

2.2 Resolution in der Aussagenlogik

Ein **Kalkül** K 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 (aus.-log.) Kalküls, vgl. Folie 26:

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

Vollständigkeit kann z.B. durch Verwendung eines vollständigen Suchverfahrens erzielt werden, vgl. Kapitel 3!

Der Resolutionskalkül

Resolution beruht auf Anwendung des Modus ponens ...

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

... in der umgeformten, verallgemeinerten Variante

$$\frac{\neg P \vee A \quad | \quad P \vee B}{A \vee B}$$

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



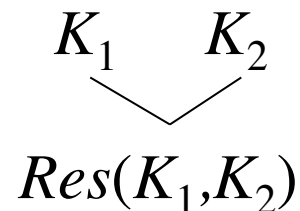
J. Alan Robinson
*1930

Die Resolutionsregel für KNF in Mengenform

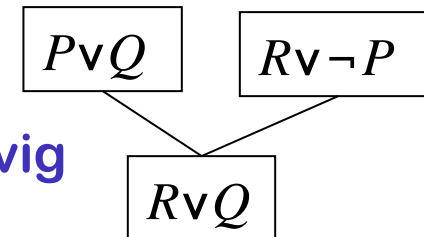
Gegeben zwei Klauseln $K_1 = \{P\} \cup K'_1$ und $K_2 = \{\neg P\} \cup K'_2$, wobei P eine beliebige Aussagevariable ist ($P, \neg P$ sind also Literale).

Die Klausel $Res(K_1, K_2) := K'_1 \cup K'_2$ heißt **Resolvente** von K_1 und K_2 .

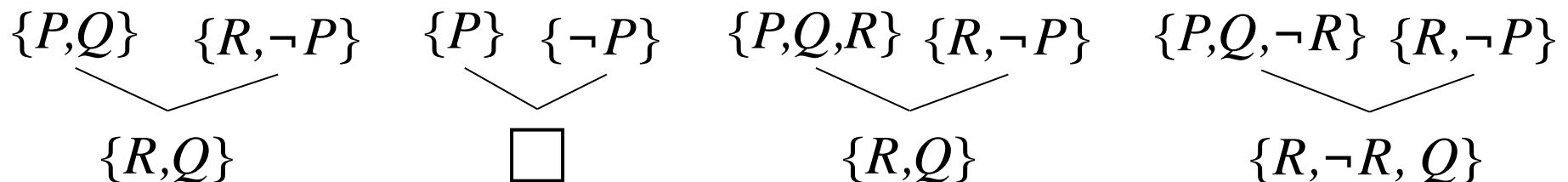
Stammbaum-Notation (von oben nach unten gerichtet zu lesen):



Notation
Russell/Norvig

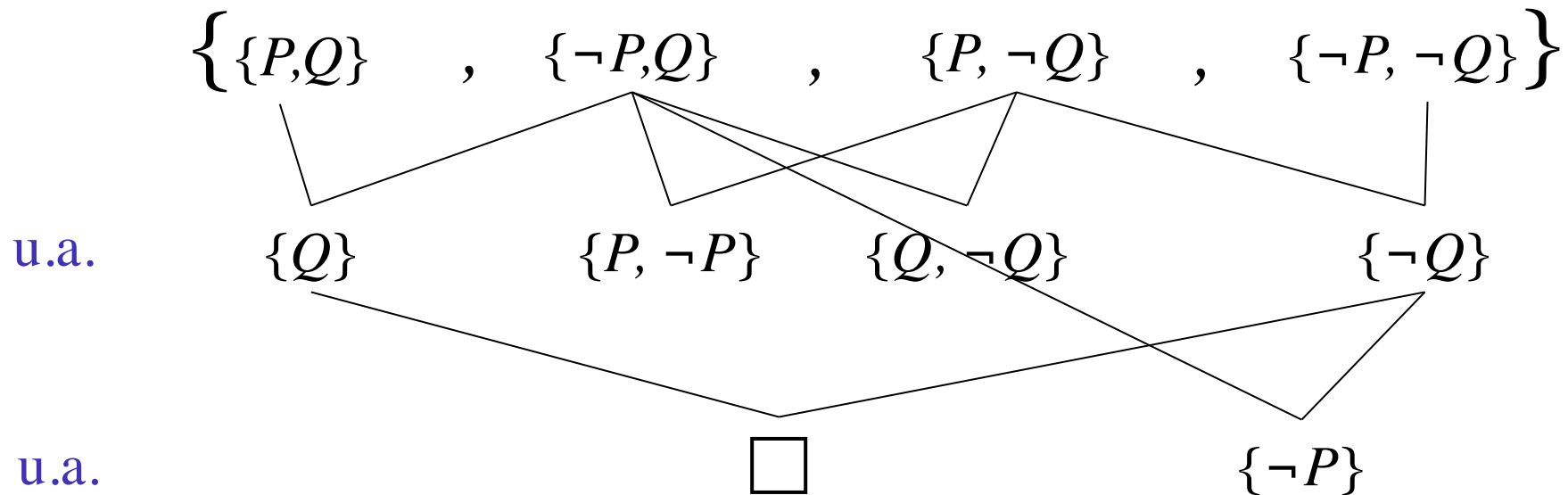


Beispiele (für KNF in Mengenform)



Beispiel für Resolventenbildung

Klauselmenge



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

Korrektheit d. 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 und 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 . 

Anwendung von Resolution immer als Widerspruchsbeweis:

Statt $KB \models \alpha$ zeige, dass $\mathcal{F} = KB \wedge \neg \alpha$ inkonsistent ist!

Das Zeichen dafür ist: Eine Resolvente irgendwann ist \square .

Algorithmus für Resolution in der AL

in R/N: englisch *propositional logic* !

```
function PL-RESOLUTION( $KB, \alpha$ ) returns incons or consistent  
   $clauses \leftarrow$  the set of clauses in the CNF representation of  $KB \wedge \neg \alpha$   
   $new \leftarrow \{ \}$   
  loop do  
    for each  $C_i, C_j$  in  $clauses$  do  
       $resolvents \leftarrow$  PL-RESOLVE( $C_i, C_j$ )  
      if  $resolvents$  contains the empty clause then return incons  
       $new \leftarrow new \cup resolvents$   
    if  $new \subseteq clauses$  then return consistent  
     $clauses \leftarrow clauses \cup new$ 
```

... bildet, ggf. bis zum Auftauchen von \square ,
die **vollständige inferenzielle Hülle** von $\mathcal{F} = KB \wedge \neg \alpha$

Beispiel

Sei \mathcal{F} die KNF von $KB \wedge \neg \alpha$.

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)

Eigenschaften von PL-RESOLUTION

PL-Resolution macht praktisch Breitensuche durch alle Klauseln, die aus der gegebenen Klauselmenge erzeugbar sind.

Also:

- ☹ Speicherbedarf: $O(2^n)$ (n Var., Tautologien+Doubletten löschen)
- ☹ Zeitbedarf: $O(2^n)$
- 😊 korrekt
- 😊 vollständig, s. folgender Satz: ...

Vollständigkeit der aussagenlog. Resolution

Sei \mathcal{F} inkonsistente Formelmenge und $\text{Res}^*(\mathcal{F}) = \bigcup_{i=0}^{\infty} \text{Res}^i(\mathcal{F})$
Dann ist $\Box \in \text{Res}^*(\mathcal{F})$

Zusammen mit der Korrektheit ergibt sich:

Resolutionssatz der Aussagenlogik

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

Beweis des Vollständigkeitssatzes

Zeige $\Box \in \text{Res}^*(\mathcal{F})$ für inkonsistentes \mathcal{F} durch Induktion ü. Variablenzahl n .

Induktionsanfang: $n=0$.

Dann muss gelten $\Box \in \mathcal{F}$, also auch $\Box \in \text{Res}^*(\mathcal{F})$.

Induktionsschritt: n beliebig fest, zeige Beh. für $(n+1)$ Variable in \mathcal{F} .

Für alle G inkonsistent, nur mit Variablen P_1, \dots, P_n gilt: $\Box \in \text{Res}^*(G)$.

Enthalte \mathcal{F} die Variablen P_1, \dots, P_{n+1} .

Konstruiere aus \mathcal{F} die Klauselmengen G_0, G_1 mit Variablen P_1, \dots, P_n :

Für G_0 streiche P_{n+1} in allen Klauseln; streiche alle Klauseln mit $\neg P_{n+1}$. (Für G_1 analog mit $\neg P_{n+1}, P_{n+1}$ vertauscht.)

G_0, G_1 sind beide inkonsistent. Denn Modell \mathcal{M} für 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 G_1 .)

Nach Induktionsvoraussetzung also $\Box \in \text{Res}^*(G_0)$ und $\Box \in \text{Res}^*(G_1)$.

...

Vollständigkeitsbeweis, Fortsetzung

Folglich gibt es in \mathcal{G}_0 eine Folge von Klauseln K_1, \dots, K_m , wobei $K_m = \square$ und für $i=1, \dots, m$: $K_i \in \mathcal{G}_0$ oder K_i ist Resolvente von K_a, K_b mit $a, b < i$.
Analog Folge $K'_1, \dots, K'_k = \square$ von Resolventen in \mathcal{G}_1 .

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

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 Einklauseln P_{n+1} und $\neg P_{n+1}$.

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



Spezialisierungen der Resolution

... sind effizienter als PL-RESOLUTION durch **Beschränkung der Auswahlmöglichkeiten** für die Elternklauseln K_i, K_j

Alle Spezialisierungen „erben“ Korrektheit!

Zum Beispiel:

- Stützmenngen-Resolution (nicht in dieser Vorlesung)
- Einklausel/Unit-Resolution
- Input-Resolution
- SLD-Resolution

Unit- und Input-Resolution

Eine Resolvente $Res(K_1, K_2)$ ist eine **Unit-Resolvente**, wenn mindestens eines der K_i eine Einklausel (Klausel aus 1 Literal) ist.

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

Unit- und Input-Resolution sind beide nicht vollständig!

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

Es gibt **vollständige** Einschränkungen der Resolution \rightarrow **hier ausgelassen!**

Äquivalenz von Input- und Unit-Resolution

Sei \mathcal{F} eine inkonsistente Klauselmengen.

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

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