# Resolution in der PL



Voraussetzung für seine volle Anwendung des Resolutionsschemas: Ausgangsformeln müssen Kausalform haben, also eine Disjunktion von Literalen (Atomformeln oder negierte Atomformeln) sein. Und: Formeln müssen ggf. in KNF überführt werden.

**Def.** (Kausal): Formel der Form

$$\forall x_1 \forall x_2 \ldots \forall x_s (P_1 \vee P_2 \vee \ldots \vee P_m)$$

wobei $P_i$ ein Literal und $x_1 \ldots x_s$ die einzigen Variablen, die in $(P_1 \vee \ldots \vee P_m)$ vorkommen.

**Bemerkung 1:** In der Logikprogrammierung Darstellung von Kauseln in der Form

$$\forall x_1 \ldots \forall x_s (A_1 \vee \ldots \vee A_k \vee \neg B_1 \vee \neg B_2 \ldots \vee \neg B_n)$$

$$\equiv \forall x_1 \ldots \forall x_s (A_1 \vee \ldots \vee A_k \leftarrow B_1 \vee \ldots \vee B_n)$$

Nachdem alle Variablen allquantifiziert sind, kann man konventionsgemäß schreiben

$$A_1, A_2, \ldots, A_n \leftarrow B_1, B_2, \ldots, B_m$$

Bemerkung 2: Die Möglichkeit zum Weglassen von Allquantoren in einer vollständig allquantifizierten Formel ist unmittelbar plausibel. Schwieriger, aber für die Herstellung von Klauselform unerlässlich ist die Elimination Existenzquantoren. Trick:
In $\forall x \exists y\, G(x,y)$ soll für jedes $x$ ein $y$ existieren, so dass $G$ erfüllt wird.
Nun Fixierung genau eines Wertes von $y$ für jeden möglichen Wert von $x$, so dass $G$ erfüllt. Damit Funktion $y = f(x)$ definierbar, die textuell überall dort substituiert wird, wo vorher $y$ in $G$ stand.
$f$ heißt Skolemfunktion, der Vorgang der Elimination des $\exists$-Quantors heißt Skolemisierung.

Beispiel a) $\exists x$ Bundespräsident$(x)$, wird umgeformt zu Bundespräsident$(s_1)$, wo $s_1$ als Platzhalter fungiert ('Skolemkonstante')

b) 'Jeder hat einen Vater':
$\forall x \exists y$ Vater$(y,x)$

wird umgeformt zu: $\forall x$ Vater$(S_2(x), x)$
wo die Skolemfunktion $S_2$ jedem Individuum x seinen Vater zugeordnet.

Damit Algorithmus angebbar, der jede PL-Formel in KNF überführt und darauf Resolution anwendbar macht).

Ausgangsformel für unser folgendes Beispiel:

$$\forall x \, (\forall y \, P(x,y) \rightarrow \neg \forall y \, (Q(x,y) \rightarrow R(x,y)))$$

| Schritt | Beispiel |
|---|---|
| 0. Ggf. Beseitigung der Äquivalenz durch $p \leftrightarrow q \equiv (p \rightarrow q) \wedge (q \rightarrow p)$ | |
| 1. Elimination der Implikation ('→') | $\forall x (\neg \forall y\, P(x,y) \vee \neg \forall y (\neg Q(x,y) \vee R(x,y)))$ |
| 2. 'Heranziehen' von Negationen durch $\neg\neg p = p$, de Morgan sowie $\neg \forall x\, P(x) \equiv \exists x \neg P(x)$ | $\forall x (\exists y \neg P(x,y) \vee \exists y (Q(x,y) \wedge \neg R(x,y)))$ |
| 3. Umbenennung von Variablen derart, daß jeder Quantor nur eine Variable bindet (hat keinen Einfluß auf Interpretation der Formel | $\forall x (\exists y \neg P(x,y) \vee \exists z (Q(x,z) \wedge \neg R(x,z)))$ |
| 4. Herstellen der sog. 'Pränex'-Normalform: Alle Quantoren werden an Anfang der Formel gestellt, ohne ihre Reihenfolge zu ändern. | $\forall x \exists y \exists z (\neg P(x,y) \vee (Q(x,z) \wedge \neg R(x,z)))$ |

5. Skolemisierung: Elimination von $\exists y$ und $\exists z$

$$\forall x \left( \neg P(x, s_1(x)) \vee \left( Q(x, s_2(x)) \wedge \neg R(x, s_2(x)) \right) \right)$$

6. Weglassen der Präfix-Allquant. (Konvention), weil alle Variablen allquantifiziert sind.

$$\neg P(x, s_1(x)) \vee \ldots\ldots$$

7. Konversion der verbleibenden 'Matrix' in eine Konjunktion von Klauseln mit Hilfe von Distributivgesetzen ....

$$\left( \neg P(x, s_1(x)) \vee Q(x, s_2(x)) \right) \wedge \left( \neg P(x, s_1(x)) \vee \neg R(x, s_2(x)) \right)$$

8. Konjunktion von Klauseln wird in Klauselmenge zusammengefaßt

$$\{ \neg P(x, s_1(x)) \vee Q(x, s_2(x)),\ \neg P(x, s_1(x)) \vee \neg R(x, s_2(x)) \}$$

9. Umbenennung von Variablen so daß kein zwei Klauseln sich auf dieselbe Variable beziehen

$$\{ \neg P(x, s_1(x)) \vee Q(x, s_2(x)),\ \neg P(y, s_1(y)) \vee \neg R(y, s_2(y)) \}$$

# Beispiel für einen Resolutionsschritt



Sei z.B.: $P(x,y) \vee P(f(a),z)$

$\neg P(f(x),g(y)) \vee Q(x,y)$

$Q(a,y)$

Vorgehensweise:

a) Auswahl zweier Prädikate, die in Klauseln einmal negiert und einmal nicht negiert vorkommen

b) (Unifikation): Belegung der Variablen links und rechts derart, daß die Ausdrücke identisch werden

Hier:

| links | rechts |
|---|---|
| $x \mapsto f(a)$ | $x \mapsto a$ |
| $z \mapsto g(y)$ | |
| $y \mapsto g(y)$ | |

womit beide Klauseln resolvierbar sind

# Prolog



Literatur: W. Clocksin, C. Mellish
Programming in Prolog, Springer

L. Sterling, E. Shapiro. The Art of Prolog, MIT-Press

Prolog: 'Horn-Klauseln + Beweis-Technik'
= elegante Programmierung.

Grundlage für die Implementierung:
Warren Abstract Machine WAM → siehe Homepage

Beispiel für Rekursion in Prolog: List-append

- Syntax: Prädikate, Konstanten, Funktionen, Variablen groß geschrieben
- Format von Listen in Prolog: [a, b, c]
- Aufspaltung von Kopf- und Restliste:
  [Head | Tail]
  ↑ Kopf ↑ Restliste
- Programm: ↓L1 + ↓L2 = ↓L3
  (1) append([], List, List)
  (2) append([Head|Tail], List2, [Head, Result]) :-
  append(Tail, List2, Result)

opa (A, B) :- vater (A, C), vater (C, B).



eine gültige Prolog-Klausel.