Kalküle

Ein **Kalkül** in der Aussagenlogik ist eine Menge von Regeln, welche aus einer Menge $\mathcal{F}$ aussagenlogischer Formeln eine Menge ebensolcher Formeln **ableiten**, Zeichen $\vdash_K$

Sinnvolle Eigenschaften eines aussagenlogischen Kalküls:

- **Korrektheit**: Nur folgerbare Formeln werden abgeleitet
- **Vollständigkeit**: Alle folgerbaren Formeln sind ableitbar

Vollständigkeit kann z.B. durch Verwendung eines vollständigen Suchverfahrens erzielt werden!

Der Resolutionskalkül

Resolution beruht auf Anwendung des Modus ponens …

$$\frac{P \Rightarrow Q \quad \Big| \quad P}{Q}$$

… in der umgeformten, verallgemeinerten Variante

$$\frac{\neg P \vee \alpha \quad \Big| \quad P \vee \beta}{\alpha \vee \beta}$$

Wir benutzen die Darstellung von Formeln in KNF,
d.h. α und β sind Disjunktionen von Literalen

J. Alan Robinson

Die Resolutionsregel

Gegeben zwei Klauseln $K_1=\{P\}\cup K'_1$ und $K_2=\{\neg P\}\cup K'_2$, wobei P eine beliebige Aussagevariable ist (P, $\neg P$ sind jeweils Literale). Die Klausel $Res(K_1,K_2)=K'_1\cup K'_2$ heißt **Resolvente** von K_1 und K_2.

Grafische Notation (von oben nach unten gerichtet zu lesen):

K_1 K_2

$Res(K_1,K_2)$

Notation Russell/Norvig

$\boxed{P\vee Q}$ $\boxed{R\vee\neg P}$

$\boxed{R\vee Q}$

Beispiele

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

$\{R,Q\}$

$\{P\}$ $\{\neg P\}$

$\square$

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

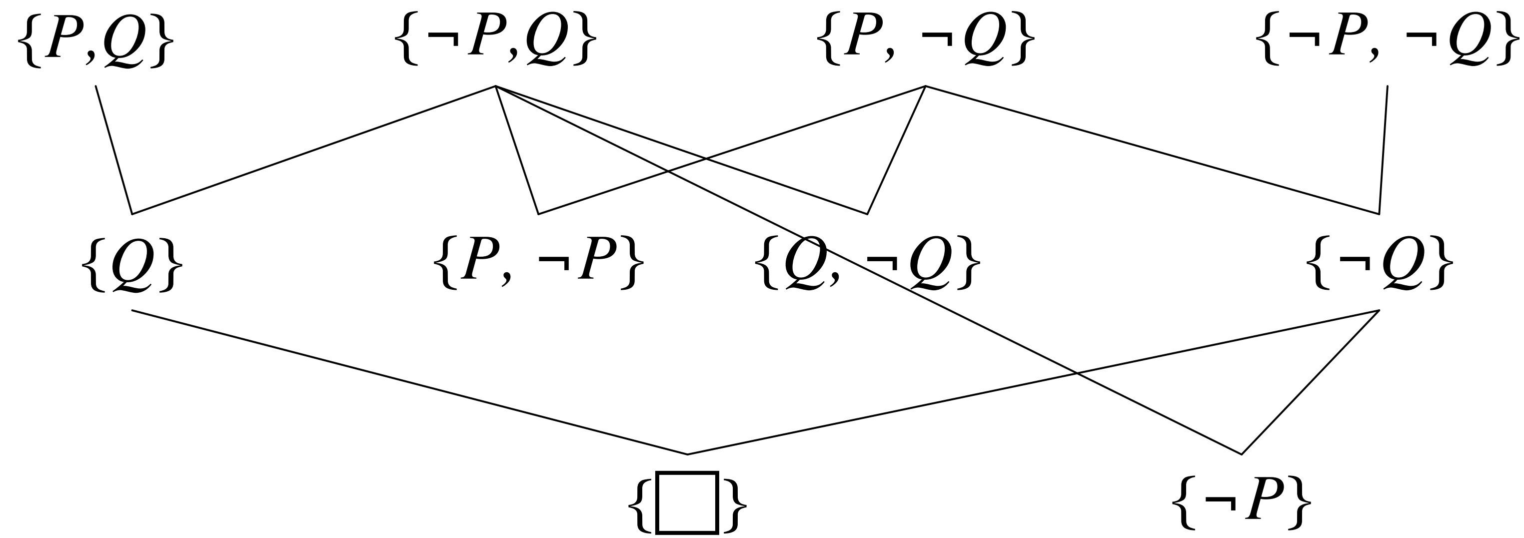
$\{R,Q\}$

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

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

Beispiel für Resolventenbildung

Klauselmenge



u.a.

u.a.

Behauptung: Alle Resolventen folgen aus ihren Elternklauseln!
Wenn das so ist, dann ist die Klauselmenge inkonsistent (wegen □)

Korrektheit der aussagenlogischen Resolution

Die Resolvente $Res(K_1,K_2)$ zweier Klauseln $K_1=\{P\}\cup K'_1$ und $K_2=\{\neg P\}\cup K'_2$ ist eine logische Folgerung aus K_1 und K_2:

$$\{K_1,K_2\} \models Res(K_1,K_2)$$

Beweis

Zeige: Jede Interpretation I, die K_1, K_2 wahr macht, macht auch $Res(K_1,K_2)$ wahr.

I macht entweder P oder $\neg P$ falsch, nimm an es sei P (dann analog für $\neg P$).

Folglich ist K_1 keine Einsklausel, folglich ist K'_1 wahr unter I.

Folglich ist $K'_1\cup K'_2 = Res(K_1,K_2)$ wahr unter I. ✓

Resolution nutzt den Satz vom Widerspruchsbeweis:
Statt $KB \models \alpha$ zeigt man, dass $\mathcal{F}=KB\wedge\neg\alpha$ inkonsistent ist.
Das Zeichen dafür ist: Eine Resolvente ist $\square$.

PL-Resolution

```
function PL-RESOLUTION(KB, α) returns true or false
   clauses ← the set of clauses in the CNF representation of KB ∧ ¬α
   new ← { }
   loop do
      for each C_i, C_j in clauses do
         resolvents ← PL-RESOLVE(C_i, C_j)
         if resolvents contains the empty clause then return true
         new ← new ∪ resolvents
      if new ⊆ clauses then return false
      clauses ← clauses ∪ new
```

… bildet bis $\square$ die **vollständige inferenzielle Hülle** von $\mathcal{F}=KB\wedge\neg\alpha$

Beispiel

Protokolliere nach Schleifendurchlauf für Klauselmenge $\mathcal{F}$:

$\text{Res}^i(\mathcal{F})$: Klauseln im i-ten Durchlauf

$\text{Res}^0(\mathcal{F}) = \mathcal{F}$

(1) $\{P,Q\}$

(2) $\{\neg P,Q\}$

(3) $\{P, \neg Q\}$

(4) $\{\neg P, \neg Q\}$

$\text{Res}^1(\mathcal{F})$

(5)	$\{Q\}$	(1,2)
(6)	$\{P\}$	(1,3)
(7)	$\{Q, \neg Q\}$	(1,4)
(8)	$\{P, \neg P\}$	(1,4)
(9)	$\{Q, \neg Q\}$	(2,3)
(10)	$\{P, \neg P\}$	(2,3)
(11)	$\{\neg P\}$	(2,4)
(12)	$\{\neg Q\}$	(3,4)

$\text{Res}^2(\mathcal{F})$

(13)-(20) = (5)-(12)

(21)	$\{P, Q\}$	(1,7)
(22)	$\{P, Q\}$	(1,8)
(23)	…	
(24)	…	
…		
(47)	$\square$	(5,12)

Vollständigkeit der aussagenlogischen Resolution

Sei $\mathcal{F}$ inkonsistente Formelmenge und $\mathrm{Res}^*(\mathcal{F}) = \bigcup_{i=0}^{\infty} \mathrm{Res}^i(\mathcal{F})$

Dann ist $\square \in \mathrm{Res}^*(\mathcal{F})$

Zusammen mit der Korrektheit ergibt sich:

Resolutionssatz der Aussagenlogik

Eine Klauselmenge $\mathcal{F}$ ist inkonsistent, gdw. $\square \in \mathrm{Res}^*(\mathcal{F})$

Beweis des Vollständigkeitssatzes

Zeige $\square \in \mathrm{Res}^*(\mathcal{F})$ für inkonsistentes $\mathcal{F}$ durch Induktion über Variablenzahl n.

Induktionsanfang: $n=0$.
Dann muss gelten $\square \in \mathcal{F}$, also auch $\square \in \mathrm{Res}^*(\mathcal{F})$.

Induktionsschritt: n beliebig fest, zeige Beh. für $(n+1)$ Variable in $\mathcal{F}$.
Für alle $\mathcal{G}$ inkonsistent, nur mit Variablen $P_1,\ldots,P_n$, gilt: $\square \in \mathrm{Res}^*(\mathcal{G})$.
Enthalte $\mathcal{F}$ die Variablen $P_1,\ldots,P_{n+1}$.
Konstruiere $\mathcal{G}_0$, $\mathcal{G}_1$ mit Variablen $P_1,\ldots,P_n$: Für $\mathcal{G}_0$ streiche P_{n+1} in allen Klauseln;
streiche alle Klauseln mit $\neg P_{n+1}$. (Für $\mathcal{G}_1$ analog mit $\neg P_{n+1}$, P_{n+1} vertauscht.)
$\mathcal{G}_0$, $\mathcal{G}_1$ sind beide inkonsistent. Denn Modell $\mathcal{M}$ für $\mathcal{G}_0$ wäre erweiterbar zu
$\mathcal{F}$-Modell $\mathcal{M}' := \mathcal{M} \cup \{P_{n+1} \mapsto 0\}$, im Widerspruch zu Inkonsistenz von $\mathcal{F}$.
(Analog für $\mathcal{G}_1$.)
Nach Induktionsvoraussetzung also $\square \in \mathrm{Res}^*(\mathcal{G}_0)$ und $\square \in \mathrm{Res}^*(\mathcal{G}_1)$.
…

Vollständigkeitsbeweis, Fortsetzung

Folglich gibt es in $\mathcal{G}_0$ eine Folge von Klauseln $K_1, \ldots, K_m$, wobei $K_m = \square$ und

für $i=1, \ldots, m$: $K_i \in \mathcal{G}_0$ oder K_i ist Resolvente von K_a, K_b mit $a,b<i$.

Analog Folge $K'_1, \ldots, K'_k = \square$ von Resolventen in $\mathcal{G}_1$.

Sind alle Klauseln K_i oder alle Klauseln K'_j in $\mathcal{F}$, gilt $\square \in \mathrm{Res}^*(\mathcal{F})$.

Andernfalls entstehen durch Wiedereinfügen der gestrichenen Literale P_{n+1} in die K_i und $\neg P_{n+1}$ in die K'_j Ableitungen der Einsklauseln P_{n+1} und $\neg P_{n+1}$.

Durch einen weiteren Resolutionsschritt leite daraus $\square$ ab, folglich $\square \in \mathrm{Res}^*(\mathcal{F})$.

✓

Spezialisierungen der Resolution

… sind effizienter als PL-RESOLUTION durch Beschränkung der Auswahlmöglichkeiten für die Elternklauseln C_i, C_j

Alle Spezialisierungen „erben“ Korrektheit!

Im Folgenden kurz beschrieben:

- Stützmengen-Resolution (*set of support*)
- Einsklausel/Unit-Resolution
- Input-Resolution
- SLD-Resolution

Stützmenge (*set of support*)

Eine Teilmenge $\mathcal{T}$ einer Klauselmenge $\mathcal{F}$ heißt **Stützmenge**, wenn $\mathcal{F}-\mathcal{T}$ erfüllbar ist.
Eine Resolvente $Res(K_1,K_2)$ ist eine **Stützmengenresolvente**, wenn für höchstens eines der K_i gilt, dass $K_i \in \mathcal{F}-\mathcal{T}$.

Vollständigkeit der Stützmengenresolution

Sei $\mathcal{F}$ eine inkonsistente Klauselmenge und $\mathcal{T} \subseteq \mathcal{F}$ eine Stützmenge. Dann gibt es eine Ableitung von $\square$ ausschließlich mit Stützmengenresolventen.

Beweis: Über Umsortierung einer beliebigen Resolutions-Ableitung von $\square$.

In der Praxis:
$\mathcal{F}-\mathcal{T}$ ist die Axiomenmenge/Datenbank; $\mathcal{T}$ die Negation der Behauptung/Anfrage

Beispiel für Stützmengen-Resolution

Aufgabe: Zeige, dass

$$\{\{\neg P,\neg Q,\neg R\}, \{R,\neg S,T\}, \{\neg P,S\}, \{Q\}, \{P\}\} \models T$$

Stützmengen-Resolution:

$\mathcal{F}=\{\{\neg P,\neg Q,\neg R\}, \{R,\neg S,T\}, \{\neg P,S\}, \{Q\}, \{P\}, \{\neg T\}\}$

$\mathcal{T}= \{\{\neg T\}\}$

$\mathcal{F}-\mathcal{T}=\{\{\neg P,\neg Q,\neg R\}, \{R,\neg S,T\}, \{\neg P,S\}, \{Q\}, \{P\}\}$

$\mathbf{Res}_{\mathbf{sos}}{}^{1}(\mathcal{F})$: $\{\{R,\neg S\}\}$ (aus $\{R,\neg S,T\}$ und $\{\neg T\}$; keine weiteren erlaubt!)

$\mathbf{Res}_{\mathbf{sos}}{}^{2}(\mathcal{F})$: $\{\{\neg P,\neg Q,\neg S\}, \{\neg P,R\}\}$ (keine weiteren erlaubt!)

$\mathbf{Res}_{\mathbf{sos}}{}^{3}(\mathcal{F})$: $\{\{\neg P,\neg Q\}, \{\neg P,\neg S\}, \{\neg Q,\neg S\}, \{R\}\}$ (keine weiteren e.!)

$\mathbf{Res}_{\mathbf{sos}}{}^{4}(\mathcal{F})$: $\{\{\neg P\}, \{\neg Q\}, \{\neg S\}, \{\neg P,\neg Q\}\}$ (keine weiteren erlaubt!)

$\mathbf{Res}_{\mathbf{sos}}{}^{5}(\mathcal{F})$: $\{\square, \ldots\}$

Unit- und Input-Resolution

Eine Resolvente $Res(K_1,K_2)$ ist eine **Unit-Resolvente**, wenn für mindestens eines der K_i eine Einsklausel ist.

Eine Resolvente $Res(K_1,K_2)$ ist eine **Input-Resolvente**, wenn für mindestens eines der K_i eine Klausel aus der Eingabe-Klauselmenge $\mathcal{F}$ ist.

Unit- und Input-Resolution sind beide nicht vollständig!

Beispiel: $\{\{P,Q\},\{\neg P,Q\},\{P,\neg Q\},\{\neg P,\neg Q\}\}$ inkons. aber so nicht widerlegbar

Äquivalenz von Input- und Unit-Resolution

Sei $\mathcal{F}$ eine inkonsistente Klauselmenge.

$\mathcal{F}$ ist unit-widerlegbar, gdw. $\mathcal{F}$ ist input-widerlegbar.

Beweis: Induktion über die Zahl der Variablen in $\mathcal{F}$.

Hornklauseln und Definite Klauseln

Eine **Hornklausel** ist eine Klausel, die höchstens 1 positives Literal enthält.

Bemerkung: Wegen $(P \wedge Q \Rightarrow R) \equiv (\neg P \vee \neg Q \vee R)$
treten Hornklauseln auf natürliche Weise in der Wissensrepräsentation auf!
Weitere Anwendung: Logikprogrammierung (PROLOG).

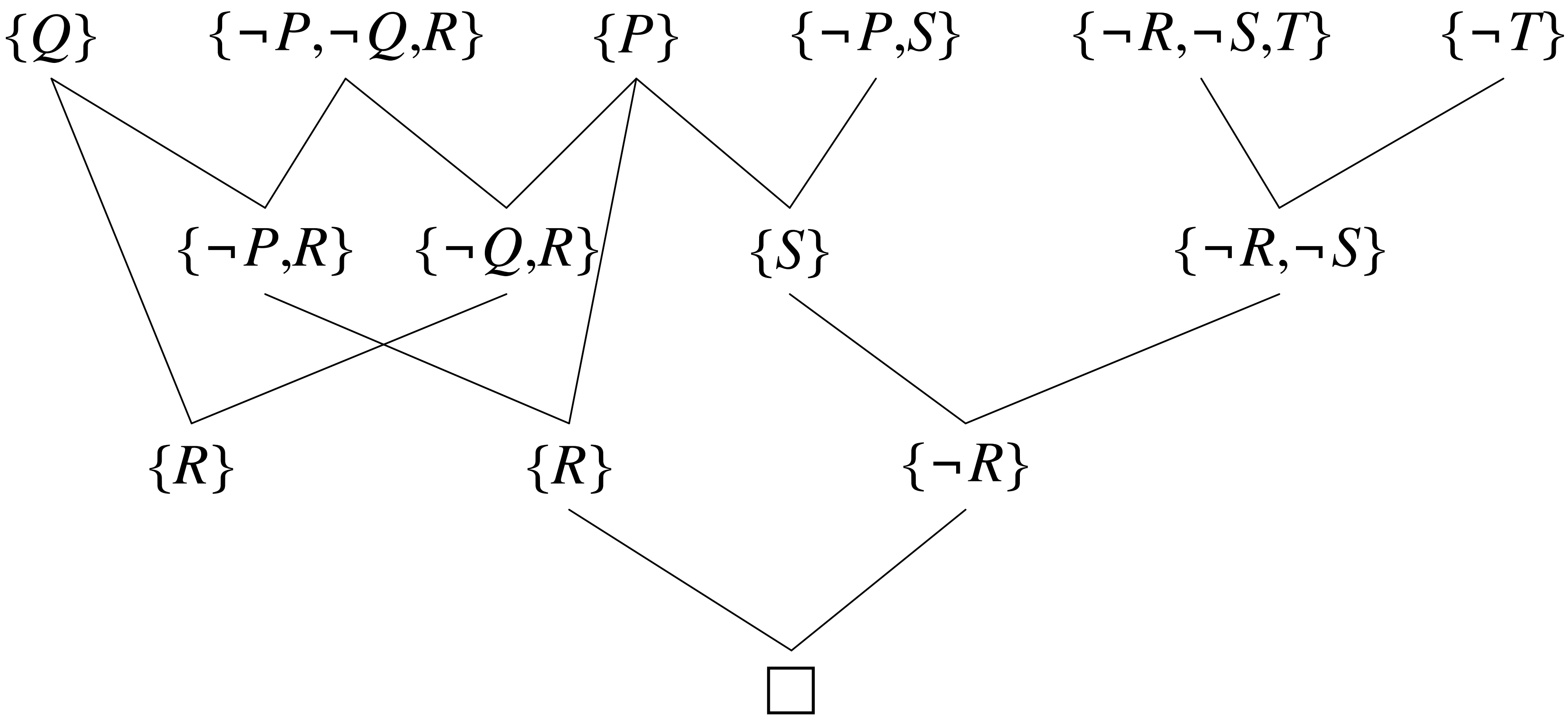
Vollständigkeit von Input- und Unit-Resolution für Hornklauseln

Sei $\mathcal{F}$ eine inkonsistente Hornklauselmenge.
Dann ist $\mathcal{F}$ unit- und input-widerlegbar.

Beweis: Für unit-Resolution direkt mit Induktion über die Variablenzahl in $\mathcal{F}$.
Für input-Resolution dann über den Äquivalenzsatz von input- und unit-Res.

Eine **definite Klausel** ist eine (Horn-)Klausel, die *genau* 1 positives Literal enthält.

Beispiel unit-Resolution



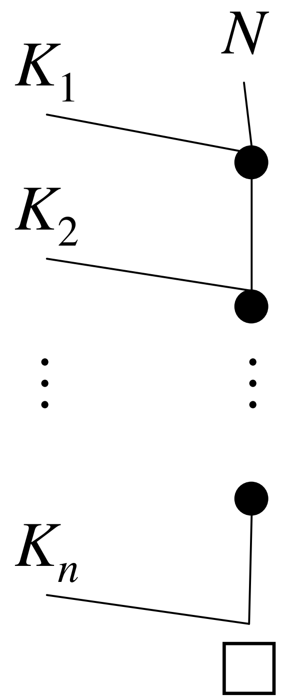
nicht alle möglichen unit-Resolutionen dargestellt!

SLD-Resolution

Eine Widerlegung der Länge n einer Hornklauselmenge $\mathcal{F}$ liegt in SLD-Resolution vor, wenn gilt:

- Die erste Resolvente $Res(K_1,N)$ hat die negative Klausel N und die definite Klausel K_1 als Elternklauseln;
- die i-te Resolvente ist $Res(K_i,Res(K_{i-1},(\ldots(Res(K_1,N))\ldots)))$
- die n-te Resolvente ist $\square$, keine frühere Resolvente ist $\square$.

SLD-Struktur



SLD steht für: *linear resolution with selection function for definite clauses*

Beispiel SLD-Resolution

Klauselmenge:

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