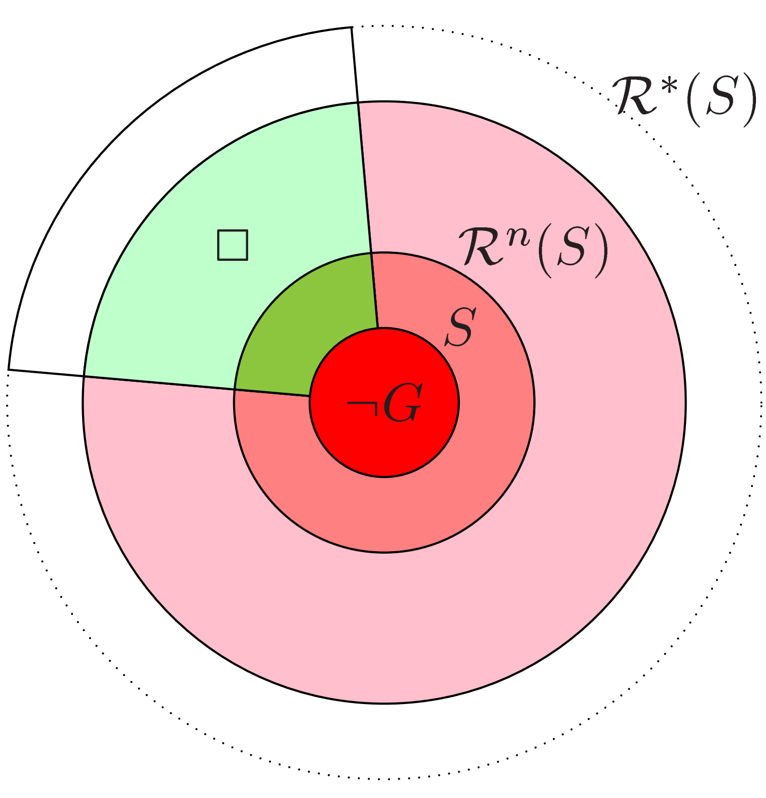
Einschränkung der Form der Resolutionsableitung

Lineare Resolution

Großer Nachteil von Resolution ist ihre Ungerichtetheit.

(Schwache) Konnektionstableaus sind dagegen zielgerichtet.
Und: Sie starten mit (der Negation) der Behauptung.

In ähnlicher Weise erzwingt lineare Resolution „Zielorientiertheit“:



Lineare Resolutionsableitungen haben „direkte Verbindung“ von erster zu letzter Klausel:

In DAG-Darstellung „linearer“ Graph.

Definition. Sei S eine Klauselmenge.

Eine **lineare Resolutionsableitung** von D_n aus S ist eine Resolutionsableitung der Form

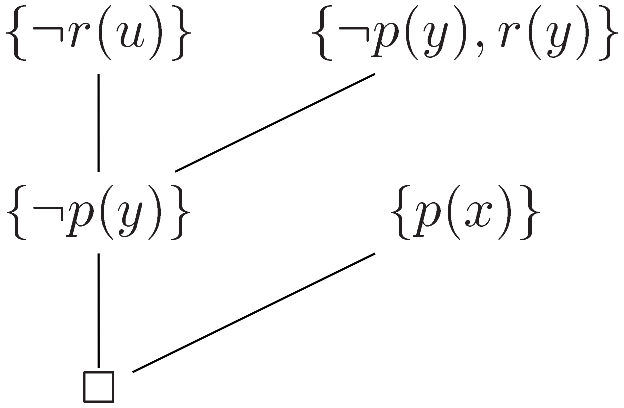
$\Gamma = \langle D_0, D_1, \ldots, D_n \rangle$,

wobei

1. $D_0 \in S$
2. für alle $i < n$ ist D_{i+1} Resolvente von D_i und $E_{i+1} \in S \cup \{D_0, \ldots, D_i\}$.

D_0 heißt **Startklausel** (von Γ),
die D_i heißen **Verbindungsklauseln**,
die E_i heißen **Seitenklauseln**. ■

Beispiel.
$S = \{\{p(x)\}, \{\neg p(y), r(y)\}, \{\neg r(u)\}, \{q(a)\}\}$



Lineare Resolution ist **nicht beweiskonfluent**.

Startklausel muß aus minimal unerfüllbarer (mu) Klauselmenge stammen.

Satz. Sei S eine unerfüllbare AL Klauselmenge, und $C \in S$ aus mu Teilmenge von S.

Dann gibt es eine lineare Resolutionsableitung von $\square$ aus S mit Startklausel C. ■

Beweisskizze:
Grundvollständigkeit:
Induktion über Überschußliterale

Lifting: Spezialfall von Lifting unbeschränkter Resolution ■

Lineare Resolution ist selbst für mu Klauselmengen nicht beweiskonfluent.

Wie Konnektionstableaus:
Backtracking/Breitensuche über Klauselauswahl notwendig

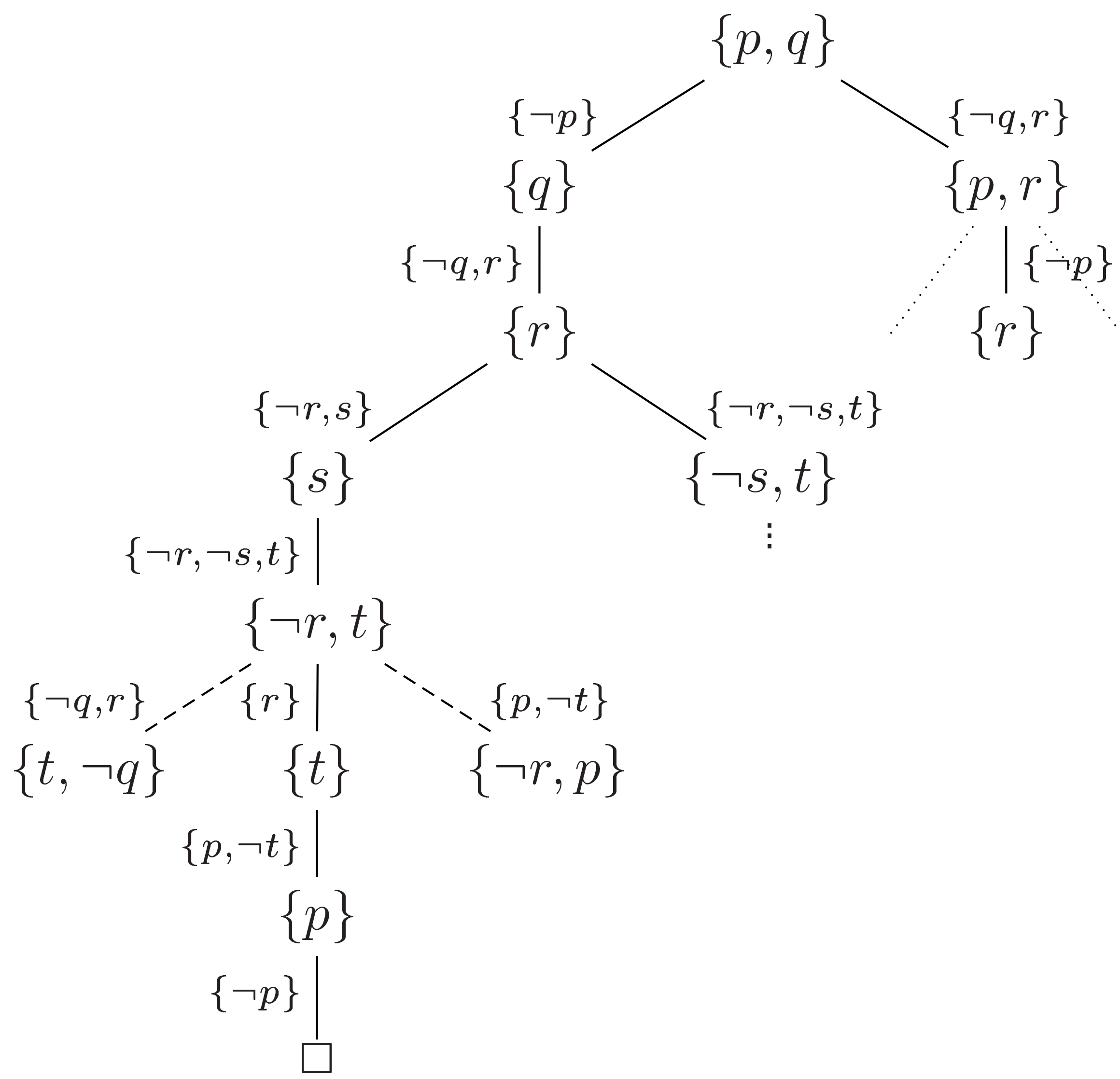
Implementierung via stufenweiser Sättigung unmöglich.

Also: Aufzählung der Ableitungen (statt der ableitbaren Klauseln).

Suchraum von LR selbst für kleine AL Beispiele enorm.

Beispiel:

$$S = \{\{p,q\}, \{\neg p\}, \{\neg q, r\}, \{\neg r, s\}, \{\neg r, \neg s, t\}, \{p, \neg t\}\}$$



Vollständiger Suchbaum > 500 Knoten, > 200 Zweige

Suchraum von linearer Resolution muß eingeschränkt werden.

Viele Varianten—eine wichtige:

SL-Resolution
linear resolution with selection function

In linearen Ableitungen zwei Arten von Resolutionsschritten:

- **Input-/Eingaberesolutionsschritt**:
  Seitenklausel aus S.

- **Ancestor/-Vorfahrresolutionsschritt**:
  Seitenklausel eine abgeleitete Klausel.

Beschränkung der verwendbaren Seitenklauseln

Definition. Sei C_i Verbindungsklausel in einer linearen AL Resolutionsableitung Γ.

Ein Literal $L \in C_i$ **ist Nachfahr** eines Literals $L \in C_j \in \Gamma$ $(j < i)$ falls $L \in C_k$ für alle $j \leq k \leq i$.

C_j ist A-**Vorfahr** von C_i falls:

1. die Seitenklausel neben C_j Eingabeklausel ist und
2. alle Literale außer dem resolvierten Literal K in C_j haben Nachfahren in C_i.

K heißt dann A-**Literal** in C_i (des Vorfahrs C_j). ■

$$
\begin{array}{lll}
C_j & \{L_1, \ldots, L_m, K\} & \{\overline{K}, \ldots\} \\
 & \quad\quad\quad\; | & \\
C_{j+1} & \{L_1, \ldots, L_m, \ldots\} & \\
 & \quad\quad\quad\; | & \\
\vdots & \quad\quad\quad\; \vdots & \\
 & \quad\quad\quad\; | & \\
C_i & \{L_1, \ldots, L_m, \ldots\} &
\end{array}
$$

Einerklauseln mit Eingabeklausel als Seitenklausel sind (triviale) A-Vorfahren aller folgenden Verbindungsklauseln.

Beispiel

$$S = \{\{p, q\}, \{\neg p\}, \{\neg q, r\}, \{\neg r, s\}, \\ \{\neg r, \neg s, t\}, \{p, \neg t\}\}$$

$$\begin{array}{rlcl}
C_1 : & \{p, q\} & & \{\neg p\} \\
C_2 : & \{q\} & & \{\neg q, r\} \\
C_3 : & \{r\} & & \{\neg r, s\} \\
C_4 : & \{s\} & & \{\neg r, \neg s, t\} \\
C_5 : & \{\neg r, t\} & & \\
C_6 : & \{t\} & & \{p, \neg t\} \\
C_7 : & \{p\} & & \{\neg p\} \\
C_8 : & \square & &
\end{array}$$

Definition. Eine lineare AL Resolutionsableitung Γ heißt t-**linear**, falls für alle Verbindungsklauseln $C_i \in \Gamma$ gilt:

1. Ancestor-Resolutionsschritte auf C_i nur mit A-Vorfahren von C_i
2. Enthält C_i ein zu einem seiner A-Literale komplementäres Literal L, dann ist C_{i+1} Ancestor-Resolvente mit resolviertem Literal L
3. A-Literale von C_i aus verschiedenen A-Vorfahren haben unterschiedliche Atome

■

ad 2.: erlaubte Ancestor-Resolutionsschritte bevorzugt.

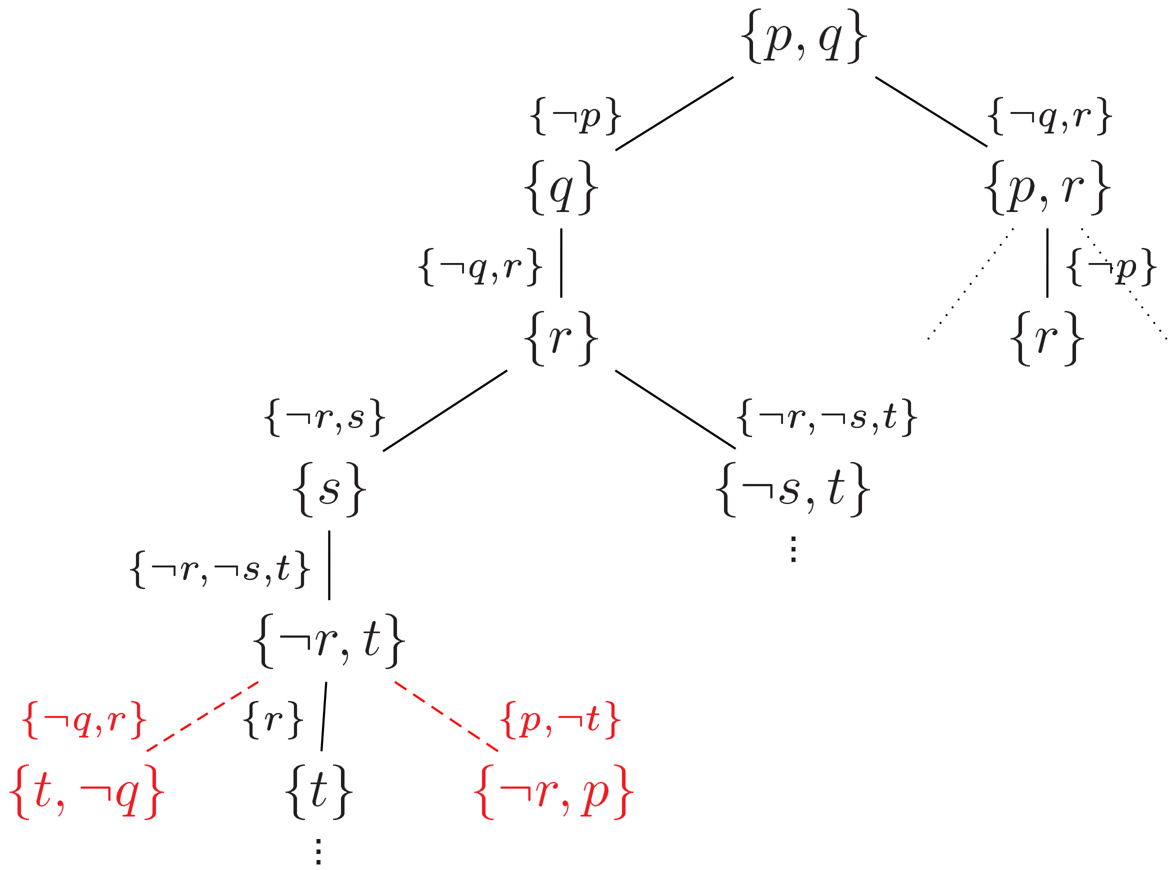
ad 3.: bestimmte wiederholte Resolutionsschritte über gleiches Literal ausgeschlossen. Vgl. Regularität.

Beschränkung der resolvierten Literale

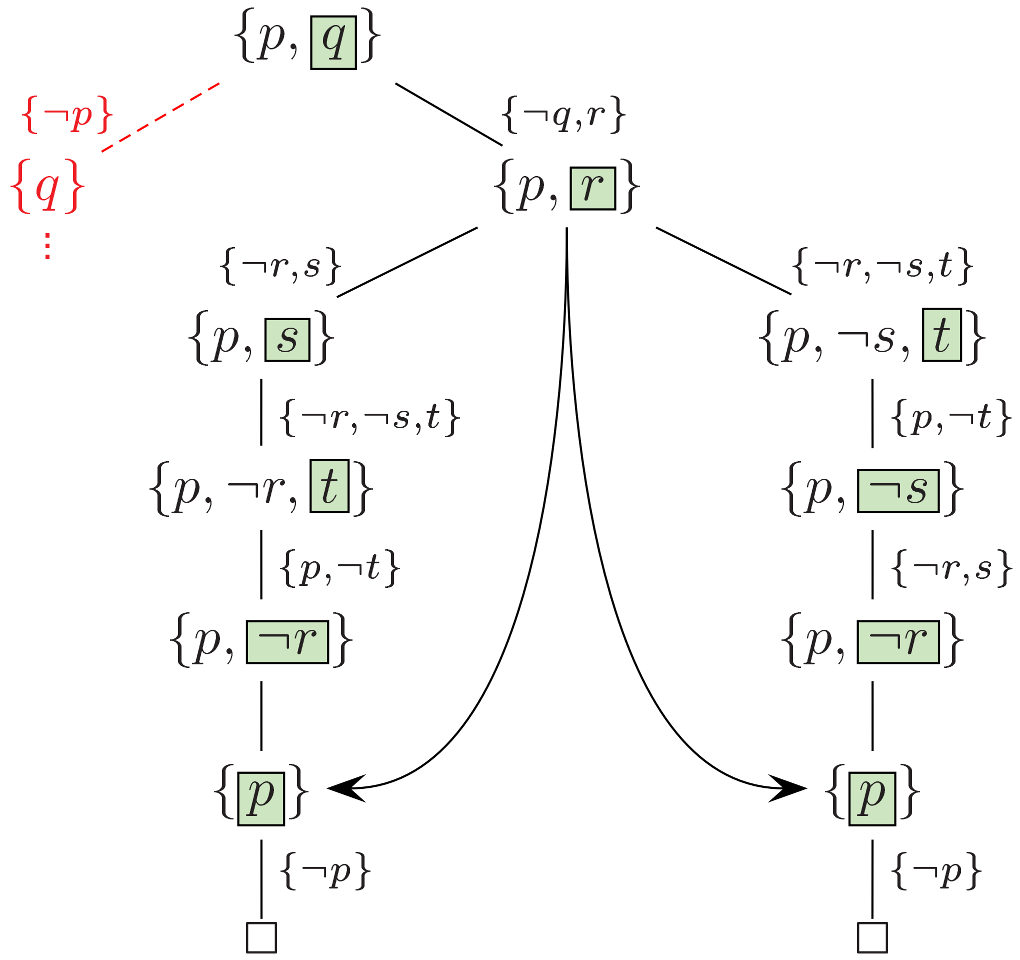
Definition. Eine **SL-Auswahlfunktion** f ordnet jeder Verbindungsklausel C eines Literal derjenigen Seitenklausel zu, deren Resolvente C ist.

Eine **SL-Resolutionsableitung** ist eine t-lineare Resolutionsableitung, wobei resolvierte Literale in Input-Resolutionsschritten durch eine SL-Auswahlfunktion bestimmt sind. ■

Suchraum bei t-linearer Ableitung kleiner:



Gesamter Suchraum von SL-Resolution bei $f(C) = \mathrm{last}(C)$



Ancestorresolutionsschritte zwingend.

Noch 12 Knoten
und nur zwei Zweige (mögliche Beweise).

Probleme von SL-Resolution.

- Definition von SL-Resolution kompliziert und schwierig zu verstehen
- Bei Erweiterung auf PL hat man verschiedene Optionen für Anwendung von Substitutionen. Zunächst ist nicht klar, welche natürlicher.
- Direkte Implementierung der Definition aufwendig.

Verbesserung durch Wechsel der Perspektive:

Tableau-/Kettennotation für SL-Resolutionsbeweise

Nurmehr Speicherung der für SL-Resolution relevanten Informationen:

A-Literale der Vorfahren der letzten Klausel und deren Nachfolger

Voraussetzung für Kettennotation:

Trennung von Faktorisierung und Resolution

Bei AL Resolution Faktorisierung
durch Mengenschreibweise implizit

Bei Kettenschreibweise Faktorisierung explizit

$\{\{\cdots\}\}$ Vielfachmenge

Definition. Sei C Vielfachmenge von Klauseln, und $L \in C$ habe Multiplizität ≥ 2.

Dann heißt $C - \{\{L\}\}$ **einfacher Faktor** von C. ■

Beispiel.

$C = \{\{p, p, \neg q, \neg q, r\}\}$

Einfache Faktoren: $\{\{p, \neg q, \neg q, r\}\}$
$\{\{p, p, \neg q, r\}\}$

Nicht aber: $\{\{p, \neg q, r\}\}$

SL$_V$-Resolution: wie SL-Resolution definiert, jedoch sind Klauseln Vielfachmengen und nach jedem Resolutionsschritt ist beliebig oft einfache Faktorisierung erlaubt.

Beispiel SL$_V$-Resolution.

$S = \{\{p, q, r\}, \{\neg p, \neg q\}, \{\neg p, q, r\}, \{p, \neg q\}, \{\neg r\}\}$

$f =$ first (Seitenklauseln vorne angefügt)

$$C_1: \ \{\{\boxed{p}, q, r\}\} \qquad \{\{\neg p, q, r\}\}$$

$$C_2: \ \{\{q, r, q, r\}\}$$

$$2 \times \mathsf{EF}$$

$$C_3: \ \{\{\boxed{q}, r\}\} \qquad \{\{\neg p, \neg q\}\}$$

$$C_4: \ \{\{\boxed{\neg p}, r\}\} \qquad \{\{p, \neg q\}\}$$

$$C_5: \ \{\{\neg q, r\}\}$$

$$C_6: \ \{\{r, r\}\}$$

$$EF$$

$$C_6: \ \{\{\boxed{r}\}\} \qquad \{\{\neg r\}\}$$

$$C_7: \ \square$$

C_4 nicht mit C_1, da q kein Nachfahre.
In C_4 ist r nicht selektiert.
C_4 nicht mit $\{p, q, r\}$,
da enstehende Klausel '„Regularität“ verletzt.

Beobachtung:

SL_V-Resolution, SL-Resolution ableitungsäquivalent

und

SL_V-Resolution ist simple Modifikation von schwacher Modellelimination

Nämlich:

Die nachfolgende Variante von ME mit
den zusätzlichen Festlegungen:

– Reduce (falls möglich) vor Extend.
– Regularität: Extend darf keine A Literale produzieren, die schon als A Literal vorhanden.

ME mit Selektionsfunktion und Faktorisierung

Die Regeln „Init" und „Contract" bleiben unverändert:

Init: Jede zu $C \in S$ gehörige Kette ist zulässig.

Contract: Ist $\langle L_1, \ldots, L_n \rangle$ eine zulässige Kette mit

– $\boxed{L_j}, \ldots, \boxed{L_n}$ vom Typ A,

dann auch

$\langle L_1, \ldots, L_{j-1} \rangle$

Die Regeln „Extend“ und „Reduce“ werden modifiziert:

Extend: Ist $\langle L_1, \ldots, L_n \rangle$ eine zulässige Kette,

- $L_r, \ldots, L_n$ vom Typ B,
- $\langle M_1, \ldots, M_m \rangle$ die zu $C \in S$ gehörige Kette,
- $f(\{L_r, \ldots, L_n\}) = L_k$ komplementär zu einem M_j $(1 \leq j \leq m)$,

dann auch

$$\langle L_1, \ldots, L_r, \ldots, L_{k-1}, L_{k+1}, \ldots, L_n, \boxed{L_k}, \\ M_1, \ldots, M_{j-1}, M_{j+1}, \ldots, M_m \rangle$$

Reduce: Ist $\langle L_1, \ldots, L_n \rangle$ eine zulässige Kette,

- $L_r, \ldots, L_n$ vom Typ B,

Ancestor Resolution:

- $\boxed{L_i}$ für ein $i < r$ vom Typ A,
- $L_i = \overline{L_r}$,

oder

Basic Factoring:

- L_i für ein $i < r$ vom Typ B,
- $L_i = L_r$

dann auch

$$\langle L_1, \ldots, L_{r-1}, L_{r+1}, \ldots, L_n \rangle.$$

Beispiel in Kettennotation.

$\boxed{p}\quad q\quad r$ Init

$q\quad r\quad \boxed{p}\quad q\quad r$ Ext

$q\quad r\quad \boxed{p}\quad q$ RedF

$q\quad r\quad \boxed{p}$ RedF

$\boxed{q}\quad r$ Contr

$\boxed{r}\quad \boxed{q}\quad \neg p$ Ext (mit $\{p, q, r\}$: $r\ \boxed{q}\ \boxed{\neg p}\ q\ r$)

$r\quad \boxed{q}\quad \boxed{\neg p}\quad \neg q$ Ext

$r\quad \boxed{q}\quad \boxed{\neg p}$ RedA

$\boxed{r}\quad 2\times$ Contr

$\boxed{r}$ Ext

$\langle\rangle$ Contr

Vergleich Modellelimination — SL$_V$-Resolution

Modellelimination mit SF und F	SL$_V$-Resolution
Folge von zulässigen Ketten	SL$_V$-Resolutionsableitung
zu $C \in S$ gehörende Kette	Eingabeklausel
Kette, die mit Typ B Literalen endet und Typ A Literale enthält	Verbindungsklausel C
Typ B Literal in Kette zu C	Literal in C
Typ A Literal in Kette zu C	A-Literal von C
Typ B Literal in rechtem Block	selektierbares Literal, Input-Resolution
adjazente Typ B Literale	Literale aus gleicher Eingabeklausel
Initiale Kette, $\langle\rangle$	Startklausel, $\Box$
Extension	Input-Resolution mit $f = \text{last}(C)$
Reduktion	Ancestor-Resolution
Kontraktion	—
Faktorisierung	einfache Faktorisierung

t-Linearität und SL-Bedingung in ME und Klauseltableaus

t-Linearität:

1. Nur A-Literale der letzten Kette/Literale auf dem aktuellen Zweig für Ancestor Resolution/Reduktion zulässig

2. Reduce/Reduktion und Faktorisierung „greedy“

3. Regularität (der Erweiterungsliterale)

SL-Bedingung:

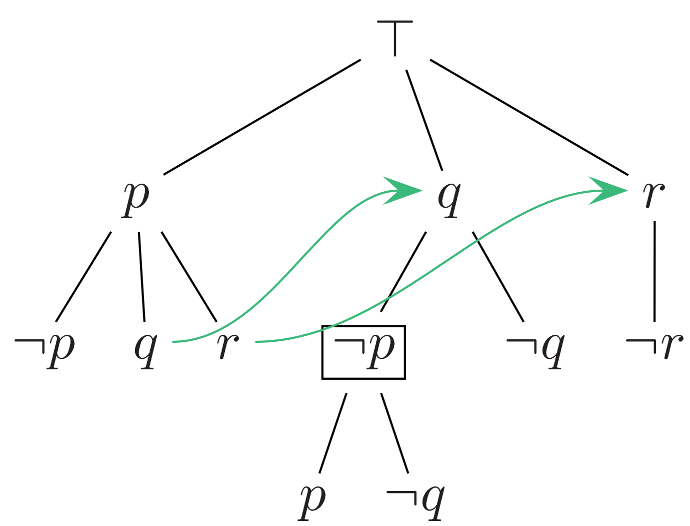
Extend nur mit rechtem Block von B-Literalen

Konnektionsbedingung

SL-Resolution
= reguläre ME mit Auswahlfkt. und Faktorisierung
= reguläre Konnektionstableaus mit Faktorisierung

(Reduktion und Faktorisierung sobald wie möglich)

Beispiel in Klauseltableaunotation.



Zur Erweiterung auf PL gibt es zwei Möglichkeiten:

1. wie bei Resolution:

- mgus nur auf Resolvente angewandt, also nur auf B-Literale bzw. Blätter
- Elternklauseln vor Verwendung bereinigt

2. wie bei Tableaus:

- mgu bei Reduktion und Extension auf die ganze bisherige Ableitung angewandt
- Bei Ancestor-Resolution keine Umbenennung der Elternklausel