# Unifikator

**Definition**:
Sei $\mathcal{S}$ eine Signatur. Ein Ausdruck $A$ heißt Literal über $\mathcal{S}$, falls er von der Form $R(t_1, t_2, \ldots, t_k)$ oder $\neg R(t_1, t_2, \ldots, t_k)$ ist, wobei $R$ ein Relationssymbol und $t_1, t_2, \ldots t_k$ Terme über $\mathcal{S}$ sind.

**Definition**:
i) Eine Substitution $s$ heißt Unifikator der Menge $\mathcal{L} = \{L_1, L_2, \ldots L_r\}$ von Literalen, falls $s(L_1) = s(L_2) = \ldots = s(L_r)$ gelten.

ii) Eine Substitution $s$ heißt allgemeinster Unifikator von $\mathcal{L}$, falls für jeden Unifikator $s'$ von $\mathcal{L}$ eine Substitution $s''$ mit $s' = s \circ s''$ existiert.

**Satz**:
Jede unifizierbare Menge von Literalen besitzt einen allgemeinsten Unifikator.

# Algorithmus für den allgemeinsten Unifikator

*Eingabe:* nichtleere Menge $\mathcal{L} = \{L_1, L_2, \ldots, L_r\}$ von Literalen

$s = id$;

`while` $(s(L_i) \neq s(L_j)$ für gewisse $1 \leq i < j \leq k)$

{ Durchsuche die Literale von $s(L_i)$ und $s(L_j)$ von links nach rechts,
bis erste Position $a$ gefunden ist, an der sich mindestens
zwei Literale unterscheiden;

`if` (keines der Zeichen ist eine Variable)

Stoppe mit "$\mathcal{L}$ ist nicht unifizierbar";

`else` { $x =$ Variable in $a$; $t =$ Term, der in $a$ beginnt;

`if` ($x$ kommt in $t$ vor)

Stoppe mit "$\mathcal{L}$ ist nicht unifizierbar";

`else` $s = s \circ [x/t]$;

}

}

Gib $s$ als allgemeinsten Unifikator aus

# Unifikationsalgorithmus – Beispiel

Literale: $L_1 = \neg P(f(z, g(a, y)), h(z))$
$L_2 = \neg P(f(f(u, v), w), h(f(a, b)))$

Unterschied an der sechsten Position: Substitution $s_1 = [z/f(u, v)]$
$s_1(L_1) = \neg P(f(f(u, v), g(a, y)), h(f(u, v)))$
$s_1(L_2) = \neg P(f(f(u, v), w), h(f(a, b)))$

Unterschied an der elften Position: Substitution $s_2 = [w/g(a, y)]$
$s_2(s_1(L_1)) = \neg P(f(f(u, v), g(a, y)), h(f(u, v)))$
$s_2(s_1(L_2)) = \neg P(f(f(u, v), g(a, y)), h(f(a, b)))$

# Unifikationsalgorithmus – Beispiel – Fortsetzung

Literale: $s_2(s_1(L_1)) = \neg P(f(f(u,v),g(a,y)),h(f(u,v)))$
$s_2(s_1(L_2)) = \neg P(f(f(u,v),g(a,y)),h(f(a,b)))$

Unterschied bei sechstletzten Buchstaben: Substitution $s_3 = [u/a]$
$s_3(s_2(s_1(L_1))) = \neg P(f(f(a,v),g(a,y)),h(f(a,v)))$
$s_3(s_2(s_1(L_2))) = \neg P(f(f(a,v),g(a,y)),h(f(a,b)))$

Unterschied bei viertletzten Buchstaben: Substitution $s_4 = [v/b]$
$s_4(s_3(s_2(s_1(L_1)))) = \neg P(f(f(a,b),g(a,y)),h(f(a,b)))$
$s_4(s_3(s_2(s_1(L_2)))) = \neg P(f(f(a,b),g(a,y)),h(f(a,b)))$

allgemeinsten Unifikator von $L_1$ und $L_2$:
$s_1 \circ s_2 \circ s_3 \circ s_4 = [z/f(u,v)] \circ [w/g(a,y)] \circ [u/a] \circ [v/b]$

# Prädikatenlogische Resolution – Definition I

**Definition**:
Es seien $\mathcal{K}_1$, $\mathcal{K}_2$ und $\mathcal{R}$ Mengen von prädikatenlogischen Literalen. Dann heißt $\mathcal{R}$ prädikatenlogische Resolvente von $\mathcal{K}_1$ und $\mathcal{K}_2$, falls folgende Bedingungen erfüllt sind:

- Es gibt Substitutionen $s_1$ und $s_2$, die nur Variablenumbenennugen sind, so dass $s_1(\mathcal{K}_1)$ und $s_2(\mathcal{K}_2)$ keine gemeinsamen Variablen haben.

- Es gibt Literale $L_1, \ldots, L_m \in s_1(\mathcal{K}_1)$, $m \geq 1$, und $L'_1, \ldots, L'_n \in s_2(\mathcal{K}_2)$, $n \geq 1$, so dass die Menge $\mathcal{L} = \{\neg L_1, \neg L_2, \ldots, \neg L_m, L'_1, L'_2, \ldots, L'_n\}$ unifizierbar ist.
  $s$ sei der allgemeinste Unifikator von $\mathcal{L}$.

- Es gilt $\mathcal{R} = s((s_1(\mathcal{K}_1) \setminus \{L_1, L_2, \ldots, L_m\}) \cup (s_2(\mathcal{K}_2) \setminus \{L'_1, L'_2, \ldots, L'_n\}))$

# Prädikatenlogische Resolution – Definition II

**Definition**:
Für eine Menge $\mathcal{F}$ von Mengen von Literalen setzen wir

$$\begin{aligned}
Res(\mathcal{F}) &= \mathcal{F} \cup \{\mathcal{R} \mid \mathcal{R} \text{ ist Resolvente gewisser } \mathcal{K} \in \mathcal{F} \text{ und } \mathcal{K}' \in \mathcal{F}\}\,, \\
Res^0(\mathcal{F}) &= \mathcal{F}\,, \\
Res^n(\mathcal{F}) &= Res(Res^{n-1}(\mathcal{F})) \text{ für } n \geq 1 \text{ und} \\
Res^*(\mathcal{F}) &= \bigcup_{n \geq 0} Res^n(\mathcal{F})\,.
\end{aligned}$$

# Prädikatenlogische Resolution – Resultate

**Lemma**: (Lifting-Lemma) Seien $K_1$ und $K_2$ zwei prädikatenlogische Klauseln und $K_1'$ und $K_2'$ zugehörige (beliebige) Grundinstanzen. Ferner sei $R'$ eine (aussagenlogische) Resolvente von $K_1'$ und $K_2'$. Dann gibt es eine prädikatenlogische Resolvente $R$ von $K_1$ und $K_2$ so, dass $R'$ Grundinstanz von $R$ ist.

$$\begin{array}{ccccccc}
K_1 & & K_2 & \quad & K_1 & & K_2 \\
 & & & & & & \\
K_1' & & K_2' & & & R & \\
 & & & & & & \\
 & R' & & & & R' &
\end{array}$$

**Satz**: Ein prädikatenlogischer Ausdruck $A = \forall x_1 \forall x_2 \ldots \forall x_n A'$ in bereinigter Skolemform, bei dem $A'$ in konjunktiver Normalform vorliegt, ist $A$ genau dann unerfüllbar, wenn die leere Menge in $Res^*(A)$ liegt.

# Lineare Resolutionen I

**Definition:** Die Resolution einer Klausel $R$ aus einer Klauselmenge $\mathcal{K}$ heißt linear, falls es Klauseln $R_0, R_1, R_2, \ldots, R_n$ so gibt, dass

$$\begin{aligned} R_0 &\in \mathcal{K}, \\ R_i &\in res(R_{i-1}, C_{i-1}) \\ &\qquad \text{mit } C_{i-1} \in \mathcal{K} \cup \{R_1, R_2, \ldots, R_{i-1}\},\ 1 \leq i \leq n \end{aligned}$$

gelten.

**Satz:** Sei $A = \forall x_1 \forall x_2 \ldots \forall x_n A'$ ein prädikatenlogischer Ausdruck in bereinigter Skolemform, bei dem $A'$ in konjunktiver Normalform vorliegt. Dann ist $A$ genau dann unerfüllbar, wenn es eine lineare Resolution für die leere Menge aus der Klauselmenge zu $A'$ gibt.

# Lineare Resolutionen II

**Definition:** Eine Klauselmenge $\mathcal{K}$ heißt minimal unerfüllbar, wenn sie unerfüllbar ist und für jede Klausel $K \in \mathcal{K}$ die Menge $\mathcal{K} \setminus \{K\}$ erfüllbar ist.

**Satz:** Sei $A = \forall x_1 \forall x_2 \ldots \forall x_n A'$ ein prädikatenlogischer Ausdruck in bereinigter Skolemform, bei dem $A'$ in konjunktiver Normalform vorliegt. Ferner sei die zu $A'$ gehörende Klauselmenge minimal unerfüllbar. Dann gibt es für jede Klausel $K$ von $A'$ eine lineare Resolution für die leere Menge aus der Klauselmenge zu $A'$, bei der $R_0 = K$ gilt.

# SDL-Resolutionen

**Definition:** i) Wir sagen, dass eine Klausel negativ ist, wenn alle Literale negierte Basisausdrücke sind. Eine Klausel heißt definit, wenn genau ein Literal ein nichtnegierter Basisausdruck ist.

ii) Eine lineare Resolution heißt SLD–Resolution, falls $R_0$ eine negative Klausel ist und $C_{i-1}$ für $1 \leq i \leq n$ eine definite Klausel ist.

**Definition:** Ein quantorenfreier prädikatenlogischer Ausdruck $A'$ in konjunktiver Normalform heißt Hornausdruck, falls jede Alternative höchstens einen nichtnegierten Basisausdruck enthält.

**Satz:** Sei $A = \forall x_1 \forall x_2 \ldots \forall x_n A'$ ein prädikatenlogischer Ausdruck in bereinigter Skolemform, bei dem $A'$ ein Hornausdruck ist. Dann ist $A$ genau dann unerfüllbar, wenn es eine SLD-Resolution für die leere Menge aus der Klauselmenge zu $A'$ gibt.

# Logik-Programme

**Definition**:

Eine Tatsachenklausel ist eine einelementige positive Klausel, d.h. sie hat die Form $\{P\}$.

Eine Prozedurklausel ist eine Klausel der Form $\{P, \neg Q_1, \neg Q_2, \ldots, \neg Q_k\}$ mit $k \geq 1$.
$P$ hei+t Prozedurkopf, und $Q_1, Q_2, \ldots, Q_k$ bilden den Prozedurkörper.

Ein Logik-Programm ist eine endliche Menge von Tatsachen- und Prozedurklauseln.

Eine Zielklausel ist eine Klausel der Form $\{\neg Q_1, \neg Q_2, \ldots, \neg Q_k\}$ mit $k \geq 1$.

# Konfigurationen und ihre Übergänge

**Definition**: Es sei $F$ ein Logik-Programm.

i) Eine Konfiguration ist ein Paar $(G, sub)$, wobei $G$ eine Zielklausel und $sub$ eine Substitution ist.

ii) Wir sagen, dass die Konfiguration $(G, sub)$ bez. $F$ in die Konfiguration $(G', sub')$ überführt wird (und schreiben $(G, sub) \vdash_F (G', sub')$), falls folgende Bedingungen erfüllt sind:

— $G = \{\neg Q_1, \neg Q_2, \ldots, \neg Q_k\}$

— es gibt in $F$ eine Klausel $K = \{P, \neg A_1, \neg A_2, \ldots, \neg A_n\}$, $n \geq 0$, und ein $i$, $1 \leq i \leq n$, so dass $B$ (nach einigen Umbenennungen) mit $Q_i$ unifizierbar ist,

— $G' = s(\{\neg Q_1, \ldots, \neg Q_{i-1}, \neg A_1, \ldots, \neg A_n, \neg Q_{i+1}, \ldots, \neg Q_k\}$, wobei $s$ der allgemeinste Unifikator von $B$ und $Q_i$ ist,

— $sub' = sub \circ s$.

# Berechnungen

**Definition**:

Es seien $F$ ein Logik-Programm und $G = \{\neg Q_1, \ldots, \neg Q_k\}$ eine Zielklausel.

i) Eine Berechnung von $F$ bei Eingabe von $G$ ist eine Folge der Form

$$(G, id) \vdash_F (G_1, sub_1) \vdash (G_2, sub_2) \vdash_F \ldots \vdash_F (G_n, sub_n) \vdash_F \ldots .$$

ii) Falls eine Rechnung endlich ist und für das letzte Glied $(G_n, sub)$ der Folge $G_n = \emptyset$ gilt, so heißt die Berechnung erfolgreich und $sub(Q_1 \wedge Q_2 \wedge \ldots \wedge Q_k)$ ist das Ergebnis der Rechnung.

$n$ ist die Länge der Berechnung.

# Korrektheit und Vollständigkeit

**Satz**:
Seien $F$ ein Logik-Programm und $G$ eine Zielklausel.
Falls es eine erfolgreiche Rechnung von $F$ bei Eingabe von $G$ gibt, so ist jede Grundinstanz des Rechenergebnisses eine Folgerung von $F$.

**Satz**:
Seien $F$ ein Logik-Programm und $G = \{\neg Q_1, \ldots, \neg Q_k\}$ eine Zielklausel.
Falls jede Grundinstanz von $(Q_1 \wedge \ldots \wedge Q_k)$eine Folgerung von $F$ ist, so gibt es eine erfolgreiche Rechnung von $F$ bei Eingabe von $G$ mit dem Ergebnis $sub(Q_1 \wedge Q_2 \wedge \ldots Q_k)$, und für jede Grundinstanz $sub'(Q_1 \wedge Q_2 \wedge \ldots Q_k)$ gibt es eine Substitution $s$ mit

$$sub'(Q_1 \wedge Q_2 \wedge \ldots Q_k) = s(sub(Q_1 \wedge Q_2 \wedge \ldots Q_k)) .$$

# Vertauschungslemma I

Seien

$$C = \{\neg C_1, \neg C_2, \ldots, \neg C_r\} \text{ und } E = \{\neg E_1, \neg E_2, \ldots, \neg E_s\}$$

mit $r \geq 0$ und $s \geq 0$ und eine Resolution

$$\begin{array}{lll}
\{\neg A_1, \neg A_2, \ldots \neg A_n\} & & \{B\} \cup C \\
\qquad\qquad\downarrow & \swarrow & \\
sub_1(\{\neg A_1, \ldots, \neg A_{i-1}, C, \neg A_{i+1}, \ldots \neg A_n\}) & & \{D\} \cup E \\
\qquad\qquad\downarrow & \swarrow & \\
sub_2(sub_1(\{\neg A_1, \ldots, \neg A_{i-1}, C, \neg A_{i+1}, \ldots \neg A_{j-1}, E, \neg A_{j+1}, \ldots \neg A_n\})) & &
\end{array}$$

gegeben.

# Vertauschungslemma II

Dann gibt es auch die Resolution

$$\begin{array}{lcl}
\{\neg A_1, \neg A_2, \dots \neg A_n\} & & \{D\} \cup E \\
\qquad\downarrow & \swarrow & \\
sub_1(\{\neg A_1, \dots, \neg A_{j-1}, C, \neg A_{j+1}, \dots \neg A_n\}) & & \{B\} \cup C \\
\qquad\downarrow & \swarrow & \\
sub_2(sub_1(\{\neg A_1, \dots, \neg A_{i-1}, C, \neg A_{i+1}, \dots \neg A_{j-1}, E, \neg A_{j+1}, \dots \neg A_n\})) & &
\end{array}$$

wobei sogar bis auf Variablenbenennungen

$$sub_1' \circ sub_2' = sub_1 \circ sub_2$$

gilt.

# Kanonische Berechnungen

**Definition**:

Seien $F$ ein Logik-Programm und $G$ eine Zielklausel.
Eine Rechnung von $F$ bei Eingabe von $G$ heißt kanonisch, falls in jeder Konfigurationsüberführung $(G', sub') \vdash_F (G'', sub'')$ der Rechnung nach dem ersten (d.h. dem am weitesten links stehenden) Literal von $G'$ resolviert wird.

**Satz**:

Seien $F$ ein Logik-Programm und $G$ eine Zielklausel.
Falls es eine erfolgreiche Rechnung $\mathcal{R}$ von $F$ bei Eingabe von $G$ gibt, so gibt es auch eine erfolgreiche kanonische Rechnung $\mathcal{R}'$ von $F$ bei Eingabe von $G$, so dass $\mathcal{R}$ und $\mathcal{R}'$ die gleiche Länge haben und das gleiche Ergebnis liefern.

# Vollständigkeit von Strategien

**Definition:** Eine Strategie heißt vollständig, wenn es für jedes Logik-Programm $F$ und jede Zielklausel $G$, für die es eine erfolgreiche Berechnung von $F$ bei Eingabe von $G$ gibt, auch eine erfolgreiche Berechnung von $F$ bei Eingabe von $G$ mittels der Strategie gibt.

**Satz:** Die Breitensuche ist eine vollständige Strategie.

**Satz:** Die Tiefensuche ist keine vollständige Strategie.