatenlogische Resolution)

Da die Klauselmenge $\{\, K_1, K_2 \,\}$ für die Formel

$$\forall X\ (p(X) \wedge \neg p(f(X))) \ \equiv\ (\forall X\ p(X)) \wedge (\forall X\ \neg p(f(X)))$$

steht, können wir K_1 und K_2 variablendisjunkt machen.

$$\begin{tikzcd}
K_1 = \{\, \underline{p(X)} \,\} \arrow[d] & & K_2 = \{\, \underline{\neg p(f(X))} \,\} \arrow[d] \\
K_1' = \{\, \underline{p(X)} \,\} \arrow[dr] & & K_2' = \{\, \underline{\neg p(f(U))} \,\} \arrow[dl] \\
& K = \Box = \emptyset &
\end{tikzcd}$$

Für $\theta_1 = [\,]$ und $\theta_2 = [X|U]$ sind die Klauseln $K_1' = K_1\theta_1 = K_1$ und $K_2' = K_2\theta_2$ variablendisjunkt. Für $\mathcal{L}_1 = K_1$ und $\mathcal{L}_2 = K_2'$ erzeugt der allgemeinste Unifikator $\theta = [X|f(U)]$ von $\mathcal{L} = \mathcal{L}_1 \cup \overline{\mathcal{L}_2}$ die Resolvente K.

Beispiel (Prädikatenlogische Resolution)

Wir betrachten die Klauselmenge $\{ K_1, K_2 \}$:

$$K_1 = \{ \underline{p(f(X))}, \neg q(Z), \underline{p(Z)} \} \qquad K_2 = \{ \underline{\neg p(X)}, r(g(X), a) \}$$

$$K = \{ \neg q(f(X)), r(g(f(X)), a) \}$$

Die Klauseln $K_1\theta_1$ und $K_2\theta_2$ sind für $\theta_1 = [\,]$ und $\theta_2 = [X|U]$ variablendisjunkt und wir setzen

$$\begin{aligned} \mathcal{L}_1 &= \{ p(f(X)), p(Z) \} \subseteq K_1\theta_1 = K_1, \\ \mathcal{L}_2 &= \{ \neg p(U) \} \subseteq K_2\theta_2 = \{ \underline{\neg p(U)}, r(g(U), a) \}. \end{aligned}$$

Dann ergibt der allgemeinste Unifikator $\theta = [U|f(X)]\,[Z|f(X)]$ von $\mathcal{L} = \mathcal{L}_1 \cup \overline{\mathcal{L}_2}$ die Resolvente K.

Die *Resolutionsmengen* werden analog zur Aussagenlogik definiert.

Definition (Resolutionsmengen)

Sei M eine prädikatenlogische Klauselmenge.

$$\begin{aligned} \mathit{Res}(M) \;&=\; M \cup \{\, K \mid K \text{ ist eine prädikatenlogische Resolvente} \\ &\qquad\qquad \text{zweier Klauseln } K_1, K_2 \in M \,\}. \end{aligned}$$

Ferner definieren wir:

$$\begin{aligned} \mathit{Res}^0(M) &= M, \\ \mathit{Res}^{n+1}(M) &= \mathit{Res}(\mathit{Res}^n(M)), \text{ für } n \geq 0, \\ \mathit{Res}^*(M) &= \bigcup_{n \geq 0} \mathit{Res}^n(M). \end{aligned}$$

Es gibt meist sehr viele Resolventen, die wir nicht alle aufzählen können. Außerdem ist das Variablendisjunkt–Machen nicht eindeutig.

Beispiel (Resolutionsmengen)

Für die Klauselmenge $M = \{ F_1, F_2, F_3 \}$ mit

$$\begin{aligned} F_1 &= \neg p(X) \vee \neg p(f(a)) \vee q(Y), \\ F_2 &= p(Y), \\ F_3 &= \neg p(g(b, X)) \vee \neg q(b), \end{aligned}$$

zur Formel $F = \forall X \, \forall Y \, (F_1 \wedge F_2 \wedge F_3)$ erhalten wir folgendes:

- Je nachdem ob wir bei der Resolution von F_1 und F_2 beide p–Literale aus F_1 gegen $F_2 = p(Y)$ resolvieren oder jeweils nur eines, erhalten wir drei unterschiedliche Resolventen:

  $$q(Y), \; \neg p(f(a)) \vee q(Y), \; \neg p(X) \vee q(Y).$$

- Die Resolution von F_2 und F_3 über die p–Literale ergibt

  $$\neg q(b).$$

- Die Resolution von F_1 und F_3 über die q–Literale ergibt

  $$\neg p(X) \vee \neg p(f(a)) \vee \neg p(g(b, X')).$$

  Um die Klauseln variablendisjunkt zu machen, haben wir X in F_3 durch X' ersetzt. Diese Substitution ist natürlich nicht eindeutig.

Wir erhalten also die folgenden Resolutionsmengen:

$$\begin{aligned}
Res^0(M) &= M, \\
Res^1(M) &= M \cup \{ \neg p(f(a)) \vee q(Y),\ \neg p(X) \vee q(Y),\ q(Y),\ \neg q(b), \\
&\qquad\qquad \neg p(X) \vee \neg p(f(a)) \vee \neg p(g(b, X')) \}, \\
Res^2(M) &= Res^1(M) \cup \{ \Box, \ldots \}.
\end{aligned}$$

$Res^2(M)$ enthält hier – neben der leeren Klausel $\Box$ – weitere Resolventen, die wir nicht alle aufzählen wollen.

Resolutionssatz der Prädikatenlogik

Sei F eine prädikatenlogische Formel in Skolemform ohne freie Variablen und mit (quantorenfreier) Matrix in KNF, und sei M die zugehörige Klauselmenge. F ist genau dann *unerfüllbar*, wenn $\square \in Res^*(M)$.

Eine Resolutionsableitung verwendet immer nur endlich viele Klauseln. Deswegen ist der Kompaktheitssatz entscheidend, der besagt, daß eine unerfüllbare Formelmenge bereits eine endliche unerfüllbare Teilmenge haben muß.

Der Beweis des Resolutionssatzes erfolgt durch Lifting des aussagenlogischen Satzes.

Lifting–Lemma

Seien K_1 und K_2 zwei prädikatenlogische Klauseln, und seien K_1' und K_2' beliebige *Grundinstanzen* von K_1 bzw. K_2.

- Sei K' eine Resolvente von K_1' und K_2' im aussagenlogischen Sinn.
- Dann gibt es eine *prädikatenlogische Resolvente* K von K_1 und K_2, so daß K' eine Grundinstanz von K ist.

$$
\begin{tikzcd}
K_1 \arrow[dr] \arrow[dd] & & K_2 \arrow[dl] \arrow[dd] \\
& K \arrow[dd] & \\
K_1' \arrow[dr] & & K_2' \arrow[dl] \\
& K' &
\end{tikzcd}
$$

Beispiel (Lifting–Lemma)

$$
\begin{tikzcd}[column sep=small]
\{\, p(f(X)), \neg q(Z), p(Z) \,\} \arrow[d, "{\theta_1 = [X|b]\,[Z|f(b)]}"'] \arrow[dr] & & \{\, \neg p(X), r(g(X), a) \,\} \arrow[d, "{\theta_2 = [X|f(b)]}"] \arrow[dl] \\
\{\, p(f(b)), \neg q(f(b)), p(f(b)) \,\} \arrow[dr] & \{\, \neg q(f(X)), r(g(f(X)), a) \arrow[d] & \{\, \neg p(f(b)), r(g(f(b)), a) \,\} \arrow[dl] \\
& \{\, \neg q(f(b)), r(g(f(b)), a) &
\end{tikzcd}
$$

Beispiel zur Resolution mit Gruppenaxiomen

Wir schreiben $p(X, Y, Z)$ anstelle von $X \circ Y = Z$ (Produkt).

(1) Abgeschlossenheit:

$$\forall X \forall Y \exists Z \; p(X, Y, Z).$$

(2) Assoziativität:

$$\forall U \forall V \forall W \; \forall X \forall Y \forall Z$$
$$(\; p(X, Y, U) \wedge p(Y, Z, V) \to (p(X, V, W) \leftrightarrow p(U, Z, W) \;).$$

äquivalent: $\forall \ldots (\; X \circ \underbrace{(Y \circ Z)}_{V} = W \; \leftrightarrow \; \underbrace{(X \circ Y)}_{U} \circ Z = W \;).$

(3) Existenz eines links–neutralen Elements (X) und Existenz von Links–Inversen (Z):

$$\exists X \, \forall Y \, (p(X, Y, Y) \, \wedge \, \exists Z \, p(Z, Y, X)).$$

äquivalent: $\exists X \, \forall Y \, ((X \circ Y = Y) \, \wedge \, \exists Z \, (Z \circ Y = X)).$

(4) Existenz von Rechts–Inversen:

$$\exists X \, \forall Y \, (p(X, Y, Y) \, \wedge \, \exists Z \, p(Y, Z, X)).$$

äquivalent: $\exists X \, \forall Y \, ((X \circ Y = Y) \, \wedge \, \exists Z \, (Y \circ Z = X)).$

Es stellt sich heraus, daß man die Existenz von Rechts–Inversen nicht separat in einem Axiom fordern muß, da sie bereits aus den anderen Gruppenaxiomen (1), (2) und (3) folgt.

Die Folgerung $\{\, (1), (2), (3) \,\} \models (4)$ gilt genau dann, wenn $\{\, (1), (2), (3), \neg(4) \,\}$ unerfüllbar ist.

Um dies zu zeigen, betrachten wir die Klauselformen.

Die Negation von Formel (4) zur Existenz von Rechts–Inversen ist äquivalent zur BPF

$$\forall X\, \exists Y\, \forall Z\, (\neg p(X, Y, Y) \vee \neg p(Y, Z, X)).$$

- Da Z nicht in der Teilformel $p(X, Y, Y)$ vorkommt, kann man in (4) den Existenzquantor $\exists Z$ nach vorne ziehen.
- Die Negation vertauscht All– und Existenzquantoren, und sie verwandelt nach De Morgan eine Konjunktion in eine Disjunktion.

Die Formel (2) zur Assoziativität wird z.B. schematisch so umgeformt:

$$
\begin{aligned}
&\forall \ldots \; (\, \alpha \rightarrow (A \leftrightarrow B) \,) \; \equiv \\
&\forall \ldots \; (\, \alpha \rightarrow ((A \wedge B) \vee (\neg A \wedge \neg B)) \,) \; \equiv \\
&\forall \ldots \; (\, \alpha \rightarrow ((A \vee \neg B) \wedge (\neg A \vee B)) \,) \; \equiv \\
&\forall \ldots \; (\, (\alpha \wedge B \rightarrow A) \; \wedge \; (\alpha \wedge A \rightarrow B) \,).
\end{aligned}
$$

Es wurde zweimal verwendet, daß ein negiertes Literal $\neg B$ im Kopf einer Regel als B in den Rumpf verschoben werden kann:

$$
\begin{aligned}
&\forall \ldots \; (\, \alpha \rightarrow (A \vee \neg B) \,) \; \equiv \\
&\forall \ldots \; (\, \neg\, \alpha \vee A \vee \neg B \,) \; \equiv \\
&\forall \ldots \; (\, \alpha \wedge B \rightarrow A \,).
\end{aligned}
$$

Wir erhalten also aus (2) zwei Implikationen.

Klauselform von $(1) \wedge (2) \wedge (3) \wedge \neg(4)$ mit Skolemfunktionen m, e, i, j:

(1) Abgeschlossenheit:

(a) $\{\, p(X, Y, m(X, Y)) \,\}$.

(2) Assoziativität:

(b) $\{\, \neg p(X, Y, U), \neg p(Y, Z, V), \neg p(X, V, W), p(U, Z, W) \,\}$,
(c) $\{\, \neg p(X, Y, U), \neg p(Y, Z, V), \neg p(U, Z, W), p(X, V, W) \,\}$.

(3) Existenz eines links–neutralen Elements e und Existenz von Links–Inversen $i(Y)$ für Gruppenelemente Y:

(d) $\{\, p(e, Y, Y) \,\}$,
(e) $\{\, p(i(Y), Y, e) \,\}$.

(4) Negation der Existenz von Rechts–Inversen:

(f) $\{\, \neg p(X, j(X), j(X)), \neg p(j(X), Z, X) \,\}$.

Resolutionsherleitung der leeren Klausel

$$
\begin{array}{l}
(f)\;\; \{\, \neg p(X, j(X), j(X)), \neg p(j(X), Z, X) \,\} \\
\downarrow \;\swarrow \quad (d)\;\; \{\, p(e, Y, Y) \,\} \\
\{\, \neg p(j(e), Z, e) \,\} \\
\downarrow \;\swarrow \quad (b)\;\; \{\, \neg p(X, Y, U), \neg p(Y, Z, V), \neg p(X, V, W), p(U, Z, W) \,\} \\
\{\, \neg p(X, Y, j(e)), \neg p(Y, Z, V), \neg p(X, V, e) \,\} \\
\downarrow \;\swarrow \quad (e)\;\; \{\, p(i(Y), Y, e) \,\} \\
\{\, \neg p(i(V), W, j(e)), \neg p(W, Z, V) \,\} \\
\downarrow \;\swarrow \quad (d)\;\; \{\, p(e, Y, Y) \,\} \\
\{\, \neg p(i(V), e, j(e)) \,\}
\end{array}
$$

$$
\begin{array}{l}
\downarrow \;\; \swarrow \quad (c) \;\; \{\, \neg p(X,Y,U), \neg p(Y,Z,V), \neg p(U,Z,W), p(X,V,W) \,\} \\
\{\, \neg p(i(V),Y,U), \neg p(Y,Z,e), \neg p(U,Z,j(e)) \,\} \\
\downarrow \;\; \swarrow \quad (d) \;\; \{\, p(e,Y,Y) \,\} \\
\{\, \neg p(i(V),Y,e), \neg p(Y,j(e),e) \,\} \\
\downarrow \;\; \swarrow \quad (e) \;\; \{\, p(i(Y),Y,e) \,\} \\
\{\, \neg p(i(V),i(j(e)),e) \,\} \\
\downarrow \;\; \swarrow \quad (e) \;\; \{\, p(i(Y),Y,e) \,\} \\
\square
\end{array}
$$

Also folgt das vierte Gruppenaxiom aus den ersten dreien:

$$(1) \wedge (2) \wedge (3) \models (4).$$

Da wir in diesem Beispiel ausschließlich Horn–Klauseln vorliegen haben, könnten wir die Resolutionsherleitung auch in *Implikationsnotation* schreiben.

Es handelt sich um eine *lineare Herleitung* von Goals.

Bis auf den ersten Schritt wird immer zuerst das rechteste Atom des Goal–Rumpfes resolviert.

Wenn man zuerst mit (b) und dann mit (d) resolvieren würde, dann wäre dies – wie bei der SLD–Resolution – immer der Fall.

Prädikatenlogik der zweiten Stufe (PL2)

Die Prädikatenlogik ist zwar ausdrucksstärker als die Aussagenlogik. Aber trotzdem kann auch im Rahmen der Prädikatenlogik nicht jede mathematische Aussage formuliert werden.

Wir erhalten eine noch größere Ausdrucksstärke, wenn wir zusätzlich *Quantifizierungen* erlauben über

- *Prädikaten*– und
- *Funktionssymbolen*.

Beispiel (PL2)

$$F = \forall p \, \exists f \, \forall X \; p(f(X)).$$

Die Axiomatisierung der natürlichen Zahlen ist z.B. erst in PL2 richtig möglich – und nicht in PL1.

Induktionsaxiom der natürlichen Zahlen in PL2

Das *Induktionsaxiom* wurde ursprünglich in der PL2 formuliert:

$$\forall p \, (\, p(0) \;\wedge\; \forall N \,(N \in \mathbb{N}_0 \wedge p(N) \to p(s(N))) \;\to \\ \forall N \,(N \in \mathbb{N}_0 \to p(N)) \,).$$

Es verwendet ein 1–stelliges Funktionssymbol s zur Bezeichnung des Nachfolgers einer natürlichen Zahl, und es quantifiziert über ein 1–stelliges Prädikatensymbol p.

Das Axiom besagt, daß jede induktive Teilmenge von $\mathbb{N}_0$ – beschrieben durch das Prädikat p – gleich $\mathbb{N}_0$ sein muß:

- Falls p für die Zahl 0 gilt, und
- falls für alle $N \in \mathbb{N}_0$ aus $p(N)$ auch $p(s(N))$ folgt,
  d.h., wenn p für N gilt, dann gilt p auch für $N + 1$,
- dann gilt p für alle $N \in \mathbb{N}_0$.

Peano–Axiome der natürlichen Zahlen

$$\text{(P1)} \quad \forall N \neg (s(N) \equiv 0),$$

$$\text{(P2)} \quad \forall N \, \forall M \, (s(N) \equiv s(M) \, \rightarrow N \equiv M),$$

$$\text{(P3)} \quad \forall p \, (\, p(0) \wedge \forall N \, (p(N) \rightarrow p(s(N))) \, \rightarrow \forall N \, p(N) \,).$$

(P3) ist eine verkürzte Version des Induktionsaxioms.

Zusätzlich muß das Gleichheitsprädikat $\equiv$ geeignet axiomatisiert werden.

Dann sind alle Modelle $\mathcal{A}$ der Gesamtformelmenge F isomorph.

Z.B. gibt es das (eindeutige) *Herbrand–Modell* mit der Grundmenge

$$HU_F \; = \; \{ \, 0, s(0), s(s(0)), \ldots \, \} \; = \; \{ \, s^n(0) \mid n \in I\!N_0 \, \}.$$

Das Herbrand–Modell repräsentiert eine natürliche Zahl $n \in I\!N$ als einen Term $s^n(0)$ mit n Auftreten des Funktionssymbols s.

Wir setzen dazu $s^0(0) = 0$ und $s^{n+1}(0) = s(s^n(0))$, für $n \in I\!N_0$.

Ein anderes Modell $\mathcal{A}$, das von *John von Neumann* vorgeschlagen wurde, repräsentiert eine natürliche Zahl $n \in I\!N$ als eine Menge s_n.

Dazu werden die Terme $t = s^n(0)$ aus HU_F folgendermaßen interpretiert:

$$\begin{aligned} 0^{\mathcal{A}} &= \emptyset, \\ s(t)^{\mathcal{A}} &= t^{\mathcal{A}} \cup \{ t^{\mathcal{A}} \}. \end{aligned}$$

Wenn wir $s_n = (s^n(0))^{\mathcal{A}}$ schreiben, dann erhalten wir:

$$\begin{aligned} s_0 &= (s^0(0))^{\mathcal{A}} = 0^{\mathcal{A}} = \emptyset, \\ s_{n+1} &= (s^{n+1}(0))^{\mathcal{A}} = (s(s^n(0)))^{\mathcal{A}} \\ &= (s^n(0))^{\mathcal{A}} \cup \{ (s^n(0))^{\mathcal{A}} \} = s_n \cup \{ s_n \}. \end{aligned}$$

Somit erhalten wir die folgende Repräsentation der natürlichen Zahlen:

$$
\begin{aligned}
s_0 &= \emptyset, \\
s_1 &= s_0 \cup \{ s_0 \} = \emptyset \cup \{ \emptyset \} = \{ \emptyset \} = \{ s_0 \}, \\
s_2 &= s_1 \cup \{ s_1 \} = \{ \emptyset, \{ \emptyset \} \} = \{ s_0, s_1 \}, \\
s_3 &= s_2 \cup \{ s_2 \} = \{ \emptyset, \{ \emptyset \}, \{ \emptyset, \{ \emptyset \} \} \} = \{ s_0, s_1, s_2 \}, \\
s_{n+1} &= s_n \cup \{ s_n \} = \{ s_0, s_1, \ldots, s_n \}.
\end{aligned}
$$

Alle Modelle $\mathcal{A}$ der Gesamtformelmenge F, die in der PL2 formuliert ist, sind isomorph (von der Struktur her äquivalent).

Man kann dagegen zeigen, daß jede Axiomatisierung der natürlichen Zahlen im Rahmen der PL1 *Nicht–Standardmodelle* aller unendlichen Kardinalitäten zuläßt – die somit auch nicht isomorph sind.

Mathematische Theorien

Eine *Theorie* ist eine Menge T von Formeln (möglicherweise beschränkt auf solche Formeln, die nur aus bestimmten vorgegebenen Bestandteilen – z.B. bestimmten Prädikaten– und Funktionssymbolen – aufgebaut ist), die gegenüber Folgerbarkeit abgeschlossen ist. D.h., falls

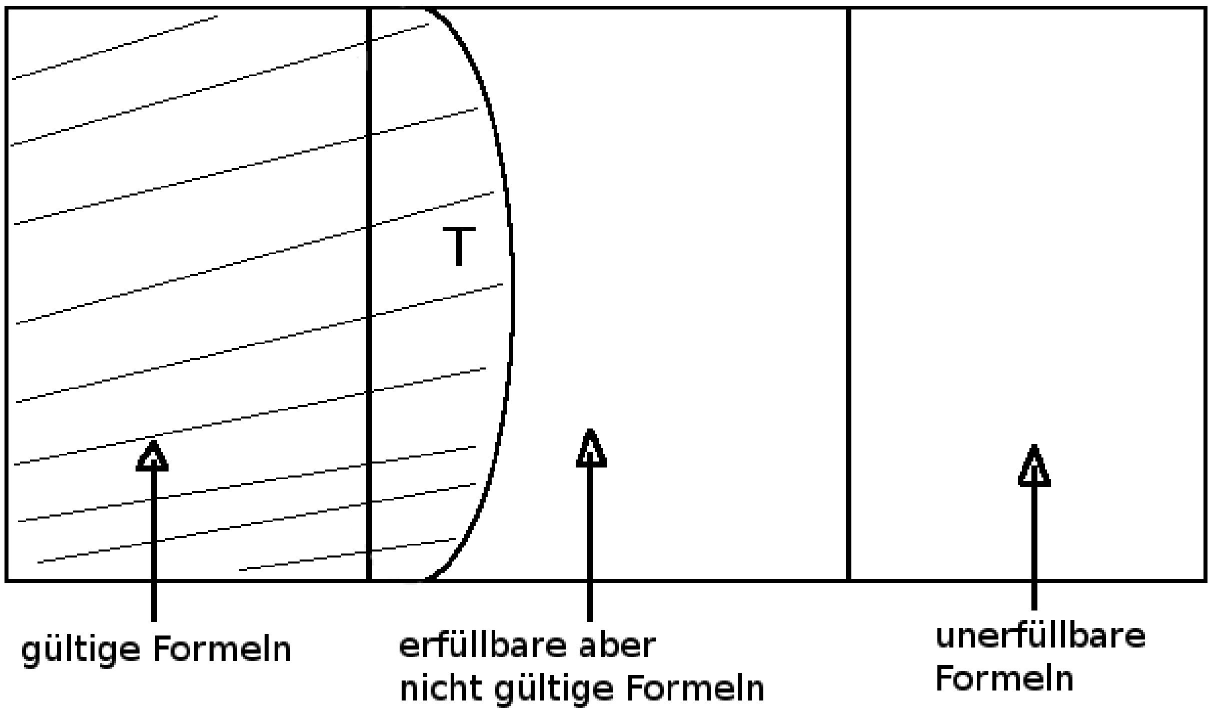
$$\{ F_1, \ldots, F_n \} \subseteq T$$

gilt und

F eine Folgerung aus $\{ F_1, \ldots, F_n \}$ ist,

dann gilt auch $F \in T$. Die Formeln $F \in T$ heißen auch *Sätze* der Theorie.

Eine Theorie muß alle gültigen Formeln enthalten. Normalerweise sollte sie keine unerfüllbaren Formeln enthalten; falls doch, so muß sie alle Formeln enthalten.



2.6 Hyperresolution für Deduktive Datenbanken

Eine prädikatenlogische Klausel $K = L_1 \vee \ldots \vee L_n$ ist eine *Hornklausel*, falls maximal ein Literal L_i positiv ist – wie in der Aussagenlogik.

Es gibt auch wieder drei interessante Spezialfälle; im folgenden seien A und $B_1, \ldots, B_m$ prädikatenlogische Atome:

- Eine *(definite) Regel* ist eine Hornklausel $K = A \vee \neg B_1 \vee \ldots \vee \neg B_m$ mit genau einem positiven Literal. Ein *(definites) Fakt* ist eine Regel $K = A$ ohne negative Literale ($m = 0$).

- Eine *Goal (Ziel, Anfrage)* ist eine Hornklausel $K = \neg B_1 \vee \ldots \vee \neg B_m$ ohne positives Literal.

In der Logikprogrammierung verwendet man – aufgrund der Regeln von De Morgan – für Hornklauseln die Implikations–Notation.

Ein *definites Logikprogramm* ist eine Menge von definiten Regeln (bzw. Fakten) in Implikations–Notation:

Fakt: A,

Regel: $A \leftarrow B_1 \wedge \ldots \wedge B_m,\ m \geq 0$,

Goal: $\leftarrow B_1 \wedge \ldots \wedge B_m,\ m \geq 0$.

Das Atom A heißt *Kopf* einer Regel $r = A \leftarrow \beta$, wobei $\beta = B_1 \wedge \ldots \wedge B_m$ eine Konjunktion von Atomen B_i, $1 \leq i \leq m$, ist. Die Konjunktion β ist der *Rumpf* von r; die B_i heißen Rumpfatome.

Falls $m = 0$ ist, so wird r mit dem Fakt A gleichgesetzt. Fakten sind also auch Regeln, und man kann dann den Implikationspfeil weglassen.

Goals werden bei der SLD–Resolution für Anfragen benutzt; in diesem Abschnitt zur Hyperresolution verwenden wir keine Goals.

DATALOG und DATALOG$^{\text{fun}}$

In der *Aussagenlogik* enthalten Hornformeln keine Variablensymbole. Als Erweiterung betrachten wir in der *Prädikatenlogik* Regeln, bei denen jedes Variablensymbol in mindestens einem Rumpfatom vorkommt. Solche Regeln nennen wir *bereichsbeschränkt*.

Bereichsbeschränkte Regeln bilden DATALOG–Programme (also spezielle definite Logikprogramme). Diese werden in *Deduktiven Datenbanken* zur Wissensrepräsentation verwendet.

- Im strengen Sinne können DATALOG–Programme P auch keine *Funktionssymbole* enthalten. Dann ist das Herbrand–Universum HU_P endlich, und es können nur endlich viele Atome abgeleitet werden.
- In praktischen Anwendungen verwendet man aber oft die Erweiterung um Funktionssymbole namens DATALOG$^{\text{fun}}$. Dann liegt es am Programmierer, Dauerschleifen bei der Ableitung zu vermeiden.

*Beispiel (*DATALOG*–Programm)*

Die folgenden Regeln bilden ein DATALOG–Programm P zur Berechnung der Vorfahren ($ancestor$) aufgrund der Eltern–Relation ($parent$):

$$ancestor(X, Y) \leftarrow parent(X, Y),$$
$$ancestor(X, Y) \leftarrow parent(X, Z) \wedge ancestor(Z, Y).$$

Zugehörige Eltern–Fakten:

$$parent(a, b), \;\; parent(b, c), \;\; parent(c, d).$$

Jeder Elternteil ist ein Vorfahre (Regel 1), und jeder Vorfahre der Eltern ist ebenfalls ein Vorfahre (Regel 2).

In PROLOG sind auch Regeln sinnvoll, bei denen Variablensymbole nur im Kopf vorkommen – wie z.B. beim Listenprädikat `member/2` und beim Prädikat `hres_iteration/3` zur Hyperresolution im nächsten Kapitel.

Hyperresolution

Die sukzessive, n–fache Anwendung der *binären Resolution* auf eine Regel $A \leftarrow B_1 \wedge \ldots \wedge B_n$ und geeignete Fakten A_i nennt man Hyperresolution. Die Rumpfatome B_i werden sukzessive mit den Fakten A_i resolviert.

$$
\begin{array}{ll}
A \vee \neg B_1 \vee \neg B_2 \vee \ldots \vee \neg B_n & A_1 \\
\Big\downarrow \theta_1 & \\
(A \vee \neg B_2 \vee \ldots \vee \neg B_n)\,\theta_1 & A_2 \\
\Big\downarrow \theta_2 & \\
\vdots & \vdots \\
\Big\downarrow & \\
(A \vee \neg B_n)\,\theta_1 \ldots \theta_{n-1} & A_n \\
\Big\downarrow \theta_n & \\
A\,\theta_1 \ldots \theta_n &
\end{array}
$$

Wir nennen das berechnete Fakt $A\,\theta_1 \ldots \theta_n$ eine Hyperresolvente.

DATALOG–Fakten enthalten keine Variablensymbole.
Deswegen muß man die Fakten A_i und die zwischendurch erhaltenen Regeln nicht variablendisjunkt machen.

Es genügt Substitutionen θ_i zu finden, so daß

$$(B_i \, \theta_1 \ldots \theta_{i-1}) \, \theta_i = A_i.$$

Wir bezeichen $\theta = \theta_1 \ldots \theta_n$ als die berechnete Substitution.

Die berechnete Hyperresolvente $A \, \theta$ ist dann automatisch auch ein Fakt ohne Variablensymbole, da ja alle Variablensymbole aus dem Regelkopf A in den Rumpfatomen B_i enthalten sind und somit von θ grundinstanziiert werden.

In den folgenden Hyperresolutionen verwenden wir die Implikations–Notation für Regeln.

Beispiel (Hyperresolution)

Für das DATALOG–Programm P mit den Regeln

$$anc(X, Y) \leftarrow par(X, Y),$$

$$anc(X, Y) \leftarrow par(X, Z) \wedge anc(Z, Y),$$

und den Eltern–Fakten

$$par(a, b),$$

$$par(b, c),$$

$$par(c, d),$$

erhalten wir folgendes:

1. Die Fakten aus P werden in Iteration 1 mittels einer Hyperresolution durch $n = 0$ Anwendungen der binären Resolution abgeleitet.

2. Dann kann in Iteration 2 aus dem Fakt $par(b, c)$ und der ersten Regel ein Fakt $anc(b, c)$ abgeleitet werden:

$$\begin{tikzcd} anc(X, Y) \leftarrow par(X, Y) \arrow[d, "{\theta_1 = [X|b]\,[Y|c]}"'] & par(b, c) \arrow[dl] \\ anc(b, c) & \end{tikzcd}$$

Die berechnete Substitution ist

$$\theta = \theta_1 = [X|b]\,[Y|c],$$

und es wird $anc(X, Y)\theta = anc(b, c)$ abgeleitet.

Ebenso werden die Fakten $anc(a, b)$ und $anc(c, d)$ abgeleitet.

3. Danach kann in Iteration 3 aus den Fakten $par(a,b)$ und $anc(b,c)$ und der zweiten Regel das Fakt $anc(a,c)$ abgeleitet werden:

$$
\begin{tikzcd}
anc(X,Y) \leftarrow par(X,Z) \wedge anc(Z,Y) \arrow[d, "\theta_1 = [X|a]\,[Z|b]"'] & par(a,b) \arrow[dl] \\
anc(a,Y) \leftarrow anc(b,Y) \arrow[d, "\theta_2 = [Y|c]"'] & anc(b,c) \arrow[dl] \\
anc(a,c) &
\end{tikzcd}
$$

Die berechnete Substitution ist

$$\theta = \theta_1\theta_2 = [X|a]\,[Z|b]\,[Y|c],$$

und es wird $anc(X,Y)\theta = anc(a,c)$ abgeleitet.

Ebenso kann das Fakt $anc(b,d)$ abgeleitet werden.

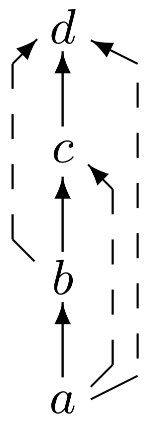
4. Zuletzt kann in Iteration 4 aus dem Fakt $anc(b, d)$ das Fakt $anc(a, d)$ abgeleitet werden:

$$
\begin{tikzcd}
anc(X,Y) \leftarrow par(X,Z) \wedge anc(Z,Y) \arrow[d, "{\theta_1 = [X|a]\,[Z|b]}"'] & par(a,b) \arrow[dl] \\
anc(a,Y) \leftarrow anc(b,Y) \arrow[d, "{\theta_2 = [Y|d]}"'] & anc(b,d) \arrow[dl] \\
anc(a,d) &
\end{tikzcd}
$$

Insgesamt haben wir also die folgenden anc–Fakten abgeleitet:

$$anc(a, b), anc(b, c), anc(c, d), anc(a, c), anc(b, d), anc(a, d).$$

In Iteration $n \geq 2$ werden die anc–Fakten zu den Pfaden der Länge $n-1$ in der Vorfahrenhierarchie abgeleitet:



Die durchgezogenen Linien zeigen die par–Fakten und die entsprechenden anc–Fakten zu den Pfaden der Länge 1.

Die gestrichelten Linien zeigen die anc–Fakten zu den Pfaden der Länge 2 und 3. Das Fakt $anc(a, d)$ zur Länge 3 wird in Iteration 4 aus $par(a, b)$ und $anc(b, d)$ abgeleitet.

Hyperresolutionsmengen $Hres(P, I)$

Sei P ein DATALOG–Programm und I eine Herbrand–Interpretation.

1. $Hres(P, I)$ ist die Menge aller Hyperresolventen, die man aus den Regeln aus P und den Fakten aus I herleiten kann:
2. Also besteht $Hres(P, I)$ aus allen Grundatomen $A\theta$, so daß es eine Regel $A \leftarrow B_1 \wedge \ldots \wedge B_n \in P$ gibt, mit $B_i\theta \in I$, für alle $1 \leq i \leq n$.
3. Man könnte auch sagen: $Hres(P, I)$ besteht aus allen Grundatomen A, so daß es eine Grundinstanz $A \leftarrow B_1 \wedge \ldots \wedge B_n$ einer Regel aus P gibt, mit $B_i \in I$, für alle $1 \leq i \leq n$.
4. Wenn $gnd(P)$ die Menge der Grundinstanzen ist, dann gilt:

$$Hres(P, I) = \{ A \mid A \leftarrow \beta \in gnd(P) \wedge I \models \beta \}.$$

Für ein festes P, schreibt man anstelle von $Hres(P, I)$ oft auch $Hres_P(I)$.

Hyperresolutionsmengen $Hres^n(P)$

Man kann auch wieder Potenzen $Hres^n(P)$ bilden:

$$\begin{aligned} Hres^0(P) &= \emptyset, \\ Hres^{n+1}(P) &= Hres(P, Hres^n(P)), \\ Hres^*(P) &= \textstyle\bigcup_{n \in \mathbb{N}_+} Hres^n(P). \end{aligned}$$

Offensichtlich gilt auch hier die Monotonie–Eigenschaft:

$$P \subseteq P' \;\wedge\; I \subseteq I' \;\Rightarrow\; Hres(P, I) \subseteq Hres(P', I').$$

Also kann man wieder durch vollständige Induktion folgendes zeigen:

$$Hres^n(P) \subseteq Hres^{n+1}(P).$$

Wir schreiben Fakten in den Hyperresolutionsmengen als Atome.

Beispiel (Hyperresolutionsmengen)

Für das DATALOG–Programm P mit den Regeln

$$anc(X, Y) \leftarrow par(X, Y),$$
$$anc(X, Y) \leftarrow par(X, Z) \wedge anc(Z, Y),$$

und den Eltern–Fakten $par(a, b),\ par(b, c),\ par(c, d)$, erhalten wir folgendes:

$$\begin{aligned}
Hres^0(P) &= \emptyset, \\
Hres^1(P) &= \{\, par(a, b),\ par(b, c),\ par(c, d) \,\}, \\
Hres^2(P) &= Hres^1(P) \cup \{\, anc(a, b),\ anc(b, c),\ anc(c, d) \,\}, \\
Hres^3(P) &= Hres^2(P) \cup \{\, anc(a, c),\ anc(b, d) \,\}, \\
Hres^n(P) &= Hres^3(P) \cup \{\, anc(a, d) \,\},\ \text{für } n \geq 4.
\end{aligned}$$

Es gilt $Hres^*(P) = Hres^4(P)$.

Die verwendeten Grundinstanzen aus $gnd\,(P)$ der Regeln sind folgende:

1. In Iteration 1 werden die Eltern–Fakten verwendet.
2. In Iteration 2 werden die folgenden Grundinstanzen verwendet:

$$anc(a,b) \leftarrow par(a,b),$$

$$anc(b,c) \leftarrow par(b,c),$$

$$anc(c,d) \leftarrow par(c,d).$$

3. In Iteration 3 werden die folgenden Grundinstanzen verwendet:

$$anc(a,c) \leftarrow par(a,b) \wedge anc(b,c),$$

$$anc(b,d) \leftarrow par(b,c) \wedge anc(c,d).$$

4. In Iteration 4 wird die folgende Grundinstanz verwendet:

$$anc(a,d) \leftarrow par(a,b) \wedge anc(b,d).$$

Hyperresolution für definite Logikprogramme

Man kann die Definition der Hyperresolutionsmenge $\mathit{Hres}(P, I)$ auf ein beliebiges definites Logikprogramm P anwenden,

$$\mathit{Hres}(P, I) = \{\, A \mid A \leftarrow \beta \in \mathit{gnd}\,(P) \;\wedge\; I \models \beta \,\},$$

selbst wenn P nicht bereichsbeschränkt ist oder Funktionssymbole enthält.

- Wenn ein Regelkopf Variablensymbole enthält, die nicht im Regelrumpf vorkommen, dann kann man diese in den zur Ableitung benutzten Grundinstanzen $A \leftarrow \beta \in \mathit{gnd}\,(P)$ mit $I \models \beta$ unabhängig von I mit allen Termen des Herbranduniversums HU_P instanziieren.
- Dann kann man manchmal unabhängig von I sehr viele Atome A ableiten.
- Wenn HU_P unendlich ist, dann kann das nicht mehr berechnet werden. In diesem Fall müßte man alternativ Nicht–Grundatome ableiten.

Beispiel (Hyperresolution ohne Bereichsbeschränktheit)

Für das definite Logikprogramm P mit den Regeln

$$anc(X, X),$$
$$anc(X, Y) \leftarrow par(X, Z) \land anc(Z, Y),$$

und den Eltern–Fakten $par(a, b),\ par(b, c),\ par(c, d)$, gibt es sehr viele Grundinstanzen des Faktes $anc(X, X)$, die man zur Ableitung benutzen kann, nämlich eines für jedes Element des Herbranduniversums $HU_P = \{\, a, b, c, d \,\}$:

$$anc(a, a),\ \ anc(b, b),\ \ anc(c, c),\ \ anc(d, d).$$

Diese sind unabhängig von I, da ihr leerer Rumpf β immer gilt.

Wenn das Herbranduniversum HU_P noch größer wäre, dann wären das entsprechend mehr.

Eigentlich müsste man die erste Regel von P mittels eines 1–stelligen Domänenprädikats $person$ *bereichsbeschränkt* machen:

$$anc(X,X) \leftarrow person(X),$$
$$anc(X,Y) \leftarrow par(X,Z) \wedge anc(Z,Y).$$

Falls der Herbranduniversum HU_P – aufgrund von weiteren Regeln in P, die wir hier nicht zeigen – neben den Konstanten a, b, c, d zu den Personen noch weitere Konstanten oder Terme enthält, so ist beim so erhaltenen DATALOG–Programm die Anwendung der ersten Regel auf Personen beschränkt. Diese könnte man wie folgt als alle Argumente X von par–Fakten definieren:

$$person(X) \leftarrow par(X,Y),$$
$$person(X) \leftarrow par(Y,X).$$

Die zweite Regel ist bereits bereichsbeschränkt. Man kann für

$$\begin{aligned} I \;=\; \{\, & par(a,b),\; par(b,c),\; par(c,d), \\ & anc(a,b),\; anc(b,c),\; anc(c,d)\,\} \end{aligned}$$

nur die folgenden Grundinstanzen der zweiten Regel zur Ableitung benutzen; diese hängen von I ab.

$$anc(a,c) \leftarrow par(a,b) \wedge anc(b,c),$$
$$anc(b,d) \leftarrow par(b,c) \wedge anc(c,d).$$

Die folgende Grundinstanz kann man dagegen für I nicht benutzen, da eines der beiden Rumpfatome – nämlich $par(a,c)$ – nicht in I liegt:

$$anc(a,d) \leftarrow par(a,c) \wedge anc(c,d).$$

Theorem 2.1 (Charakterisierung von Herbrand–Modellen)

Sei P ein definites Logikprogramm. Eine Herbrand–Interpretation I von P ist ein Herbrand–Modell von P, g.d.w. gilt $Hres(P, I) \subseteq I$.

Beweis:

“$\Leftarrow$” Sei I eine Herbrand–Interpretation von P mit $Hres(P, I) \subseteq I$.
Sei $A \leftarrow \beta$ eine Grundinstanz einer Regel aus P mit $I \models \beta$.
Dann gilt $A \in Hres(P, I)$, und wegen $Hres(P, I) \subseteq I$ folgt $A \in I$.
Also ist I ein Herbrand–Modell von P.

“$\Rightarrow$” Sei I ein Herbrand–Modell von P.
Für jedes $A \in Hres(P, I)$ gibt es eine Grundinstanz einer Regel $A \leftarrow \beta$ aus P mit $I \models \beta$. Da I ein Herbrand–Modell von P ist, folgt sofort $A \in I$. Also gilt $Hres(P, I) \subseteq I$. $\square$

Minimales Herbrand–Modell $\mathcal{M}_P$

Für ein definites Logikprogramm P kann man entsprechend auch die Hyperresolutionsmengen $\mathit{Hres}^n(P)$ und $\mathit{Hres}^*(P)$ definieren.

Dann kann man für die Herbrand–Modelle und für $\mathcal{M}_P = \mathit{Hres}^*(P)$ folgendes zeigen:

1. Jedes Herbrand–Modell muß $\mathcal{M}_P$ enthalten.

2. $\mathcal{M}_P$ ist ein Herbrand–Modell. Dies gilt, da $\mathit{Hres}(P, \mathcal{M}_P) \subseteq \mathcal{M}_P$.

Also ist $\mathcal{M}_P$ das eindeutige minimale Herbrand–Modell von P.

Für DATALOG–Programme ist $\mathcal{M}_P$ endlich, und es kann effizient berechnet werden. Im allgemeinen kann $\mathcal{M}_P$ unendlich sein.

Beispiel (Minimales Herbrand–Modell)

Für das DATALOG$^{\mathrm{fun}}$–Programm P mit den Regeln

$$natural(0),$$
$$natural(s(N)) \leftarrow natural(N),$$

zu den natürlichen Zahlen, welches ein 1–stelliges Funktionssymbol s zur Codierung des Nachfolgers (successor) einer natürlichen Zahl N enthält, ist das minimale Herbrand–Modell $\mathcal{M}_P$ unendlich, und es wird durch die folgenden Hyperresolutionsmengen approximiert:

$$\begin{aligned} Hres^0(P) &= \emptyset, \\ Hres^1(P) &= \{\, natural(0)\, \}, \\ Hres^n(P) &= \{\, natural(s^m(0)) \,|\, 0 \leq m \leq n-1 \,\}, \text{ für } n \in I\!N_0. \end{aligned}$$

Es gilt $\mathcal{M}_P = Hres^*(P) = \{\, natural(s^m(0)) \,|\, m \in I\!N_0 \,\}$.

2.7 SLD–Resolution

Die SLD–Resolution (Linear Resolution with Selection Function for Definite Clauses) wurde ursprünglich zum automatischen Theorembeweisen für allgemeine Klauseln verwendet (vgl. J.A. Robinson, 1965). Wir verwenden sie hier für definite Regeln und Goals.

SLD–Resolvente

Sei s eine Selektionsfunktion, die aus jedem Goal ein Atom auswählt.

- Sei $G = \leftarrow \beta_1 \wedge B \wedge \beta_2$ ein Goal und $r = A \leftarrow \beta$ eine definite Regel. Dabei seien β_1 und β_2 Konjunktionen von Atomen und s selektiere das Atom $s(G) = B$.
- Falls $\text{M}{\small\text{GU}}(A, B) = \theta$ existiert, dann heißt $G' = \leftarrow (\, \beta_1 \wedge \beta \wedge \beta_2 \,)\, \theta$ eine *SLD–Resolvente* von G und r bezüglich s.

Falls r ein Fakt ist (d.h. β ist leer), dann ist G' um ein Atom kürzer als G.

Motivation durch die binäre Resolution der PL1

Die SLD–Resolution ergibt sich aus der binären Resolution der PL1.

Bekanntlich ist ein Goal $G = \leftarrow \beta_1 \wedge B \wedge \beta_2$ äquivalent zu einer Disjunktion

$$\alpha \;=\; \neg\beta_1 \vee \neg B \vee \neg\beta_2,$$

und eine Regel $r = A \leftarrow \beta$ ist äquivalent zu einer Disjunktion

$$\delta \;=\; A \vee \neg\beta.$$

Durch die bekannte binäre Resolution erhält man mittels eines allgemeinsten Unifikators θ von A und B eine neue Disjunktion α', die wieder einem Goal $G' = \leftarrow (\ \beta_1 \wedge \beta \wedge \beta_2\ )\,\theta$ entspricht:

$$\alpha' \;=\; (\neg\beta_1 \vee \neg\beta \vee \neg\beta_2)\theta.$$

SLD–Widerlegung

Sei P ein definites Logikprogramm und Q eine Anfrage.

- Eine *SLD–Widerlegung* von $\langle\, P, Q \,\rangle$ ist eine endliche Folge von Goals $\langle\, G_i \,\rangle_{0 \leq i \leq n}$, wobei $G_0 = Q$ und $G_n = \Box$, zusammen mit einer Folge $\langle\, r_i \,\rangle_{1 \leq i \leq n}$ von Regeln und einer Folge $\langle\, \theta_i \,\rangle_{1 \leq i \leq n}$ von Substitutionen, wobei für alle $1 \leq i \leq n$ gilt:
  - r_i ist eine Variante einer Regel aus P,
  - G_i ist eine SLD–Resolvente aus G_{i-1} und r_i bezüglich s, und
  - θ_i ist der benutzte allgemeinste Unifikator.

  $\theta = \theta_1 \cdot \theta_2 \cdot \ldots \cdot \theta_n$ heißt *Antwortsubstitution*.

- Die SLD–Widerlegung einer atomaren Anfrage $Q = \;\leftarrow A$ ist äquivalent zu dem Beweis, daß die Instanz $A\theta$, welche man *Antwort* nennt, aus P abgeleitet werden kann.

- Ein *SLD–Widerlegungsbaum* veranschaulicht eine SLD–Widerlegung:

$$
\begin{tikzcd}
G_0 \arrow[d] & r_1 \arrow[dl] \\
G_1 \arrow[d] & r_2 \arrow[dl] \\
G_2 \arrow[d] & r_3 \arrow[dl] \\
G_3 & \vdots \\
\vdots \arrow[d] & r_n \arrow[dl] \\
G_n = \Box &
\end{tikzcd}
$$

Man kann eine SLD–Widerlegung natürlich auch einfach als eine lineare Kette $G_0 \xrightarrow{r_1} G_1 \xrightarrow{r_2} G_2 \xrightarrow{r_3} \ldots \xrightarrow{r_n} G_n = \Box$ repräsentieren, wenn man die Kanten mit den verwendeten Regeln markiert.

Beispiel (SLD–Resolution)

Für das DATALOG–Programm $P = \{\, r,\ f_1,\ f_2 \,\}$, mit

$$\begin{aligned} r &= p \leftarrow a \wedge b, \\ f_1 &= a, \\ f_2 &= b, \end{aligned}$$

erhalten wir mit der Selektionsfunktion s_l, welche stets das linkeste (erste) Atom eines Goals selektiert, eine SLD–Widerlegung zum Goal $Q = \ \leftarrow p$.

$$\begin{tikzcd}
\leftarrow p \arrow[d] & r \arrow[dl] \\
\leftarrow a \wedge b \arrow[d] & f_1 \arrow[dl] \\
\leftarrow b \arrow[d] & f_2 \arrow[dl] \\
\Box &
\end{tikzcd}$$

In Klauselschreibweise

$$\begin{aligned} r &= p \vee \neg a \vee \neg b, \\ f_1 &= a, \\ f_2 &= b, \end{aligned}$$

erhalten wir folgende Folge von binären Resolutionen zum Goal $Q = \neg p$:

$$\begin{tikzcd}
\neg p \arrow[d] & r \arrow[dl] \\
\neg a \vee \neg b \arrow[d] & f_1 \arrow[dl] \\
\neg b \arrow[d] & f_2 \arrow[dl] \\
\Box &
\end{tikzcd}$$

Wir würden für das DATALOG–Programm $P = \{\, r,\ f_1,\ f_2 \,\}$, mit

$$
\begin{aligned}
r &= p \leftarrow a \wedge b, \\
f_1 &= a, \\
f_2 &= b,
\end{aligned}
$$

zum Vergleich die folgenden Hyperresolutionsmengen erhalten:

$$
\begin{aligned}
\mathit{Hres}^0(P) &= \emptyset, \\
\mathit{Hres}^1(P) &= \{\, a,\ b \,\}, \\
\mathit{Hres}^n(P) &= \mathit{Hres}^1(P) \cup \{\, p \,\},\ \text{für } n \geq 2.
\end{aligned}
$$

Die obige Antwort p aus der SLD–Resolution liegt in $\mathit{Hres}^2(P)$.

Nicht–Lineare Resolution

Für die inkonsistente Klauselmenge

$$\mathcal{F} = \{\, p \lor q,\ p \lor \neg q,\ \neg p \lor q,\ \neg p \lor \neg q \,\}$$

gibt es keine lineare Widerlegung, aber folgende Resolutionsableitung:

$$\begin{tikzcd}
p \lor q \arrow[dr] & & p \lor \neg q \arrow[dl] & & \neg p \lor q \arrow[dr] & & \neg p \lor \neg q \arrow[dl] \\
& p \arrow[drr] & & & & \neg p \arrow[dll] & \\
& & & \square & & &
\end{tikzcd}$$

Diese Klauselmenge ist keine Hornklauselmenge.

Für eine inkonsistente Hornklauselmenge gibt es immer eine lineare Ableitung der leeren Klausel, die mit einem Goal (Integritätsbedingung) startet.

Alternative Notation für Substitutionen

Wir schreiben eine Substitution $[X_1|t_1, \ldots X_n|t_n]$, bei der alle Ersetzungen parallel angewendet werden, als Menge $\theta = \{ X_1 \mapsto t_1, \ldots, X_n \mapsto t_n \}$.

θ definiert eine Abbildung von einer Struktur t auf eine neue Struktur $t\theta$:

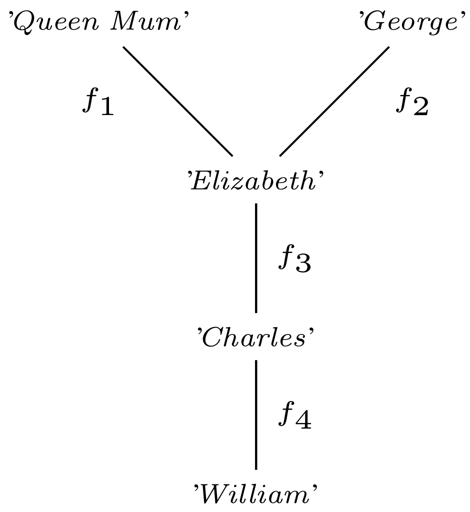
- Für eine Konstante t gilt $t\theta = t$.
- Für eine Variable $t = X$ gilt: falls es eine Ersetzung $X_i \mapsto t_i \in \theta$ gibt, mit $X = X_i$, so erhalten wir $t\theta = t_i$, und ansonsten $t\theta = t$.
- Für eine komplexe Struktur gilt $f(t_1, \ldots, t_n)\theta = f(t_1\theta, \ldots, t_n\theta)$.

Dann entspricht die Substitution $\theta_1 = [X|Z][Z|Y]$ der Substitution $[X|Y, Z|Y]$, da X zuerst durch Y und dieses dann durch Z ersetzt wird.
Sie ist damit verschieden von $\theta_2 = [X|Z, Z|Y]$.
Für $t = f(X, Y, Z)$ gilt z.B. $t\theta_1 = f(Y, Y, Y)$ und $t\theta_2 = f(Z, Y, Y)$.

Beispiel (SLD–Resolution)

Wir betrachten das nicht–rekursive, definite Logikprogramm $P = \{ r,\ f_1,\ f_2,\ f_3,\ f_4 \}$ und die Anfrage $Q = \ \leftarrow grandparent(X, Z)$:

$$\begin{aligned} r &= grandparent(X, Z) \leftarrow parent(X, Y) \wedge parent(Y, Z), \\ f_1 &= parent(\text{'}Elizabeth\text{'}, \text{'}Queen\ Mum\text{'}), \\ f_2 &= parent(\text{'}Elizabeth\text{'}, \text{'}George\text{'}), \\ f_3 &= parent(\text{'}Charles\text{'}, \text{'}Elizabeth\text{'}), \\ f_4 &= parent(\text{'}William\text{'}, \text{'}Charles\text{'}). \end{aligned}$$



Aus $G_0 = Q$ erhält man mittels r und $\theta_1 = \emptyset$ die Resolvente

$$G_1 = \ \leftarrow parent(X, Y) \wedge parent(Y, Z).$$

Daraus kann man mit der Selektionsfunktion s_l, welche stets das linkeste Literal eines Goals auswählt, z.B. folgende SLD–Widerlegungen erhalten:

- Mittels f_3 und $\theta_2 = \{ X \mapsto \text{'}Charles\text{'}, Y \mapsto \text{'}Elizabeth\text{'} \}$ erhalten wir

  $$G_2 = \ \leftarrow parent(\text{'}Elizabeth\text{'}, Z).$$

  – Mittels f_1 und $\theta_3 = \{ Z \mapsto \text{'}Queen\ Mum\text{'} \}$ erhalten wir $G_3 = \Box$ und die Antwortsubstitution

  $$\theta = \{ X \mapsto \text{'}Charles\text{'}, Y \mapsto \text{'}Elizabeth\text{'}, Z \mapsto \text{'}Queen\ Mum\text{'} \}.$$

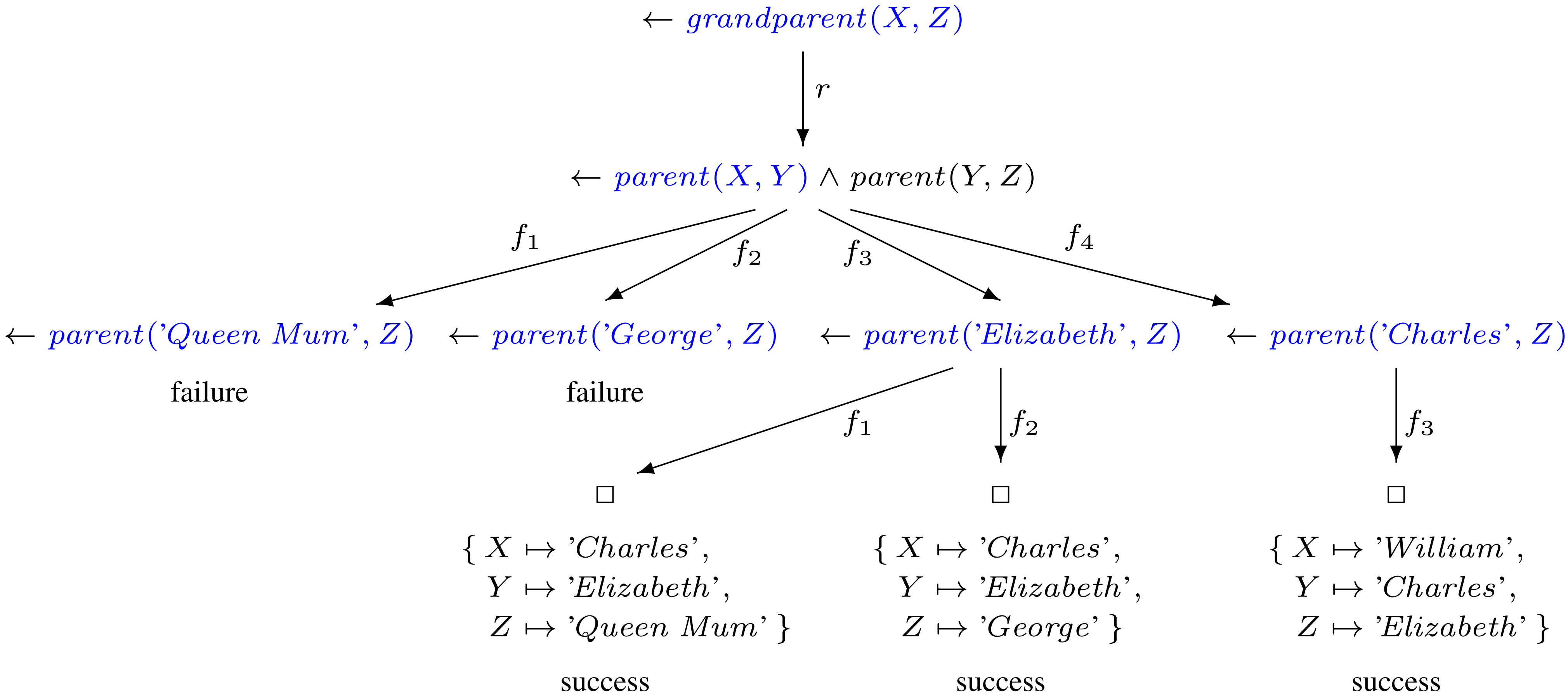
  – Mittels f_2 und $\theta_3 = \{ Z \mapsto \text{'}George\text{'} \}$ erhalten wir ebenfalls $G_3 = \Box$; nun ist die Antwortsubstitution

  $$\theta' = \{ X \mapsto \text{'}Charles\text{'}, Y \mapsto \text{'}Elizabeth\text{'}, Z \mapsto \text{'}George\text{'} \}.$$

- Mittels f_4 und f_3 erhalten wir aus G_1 die Antwortsubstitution

  $$\theta'' = \{ X \mapsto \text{'}William\text{'}, Y \mapsto \text{'}Charles\text{'}, Z \mapsto \text{'}Elizabeth\text{'} \}.$$

Wir können alle SLD–Widerlegungen im SLD–Baum $\mathcal{T}^{s_l}_{P,Q}$ zusammenfassend veranschaulichen. An den Blättern werden die jeweiligen Antwortsubstitutionen angegeben.



SLD–Bäume

Sei P ein definites Logikprogramm, Q eine Anfrage und s eine Selektionsfunktion.

Der SLD–Baum $\mathcal{T}^s_{P,Q}$ faßt alle möglichen SLD–Widerlegungen zusammen:

- Die Knoten sind mit Goals markiert,
  und die Kanten sind mit Regeln $r \in P$ markiert.
- Die Wurzel ist mit der Anfrage Q markiert.
- Ein Knoten v, der mit einem Goal G markiert ist, hat für jede SLD–Resolvente G', die man aus G und einer Regel $r \in P$ bezüglich s bilden kann, einen Nachfolgerknoten v'.
- Dieser ist mit G' markiert, und die Kante $\langle\, v, v' \,\rangle$ ist mit r markiert.

Jeder mit dem Goal $G = \Box$ endende Ast eines SLD–Baumes entspricht einer SLD–Widerlegung bzw. einem SLD–Widerlegungsbaum:

- Die Blätter dieser Äste werden als erfolgreich (*success*) markiert.
- Alle anderen Blätter werden als erfolglos (*failure*) markiert.
- Daneben kann es noch *unendliche Äste* im Baum geben.

Zwei SLD–Bäume $\mathcal{T}^{s_1}_{P,Q}$ und $\mathcal{T}^{s_2}_{P,Q}$ zu unterschiedlichen Selektionsfunktionen s_1, s_2 haben immer dieselben Antworten (success–Äste).

Sie unterscheiden sich aber in der Regel hinsichtlich ihrer failure–Äste bzw. ihrer unendlichen Äste.

Vollständigkeit und Korrektheit der SLD–Resolution

Sei P ein definites Logikprogramm und $Q = \leftarrow A$ eine atomare Anfrage.

1. Es gibt eine SLD–Widerlegung mit der Antwort $A\theta$, genau dann wenn alle Grundinstanzen von $A\theta$ im minimalen Modell $\mathcal{M}_P$ von P liegen.
2. Dieses Resultat ist unabhängig von der verwendeten Selektionsfunktion s.

PROLOG verwendet immer die Selektionsfunktion s_l, welche stets das linkeste (erste) Literal eines Goals auswählt. Durch Umstellung der Atome in den Regelrümpfen kann der Programmierer de facto eine andere Selektionsfunktion simulieren.

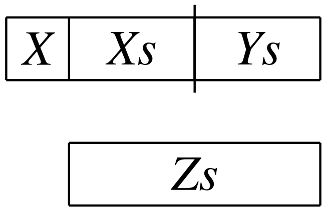
Die Abarbeitung von PROLOG–Programmen verwendet Tiefensuche und Backtracking im SLD–Baum.

Listenkonkatenation in der Logikprogrammierung

In der Logik dienen Terme als Datenstruktur. Eine Liste $[a, b, c]$ wird als Term $list(a, list(b, list(c, nil)))$ repräsentiert. Dabei ist $list$ ein 2–stelliges Funktionssymbol, und nil ist ein 0–stelliges Funktionssymbol, welches das Listenende markiert.

Die folgenden Regeln können zwei Listen Xs und Ys mittels des Aufrufs $\leftarrow append(Xs, Ys, Zs)$ zu einer Liste Zs konkatenieren:

$$
\begin{array}{l}
append(list(X, Xs), Ys, list(X, Zs)) \leftarrow \\
\quad append(Xs, Ys, Zs), \\
append(nil, Ys, Ys).
\end{array}
$$



In der gezeigten kompakten Fassung der ersten Regel erfolgt die Listenzerlegung und der Listenaufbau in der Kopfzeile:

$$\begin{array}{l} \mathit{append}(\mathit{list}(X, Xs), Ys, \mathit{list}(X, Zs)) \leftarrow \\ \quad \mathit{append}(Xs, Ys, Zs). \end{array}$$

Die folgende ausführlichere Fassung hat dafür separate Zeilen. Außerdem verdeutlicht die funktionale Notation $Zs <= \mathit{append}(Xs, Ys)$ des Systems `Declare`, daß Zs durch die Konkatenation von Xs und Ys berechnet wird:

$$\begin{array}{ll} \mathit{append}(Xs', Ys, Zs') \leftarrow & \\ \quad Xs' = \mathit{list}(X, Xs) \wedge & \text{% Listenzerlegung} \\ \quad Zs <= \mathit{append}(Xs, Ys) \wedge & \\ \quad Zs' = \mathit{list}(X, Zs). & \text{% Listenaufbau} \end{array}$$

In der relationalen Notation $\mathit{append}(Xs, Ys, Zs)$ war die Ausgabe Zs dagegen lediglich das letzte Argument – und ansonsten gleichberechtigt mit der Eingabe Xs und Ys.

Beispiel (SLD–Widerlegungsbaum)

$$\begin{array}{cl}
\leftarrow app(list(a, list(b, nil)), list(c, nil), Zs) & r_1 \\
\downarrow \quad \theta_1 & \\
\leftarrow app(list(b, nil), list(c, nil), Zs_1) & r_2 \\
\downarrow \quad \theta_2 & \\
\leftarrow app(nil, list(c, nil), Zs_2) & f_3 \\
\downarrow \quad \theta_3 & \\
\square &
\end{array}$$

$$\begin{array}{lcl}
r_1 & = & app(list(X_1, Xs_1), Ys_1, list(X_1, Zs_1)) \leftarrow app(Xs_1, Ys_1, Zs_1), \\
r_2 & = & app(list(X_2, Xs_2), Ys_2, list(X_2, Zs_2)) \leftarrow app(Xs_2, Ys_2, Zs_2), \\
f_3 & = & app(nil, Ys_3, Ys_3).
\end{array}$$

$$\begin{array}{lcl}
\theta_1 & = & \{ X_1 \mapsto a,\ Xs_1 \mapsto list(b, nil),\ Ys_1 \mapsto list(c, nil),\ Zs \mapsto list(a, Zs_1) \}, \\
\theta_2 & = & \{ X_2 \mapsto b,\ Xs_2 \mapsto nil,\ Ys_2 \mapsto list(c, nil),\ Zs_1 \mapsto list(b, Zs_2) \}, \\
\theta_3 & = & \{ Ys_3 \mapsto list(c, nil),\ Zs_2 \mapsto Ys_3 = list(c, nil) \}.
\end{array}$$

In obigem SLD–Widerlegungsbaum wurde das Prädikatensymbol $append$ mit app abgekürzt.

Im Endergebnis $\theta = \theta_1 \cdot \theta_2 \cdot \theta_3$ gilt

$$\begin{aligned} Zs &\mapsto list(a, Zs_1) \mapsto list(a, list(b, Zs_2)) \\ &\mapsto list(a, list(b, list(c, nil))). \end{aligned}$$

Die berechnete Antwort ist also

$$\begin{aligned} &append(list(a, list(b, nil)), list(c, nil), Zs)\theta = \\ &append(list(a, list(b, nil)), list(c, nil), list(a, list(b, list(c, nil)))), \end{aligned}$$

denn die Konkatenation der Listen $[a, b]$ und $[c]$ liefert die Liste $[a, b, c]$.