

Formale Grundlagen der Informatik 3  
Kapitel 2  
Aussagenlogik  
*Resolution*

Frank Heitmann  
heitmann@informatik.uni-hamburg.de

19. Oktober 2015

# Motivation

Wir haben letztes Mal Syntax und Semantik der Aussagenlogik eingeführt. Zentral ist nun die Frage

Wie löst man das Erfüllbarkeitsproblem?

## Wichtige Anmerkung

Etliche Fragen lassen sich auf das (Un-)Erfüllbarkeitsproblem zurückführen.

# Motivation

Wir wollen nun die **Resolution** betrachten, ein **spezielles Ableitungsverfahren**, das mit KNFs arbeitet.

## Anmerkung

Die Resolution ist ein **Widerlegungsverfahren**, d.h. es ist möglich eine Formel auf Unerfüllbarkeit zu testen.

# Motivation

Ableitungsverfahren arbeiten **rein syntaktisch**.

Bei der Resolution

- wird aus zwei Formeln
  - sofern sie die Voraussetzungen (gewisse syntaktische Eigenschaften) erfüllen
- eine neue Formel generiert

## Sneak Peek

$$\begin{array}{ccc} \neg B \vee C \vee A & & A \vee B \vee D \\ \downarrow & \swarrow & \\ C \vee A \vee A \vee D & & \end{array}$$

# Mengendarstellung

Resolution arbeitet mit Formeln in KNF und benutzt eine **Mengendarstellung**

- Ist  $K = \bigvee_{i=1}^m L_i$  eine Klausel, dann ist  $\mathbf{K} = \{L_1, \dots, L_m\}$  die Mengendarstellung von  $K$ .
- Ist  $F = (\bigwedge_{i=1}^n (\bigvee_{j=1}^{m_i} L_{i,j}))$  eine Formel in KNF, dann ist  $\mathbf{F} = \{\{L_{1,1}, \dots, L_{1,m_1}\}, \dots, \{L_{n,1}, \dots, L_{n,m_n}\}\}$  die Mengendarstellung von  $F$ .
- Die leere Klausel (Mengendarstellung zu  $\perp$ ) wird mit  $\square$  bezeichnet.
- Die Wahrheitswerteberechnung wird angepasst:
  - $\mathcal{A}(\mathbf{K}) = \max(\{\mathcal{A}(L) \mid L \in \mathbf{K}\})$
  - $\mathcal{A}(\mathbf{F}) = \min(\{\mathcal{A}(\mathbf{K}) \mid \mathbf{K} \in \mathbf{F}\})$
  - $\mathcal{A}(\square) = 0$

# Mengendarstellung - Beispiel

$$F = (\neg A \vee B) \wedge E \wedge (\neg G \vee \neg C)$$

Dann ist

- $K_1 = \neg A \vee B$  und  $\mathbf{K}_1 = \{\neg A, B\}$
- $K_2 = E$  und  $\mathbf{K}_2 = \{E\}$
- $K_3 = \neg G \vee \neg C$  und  $\mathbf{K}_3 = \{\neg G, \neg C\}$
- $\mathbf{F} = \{\{\neg A, B\}, \{E\}, \{\neg G, \neg C\}\}$

## Anmerkung

In der Mengendarstellung sind die Junktoren nicht mehr explizit zu sehen.

# Die Resolutionsregel

## Definition (Resolvente)

Seien  $K_1$  und  $K_2$  Klauseln (in Mengendarstellung) und  $L$  ein Literal mit  $L \in K_1$  und  $\bar{L} \in K_2$ . Dann heißt die (Literal-)Menge  $R = (K_1 - \{L\}) \cup (K_2 - \{\bar{L}\})$  **Resolvente** von  $K_1$  und  $K_2$  (bzgl.  $L$ ).

Dabei ist  $\bar{L} = A$ , falls  $L = \neg A$  und  $\bar{L} = \neg A$ , falls  $L = A$ .

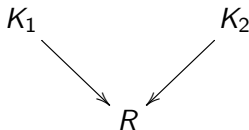
Im Falle  $K_1 = \{L\}$  und  $K_2 = \{\neg L\}$  ist  $R = \emptyset$  und wird durch  $\square$  symbolisiert.

# Die Resolutionsregel

## Definition (Resolvente)

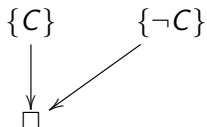
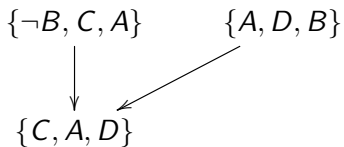
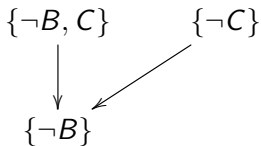
Seien  $K_1$  und  $K_2$  Klauseln (in Mengendarstellung) und  $L$  ein Literal mit  $L \in K_1$  und  $\bar{L} \in K_2$ . Dann heißt die (Literal-)Menge  $R = (K_1 - \{L\}) \cup (K_2 - \{\bar{L}\})$  **Resolvente** von  $K_1$  und  $K_2$  (bzgl.  $L$ ).

- Resolventenbildung als Ableitung:  $\{K_1, K_2\} \vdash_{res} R$
- Darstellung als Diagramm:





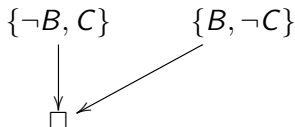
# Beispiele



# Beispiele

## Wichtige Anmerkung

Das unten stehende auf keinen Fall!



## Wichtige Anmerkung

Wirklich nicht! Nein, nein, nein! Es wird immer nur genau ein Literal aus der einen Mengen und ein (komplementäres) Literal aus der anderen Menge entfernt!

# Beispiele

## Wichtige Anmerkung

Das unten stehende auch nicht!

$$\{\neg B, B\}$$



## Wichtige Anmerkung

Zur Resolventenbildung werden zwei Klauseln benötigt!

# Resolventenmengen

Wir führen mehrschrittige Resolventenbildung ein:

## Definition

Sei  $F$  eine Formel in KNF dargestellt als Klauselmenge.

$$Res(F) := F \cup \{R \mid R \text{ ist Resolvente zweier Klauseln aus } F\}$$

$$Res^0(F) := F$$

$$Res^{n+1}(F) := Res(Res^n(F))$$

$$Res^*(F) := \bigcup_{i \geq 0} Res^i(F)$$

# Resolutionssatz (Vorschau)

Gleich zeigen wir:

## Satz (Resolutionssatz)

*Eine Klauselmeng*e  $F$  *ist unerfüllbar genau dann, wenn*  
 $\square \in \text{Res}^*(F)$  *gilt.*

Da dieser Satz gilt, werden die ganzen Dinge oben überhaupt nur eingeführt!

# Beispiel

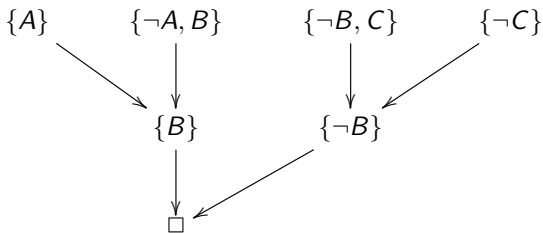
## Beispiel

Wir wollen die folgende Formel auf (Un-)Erfüllbarkeit testen:

$$(\neg A \vee B) \wedge (\neg B \vee C) \wedge A \wedge \neg C$$

Als Klauselmenge:

$$\{\{\neg A, B\}, \{\neg B, C\}, \{A\}, \{\neg C\}\}$$



# Das weitere

Jetzt:

- zeigen wir den Resolutionssatz und
- leiten daraus einen Algorithmus zum Test auf Unerfüllbarkeit ab.

# Mengendarstellung - Wiederholung

Resolution arbeitet mit Formeln in KNF und benutzt eine **Mengendarstellung**

- Ist  $K = \bigvee_{i=1}^m L_i$  eine Klausel, dann ist  $\mathbf{K} = \{L_1, \dots, L_m\}$  die Mengendarstellung von  $K$ .
- Ist  $F = (\bigwedge_{i=1}^n (\bigvee_{j=1}^{m_i} L_{i,j}))$  eine Formel in KNF, dann ist  $\mathbf{F} = \{\{L_{1,1}, \dots, L_{1,m_1}\}, \dots, \{L_{n,1}, \dots, L_{n,m_n}\}\}$  die Mengendarstellung von  $F$ .
- Die leere Klausel (Mengendarstellung zu  $\perp$ ) wird mit  $\square$  bezeichnet.
- Die Wahrheitswerteberechnung wird angepasst:
  - $\mathcal{A}(\mathbf{K}) = \max(\{\mathcal{A}(L) \mid L \in \mathbf{K}\})$
  - $\mathcal{A}(\mathbf{F}) = \min(\{\mathcal{A}(\mathbf{K}) \mid \mathbf{K} \in \mathbf{F}\})$
  - $\mathcal{A}(\square) = 0$



# Ablauf

## Lemma (Resolutionslemma)

*Sei  $F$  eine Formeln in KNF als Klauselmengenmenge und  $R$  eine Resolvente zweier Klauseln  $K_1$  und  $K_2$  in  $F$ . Dann sind  $F$  und  $F \cup \{R\}$  äquivalent.*

## Lemma

*Sei  $F$  eine Formel in KNF als Klauselmengenmenge. Dann gilt*

$$F \equiv \text{Res}^1(F) \equiv \text{Res}^2(F) \equiv \dots \equiv \text{Res}^n(F) \equiv \dots \equiv \text{Res}^*(F)$$

## Satz (Resolutionssatz)

*Eine Klauselmengenmenge  $F$  ist unerfüllbar genau dann, wenn  $\square \in \text{Res}^*(F)$  gilt.*

# Resolutionslemma

## Satz (Resolutionslemma)

Sei  $F$  eine Formeln in KNF und  $R$  eine Resolvente zweier Klauseln  $K_1$  und  $K_2$  in  $F$ . Dann sind  $F$  und  $F \cup \{R\}$  äquivalent.

## Beweis.

Sei  $\mathcal{A}$  eine Belegung. Gilt  $\mathcal{A} \models F \cup \{R\}$ , so gilt sofort auch  $\mathcal{A} \models F$ . (Warum?) Gelte umgekehrt  $\mathcal{A} \models F$ , d.h. es ist  $\mathcal{A} \models K$  für alle Klauseln  $K \in F$ . Sei  $R = (K_1 - \{L\}) \cup (K_2 - \{\bar{L}\})$  mit  $K_1, K_2 \in F$  und  $L \in K_1$ ,  $\bar{L} \in K_2$ . Nun gibt es zwei Fälle:

- 1  $\mathcal{A} \models L$ . Dann folgt  $\mathcal{A} \models K_2 - \{\bar{L}\}$  aus  $\mathcal{A} \models K_2$  und  $\mathcal{A} \not\models \bar{L}$ .
- 2  $\mathcal{A} \not\models L$ . Dann folgt  $\mathcal{A} \models K_1 - \{L\}$  aus  $\mathcal{A} \models K_1$ .

In beiden Fällen folgt  $\mathcal{A} \models R$  und damit insgesamt  $\mathcal{A} \models F \cup \{R\}$ .  $\square$

# Ein Lemma/Korollar

## Lemma

*Sei  $F$  eine Formel in KNF als Klauselmengung. Dann gilt*

$$F \equiv \text{Res}^1(F) \equiv \text{Res}^2(F) \equiv \dots \equiv \text{Res}^n(F) \equiv \dots \equiv \text{Res}^*(F)$$

## Beweis.

Folgt sofort aus dem Resolutionslemma von eben ...



# Der Resolutionssatz

## Satz (Resolutionssatz)

*Eine Klauselmengemenge  $F$  ist unerfüllbar genau dann, wenn  $\square \in \text{Res}^*(F)$  gilt.*

## Beweis

Angenommen  $\square \in \text{Res}^*(F)$ . Wir wollen zeigen, dass  $F$  unerfüllbar ist. Zunächst kann  $\square$  nur durch die Resolution zweier Klauseln  $K_1 = \{L\}$  und  $K_2 = \{\bar{L}\}$  entstanden sein. Wegen der Definition von  $\text{Res}^*(F)$  muss es wegen  $\square \in \text{Res}^*(F)$  ferner ein  $n$  geben mit  $\square \in \text{Res}^n(F)$  und damit auch  $K_1, K_2 \in \text{Res}^n(F)$ . Da keine Belegung sowohl  $K_1$  als auch  $K_2$  erfüllen kann, ist  $\text{Res}^n(F)$  unerfüllbar und damit wegen  $F \equiv \text{Res}^n(F)$  aus dem vorherigen Lemma auch  $F$ .

Dies zeigt die *Korrektheit* der Resolution.

# Fortsetzung des Beweises

## Beweis

Sei nun  $F$  als unerfüllbar angenommen. Wir wollen  $\square \in \text{Res}^*(F)$  zeigen. Dies gelingt mittels Induktion über die Anzahl  $n$  der in  $F$  vorkommenden atomaren Formeln.

**Induktionsanfang.** ist  $n = 0$ , so ist  $F = \{\square\}$  und wegen  $F \subseteq \text{Res}^*(F)$  ist  $\square \in F \subseteq \text{Res}^*(F)$ .

**Induktionsannahme.** Angenommen für jede Klauselmenge  $G$ , die nur  $n$  atomare Formeln enthält und die unerfüllbar ist, gilt  $\square \in \text{Res}^*(G)$ .

**Induktionsschritt.** Sei  $F$  eine unerfüllbare Formelmengung, die die atomaren Formeln  $A_1, \dots, A_{n+1}$  enthält. Zunächst konstruieren wir zwei Formeln  $F_0$  und  $F_1$  wie folgt:

- Idee: Für  $F_0$  belegen wir  $A_{n+1}$  mit 0 und vereinfachen.  $F_0$  entsteht aus  $F$ , indem wir jedes Vorkommen von  $A_{n+1}$  in einer Klausel streichen und bei Vorkommen von  $\neg A_{n+1}$  die ganze Klausel streichen.
- Idee: Für  $F_1$  belegen wir  $A_{n+1}$  mit 1 und vereinfachen. ...

# Fortsetzung des Beweises

## Beweis

$F_1$  entsteht aus  $F$ , indem wir jedes Vorkommen von  $\neg A_{n+1}$  in einer Klausel streichen und bei Vorkommen von  $A_{n+1}$  die ganze Klausel streichen.

Auf  $F_0$  und  $F_1$  ist nun die Induktionsannahme anwendbar **wenn** wir denn zeigen können, dass sie unerfüllbar sind!

Angenommen  $F_0$  hat eine Belegung  $\mathcal{A}$ , die  $F_0$  erfüllt. Dann ist aber  $\mathcal{A}'$  mit  $\mathcal{A}'(B) = \mathcal{A}(B)$  für die  $B \in \{A_1, \dots, A_n\}$  und  $\mathcal{A}'(A_{n+1}) = 0$  eine erfüllende Belegung für  $F$  im Widerspruch zur Unerfüllbarkeit von  $F$ . (Analog für  $F_1$ .)

Es gilt also  $\square \in \text{Res}^*(F_0)$  und  $\square \in \text{Res}^*(F_1)$  nach Induktionsannahme.

# Fortsetzung des Beweises

## Beweis

D.h., es gibt für  $F_0$  Klauseln  $K_1, \dots, K_m$  mit

- $K_m = \square$  und für  $i = 1, \dots, m$  gilt
- $K_i \in F_0$  oder  $K_i$  ist Resolvente von  $K_a, K_b$ ,  $a, b < i$

Ebenso gibt es für  $F_1$  Klauseln  $K'_1, \dots, K'_t$ .

Wir stellen nun durch Wiedereinführung von  $A_{n+1}$  in die Klauseln  $K_1, \dots, K_m$  dort, wo es gestrichen wurde, die ursprünglichen Klauseln wieder her und erhalten so eine Folge von Klauseln, die  $\{A_{n+1}\} \in \text{Res}^*(F)$  bezeugt.

Zwei Anmerkungen:

- 1 Gilt hier bereits  $\square \in \text{Res}^*(F)$ , so sind wir bereits fertig! Dies tritt auf, wenn in keiner der Klauseln  $K_1, \dots, K_m$  ein  $A_{n+1}$  auftritt.
- 2 Zur Herstellung von  $F_0$  wurden die Klauseln mit  $\neg A_{n+1}$  gestrichen, diese treten also bei den  $K_1, \dots, K_m$  nicht auf!

# Fortsetzung des Beweises

## Beweis.

Ebenso erhält man aus den  $K'_1, \dots, K'_t$  durch Wiedereinführen von  $\neg A_{n+1}$  dort, wo es gelöscht wurde eine Folge, die  $\{\neg A_{n+1}\} \in \text{Res}^*(F)$  bezeugt (Mit analogen Anmerkungen wie eben.)

Insg. haben wir damit  $\{A_{n+1}\} \in \text{Res}^*(F)$  und  $\{\neg A_{n+1}\} \in \text{Res}^*(F)$ . Die Resolvente dieser beiden Klauseln bezeugt dann  $\square \in \text{Res}^*(F)$ .

Dies zeigt die *Vollständigkeit* der Resolution und schließt den Beweis des Resolutionssatzes ab. □



# Anmerkung

## Nebenbemerkung

Im Beweis gehen wir von einer endlichen Klauselmenge  $F$  aus. Der Beweis geht auch mit einer unendlichen Klauselmenge benötigt dann aber den Endlichkeitssatz, den wir bisher noch nicht gezeigt haben (siehe *Logik für Informatiker* von Uwe Schöning, Kapitel 1.4).

## Endlichkeitssatz

Eine Menge  $M$  von Formeln ist genau dann erfüllbar, wenn jede endliche Teilmenge von  $M$  erfüllbar ist.

# Resolutionsalgorithmus

Ein Algorithmus zur Resolutionsbildung:

---

## Algorithmus 1 Resolutionsalgorithmus

---

- 1: *Input*: Eine Formel  $F$  in KNF
  - 2: Bilde zu  $F$  eine Klauselmenge (auch  $F$  genannt)
  - 3: **do**
  - 4:    $G := F$
  - 5:    $F := Res(F)$
  - 6: **while**  $(\square \in F)$  **or**  $(F = G)$
  - 7: **if**  $\square \in F$  **then**
  - 8:   **return**  $F$  ist unerfüllbar
  - 9: **else**
  - 10:   **return**  $F$  ist erfüllbar
  - 11: **end if**
-

# Termination

Der Resolutionsalgorithmus terminiert aufgrund des folgenden Satzes:

## Satz

Für jede endliche Klauselmeng  $F$  gibt es ein  $k \geq 0$  mit

$$\text{Res}^k(F) = \text{Res}^{k+1}(F) = \dots = \text{Res}^*(F)$$

## Beweis.

Enthalte  $F$  die  $n$  atomaren Formeln  $A_1, \dots, A_n$ . Da jede dieser  $n$  Formeln in einer Klausel positiv vorkommen kann, negativ vorkommen kann, positiv und negativ vorkommen kann oder gar nicht vorkommen kann, gibt es vier Möglichkeiten wie sie in einer Klausel auftreten kann und damit maximal  $4^n$  viele Klauseln, die aus  $A_1, \dots, A_n$  aufgebaut sein können und die in  $\text{Res}^*(F)$  vorkommen können.

Da in einem Schritt von  $\text{Res}^i(F)$  zu  $\text{Res}^{i+1}(F)$  mindestens eine Klausel hinzukommen muss, muss  $k \leq 4^n$  gelten.  $\square$

# Zum Algorithmus

- Man muss nicht alle Resolventen bilden, wenn man  $\square \in Res^*(F)$  zeigen will!
- Der Resolutionsalgorithmus ist in einigen Fällen sehr schnell
- in anderen müssen (exponentiell) viele Klauseln erzeugt werden.
- I.A. ist der Algorithmus also bzgl. der Laufzeit mit dem Aufstellen der Wahrheitstabellen vergleichbar.
- Dies ist auch wenig überraschend: *SAT* ist *NP*-vollständig!

Dennoch gibt es nun Verfahren, deren Ziel es ist, die Resolution zu verbessern. Diese Verfahren sind meist wieder korrekt und vollständig, schränken aber die Möglichkeiten Resolventen zu bilden ein - und werden dadurch meist schneller und übersichtlicher. I.A. benötigen sie aber wie oben exponentielle Laufzeit.

# Verfeinerungen

## Definition (P-Resolution)

Eine Klausel heißt *positiv*, wenn sie nur positive Literale enthält. Bei der **P-Resolution** darf nur dann eine Resolvente aus den Klauseln  $K_1$  und  $K_2$  gebildet werden, wenn eine der beiden Klauseln positiv ist.

## Definition (N-Resolution)

Eine Klausel heißt *negativ*, wenn sie nur negative Literale enthält. Bei der **N-Resolution** darf nur dann eine Resolvente aus den Klauseln  $K_1$  und  $K_2$  gebildet werden, wenn eine der beiden Klauseln negativ ist.

## Satz

*P- und N-Resolution ist vollständig (und weiterhin korrekt).*

# Verfeinerung

## Definition (Einheitsresolution)

Bei der **Einheitsresolution** darf nur dann eine Resolvente aus den Klauseln  $K_1$  und  $K_2$  gebildet werden, wenn mindestens eine der beiden Klauseln aus nur genau einem Literal besteht.

## Satz

*Einheitsresolution ist vollständig für die Klasse der Hornformeln, aber nicht für beliebige KNF-Formeln. (Korrekt ist sie weiterhin.)*

# Übungsaufgabe

## Übungsaufgaben

- 1 Zeigen Sie, dass P-Resolution korrekt und vollständig ist.
- 2 Zeigen Sie, dass Einheitsresolution korrekt ist, aber nicht vollständig.

Korrektheit:

Wenn  $\square \in Res^*(F)$ , dann ist  $F$  unerfüllbar.

Vollständigkeit:

Wenn  $F$  unerfüllbar ist, dann ist  $\square \in Res^*(F)$ .

# Wiederholung

## Zusammenfassung bisher

Wir hatten:

- Syntax der Aussagenlogik inkl.
  - strukturelle Induktion
  - strukturelle Rekursion
- Semantik der Aussagenlogik
  - Belegung, Wahrheitstafel
  - Kategorien (Tautologie etc.)



# Wiederholung

Weitere wichtige (semantische) Begriffe waren

- Folgerbarkeit
- Äquivalenz

(inkl. der zugehörigen Sätze).

Darauf aufbauen dann:

- Herstellung einer KNF/DNF
  - durch Äquivalenzumformungen
  - mittels Wahrheitstafeln

# Wiederholung

Die effiziente Berechnung von (Un-)Erfüllbarkeit und Folgerbarkeiten rückte dann ins Zentrum. Dazu haben wir jetzt zunächst

- Resolution (spezielles Ableitungsverfahren)

kennengelernt.

Demnächst betrachten wir noch

- natürliche Deduktion und
- Tableauxbeweise

## Literaturhinweis

Das heutige Thema Resolution findet man im Buch *Logik für Informatiker* von Uwe Schöning, an dem sich dieser Vorlesungsteil orientierte.