**Beispiel.**
PL Klauseln $C = \{p(x), p(f(y))\}$, $D = \{\neg p(x), q(x)\}$.

Grundsubstitutionen
$\rho = \{x \leftarrow f(a),\ y \leftarrow a\}$, $\tau = \{x \leftarrow f(a)\}$.

$C\rho = \{p(f(a))\}$, $D\tau = \{\neg p(f(a)), q(f(a))\}$.

haben AL Resolvente $E = \{q(f(a))\}$.

Offensichtlich gilt $E = (C - L)\rho \cup (D - M)\tau$ mit

- $L = C$,
- $M = \{\neg p(x)\}$,
- $|L\rho| = |M\tau| = 1$ und
- $L\rho = \overline{M\tau}$.

Konstruiere PL Resolutionsschritt (mit Faktorisierung) aus $C$ und $D$, dessen Resolvente $E$ als Grundinstanz hat.

1. Variablenumbenennung, damit $C$ und $D$ variablen-disjunkt:
   $C\xi_C$, $D\eta_D$ mit $\xi_C = \{\}$ und $\eta_D = \{x \leftarrow u\}$.

2. Aus $\rho$ und $\tau$ Unifikator $\theta$ von $\{L\xi_C, \overline{M\eta_D}\}$:

   $$\theta = \{x \leftarrow f(a),\, y \leftarrow a,\, u \leftarrow f(a)\}$$

   Damit: $E = (C - L)\xi_C\theta \cup (D - M)\eta_D\theta$.

3. Faktorisierung von $C\xi_C$, $D\eta_D$ mit
   $\nu_C = \{x \leftarrow f(y)\}$ und $\mu_D = \{\}$.

4. Mgu $\sigma_N$ der resolvierten Literale in umbenannten Faktoren:

   $$N = \{L\xi_C\nu_C, \overline{M\eta_D\mu_D}\} = \{p(f(y)), \neg p(u)\} \ .$$

   Ergibt $\sigma_N = \{u \leftarrow f(y)\}$.

5. Da $\sigma_N$ mgu ist, kann es zu $\theta$ erweitert werden mit $\lambda = \{y \leftarrow a\}$. Also $\theta = (\nu_C \cup \mu_D)\sigma_N\lambda$.

| | | |
|---|---|---|
| | $C = \{p(x), p(f(y))\}$ | $D = \{\neg p(x), q(x)\}$ |
| Umben. | $\xi_C = \{\}$ | $\eta_D = \{x \leftarrow u\}$ |
| | $\{p(x), p(f(y))\}$ | $\{\neg p(u), q(u)\}$ |
| Fakt. | $\nu_C = \{x \leftarrow f(y)\}$ | $\mu_D = \{\}$ |
| | $\{p(f(y))\}$ | $\{\neg p(u), q(u)\}$ |
| PL Res. | $\sigma_N = \{u \leftarrow f(y)\}$ | |
| | $E' = \{q(f(y))\}$ | |
| Inst. | $\lambda = \{y \leftarrow a\}$ | |
| | $E' = \{q(f(a))\}$ | |

Dieses Beispiel im allgemeinen Fall nachvollziehen.

**Beweis: Lifting Lemma.**

$E \in \mathcal{R}(T(S))$ und nicht in $T(S)$, sonst trivial.

$E$ AL Resolvente von Grundklauseln $C\rho$ und $D\tau$ aus $T(S)$, wobei $C, D \in S$.

Also: $E = (C - L)\rho \cup (D - M)\tau$ für
nichtleere $L \subseteq C$, $M \subseteq D$.

Außerdem $\overline{L\rho} = M\tau$ und $|L\rho| = |M\tau| = 1$.

$\rho = \{v_1 \leftarrow t_1, \ldots, v_k \leftarrow t_k\}$ und
$\tau = \{u_1 \leftarrow w_1, \ldots, u_m \leftarrow w_m\}$, wobei
Var$(C) = \{v_1, \ldots, v_k\}$ und Var$(D) = \{u_1, \ldots, u_m\}$.

$\xi_C$, $\eta_D$ Variablenumbennenungen von $C$ und $D$ mit frischen Variablen $\{x_1, \ldots, x_k\}$, $\{y_1, \ldots, y_m\}$.

$\theta = \{x_1 \leftarrow t_1, \ldots, x_k \leftarrow t_k, y_1 \leftarrow w_1, \ldots, y_m \leftarrow w_m\}$

Damit: $E = (C - L)\xi_C\theta \cup (D - M)\eta_D\theta$,
sowie $L\xi_C\theta = L\rho$ und $M\eta_D\theta = M\tau$.

Insbesondere: $L\xi_C \cup \overline{M\eta_D}$ von $\theta$ unifiziert.

Es gibt mgus $\nu_C$ von $L\xi_C$, $\mu_D$ von $\overline{M\eta_D}$, sowie mgu $\sigma_N$ von $N = L\xi_C\nu_C \cup \overline{M\eta_D\mu_D}$.

Es gibt Substitution $\lambda : \text{Var} \to T$, so daß
$\theta = \lambda \circ \sigma_N \circ (\nu_C \cup \mu_D)$

Insbesondere: $|L\xi_C\nu_C| = |M\eta_D\mu_D| = 1$ und
$L\xi_C\nu_C\sigma_N\lambda = L\rho$ sowie
$M\eta_D\mu_D\sigma_N\lambda = M\tau$.

Also: $\overline{L\xi_C\nu_C\sigma_N} = M\eta_D\mu_D\sigma_N$.

- $C' = C\xi_C$ und $D' = D\eta_D$ durch gebundene Umbenennung aus $C$ und $D$ entstanden;
- $\mathsf{Var}(C') \cap \mathsf{Var}(D') = \emptyset$;
- $C'\nu_C$ und $D'\mu_D$ Faktoren von $C'$ und $D'$;
- $\sigma_N$ mgu der Literale in Einerklauseln $L' \in L\xi_C\nu_C$ und $M' \in M\eta_D\mu_D$;
- $E' = ((C'\nu_C - L') \cup (D'\mu_D - M'))\sigma_N$ ist Resolvente mit Faktorisierung von $C'$ und $D'$.

Damit $E' \in \mathcal{R}(S)$ und

$$
\begin{aligned}
E &= (C - L)\rho \cup (D - M)\tau \\
&= (C - L)\xi_C\theta \cup (D - M)\eta_D\theta \\
&= ((C' - L\xi_C)(\nu_C \cup \mu_D) \cup \\
&\qquad (D' - M\eta_D)(\nu_C \cup \mu_D))\sigma_N\lambda \\
&= ((C'\nu_C - L') \cup (D'\mu_D - M'))\sigma_N\lambda \\
&= E'\lambda \in T(\mathcal{R}(S))
\end{aligned}
$$

Daraus erhält man durch einfache Induktion über $n$ die benötigte Eigenschaft:

Für beliebige PL $\Sigma$-Klauselmengen $S$, $T \subseteq \mathsf{Term}^0_\Sigma$ und $n \in \mathbb{N}$ gilt:

$$\mathcal{R}^n(T(S)) \subseteq T(\mathcal{R}^n(S))$$

■

# Grundvollständigkeit

Eine klassische Beweistechnik:

## Semantische Bäume

[Robinson (1968); Kowalski & Hayes (1969)]

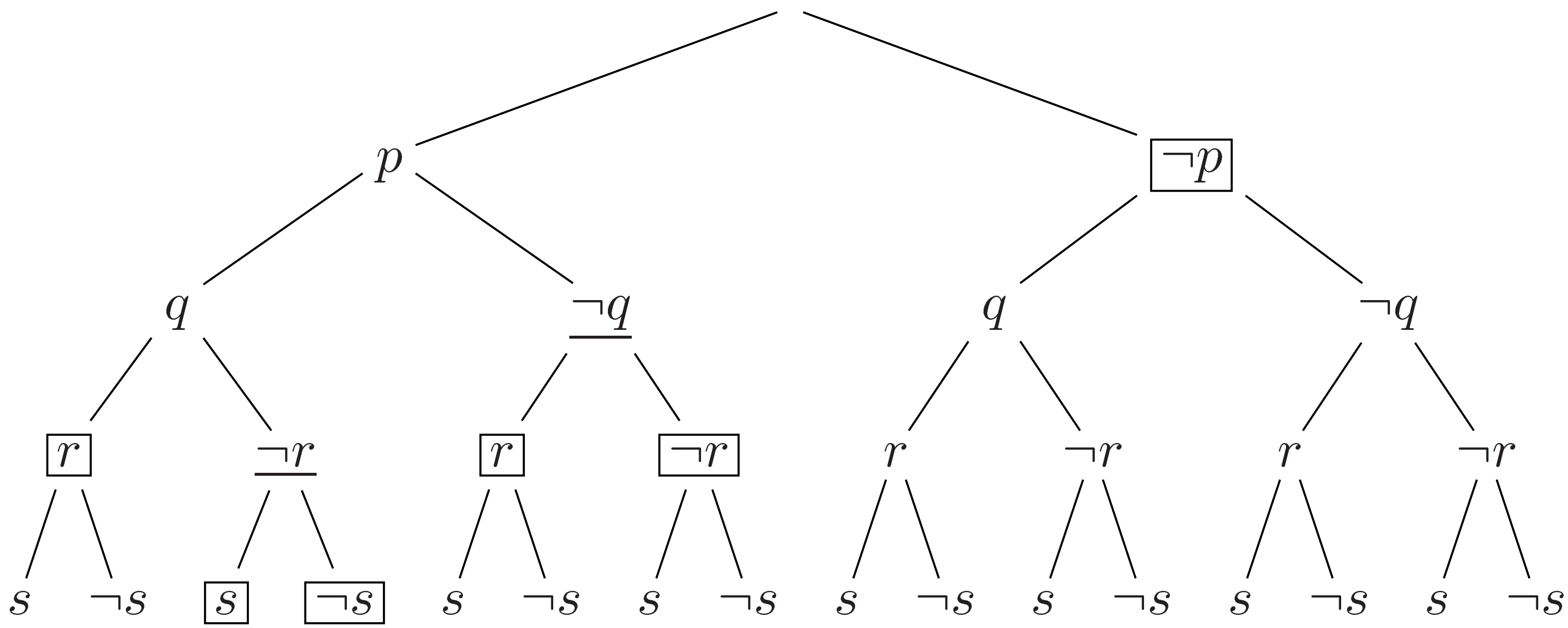
**Definition.** Sei $\langle p_0, p_1, \ldots \rangle$ beliebige Aufzählung einer AL Signatur $\Sigma$. Ein **semantischer Baum** $ST_\Sigma$ über $\Sigma$ ist markierter, binärer Baum mit:

1. Wurzel ist unmarkiert.
2. Nachfolgerknoten der Wurzel mit $p_0$ und $\neg p_0$ markiert.
3. Ist Knoten mit $p_i$ oder $\neg p_i$ markiert, sind Nachfolgerknoten mit $p_{i+1}$ und $\neg p_{i+1}$ markiert.

■

Semantische Bäume repräsentieren alle möglichen $\Sigma$-Interpretationen.

Semantischer Baum für $\Sigma_S = \{p, q, r, s\}$



**Definition.** Sei $ST_\Sigma$ semantischer Baum für $\Sigma$.

**Widerlegungsmenge** eines Knotens $\nu$ von $ST_\Sigma$:
alle Markierungen oberhalb und einschließlich von $\nu$.

AL Klausel $C$ **schlägt fehl** bei $\nu$ falls für alle Literale $L \in C$ Markierung $\overline{L}$ in Widerlegungsmenge von $\nu$.

$\nu$ ist **Fehlschlagsknoten** für AL Klauselmenge $S$ falls

1. ein $C \in S$ bei $\nu$ fehlschlägt und
2. *kein* $C \in S$ oberhalb von $\nu$ fehlschlägt.

**Inferenzknoten**: ein Knoten, dessen beide Nachfolger Fehlschlagsknoten sind.

$ST_\Sigma$ **geschlossen** für $S$ falls jeder Pfad Fehlschlagsknoten für $S$ enthält. ■

Auf voriger Folie für Beispiel
$$S = \{\{p\}, \{r, \neg p, q\}, \{s, \neg q\}, \{\neg q, \neg s\}, \{\neg q, \neg r\}, \{\neg r, q\}\}$$

Fehlschlagsknoten gerahmt.
Inferenzknoten unterstrichen.

Die Pfade ohne Fehlschlagsknoten für $S$ repräsentieren die Modelle von $S$:

**Lemma.** Ist endliche AL Klauselmenge $S$ über $\Sigma$ unerfüllbar, so ist jeder semantische Baum über $\Sigma$ geschlossen für $S$. ■

**Beweis:** (Lemma)
$\pi$ beliebiger Pfad ohne Fehlschlagsknoten für $S$ auf beliebigem semantischem Baum für $\Sigma$.

Jedes $p \in \Sigma$ genau einmal entweder positiv oder negativ unter den Markierungen $M$ von $\pi$.

$M$ induziert $\Sigma$-Interpretation durch

$$I(p) = \begin{cases} W & p \in M \\ F & \text{sonst} \end{cases}$$

$S$ unerfüllbar, also $I \not\models S$,
also muß $\text{val}_I(C) = F$ für ein $C \in S$ gelten.

Komplement jedes Literals von $C$ in $M$ (nach Def. von $I$). Also muß es Fehlschlagsknoten auf $\pi$ geben—Widerspruch. ■

**Satz.** Sei $S$ unerfüllbare AL Klauselmenge. Dann gilt $S \vdash_{RA} \Box$. ■

**Beweis:** Sei $ST$ geschlossener semantischer Baum für $S$ nach Lemma.

Jeder geschlossene semantische Baum hat Inferenzknoten. Sei $\nu$ ein Inferenzknoten maximaler Tiefe in $ST$.

Nachfolger von $\nu$ sind Fehlschlagsknoten mit $p$ und $\neg p$ markiert.

$C, D \in S$ die Klauseln, die bei Nachfolgern von $\nu$ fehlschlagen.

$C = \{\neg p\} \cup C'$ und $D = \{p\} \cup D'$. $C$ und $D$ AL zu $E = C' \cup D'$ resolviert.

$E$ schlägt bei $\nu$ fehl, da Komplemente aller Literale in $C \cup D$ außer $p$ und $\neg p$ schon in Widerlegungsmenge von $\nu$.

$S \cup \{E\}$ hat geschlossenen semantischen Baum mit weniger Fehlschlagsknoten als $ST$.

Geschlossener, semantischer Baum hat endlich viele Fehlschlagsknoten, also gibt es AL Resolutionsableitung $\langle C_1, \ldots, C_n \rangle$ aus $S$ und $\overline{S} = S \cup \bigcup_{i=1}^{n} \{C_i\}$ hat geschlossenen semantischen Baum mit einzigem Fehlschlagsknoten Wurzel.

Nur $\Box$ kann für diesen semantischen Baum fehlschlagen, also $\Box \in \overline{S}$. ■

# Anwendung auf Beispiel ergibt die Ableitung:

$$\{q, \neg r\} \quad \{\neg p, q, r\} \quad \{\neg q, s\} \quad \{\neg q, \neg s\}$$

$$\{\neg p, q\} \qquad\qquad \{\neg q\}$$

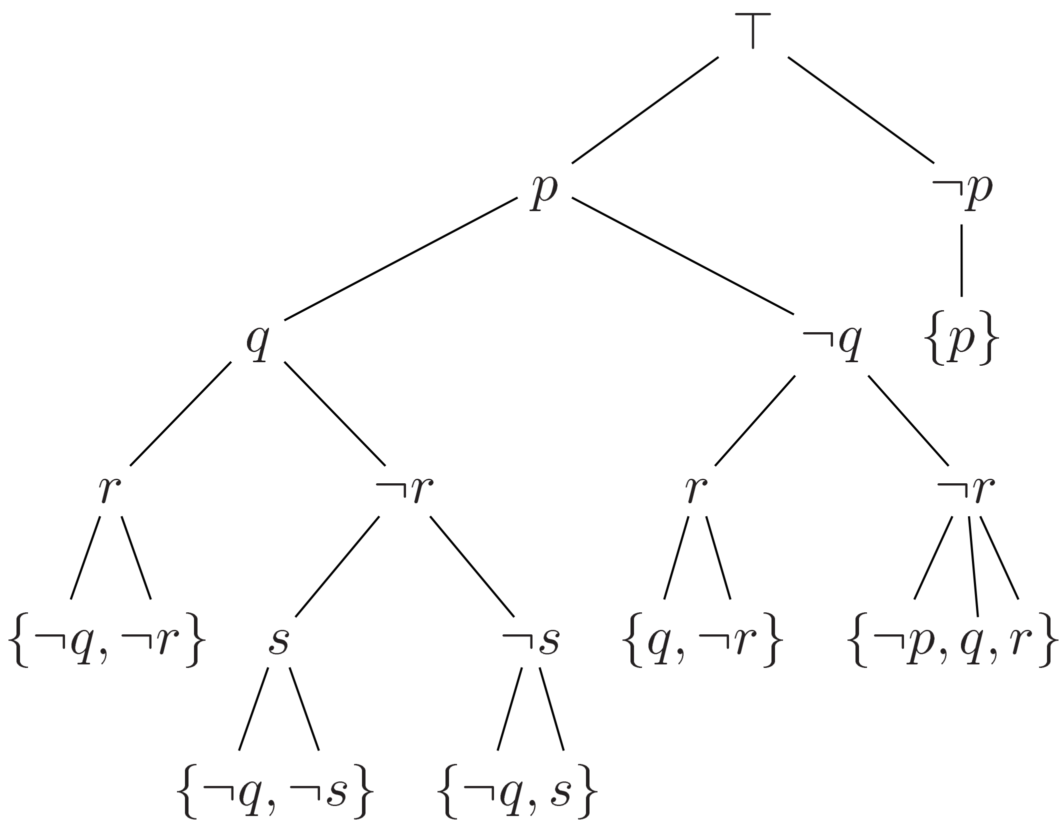
$$\{\neg p\} \quad \{p\}$$

$$\Box$$

Resolutionsableitung aus Beweis von Satz i.a. nicht Kürzestmögliche, da Länge abhängig von gewählter Ordnung auf $\Sigma$.

Kann sogar exponentiell länger sein als Resolutionsableitung mit anderer Ordnung: Goerdt (1993).

Semantische Bäume als reguläre Klauseltableaux mit atomarem Schnitt, wobei die Erweiterungsregel nur erlaubt, sofern alle neuen Zweige abgeschlossen werden können.



# Implementierung von Resolution

**Notation:** $\mathcal{R}^*(S) = \bigcup_{i=1}^{\infty} \mathcal{R}^i(S)$

Vollständige Implementierung erfordert, daß jedes Element von $\mathcal{R}^*(S)$ (modulo gebundener Umbenennung) nach endlich vielen Schritten generiert ist.

Einfachste Strategie:
*Stufenweise Saturierung* (level saturation)

```
S := [];
while ¬ eof  do
  S := append(S,[read_next_clause]);

while □ ∉ S do
  S := R(S);
```

Nachteile:

1. Zu grobkörnig; Verfeinerungen beziehen sich i.a. auf einzelne Resolutionsschritte.

2. # Literale in $S$ wächst von einer Stufe zur nächsten schnell an (kubisch).

**Besser:**
Jeweils die neuen Resolventen einer einzelnen Klausel mit den übrigen Klauseln berechnen:

*while* $\Box \notin$ S *do begin*
  C := first(S);
  T := $\{$E$|$E Resolvente von C,D$\in$S, E$\notin$S$\}$;
  S := append(rest(S),append(T,[C]))
*end*;

Selbstresolution möglich.

**Nachteil:**
derselbe Resolutionsschritt kann mehrfach vorkommen, wenngleich die Resolvente nur einmal behalten wird

Verbesserung später: Stützmengenstrategie

Implementierung mit Schlange ähnlich zu fairer Implementierung von `select_clause`

Prozedur ist einfacher als Klauseltableauxprozedur

Offenbar beweiskonfluent

# Verfeinerungen von Resolution

1. Löschen von Klauseln, die nicht zur Ableitung von $\Box$ beitragen.
   (a) Pure Literale, Klauseln
   (b) Tautologien
   (c) Subsumtion
2. Einschränkung der Möglichkeiten zur Bildung neuer Resolventen.
   (a) geordnete Resolution
   (b) Lock-Resolution
3. Einschränkung der Form der Resolutionsableitung.
   (a) Lineare Resolution
   (b) Input Resolution
   (c) Unit Resolution
4. Kombination von 2. und 3.: semantische Resolution.
   (a) Hyperresolution
   (b) Stützmengenstrategie (Set of support — SOS)
   (c) UR-Resolution
   (d) Geordnete Resolution mit Selektionsfunktion

# Löschstrategien

**Definition.** $S$ PL Klauselmenge. $C \in S$ heißt **pur** in $S$, falls es ein $L \in C$ gibt, so daß $L$ kein Link mit irgendeiner Klausel aus $S$ (einschließlich $C$) bildet.

$C \in S$ heißt **tautologisch** falls $C$ eine AL Tautologie ist.

$C$ **subsumiert** $D$ falls es eine Substitution $\sigma$ gibt, so daß $C\sigma \subseteq D$. ∎

Wir führen vier Löschoperatoren $D_x$, wobei $x \in \{\mathsf{pur}, \mathsf{taut}, \mathsf{sub}, \overline{\mathsf{sub}}\}$, auf Klauselmengen ein:

$D_{\mathsf{pur}}(S) = S - \{C \in S \mid C \text{ pur in } S\}$.

$D_{\mathsf{taut}}(S) = S - \{C \in S \mid C \text{ tautologisch}\}$.

$$D_{\mathsf{sub}}(S,T) = S - \{D \in S \mid \exists C \in T : C \text{ subsumiert } D\}.$$

$$D_{\overline{\mathsf{sub}}}(S) = S - \{D \in S \mid \exists C \in S : C \neq D \;\; C \text{ subsumiert } D\}.$$

$\neq$ modulo geb. Umbenennung, sonst $D_{\overline{\mathsf{sub}}}(S) = \emptyset$

Alle Löschoperatoren erhalten Modelle, da sie keine neuen Klauseln hinzufügen.

## Beispiele:

S:

$C_1 = \{p(x,z), q(z)\}$ $C_2 = \{q(a), p(b,a), r(u)\}$

$C_3 = \{p(x,y), \neg r(f(x))\}$ $C_4 = \{\neg p(y,y), \neg q(y)\}$

$C_5 = \{p(z,x), r(g(z))\}$ $C_6 = \{\neg r(f(x)), \neg q(z)\}$

$C_5$ pur in $S$ (zweites Literal)

$C_2$ von $C_1$ subsumiert

$C_1$ und $C_4$ haben PL Resolvente $RF(C_1, 1; C_4, 1) = \{q(y), \neg q(y)\}$, die tautologisch ist.

Weitere Resolvente
$RF(C_1, 2; C_4, 2) = \{p(x,y), \neg p(y,y)\}$
nicht tautologisch, da $x$ und $y$ verschieden instantiiert werden können.

$RF(C_3, 1; C_4, 1) = \{\neg r(f(y)), \neg q(y)\}$ wird von $C_6$ subsumiert und bei Vorwärtssubsumtion gelöscht.

$D_{\text{sub}}$ ist Subsumtion bzgl. Klauseln in $T$

$D_{\overline{\text{sub}}}$ ist totale Subsumtion. $C \neq D$ gebraucht, da jede Klausel sich selbst subsumiert.

# Implementierung der Löschregeln

(i) Als Vorverarbeitungsschritt.

(ii) Nach Berechnung neuer Resolventen:

```
while □ ∉ S do begin
  C := first(S);
  T := {E | E Resolvente von C,D ∈ S, E ∉ S};
  Ξ;
  S := append(rest(S),append(T,[C]))
end;
```

Dabei ist $\Xi$

T := $D_{\mathtt{pur}}$(T)

T := $D_{\mathtt{taut}}$(T)

T := $D_{\mathtt{sub}}$(T,S) Vorwärtssubsumtion

S := $D_{\mathtt{sub}}$(S,T) Rückwärtssubsumtion

oder Kombinationen davon.

# Implementierung totaler Subsumtion

$\texttt{S} := D_{\overline{\texttt{sub}}}(\texttt{S} \cup \texttt{T}) \cap \texttt{S};$
$\texttt{T} := D_{\overline{\texttt{sub}}}(\texttt{S} \cup \texttt{T}) \cap \texttt{T};$
*if* $\texttt{C} \in \texttt{S}$ *then*
  S := append(rest(S),append(T,[C]))
*else*
  S := append(S,T);

**Satz.** [Garey & Johnson (1979)]

Die Subsumtionsrelation auf PL Klauseln ist entscheidbar: sie ist NP-vollständig. ■

Spezielle Verwendung von Resolution als Entscheidungsverfahren für Subsumtion.

**Verfahren:**

Man betrachtet spezielleres Problem:

Literal $L$ ist ($\sigma$-)**Instanz** eines Literals $K$, falls es Substitution $\sigma$ gibt, so daß $L = K\sigma$.

**Proposition.** $L$ ist Instanz von $K$ gdw $\{L\theta, K\}$ unifizierbar, wobei $\theta = \{x_1 \leftarrow a_1, \ldots, x_n \leftarrow a_n\}$, Var$(L) = \{x_1, \ldots, x_n\}$ und $\{a_1, \ldots, a_n\}$ neue Konstanten. ■

Beweis: Übung. ■

$C$, $D = \{L_1, \ldots, L_m\}$ PL Klauseln.

$C$ subsumiert $D$ gdw es Substitution $\sigma$ gibt, so daß $C\sigma \subseteq D$

gdw

es Substitution $\sigma$ gibt, so daß für jedes $K \in C$ es ein $L \in D$ gibt, das $\sigma$-Instanz von $K$ ist

gdw (Proposition)

jedes $K \in C$ mit $L_i\theta_i$ simultan unifizierbar ist, wobei die in $\theta_i$ eingeführten Konstanten disjunkt

gdw (Übung)

$\Box \in \mathcal{R}^*(\{\{\neg L_1\theta_1\}, \ldots, \{\neg L_m\theta_m\}, C\})$, wobei keine Resolutionschritte innerhalb von $C$ oder von $\{\{\neg L_1\theta_1\}, \ldots, \{\neg L_m\theta_m\}\}$ erlaubt. ■

**Nachteil:**
Schlimmstenfalls überexponentieller Aufwand.

Besserer Algorithmus von Gottlob & Leitsch (1985).

**Beispiel.**
$C = \{p(x,y), q(z)\} \quad D = \{q(a), p(b,z), r(u)\}$

$\Box \in \mathcal{R}^*(\{\{\neg q(a)\}, \{\neg p(b,c)\}, \{\neg r(d)\}, C\})$ ?

$C$ und $\{\neg q(a)\}$ resolvieren zu $\{p(x,y)\}$, welches mit $\{\neg p(b,c)\}$ zu $\Box$ resolviert.

Also wird $D$ von $C$ subsumiert.

Für $C' = \{p(x,y), q(x)\}$, ergibt Resolution von $C'$ mit $\{\neg q(a)\}$ Resolvente $\{p(a,y)\}$, auf die kein weiterer Resolutionsschritt anwendbar.

Für $C'$ und $\{\neg p(b,c)\}$ ähnlich,
also subsumiert $C'$ nicht $D$.