Inferenzmethoden

Einheit 8

Alternative Deduktionsverfahren II: Resolution und Varianten

Verdichtung semantischer Bäume

Verdichtung semantischer Bäume

Gegeben: Menge von Klauseln $\{c_1, \ldots, c_n\}$, die eine zu widerlegende Formel in konjunktiver Normalform repräsentieren

Verdichtung semantischer Bäume

Gegeben: Menge von Klauseln $\{c_1, \ldots, c_n\}$, die eine zu widerlegende Formel in konjunktiver Normalform repräsentieren

Regeln: Resolutionsregel: Wähle zwei Klauseln $L_{11} \vee \ldots \vee L_{1m} \vee Pt_1$ und $L_{21} \vee \ldots \vee L_{2k} \vee \neg Pt_2$, so daß t_1 und t_2 unifizierbar sind

Ergänze $\sigma(L_{11} \vee \ldots \vee L_{1m} \vee L_{21} \vee \ldots \vee L_{2k})$ zur Klauselmenge, wobei σ der allgemeinste Unifikator von t_1 und t_2 ist.

Umbenennung der Variablen garantiert, daß Elternklauseln verschiedene Variablen haben

Verdichtung semantischer Bäume

Gegeben: Menge von Klauseln $\{c_1, \ldots, c_n\}$, die eine zu widerlegende Formel in konjunktiver Normalform repräsentieren

Regeln: Resolutionsregel: Wähle zwei Klauseln $L_{11} \vee \ldots \vee L_{1m} \vee Pt_1$ und $L_{21} \vee \ldots \vee L_{2k} \vee \neg Pt_2$, so daß t_1 und t_2 unifizierbar sind
Ergänze $\sigma(L_{11} \vee \ldots \vee L_{1m} \vee L_{21} \vee \ldots \vee L_{2k})$ zur Klauselmenge, wobei σ der allgemeinste Unifikator von t_1 und t_2 ist.
Umbenennung der Variablen garantiert, daß Elternklauseln verschiedene Variablen haben

Ziel: Die Klauselmenge enthält die leere Klausel

Verdichtung semantischer Bäume

Gegeben: Menge von Klauseln $\{c_1, \ldots, c_n\}$, die eine zu widerlegende Formel in konjunktiver Normalform repräsentieren

Regeln: Resolutionsregel: Wähle zwei Klauseln $L_{11} \vee \ldots \vee L_{1m} \vee Pt_1$ und $L_{21} \vee \ldots \vee L_{2k} \vee \neg Pt_2$, so daß t_1 und t_2 unifizierbar sind

Ergänze $\sigma(L_{11} \vee \ldots \vee L_{1m} \vee L_{21} \vee \ldots \vee L_{2k})$ zur Klauselmenge, wobei σ der allgemeinste Unifikator von t_1 und t_2 ist.

Umbenennung der Variablen garantiert, daß Elternklauseln verschiedene Variablen haben

Ziel: Die Klauselmenge enthält die leere Klausel

Rechtfertigung: Die Formel $\sigma(L_{11} \vee \ldots \vee L_{1m} \vee L_{21} \vee \ldots \vee L_{2k})$ ist genau dann erfüllbar, wenn $L_{11} \vee \ldots \vee L_{1m} \vee Pt_1$ und $L_{21} \vee \ldots \vee L_{2k} \vee \neg Pt_2$ erfüllbar sind. Da die leere Klausel unerfüllbar ist, folgt aus der Ableitbarkeit der leeren Klausel die Unerfüllbarkeit der ursprünglichen Klauselmenge, also die von F.

Resolutions-Ableitung für $P \wedge Q \vee \neg P \wedge Q \wedge R \vee \neg Q \wedge R \vee \neg R$

$\{P, Q\}$ $\{\neg P, Q, R\}$ $\{\neg Q, R\}$ $\{\neg R\}$

Resolutions-Ableitung für $P \wedge Q \vee \neg P \wedge Q \wedge R \vee \neg Q \wedge R \vee \neg R$

$\{\boxed{P}, Q\}$ $\quad$ $\{\boxed{\neg P}, Q, R\}$ $\quad$ $\{\neg Q, R\}$ $\quad$ $\{\neg R\}$

$\{Q, R\}$

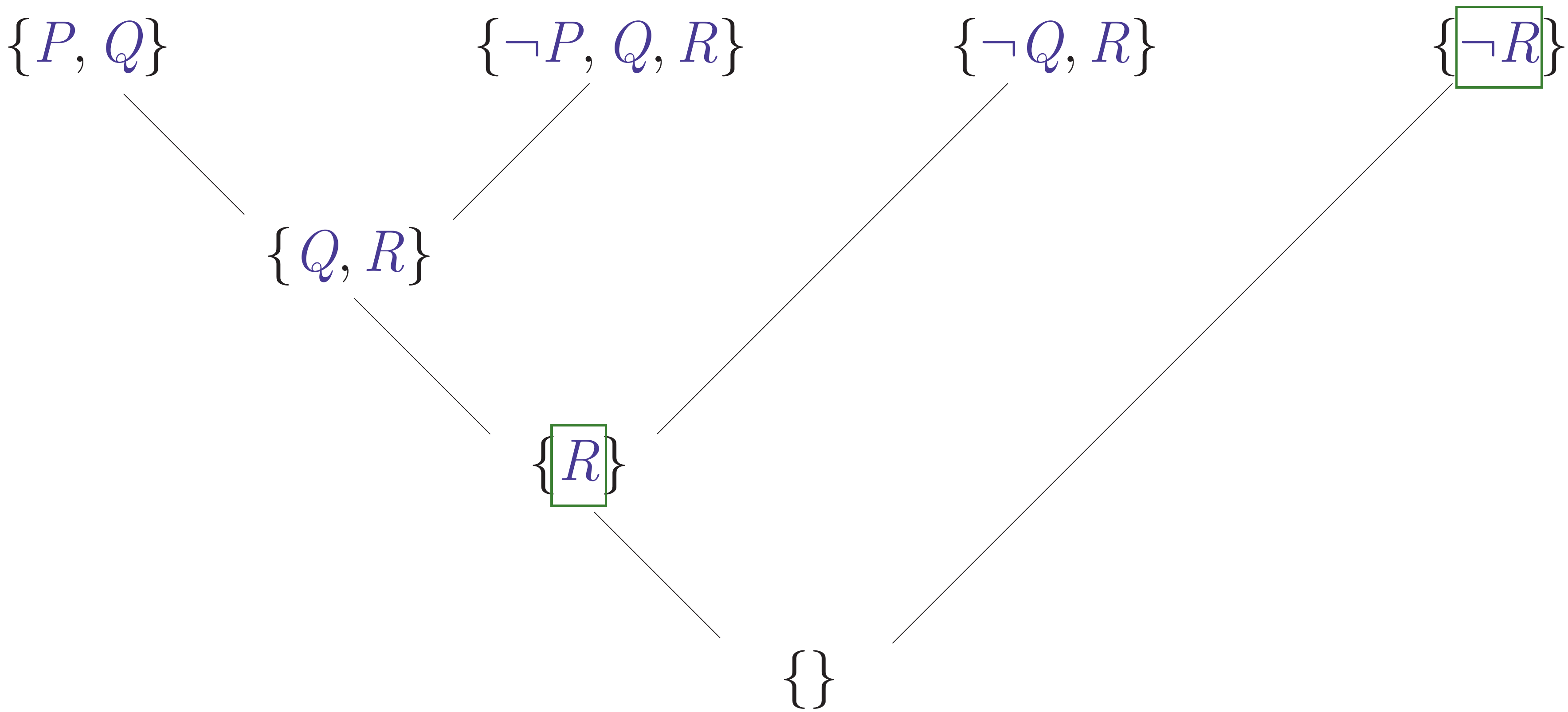
Resolutions-Ableitung für $P \wedge Q \vee \neg P \wedge Q \wedge R \vee \neg Q \wedge R \vee \neg R$

$\{P, Q\}$ $\quad$ $\{\neg P, Q, R\}$ $\quad$ $\{\boxed{\neg Q}, R\}$ $\quad$ $\{\neg R\}$

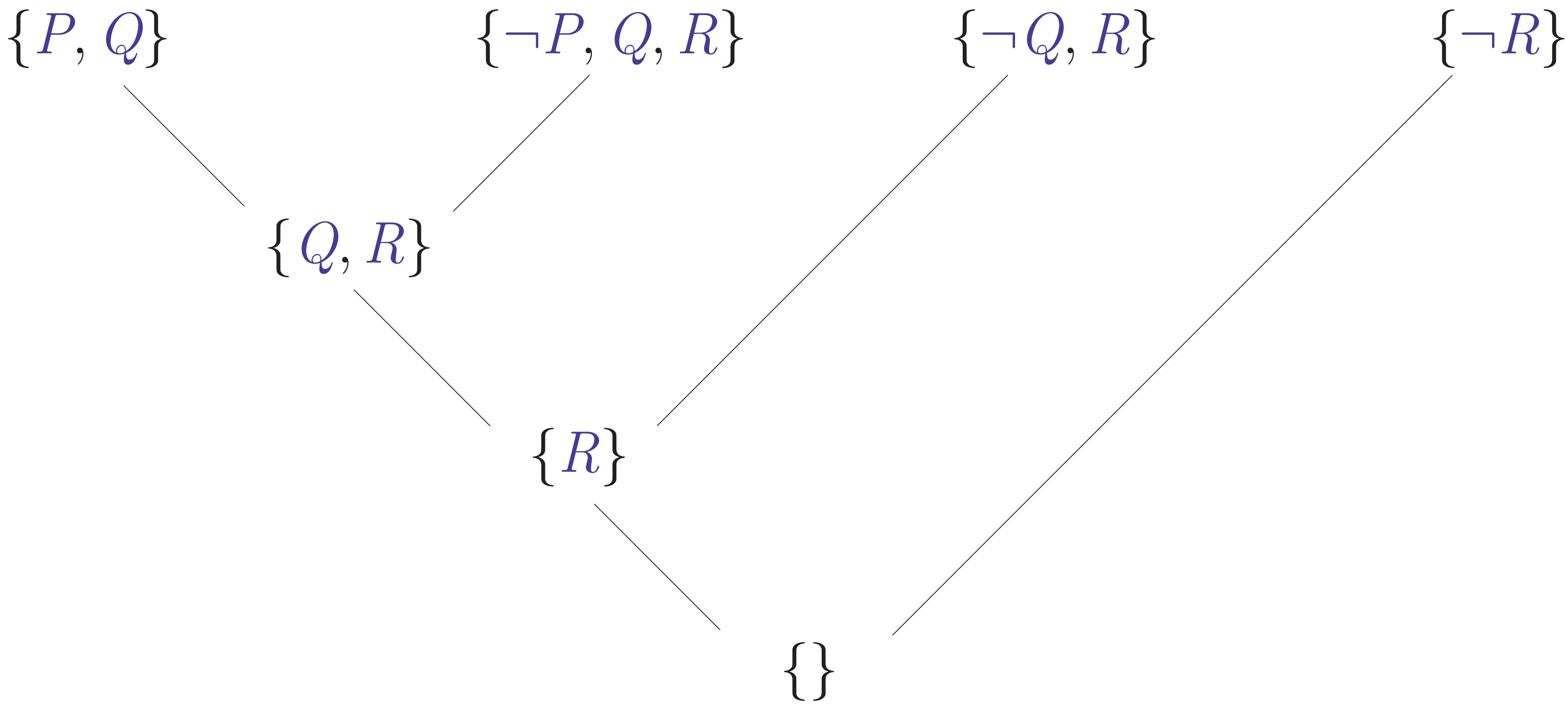
$\{\boxed{Q}, R\}$

$\{R\}$

Resolutions-Ableitung für $P \wedge Q \vee \neg P \wedge Q \wedge R \vee \neg Q \wedge R \vee \neg R$



Resolutions-Ableitung für $P \wedge Q \vee \neg P \wedge Q \wedge R \vee \neg Q \wedge R \vee \neg R$



Lineare, Input-, SLD-, P_1- und einfache Hyper-Resolution

Resolution in Matrix-Denkweise

Gegeben: Matrix $\{c_1, \ldots, c_n\}$ mit $c_i = \{L_{i1}, \ldots, L_{im_i}\}$

Menge von Konnektionen $\{k_1, \ldots, k_m\}$ mit $k_j = \{P_{i_j}{}^T, P_{i'_j}{}^F\}$

Resolution in Matrix-Denkweise

Gegeben: Matrix $\{c_1, \ldots, c_n\}$ mit $c_i = \{L_{i1}, \ldots, L_{im_i}\}$

Menge von Konnektionen $\{k_1, \ldots, k_m\}$ mit $k_j = \{P_{i_j}{}^T, P_{i'_j}{}^F\}$

Start: n Knoten markiert mit den Klauseln $c_i = \{L_{i1}, \ldots, L_{im_i}\}$

Resolution in Matrix-Denkweise

Gegeben: Matrix $\{c_1, \ldots, c_n\}$ mit $c_i = \{L_{i1}, \ldots, L_{im_i}\}$
Menge von Konnektionen $\{k_1, \ldots, k_m\}$ mit $k_j = \{P_{i_j}{}^T, P_{i'_j}{}^F\}$

Start: n Knoten markiert mit den Klauseln $c_i = \{L_{i1}, \ldots, L_{im_i}\}$

Regeln: Wähle Konnektion $k_j = \{P_{i_j}{}^T, P_{i'_j}{}^F\}$
und 2 Knoten markiert mit $K_1 \cup \{P_{i_j}{}^T\}$ und $K_2 \cup \{P_{i'_j}{}^F\}$.
(Prädikatenlogische Form: mit σ instantiierte Konnektion)

Generiere einen Nachfolger, markiert mit $K_1 \cup K_2$

Resolution in Matrix-Denkweise

Gegeben: Matrix $\{c_1, \ldots, c_n\}$ mit $c_i = \{L_{i1}, \ldots, L_{im_i}\}$
Menge von Konnektionen $\{k_1, \ldots, k_m\}$ mit $k_j = \{P_{i_j}{}^T, P_{i'_j}{}^F\}$

Start: n Knoten markiert mit den Klauseln $c_i = \{L_{i1}, \ldots, L_{im_i}\}$

Regeln: Wähle Konnektion $k_j = \{P_{i_j}{}^T, P_{i'_j}{}^F\}$
und 2 Knoten markiert mit $K_1 \cup \{P_{i_j}{}^T\}$ und $K_2 \cup \{P_{i'_j}{}^F\}$.
(Prädikatenlogische Form: mit σ instantiierte Konnektion)

Generiere einen Nachfolger, markiert mit $K_1 \cup K_2$

Ziel: Ein Knoten, der mit $\{\}$ markiert ist

Resolution in Matrix-Denkweise

Gegeben: Matrix $\{c_1, \ldots, c_n\}$ mit $c_i = \{L_{i1}, \ldots, L_{im_i}\}$

Menge von Konnektionen $\{k_1, \ldots, k_m\}$ mit $k_j = \{P_{i_j}{}^T, P_{i'_j}{}^F\}$

Start: n Knoten markiert mit den Klauseln $c_i = \{L_{i1}, \ldots, L_{im_i}\}$

Regeln: Wähle Konnektion $k_j = \{P_{i_j}{}^T, P_{i'_j}{}^F\}$

und 2 Knoten markiert mit $K_1 \cup \{P_{i_j}{}^T\}$ und $K_2 \cup \{P_{i'_j}{}^F\}$.

(Prädikatenlogische Form: mit σ instantiierte Konnektion)

Generiere einen Nachfolger, markiert mit $K_1 \cup K_2$

Ziel: Ein Knoten, der mit $\{\}$ markiert ist

Rechtfertigung: Die Klausel $K_1 \cup K_2$ ist genau dann komplementär zur Restmatrix, wenn dies für die Matrix $\{\, K_1 \cup \{P_{i_j}{}^T\},\ K_2 \cup \{P_{i'_j}{}^F\}\,\}$ gilt

Die leere Klausel macht jede Restmatrix komplementär (es gibt keine Pfade)

Resolutions-Ableitung für $P \wedge Q \vee \neg P \wedge Q \wedge R \vee \neg Q \wedge R \vee \neg R$

$\{P^T, Q^T\}$ $\{P^F, Q^T, R^T\}$ $\{Q^F, R^T\}$ $\{R^F\}$

Resolutions-Ableitung für $P \wedge Q \vee \neg P \wedge Q \wedge R \vee \neg Q \wedge R \vee \neg R$

$$\{\boxed{P^{T}}, Q^{T}\} \qquad \{\boxed{P^{F}}, Q^{T}, R^{T}\} \qquad \{Q^{F}, R^{T}\} \qquad \{R^{F}\}$$

$$\{Q^{T}, R^{T}\}$$

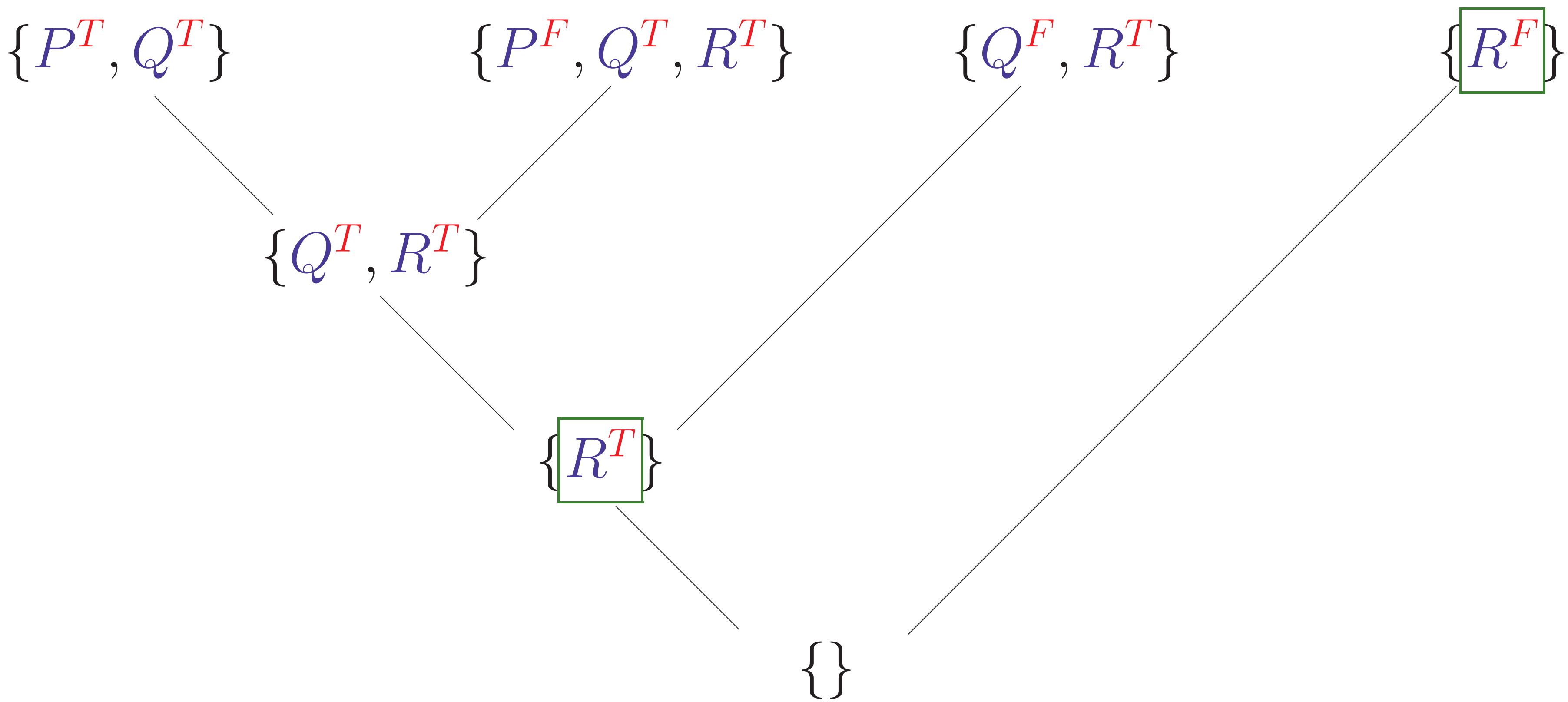
Resolution-Ableitung für $P \wedge Q \vee \neg P \wedge Q \wedge R \vee \neg Q \wedge R \vee \neg R$

$\{P^T, Q^T\}$ $\quad$ $\{P^F, Q^T, R^T\}$ $\quad$ $\{\boxed{Q^F}, R^T\}$ $\quad$ $\{R^F\}$

$\{\boxed{Q^T}, R^T\}$

$\{R^T\}$

Resolutions-Ableitung für $P \wedge Q \vee \neg P \wedge Q \wedge R \vee \neg Q \wedge R \vee \neg R$



Resolutionsstrategien und -varianten

- **Linear Resolution**: Resolvente ist Elternklausel des nächsten Schrittes

Resolutionsstrategien und -varianten

- **Linear Resolution**: Resolvente ist Elternklausel des nächsten Schrittes
- **Input Resolution**: Eine Elternklausel ist Eingabeklausel i.a. unvollständig
  Linear Input Resolution ist vollständig für Hornklauseln

Resolutionsstrategien und -varianten

- **Linear Resolution**: Resolvente ist Elternklausel des nächsten Schrittes
- **Input Resolution**: Eine Elternklausel ist Eingabeklausel   i.a. unvollständig
  Linear Input Resolution ist vollständig für Hornklauseln
- **Unit-Resolution**: Eine Elternklausel ist Einerklausel
  (Auch als **Unit-Resulting Resolution**)   vollständig für Hornklauseln

Resolutionsstrategien und -varianten

- **Linear Resolution**: Resolvente ist Elternklausel des nächsten Schrittes
- **Input Resolution**: Eine Elternklausel ist Eingabeklausel   i.a. unvollständig

  Linear Input Resolution ist vollständig für Hornklauseln
- **Unit-Resolution**: Eine Elternklausel ist Einerklausel

  (Auch als **Unit-Resulting Resolution**)   vollständig für Hornklauseln
- **P_1/N_1-Resolution**: Eine Elternklausel ist positiv (negativ)   vollständig

Resolutionsstrategien und -varianten

- **Linear Resolution**: Resolvente ist Elternklausel des nächsten Schrittes
- **Input Resolution**: Eine Elternklausel ist Eingabeklausel i.a. unvollständig

  Linear Input Resolution ist vollständig für Hornklauseln
- **Unit-Resolution**: Eine Elternklausel ist Einerklausel

  (Auch als **Unit-Resulting Resolution**) vollständig für Hornklauseln
- **P_1/N_1-Resolution**: Eine Elternklausel ist positiv (negativ) vollständig
- **Set-of-support**: Eine Elternklausel hat Vorfahren in einer Stützmenge

  vollständig genau dann, wenn Restklauselmenge erfüllbar

Resolutionsstrategien und -varianten

- **Linear Resolution**: Resolvente ist Elternklausel des nächsten Schrittes
- **Input Resolution**: Eine Elternklausel ist Eingabeklausel i.a. unvollständig
  Linear Input Resolution ist vollständig für Hornklauseln
- **Unit-Resolution**: Eine Elternklausel ist Einerklausel
  (Auch als **Unit-Resulting Resolution**) vollständig für Hornklauseln
- **P_1/N_1-Resolution**: Eine Elternklausel ist positiv (negativ) vollständig
- **Set-of-support**: Eine Elternklausel hat Vorfahren in einer Stützmenge
  vollständig genau dann, wenn Restklauselmenge erfüllbar
- **SLD Resolution** lineare Verkettung, sehr effizient
  Linear resolution with **S**election function restricted to **D**efinite clauses.

Resolutionsstrategien und -varianten

- **Linear Resolution**: Resolvente ist Elternklausel des nächsten Schrittes
- **Input Resolution**: Eine Elternklausel ist Eingabeklausel   i.a. unvollständig
  Linear Input Resolution ist vollständig für Hornklauseln
- **Unit-Resolution**: Eine Elternklausel ist Einerklausel
  (Auch als **Unit-Resulting Resolution**)   vollständig für Hornklauseln
- **P_1/N_1-Resolution**: Eine Elternklausel ist positiv (negativ)   vollständig
- **Set-of-support**: Eine Elternklausel hat Vorfahren in einer Stützmenge
  vollständig genau dann, wenn Restklauselmenge erfüllbar
- **SLD Resolution**   lineare Verkettung, sehr effizient
  Linear resolution with **S**election function restricted to **D**efinite clauses.
- **LUSH Resolution**:
  Linear resolution with **U**nrestricted **S**election function on **H**orn clauses

Resolutionsstrategien und -varianten

- **Linear Resolution**: Resolvente ist Elternklausel des nächsten Schrittes
- **Input Resolution**: Eine Elternklausel ist Eingabeklausel i.a. unvollständig
  Linear Input Resolution ist vollständig für Hornklauseln
- **Unit-Resolution**: Eine Elternklausel ist Einerklausel
  (Auch als **Unit-Resulting Resolution**) vollständig für Hornklauseln
- **P_1/N_1-Resolution**: Eine Elternklausel ist positiv (negativ) vollständig
- **Set-of-support**: Eine Elternklausel hat Vorfahren in einer Stützmenge
  vollständig genau dann, wenn Restklauselmenge erfüllbar
- **SLD Resolution** lineare Verkettung, sehr effizient
  Linear resolution with **S**election function restricted to **D**efinite clauses.
- **LUSH Resolution**:
  Linear resolution with **U**nrestricted **S**election function on **H**orn clauses
- **Hyperresolution**: Wähle Klausel mit negativen Literalen und zu jedem negativen Literal eine positive Klausel, die dieses enthält. Resolviere gleichzeitig mit allen Klauseln. vollständig

Resolutionsstrategien und -varianten

- **Linear Resolution**: Resolvente ist Elternklausel des nächsten Schrittes
- **Input Resolution**: Eine Elternklausel ist Eingabeklausel   i.a. unvollständig
  Linear Input Resolution ist vollständig für Hornklauseln
- **Unit-Resolution**: Eine Elternklausel ist Einerklausel
  (Auch als **Unit-Resulting Resolution**)   vollständig für Hornklauseln
- **P_1/N_1-Resolution**: Eine Elternklausel ist positiv (negativ)   vollständig
- **Set-of-support**: Eine Elternklausel hat Vorfahren in einer Stützmenge
  vollständig genau dann, wenn Restklauselmenge erfüllbar
- **SLD Resolution**   lineare Verkettung, sehr effizient
  Linear resolution with **S**election function restricted to **D**efinite clauses.
- **LUSH Resolution**:
  Linear resolution with **U**nrestricted **S**election function on **H**orn clauses
- **Hyperresolution**: Wähle Klausel mit negativen Literalen und zu jedem negativen Literal eine positive Klausel, die dieses enthält. Resolviere gleichzeitig mit allen Klauseln.   vollständig
- **Konnektionsgraphen Resolution**: Vermeide mehrfache Resolution über dieselbe Konnektion durch Markierung unbenutzter Konnektionen in Klauselgraphen   aufwendig

Lineare Input P_1-Resolution

$\{P^T, Q^T\}$ $\{P^F, Q^T, R^T\}$ $\{Q^F, R^T\}$ $\{R^F\}$

Lineare Input P_1-Resolution

$$\{\boxed{P^T}, Q^T\} \qquad \{\boxed{P^F}, Q^T, R^T\} \qquad \{Q^F, R^T\} \qquad \{R^F\}$$

$$\{Q^T, R^T\}$$

Lineare Input P_1-Resolution

$\{P^T, Q^T\}$ $\qquad$ $\{P^F, Q^T, R^T\}$ $\qquad$ $\{\boxed{Q^F}, R^T\}$ $\qquad$ $\{R^F\}$

$\{\boxed{Q^T}, R^T\}$

$\{R^T\}$

Lineare Input P_1-Resolution

$$\{P^T, Q^T\} \qquad \{P^F, Q^T, R^T\} \qquad \{Q^F, R^T\} \qquad \{\boxed{R^F}\}$$

$$\{Q^T, R^T\}$$

$$\{\boxed{R^T}\}$$

$$\{\}$$

Unit-Resolution

Eine Elternklausel ist immer eine Einerklausel

$\{P^T, Q^T\}$ $\{P^F, Q^T, R^T\}$ $\{Q^F, R^T\}$ $\{R^F\}$

Unit-Resolution

Eine Elternklausel ist immer eine Einerklausel

$\{P^T, Q^T\}$ $\qquad$ $\{P^F, Q^T, R^T\}$ $\qquad$ $\{Q^F, R^T\}$ $\qquad$ $\{R^F\}$

$$\{P^F, Q^T\}$$

Unit-Resolution

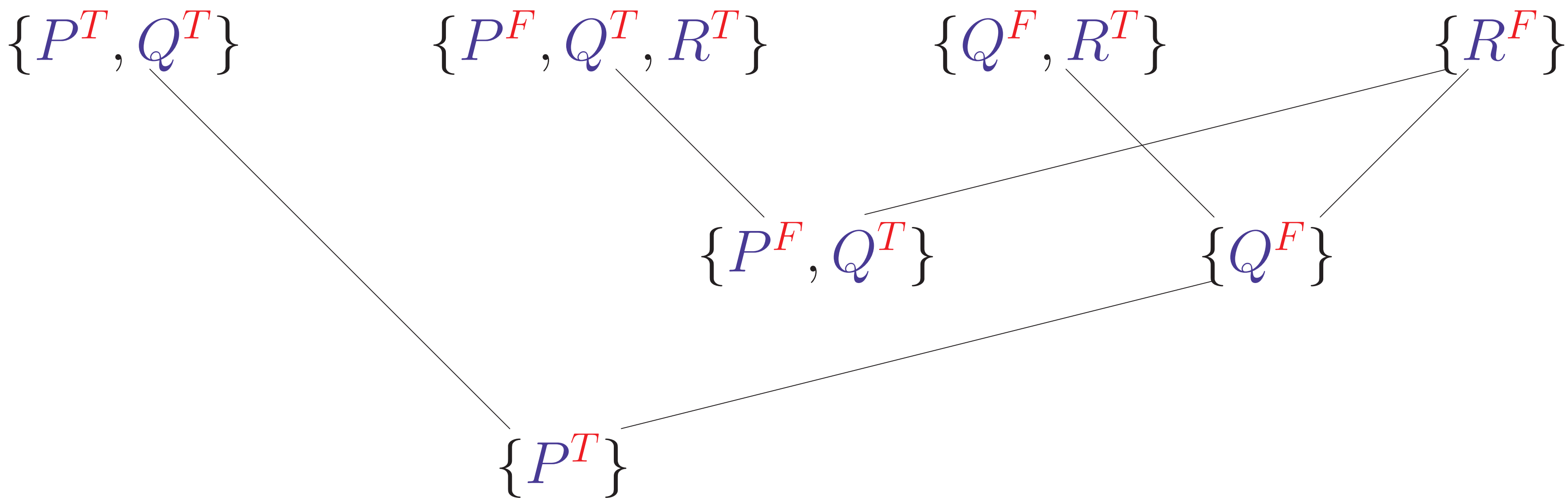
Eine Elternklausel ist immer eine Einerklausel

$$\{P^T, Q^T\} \qquad \{P^F, Q^T, R^T\} \qquad \{Q^F, R^T\} \qquad \{R^F\}$$

$$\{P^F, Q^T\} \qquad \{Q^F\}$$

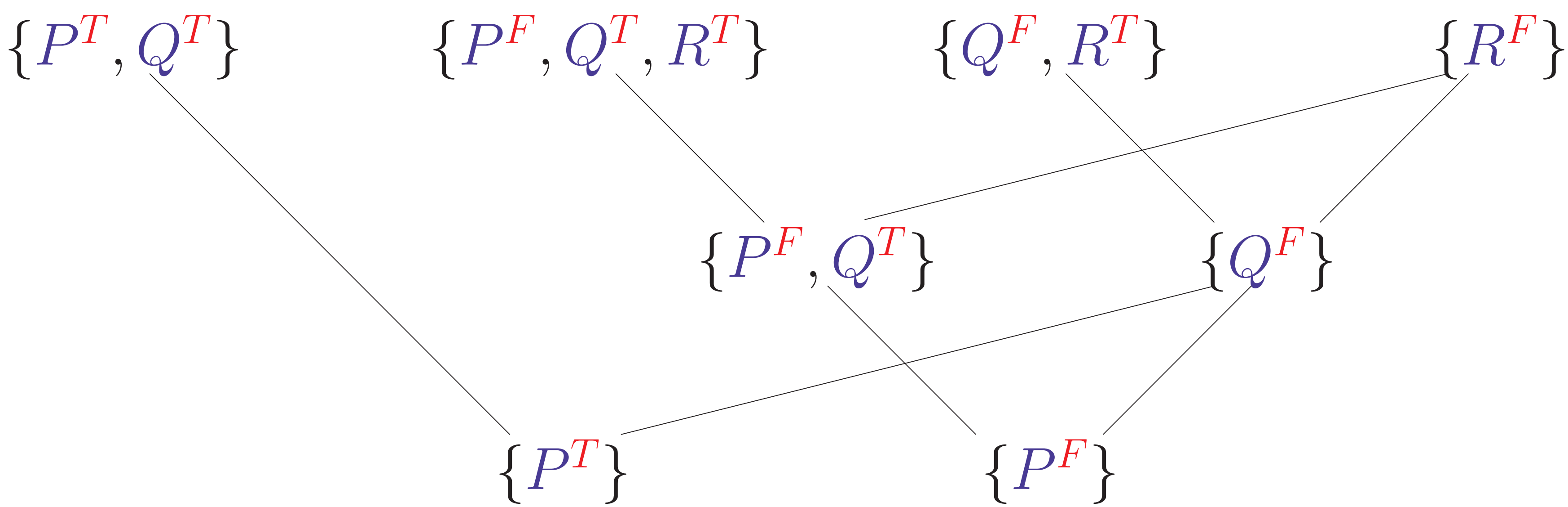
Unit-Resolution

Eine Elternklausel ist immer eine Einerklausel

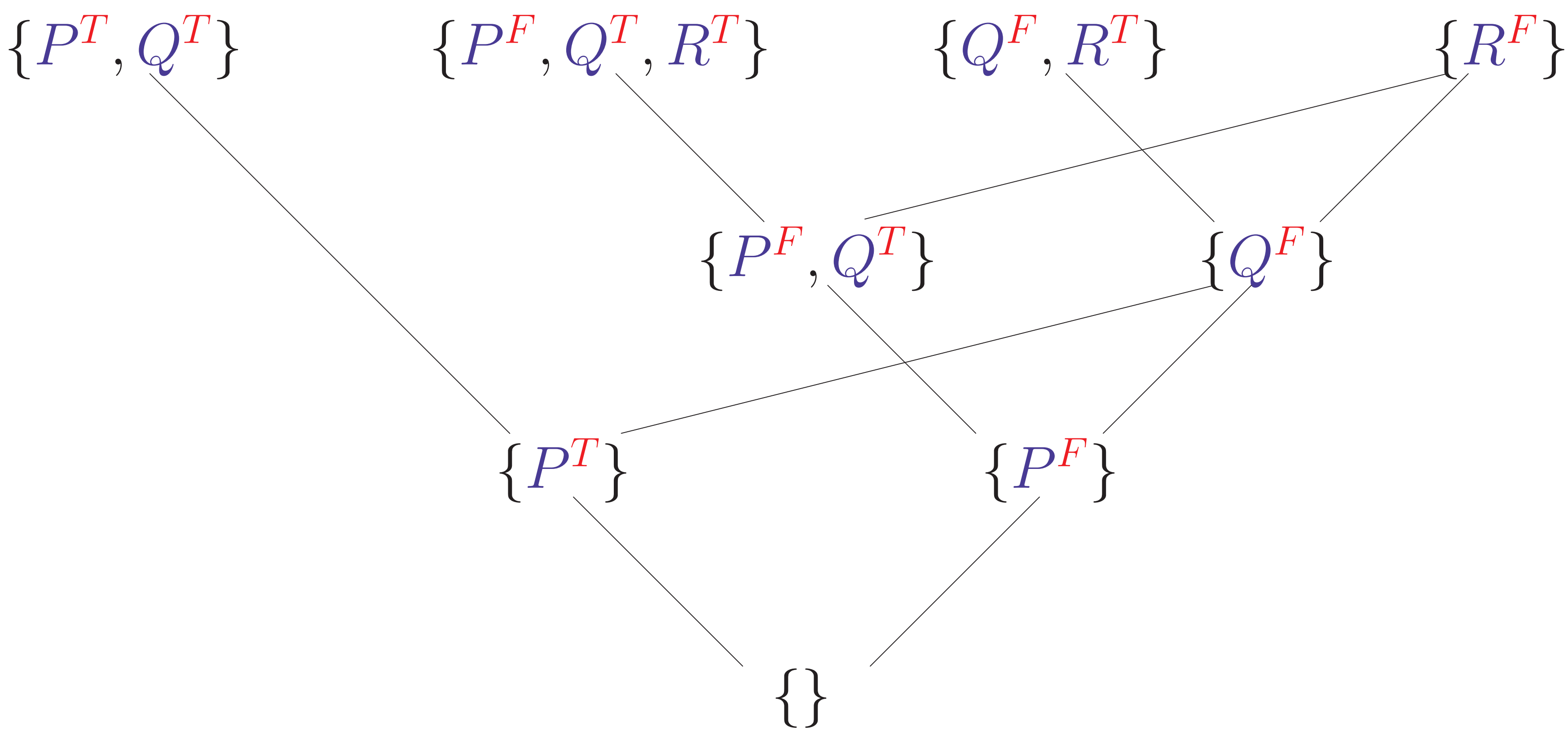


Unit-Resolution

Eine Elternklausel ist immer eine Einerklausel



Eine Elternklausel ist immer eine Einerklausel



Unit-Resulting-Resolution

Strategische Variante der Unit-Resulution

Unit-Resulting-Resolution

Strategische Variante der Unit-Resulution

Gegeben: Matrix $\{c_1, \ldots, c_n\}$ mit $c_i = \{L_{i1}, \ldots, L_{im_i}\}$
Menge von Konnektionen $\{k_1, \ldots, k_m\}$ mit $k_j = \{P_{i_j}{}^T, P_{i'_j}{}^F\}$

Strategische Variante der Unit-Resulution

Gegeben: Matrix $\{c_1, \ldots, c_n\}$ mit $c_i = \{L_{i1}, \ldots, L_{im_i}\}$
Menge von Konnektionen $\{k_1, \ldots, k_m\}$ mit $k_j = \{P_{i_j}{}^T, P_{i'_j}{}^F\}$

Start: n Knoten markiert mit Klauseln $c_i = \{L_{i1}, \ldots, L_{im_i}\}$

Strategische Variante der Unit-Resulution

Gegeben: Matrix $\{c_1, \ldots, c_n\}$ mit $c_i = \{L_{i1}, \ldots, L_{im_i}\}$
Menge von Konnektionen $\{k_1, \ldots, k_m\}$ mit $k_j = \{P_{i_j}{}^T, P_{i'_j}{}^F\}$

Start: n Knoten markiert mit Klauseln $c_i = \{L_{i1}, \ldots, L_{im_i}\}$

Regeln: Wähle Knoten markiert mit Klausel $\{L_1, \ldots, L_k\}$ für ein $k \geq 1$ und $k-1$ Einerklauseln $\{\overline{L_1}\}, \ldots, \{\overline{L_{k-1}}\}$

Generiere als Resolvente die Einerklausel $\{L_k\}$

Unit-Resulting-Resolution

Strategische Variante der Unit-Resulution

Gegeben: Matrix $\{c_1, \ldots, c_n\}$ mit $c_i = \{L_{i1}, \ldots, L_{im_i}\}$
Menge von Konnektionen $\{k_1, \ldots, k_m\}$ mit $k_j = \{{P_{i_j}}^T, {P_{i'_j}}^F\}$

Start: n Knoten markiert mit Klauseln $c_i = \{L_{i1}, \ldots, L_{im_i}\}$

Regeln: Wähle Knoten markiert mit Klausel $\{L_1, \ldots, L_k\}$ für ein $k{\geq}1$ und $k{-}1$ Einerklauseln $\{\overline{L_1}\}, \ldots, \{\overline{L_{k-1}}\}$

Generiere als Resolvente die Einerklausel $\{L_k\}$

Ziel: Ein Knoten, der mit $\{\}$ markiert ist

Strategische Variante der Unit-Resulution

Gegeben: Matrix $\{c_1, \ldots, c_n\}$ mit $c_i = \{L_{i1}, \ldots, L_{im_i}\}$
Menge von Konnektionen $\{k_1, \ldots, k_m\}$ mit $k_j = \{P_{i_j}{}^T, P_{i'_j}{}^F\}$

Start: n Knoten markiert mit Klauseln $c_i = \{L_{i1}, \ldots, L_{im_i}\}$

Regeln: Wähle Knoten markiert mit Klausel $\{L_1, \ldots, L_k\}$ für ein $k{\geq}1$ und $k{-}1$ Einerklauseln $\{\overline{L_1}\}, \ldots, \{\overline{L_{k-1}}\}$

Generiere als Resolvente die Einerklausel $\{L_k\}$

Ziel: Ein Knoten, der mit $\{\}$ markiert ist

Rechtfertigung: Die Regel entspricht mehreren Unit-Resolutionsschritten, die in einem Schritt ausgeführt werden.

Vollständig nur für Hornklauseln

Unit-Resulting-Resolution

$\{P^T, Q^T\}$ $\{P^F, Q^T, R^T\}$ $\{Q^F, R^T\}$ $\{R^F\}$

Unit-Resulting-Resolution

$$\{P^T, Q^T\} \qquad \{P^F, Q^T, R^T\} \qquad \{Q^F, R^T\} \qquad \{R^F\}$$

$$\{Q^F\}$$

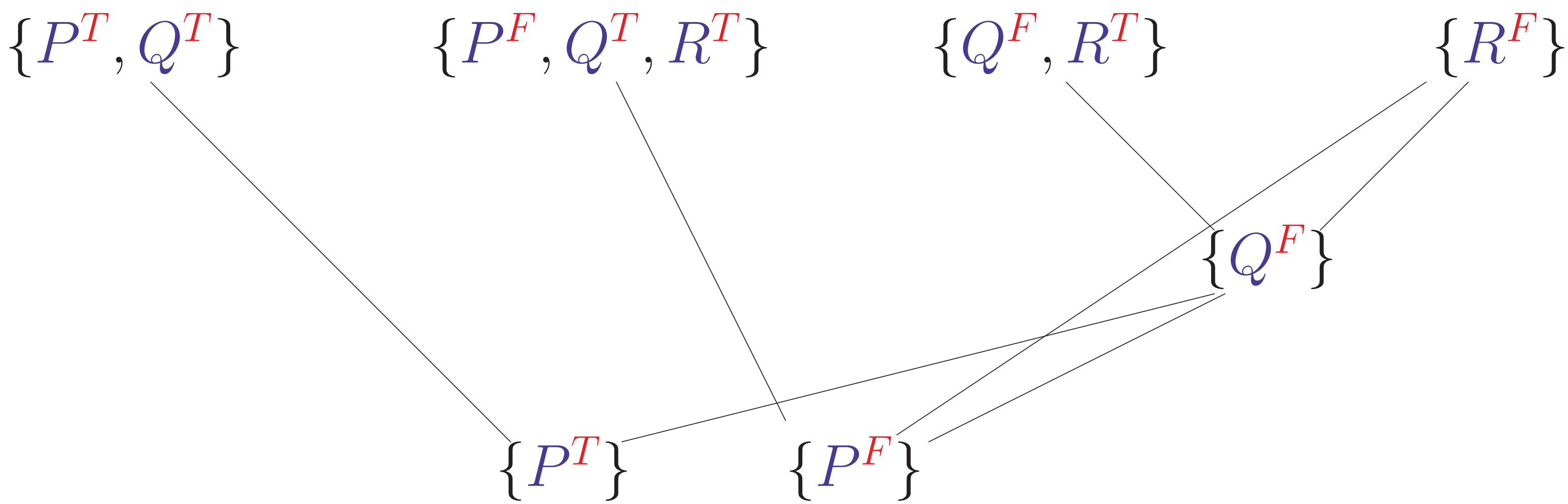
Unit-Resulting-Resolution

$\{P^T, Q^T\}$ $\{P^F, Q^T, R^T\}$ $\{Q^F, R^T\}$ $\{R^F\}$

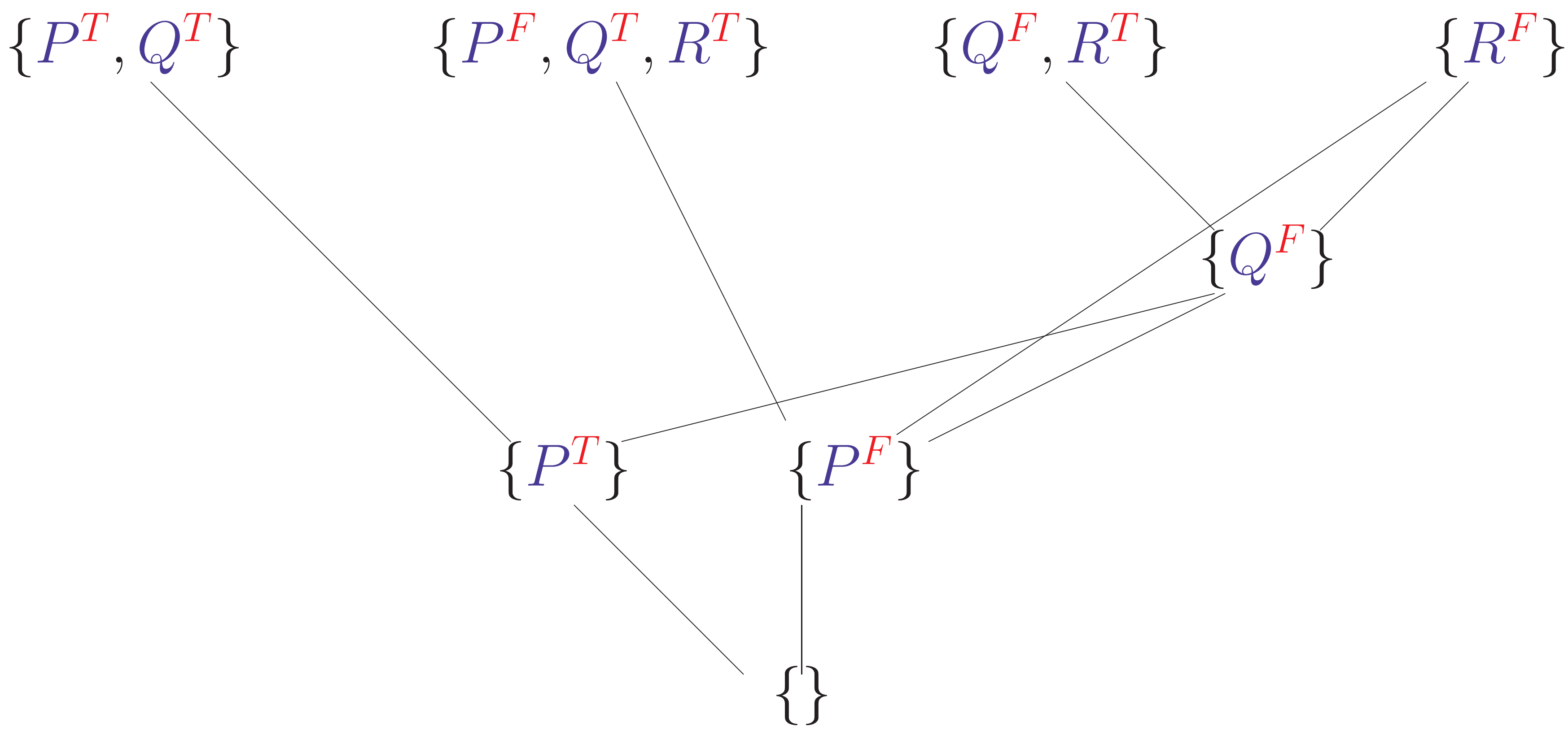
$\{Q^F\}$

$\{P^F\}$

Unit-Resulting-Resolution



Unit-Resulting-Resolution



Set-of-support Resolution

$$
\begin{tikzcd}[column sep=small]
\{P^T, Q^T\} \arrow[dr, dash] & & \{P^F, Q^T, R^T\} \arrow[dl, dash] & & \{Q^F, R^T\} \arrow[ddll, dash] & & \{R^F\} \arrow[dddlll, dash] \\
& \{Q^T, R^T\} \arrow[dr, dash] & & & & & \\
& & \{R^T\} \arrow[dr, dash] & & & & \\
& & & \{\} & & &
\end{tikzcd}
$$

Set-of-support Resolution

$$\{P^T, Q^T\} \qquad \{P^F, Q^T, R^T\} \qquad \{Q^F, R^T\} \qquad \{R^F\}$$

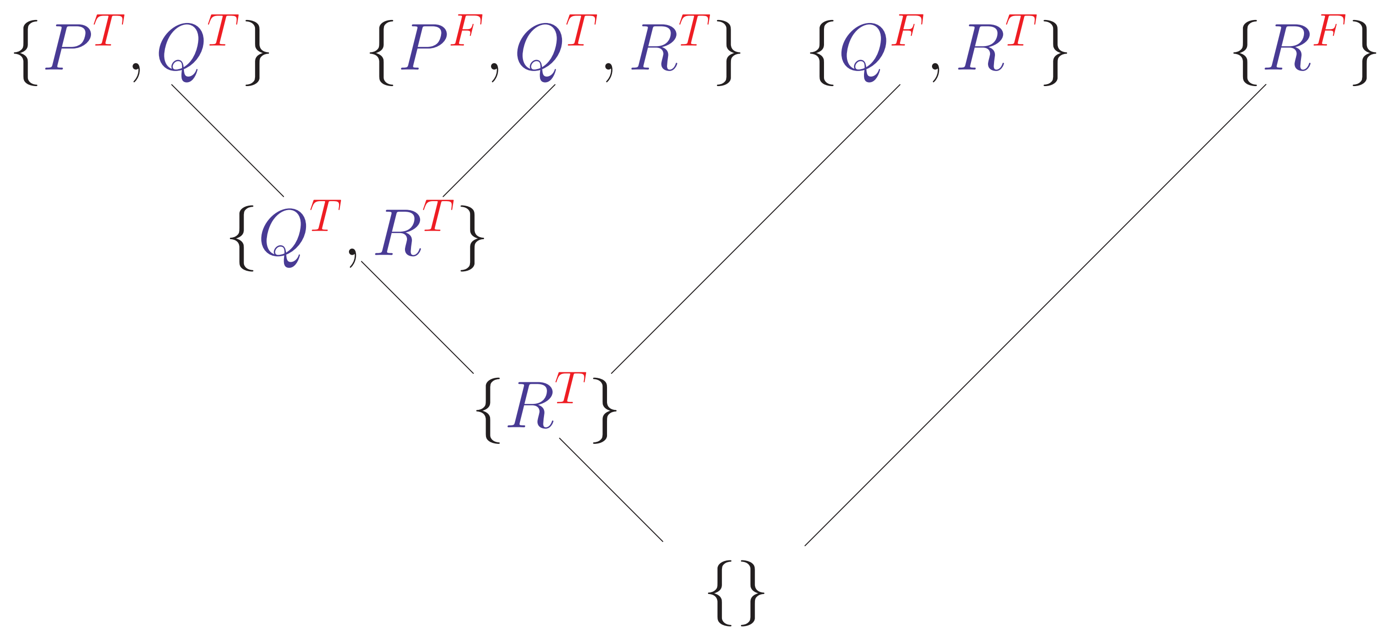
$$\{Q^T, R^T\}$$

$$\{R^T\}$$

$$\{\}$$

- **Stützmenge: Menge beweisrelevanter Klauseln**
  - Formel ohne diese Menge nicht beweisbar
  - Beweisziel sollte enthalten sein
  - Im Beispiel: $\{\ P^T, Q^T\ \}$

Set-of-support Resolution



- **Stützmenge: Menge beweisrelevanter Klauseln**
  - Formel ohne diese Menge nicht beweisbar
  - Beweisziel sollte enthalten sein
  - Im Beispiel: $\{ P^T, Q^T \}$
- **Strategie:** resolviere mit Nachkommen der Stützmenge
  - Vollständig genau dann, wenn Restklauselmenge erfüllbar

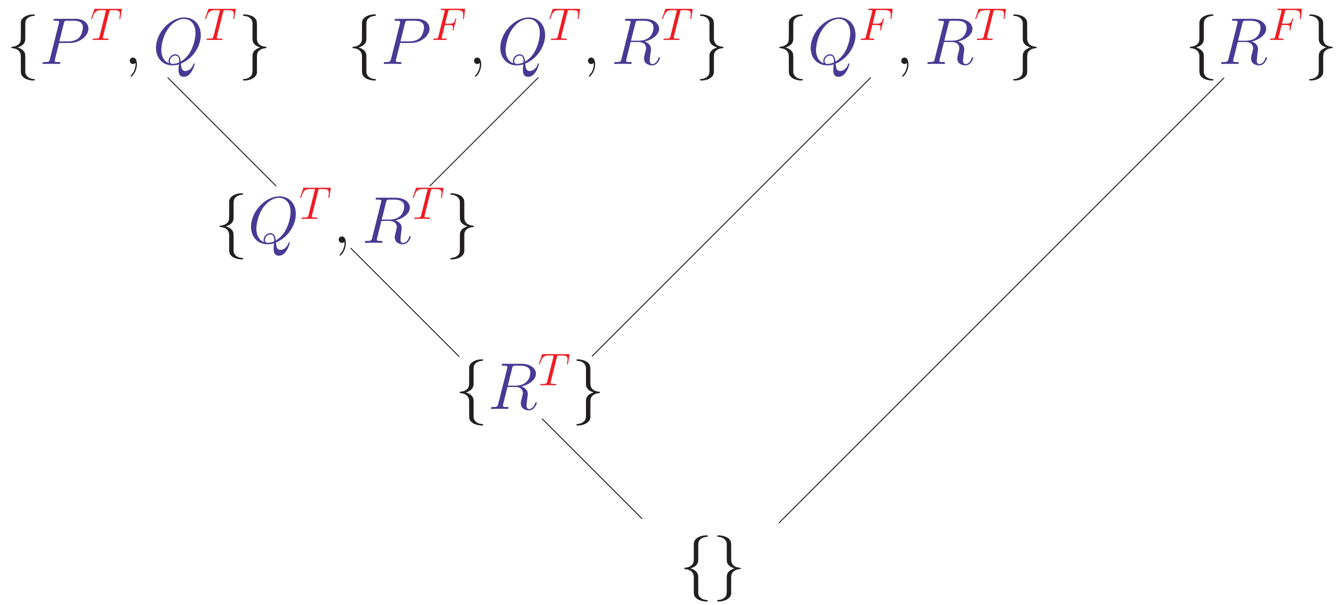
SLD-Resolution

$$\{P^T, Q^T\} \qquad \{P^F, Q^T, R^T\} \qquad \{Q^F, R^T\} \qquad \{R^F\}$$

$$\{Q^T, R^T\}$$

$$\{R^T\}$$

$$\{\}$$



- **Lineare Input Resolution**

  + Wahl des Literals immer aus "neu" erzeugten Literalen

  + Wahl der Inputklausel aus "Definiten Klauseln"

  (Hornklauseln mit genau einem negativen Literal)

SLD-Resolution

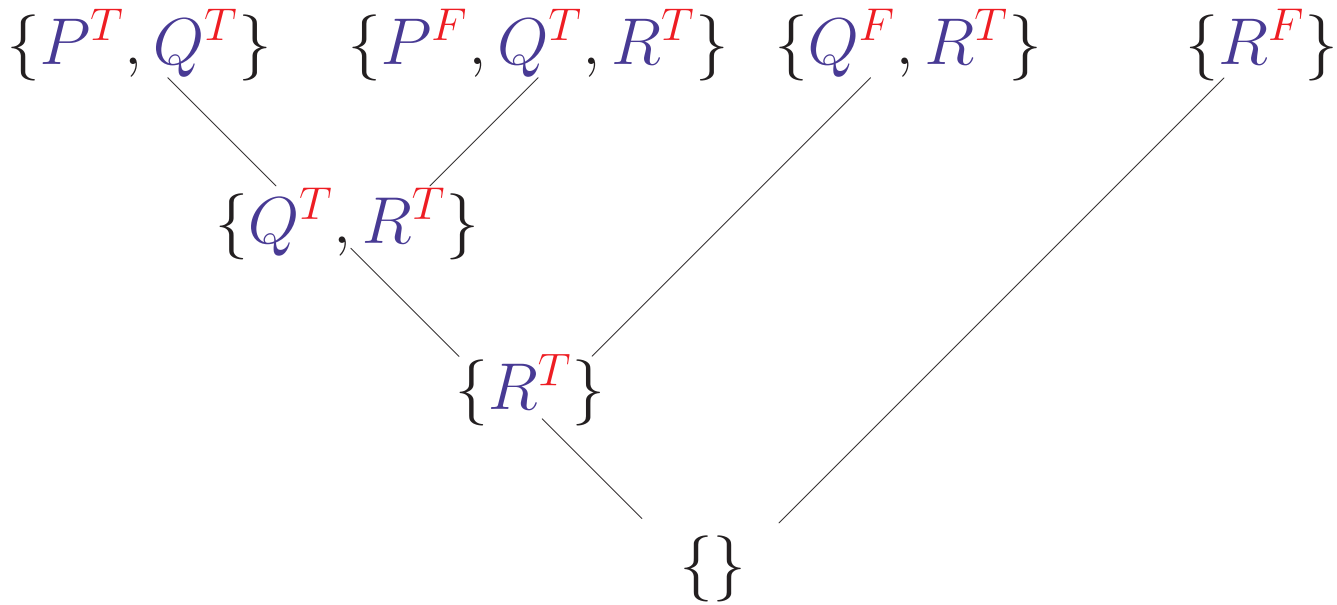
$\{P^T, Q^T\}$ $\{P^F, Q^T, R^T\}$ $\{Q^F, R^T\}$ $\{R^F\}$

$\{Q^T, R^T\}$

$\{R^T\}$

$\{\}$

- **Lineare Input Resolution**
  - \+ Wahl des Literals immer aus "neu" erzeugten Literalen
  - \+ Wahl der Inputklausel aus "Definiten Klauseln"
    (Hornklauseln mit genau einem negativen Literal)
  - – Sehr effizient durch lineare Verkettung der Beweisschritte



- **Lineare Input Resolution**
  - \+ Wahl des Literals immer aus "neu" erzeugten Literalen
  - \+ Wahl der Inputklausel aus "Definiten Klauseln"
    (Hornklauseln mit genau einem negativen Literal)
  - – Sehr effizient durch lineare Verkettung der Beweisschritte
- **Grundlage von Prolog (negative Repräsentation)**
  - – Programmiersprache mit Hornklauseln – nicht Beweistechnik

Hyper-Resolution

Gegeben: Matrix $\{c_1, \ldots, c_n\}$ mit $c_i = \{L_{i1}, \ldots, L_{im_i}\}$
Menge von Konnektionen $\{k_1, \ldots, k_m\}$ mit $k_j = \{P_{i_j}{}^T, P_{i'_j}{}^F\}$

Hyper-Resolution

Gegeben: Matrix $\{c_1, \ldots, c_n\}$ mit $c_i = \{L_{i1}, \ldots, L_{im_i}\}$
Menge von Konnektionen $\{k_1, \ldots, k_m\}$ mit $k_j = \{P_{i_j}{}^T, P_{i'_j}{}^F\}$

Start: n Knoten markiert mit Klauseln $c_i = \{L_{i1}, \ldots, L_{im_i}\}$

Hyper-Resolution

Gegeben: Matrix $\{c_1, \ldots, c_n\}$ mit $c_i = \{L_{i1}, \ldots, L_{im_i}\}$
Menge von Konnektionen $\{k_1, \ldots, k_m\}$ mit $k_j = \{P_{i_j}{}^T, P_{i'_j}{}^F\}$

Start: n Knoten markiert mit Klauseln $c_i = \{L_{i1}, \ldots, L_{im_i}\}$

Regeln: Wähle Knoten markiert mit Klausel $K \cup \{P_1{}^F, \ldots, P_k{}^F\}$ für ein $k \geq 1$, wobei K positive Teilklausel. (Nukleus).
Wähle rein positive Klauseln $K_1 \cup \{P_1{}^T\}, \ldots K_k \cup \{P_k{}^T\}$ (Elektronen)

Generiere als Resolvente die (positive) Klausel $K \cup K_1 \ldots K_k$

Hyper-Resolution

Gegeben: Matrix $\{c_1, \ldots, c_n\}$ mit $c_i = \{L_{i1}, \ldots, L_{im_i}\}$
Menge von Konnektionen $\{k_1, \ldots, k_m\}$ mit $k_j = \{P_{i_j}{}^T, P_{i'_j}{}^F\}$

Start: n Knoten markiert mit Klauseln $c_i = \{L_{i1}, \ldots, L_{im_i}\}$

Regeln: Wähle Knoten markiert mit Klausel $K \cup \{P_1{}^F, \ldots, P_k{}^F\}$ für ein $k \geq 1$, wobei K positive Teilklausel. (Nukleus).
Wähle rein positive Klauseln $K_1 \cup \{P_1{}^T\}, \ldots K_k \cup \{P_k{}^T\}$ (Elektronen)
Generiere als Resolvente die (positive) Klausel $K \cup K_1 \ldots K_k$

Ziel: Ein Knoten, der mit $\{\}$ markiert ist

Hyper-Resolution

Gegeben: Matrix $\{c_1, \ldots, c_n\}$ mit $c_i = \{L_{i1}, \ldots, L_{im_i}\}$
Menge von Konnektionen $\{k_1, \ldots, k_m\}$ mit $k_j = \{P_{i_j}{}^T, P_{i'_j}{}^F\}$

Start: n Knoten markiert mit Klauseln $c_i = \{L_{i1}, \ldots, L_{im_i}\}$

Regeln: Wähle Knoten markiert mit Klausel $K \cup \{P_1{}^F, \ldots, P_k{}^F\}$ für ein $k \geq 1$, wobei K positive Teilklausel. (Nukleus).
Wähle rein positive Klauseln $K_1 \cup \{P_1{}^T\}, \ldots K_k \cup \{P_k{}^T\}$ (Elektronen)
Generiere als Resolvente die (positive) Klausel $K \cup K_1 \ldots K_k$

Ziel: Ein Knoten, der mit $\{\}$ markiert ist

Rechtfertigung: Die Hyperresolutionsregel entspricht mehreren P_1-Resolutionsschritten, die in einem Schritt ausgeführt werden.

Hyper-Resolution

Gegeben: Matrix $\{c_1, \ldots, c_n\}$ mit $c_i = \{L_{i1}, \ldots, L_{im_i}\}$
Menge von Konnektionen $\{k_1, \ldots, k_m\}$ mit $k_j = \{P_{i_j}{}^T, P_{i'_j}{}^F\}$

Start: n Knoten markiert mit Klauseln $c_i = \{L_{i1}, \ldots, L_{im_i}\}$

Regeln: Wähle Knoten markiert mit Klausel $K \cup \{P_1{}^F, \ldots, P_k{}^F\}$ für ein $k \geq 1$, wobei K positive Teilklausel. (Nukleus).
Wähle rein positive Klauseln $K_1 \cup \{P_1{}^T\}, \ldots K_k \cup \{P_k{}^T\}$ (Elektronen)

Generiere als Resolvente die (positive) Klausel $K \cup K_1 \ldots K_k$

Ziel: Ein Knoten, der mit $\{\}$ markiert ist

Rechtfertigung: Die Hyperresolutionsregel entspricht mehreren P_1-Resolutionsschritten, die in einem Schritt ausgeführt werden.

Negative Variante ebenfalls möglich

Hyper-Resolution für $P \wedge Q \vee \neg P \wedge Q \wedge \neg R \vee \neg Q \wedge \neg R \vee R$

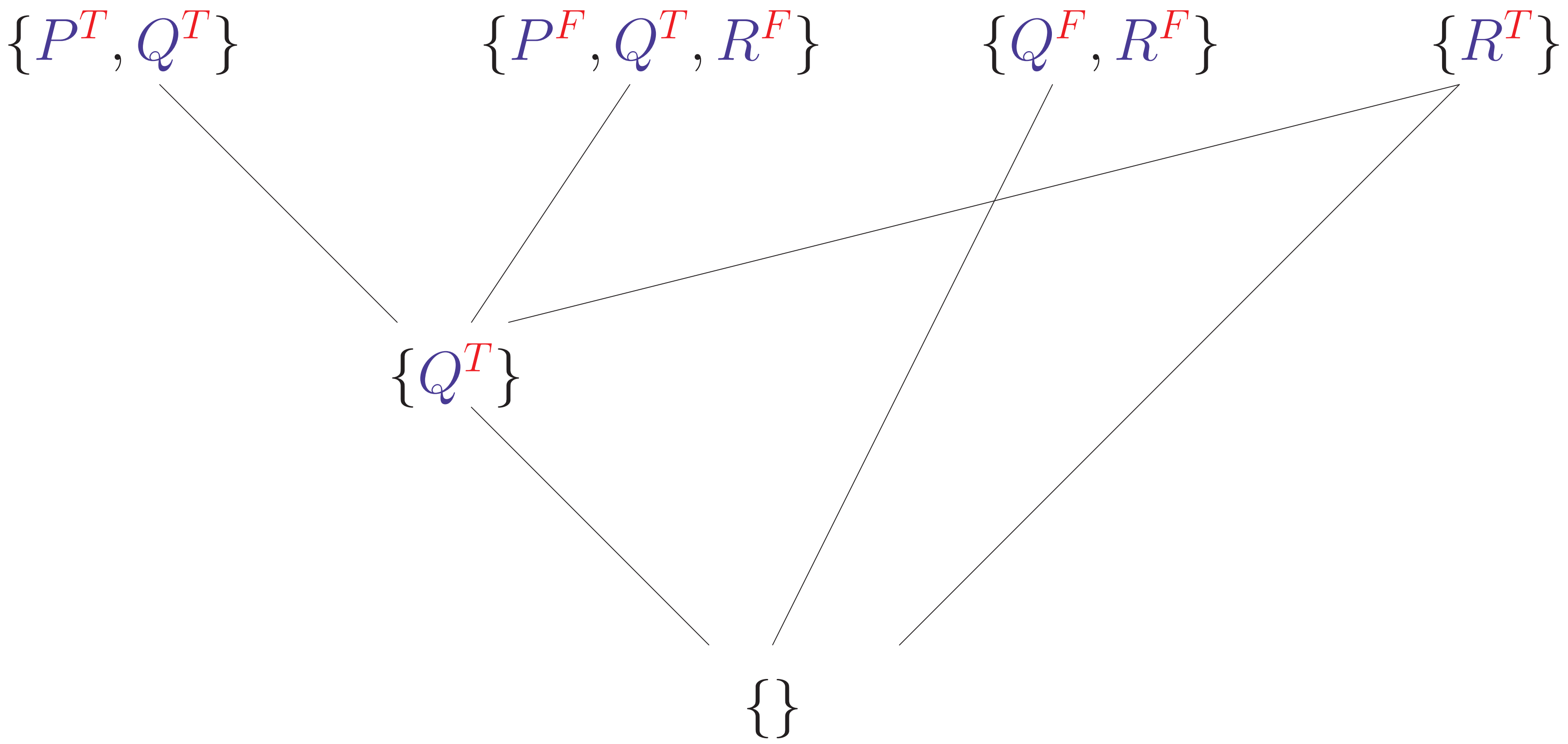
$\{P^T, Q^T\}$ $\{P^F, Q^T, R^F\}$ $\{Q^F, R^F\}$ $\{R^T\}$

Hyper-Resolution für $P \wedge Q \vee \neg P \wedge Q \wedge \neg R \vee \neg Q \wedge \neg R \vee R$

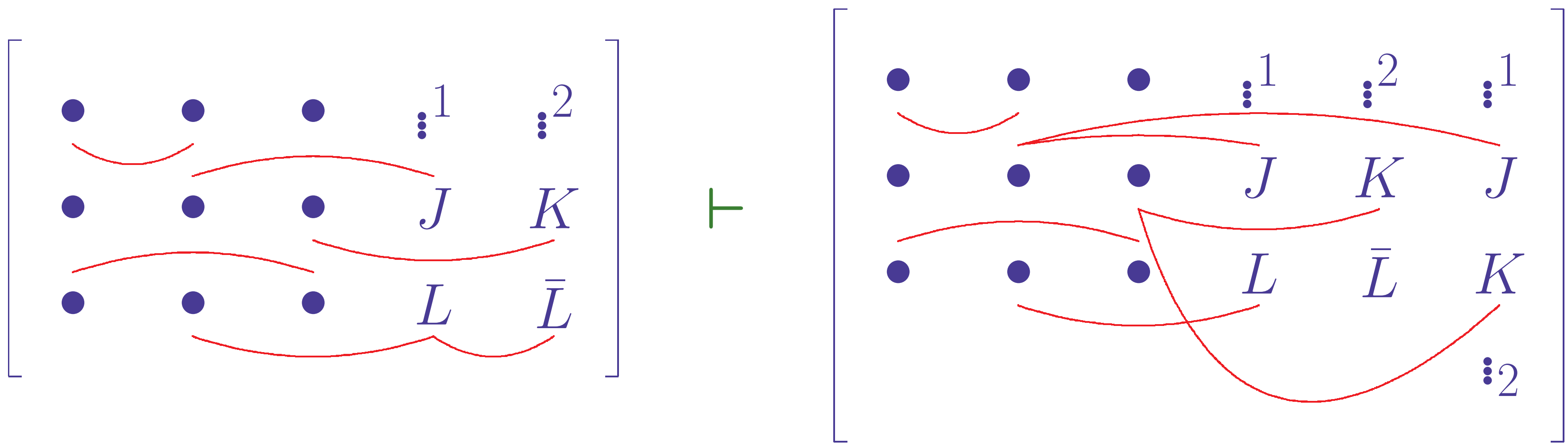
$\{P^T, Q^T\}$ $\{P^F, Q^T, R^F\}$ $\{Q^F, R^F\}$ $\{R^T\}$

$\{Q^T\}$

Hyper-Resolution für $P \wedge Q \vee \neg P \wedge Q \wedge \neg R \vee \neg Q \wedge \neg R \vee R$

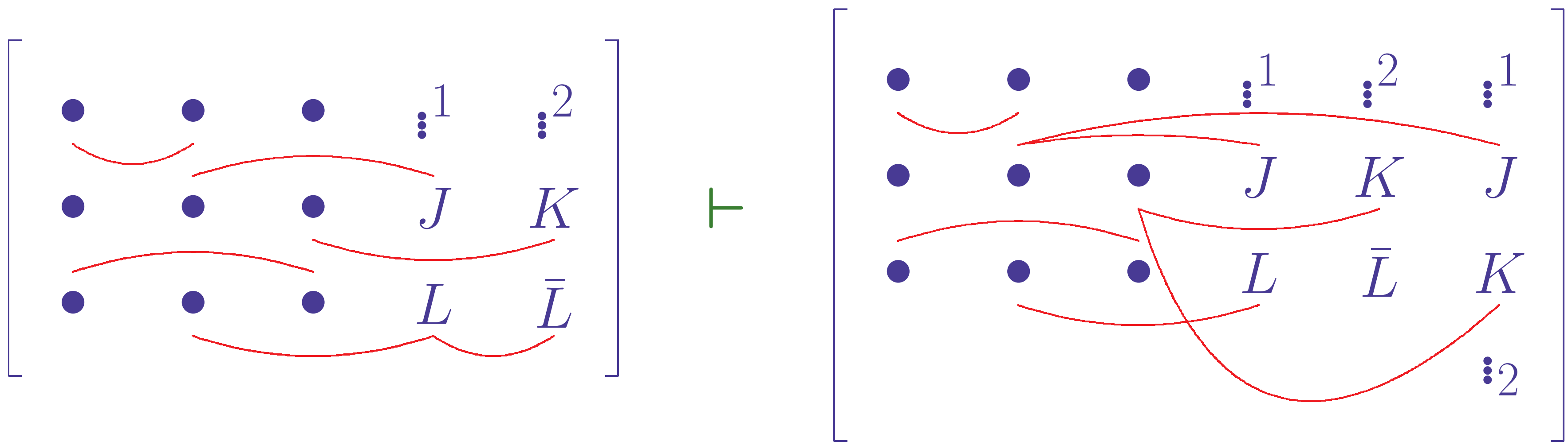


Konnektionsgraphen Resolution



- **Verhindere Mehrfachresolution über gleiche Konnektion**

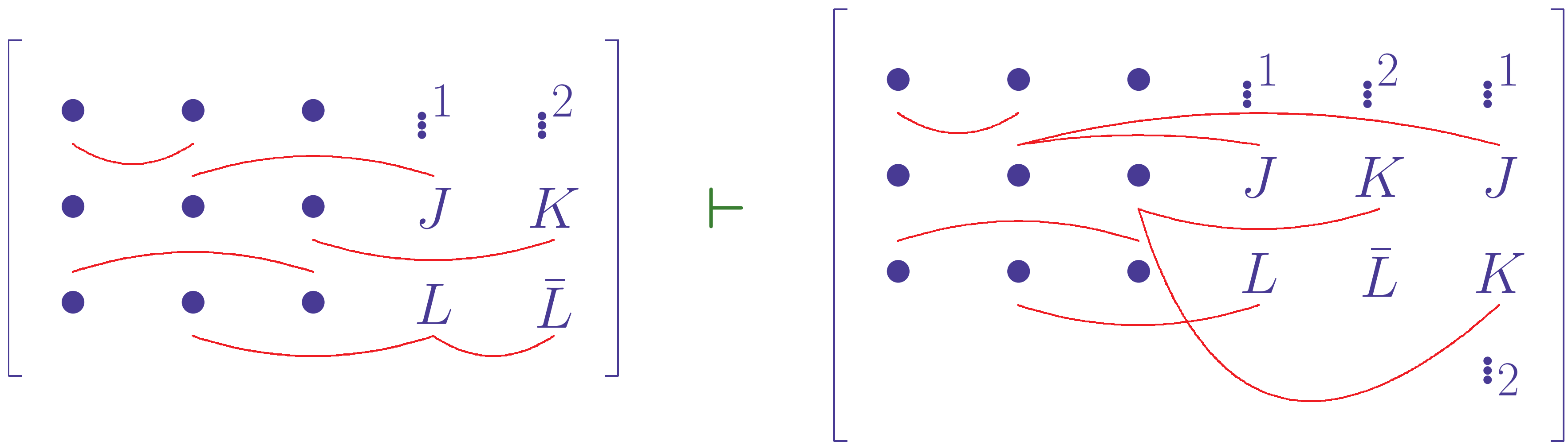
Konnektionsgraphen Resolution



- **Verhindere Mehrfachresolution über gleiche Konnektion**

- **Markiere unbenutzte Konnektionen in Klauselgraphen**
  Vererbe diese bei Resolutionschritt

Konnektionsgraphen Resolution



- **Verhindere Mehrfachresolution über gleiche Konnektion**

- **Markiere unbenutzte Konnektionen in Klauselgraphen**
  Vererbe diese bei Resolutionschritt

  Technisch sehr aufwendig

Konsolution

Verallgemeinerung von Resolution und Extensionsverfahren

Konsolution

Verallgemeinerung von Resolution und Extensionsverfahren

Gegeben: Matrix $\{c_1, \ldots, c_n\}$ mit $c_i = \{L_{i1}, \ldots, L_{im_i}\}$
Menge von Konnektionen $\{k_1, \ldots, k_m\}$ mit $k_j = \{P_{i_j}{}^T, P_{i'_j}{}^F\}$

Konsolution

Verallgemeinerung von Resolution und Extensionsverfahren

Gegeben: Matrix $\{c_1, \ldots, c_n\}$ mit $c_i = \{L_{i1}, \ldots, L_{im_i}\}$
Menge von Konnektionen $\{k_1, \ldots, k_m\}$ mit $k_j = \{P_{i_j}{}^T, P_{i'_j}{}^F\}$

Start: n Knoten markiert mit einelementigen Pfaden $\{\{L_{i1}\}, \ldots, \{L_{im_i}\}\}$

Konsolution

Verallgemeinerung von Resolution und Extensionsverfahren

Gegeben: Matrix $\{c_1, \ldots, c_n\}$ mit $c_i = \{L_{i1}, \ldots, L_{im_i}\}$

Menge von Konnektionen $\{k_1, \ldots, k_m\}$ mit $k_j = \{P_{i_j}{}^{T}, P_{i'_j}{}^{F}\}$

Start: n Knoten markiert mit einelementigen Pfaden $\{\{L_{i1}\}, \ldots, \{L_{im_i}\}\}$

Regeln: **Konsolution**:

Wähle zwei Knoten markiert mit Pfadmengen $\mathcal{P}$ und $\mathcal{Q}$

Generiere einen Nachfolger markiert mit dem Pfadmengenprodukt

$$\mathcal{P}\mathcal{Q} = \{p \cup q \mid p \in \mathcal{P}, q \in \mathcal{Q}, p \cup q \text{ nicht komplementär}\}$$

Vereinfachung: Ersetze Markierung $\{p_1, \ldots, p_m\}$ eines Knotens durch $\{p'_1, \ldots, p'_m\}$ wobei die p'_i Teilpfade der Pfade p_i sind

Konsolution

Verallgemeinerung von Resolution und Extensionsverfahren

Gegeben: Matrix $\{c_1, \ldots, c_n\}$ mit $c_i = \{L_{i1}, \ldots, L_{im_i}\}$
Menge von Konnektionen $\{k_1, \ldots, k_m\}$ mit $k_j = \{P_{i_j}{}^T, P_{i'_j}{}^F\}$

Start: n Knoten markiert mit einelementigen Pfaden $\{\{L_{i1}\}, \ldots, \{L_{im_i}\}\}$

Regeln: **Konsolution**:
Wähle zwei Knoten markiert mit Pfadmengen $\mathcal{P}$ und $\mathcal{Q}$
Generiere einen Nachfolger markiert mit dem Pfadmengenprodukt

$$\mathcal{P}\mathcal{Q} = \{p \cup q \mid p \in \mathcal{P}, q \in \mathcal{Q}, p \cup q \text{ nicht komplementär}\}$$

Vereinfachung: Ersetze Markierung $\{p_1, \ldots, p_m\}$ eines Knotens durch $\{p'_1, \ldots, p'_m\}$ wobei die p'_i Teilpfade der Pfade p_i sind

Ziel: Ein Knoten, der mit $\{\}$ markiert ist

Konsolution

Verallgemeinerung von Resolution und Extensionsverfahren

Gegeben: Matrix $\{c_1,\ldots,c_n\}$ mit $c_i = \{L_{i1},\ldots,L_{im_i}\}$
Menge von Konnektionen $\{k_1,\ldots,k_m\}$ mit $k_j = \{P_{i_j}{}^T, P_{i'_j}{}^F\}$

Start: n Knoten markiert mit einelementigen Pfaden $\{\{L_{i1}\},\ldots,\{L_{im_i}\}\}$

Regeln: **Konsolution**:
Wähle zwei Knoten markiert mit Pfadmengen $\mathcal{P}$ und $\mathcal{Q}$
Generiere einen Nachfolger markiert mit dem Pfadmengenprodukt

$$\mathcal{P}\mathcal{Q} = \{p{\cup}q \mid p\in\mathcal{P}, q\in\mathcal{Q}, p{\cup}q \text{ nicht komplementär}\}$$

Vereinfachung: Ersetze Markierung $\{p_1,\ldots,p_m\}$ eines Knotens durch $\{p'_1,\ldots,p'_m\}$ wobei die p'_i Teilpfade der Pfade p_i sind

Ziel: Ein Knoten, der mit $\{\}$ markiert ist

Rechtfertigung: Knoten codieren Teilpfade, die noch nicht als komplementär nachgeweisen sind. Konsolution verkettet zwei Pfade und streicht die komplementären. Vereinfachung gestattet die Betrachtung beliebiger Teilpfade. Das Verfahren endet, wenn kein Pfad übrigbleibt, der nicht komplementär ist.

Konsolutions-Ableitung für $P \wedge Q \vee \neg P \wedge Q \wedge R \vee \neg Q \wedge R \vee \neg R$

$\{\{P^T\},\{Q^T\}\}$ $\{\{P^F\},\{Q^T\},\{R^T\}\}$ $\{\{Q^F\},\{R^T\}\}$ $\{\{R^F\}\}$

Konsolutions-Ableitung für $P \wedge Q \vee \neg P \wedge Q \wedge R \vee \neg Q \wedge R \vee \neg R$

$\{\{\boxed{P^T}\},\{Q^T\}\}$ $\quad$ $\{\{\boxed{P^F}\},\{Q^T\},\{R^T\}\}$ $\quad$ $\{\{Q^F\},\{R^T\}\}$ $\quad$ $\{\{R^F\}\}$

$\{\{P^T,Q^T\},\{P^T,R^T\},\{Q^T,P^F\},\{Q^T\},\{Q^T,R^T\}\}$

Konsolutions-Ableitung für $P \wedge Q \vee \neg P \wedge Q \wedge R \vee \neg Q \wedge R \vee \neg R$

$$\{\{P^T\},\{Q^T\}\} \qquad \{\{P^F\},\{Q^T\},\{R^T\}\} \qquad \{\{\boxed{Q^F}\},\{\boxed{R^T}\}\} \qquad \{\{R^F\}\}$$

$$\{\{P^T,Q^T\},\{P^T,\boxed{R^T}\},\{\boxed{Q^T},P^F\},\{\boxed{Q^T}\},\{\boxed{Q^T},\boxed{R^T}\}\}$$

$$\{\{P^T,Q^T,R^T\},\{P^T,Q^F,R^T\},\{P^T,R^T\},\{Q^T,P^F,R^T\},\{Q^T,R^T\}\}$$

Konsolutions-Ableitung für $P \wedge Q \vee \neg P \wedge Q \wedge R \vee \neg Q \wedge R \vee \neg R$

$$\{\{P^T\},\{Q^T\}\} \qquad \{\{P^F\},\{Q^T\},\{R^T\}\} \qquad \{\{Q^F\},\{R^T\}\} \qquad \{\{\boxed{R^F}\}\}$$

$$\{\{P^T,Q^T\},\{P^T,R^T\},\{Q^T,P^F\},\{Q^T\},\{Q^T,R^T\}\}$$

$$\{\{P^T,Q^T,\boxed{R^T}\},\{P^T,Q^F,\boxed{R^T}\},\{P^T,\boxed{R^T}\},\{Q^T,P^F,\boxed{R^T}\},\{Q^T,\boxed{R^T}\}\}$$

$$\{\}$$

Deduktionsverfahren: Übersicht

- **Extensionsverfahren**: Folge Konnektionen und halte als aktuellen Pfad, was noch nicht als komplentär erkannt wurde. Halte unbetrachtete Literale der beteiligten Klauseln auf dem Stack.

Deduktionsverfahren: Übersicht

- **Extensionsverfahren**: Folge Konnektionen und halte als aktuellen Pfad, was noch nicht als komplentär erkannt wurde. Halte unbetrachtete Literale der beteiligten Klauseln auf dem Stack.
- **Analytische Tableaux**: andere Technik um Pfade aufzuschreiben und zu stoppen, wenn sie komplementär sind.

Deduktionsverfahren: Übersicht

- **Extensionsverfahren**: Folge Konnektionen und halte als aktuellen Pfad, was noch nicht als komplentär erkannt wurde. Halte unbetrachtete Literale der beteiligten Klauseln auf dem Stack.
- **Analytische Tableaux**: andere Technik um Pfade aufzuschreiben und zu stoppen, wenn sie komplementär sind.
- **Modellelimination**: Elimination von Modellen, welche die Matrix falsifizieren könnten

Deduktionsverfahren: Übersicht

- **Extensionsverfahren**: Folge Konnektionen und halte als aktuellen Pfad, was noch nicht als komplentär erkannt wurde. Halte unbetrachtete Literale der beteiligten Klauseln auf dem Stack.
- **Analytische Tableaux**: andere Technik um Pfade aufzuschreiben und zu stoppen, wenn sie komplementär sind.
- **Modellelimination**: Elimination von Modellen, welche die Matrix falsifizieren könnten
- **Resolution**: leite aus zwei Klauseln mit komplementären Literalen eine Resolvente ab, die erfüllbar ist, genau dann wenn die Teilmatrix der Elternklauseln erfüllbar ist

Deduktionsverfahren: Übersicht

- **Extensionsverfahren**: Folge Konnektionen und halte als aktuellen Pfad, was noch nicht als komplentär erkannt wurde. Halte unbetrachtete Literale der beteiligten Klauseln auf dem Stack.
- **Analytische Tableaux**: andere Technik um Pfade aufzuschreiben und zu stoppen, wenn sie komplementär sind.
- **Modellelimination**: Elimination von Modellen, welche die Matrix falsifizieren könnten
- **Resolution**: leite aus zwei Klauseln mit komplementären Literalen eine Resolvente ab, die erfüllbar ist, genau dann wenn die Teilmatrix der Elternklauseln erfüllbar ist
- **Semantische Bäume**: Beginne alle Modelle aufzuschreiben und stoppe, wenn ein Modell durch eine Klausel erfüllt (widerlegt) wird.

Deduktionsverfahren: Übersicht

- **Extensionsverfahren**: Folge Konnektionen und halte als aktuellen Pfad, was noch nicht als komplentär erkannt wurde. Halte unbetrachtete Literale der beteiligten Klauseln auf dem Stack.
- **Analytische Tableaux**: andere Technik um Pfade aufzuschreiben und zu stoppen, wenn sie komplementär sind.
- **Modellelimination**: Elimination von Modellen, welche die Matrix falsifizieren könnten
- **Resolution**: leite aus zwei Klauseln mit komplementären Literalen eine Resolvente ab, die erfüllbar ist, genau dann wenn die Teilmatrix der Elternklauseln erfüllbar ist
- **Semantische Bäume**: Beginne alle Modelle aufzuschreiben und stoppe, wenn ein Modell durch eine Klausel erfüllt (widerlegt) wird.
- **Konsolution**: Notiere *alle* Pfadmengen, die durch die Vorgängerklauseln gehen und noch nicht als komplementär nachgewiesen wurden.

Deduktionsverfahren: Übersicht

- **Extensionsverfahren**: Folge Konnektionen und halte als aktuellen Pfad, was noch nicht als komplentär erkannt wurde. Halte unbetrachtete Literale der beteiligten Klauseln auf dem Stack.
- **Analytische Tableaux**: andere Technik um Pfade aufzuschreiben und zu stoppen, wenn sie komplementär sind.
- **Modellelimination**: Elimination von Modellen, welche die Matrix falsifizieren könnten
- **Resolution**: leite aus zwei Klauseln mit komplementären Literalen eine Resolvente ab, die erfüllbar ist, genau dann wenn die Teilmatrix der Elternklauseln erfüllbar ist
- **Semantische Bäume**: Beginne alle Modelle aufzuschreiben und stoppe, wenn ein Modell durch eine Klausel erfüllt (widerlegt) wird.
- **Konsolution**: Notiere *alle* Pfadmengen, die durch die Vorgängerklauseln gehen und noch nicht als komplementär nachgewiesen wurden.
- **Maslov-Verfahren (Inverse Methode)**: Verarbeite Klauseln und leite dabei – beginnend mit den Konnektionen – Teilpfade ab, die alle offenen Pfade garantiert komplementär machen.

Deduktionsverfahren: Übersicht

- **Extensionsverfahren**: Folge Konnektionen und halte als aktuellen Pfad, was noch nicht als komplentär erkannt wurde. Halte unbetrachtete Literale der beteiligten Klauseln auf dem Stack.
- **Analytische Tableaux**: andere Technik um Pfade aufzuschreiben und zu stoppen, wenn sie komplementär sind.
- **Modellelimination**: Elimination von Modellen, welche die Matrix falsifizieren könnten
- **Resolution**: leite aus zwei Klauseln mit komplementären Literalen eine Resolvente ab, die erfüllbar ist, genau dann wenn die Teilmatrix der Elternklauseln erfüllbar ist
- **Semantische Bäume**: Beginne alle Modelle aufzuschreiben und stoppe, wenn ein Modell durch eine Klausel erfüllt (widerlegt) wird.
- **Konsolution**: Notiere *alle* Pfadmengen, die durch die Vorgängerklauseln gehen und noch nicht als komplementär nachgewiesen wurden.
- **Maslov-Verfahren (Inverse Methode)**: Verarbeite Klauseln und leite dabei – beginnend mit den Konnektionen – Teilpfade ab, die alle offenen Pfade garantiert komplementär machen.
- **Aussagenlogische Reduktionstechniken**: Anwendung von Reduktionstechniken und Aufspaltung von Matrizen