Formale Grundlagen der Informatik 1
Kapitel 16

Resolution

Frank Heitmann
heitmann@informatik.uni-hamburg.de

31. Mai 2016

Motivation

Motivation

- Wir benötigen einen (Un-)Erfüllbarkeitstest für aussagenlogische Formeln.
- Mit dem könn(t)en auch $F \equiv G$ und $F \models G$ entschieden werden!
- Dies geht zwar immer mit Wahrheitstafeln, dauert dann aber lange.
- Hornformeln erlauben einen schnellen Test, schränken die möglichen Formeln aber ein.
- Wir gehen nun anders vor ...

Motivation

Motivation

- Wir benötigen einen (Un-)Erfüllbarkeitstest für aussagenlogische Formeln.
- Mit dem könn(t)en auch $F \equiv G$ und $F \models G$ entschieden werden!
- Dies geht zwar immer mit Wahrheitstafeln, dauert dann aber lange.
- Hornformeln erlauben einen schnellen Test, schränken die möglichen Formeln aber ein.
- Wir gehen nun anders vor ...

Motivation

Motivation

- Wir benötigen einen (Un-)Erfüllbarkeitstest für aussagenlogische Formeln.
- Mit dem könn(t)en auch $F \equiv G$ und $F \models G$ entschieden werden!
- Dies geht zwar immer mit Wahrheitstafeln, dauert dann aber lange.
- Hornformeln erlauben einen schnellen Test, schränken die möglichen Formeln aber ein.
- Wir gehen nun anders vor ...

Motivation

Motivation

- Wir benötigen einen (Un-)Erfüllbarkeitstest für aussagenlogische Formeln.
- Mit dem könn(t)en auch $F \equiv G$ und $F \models G$ entschieden werden!
- Dies geht zwar immer mit Wahrheitstafeln, dauert dann aber lange.
- Hornformeln erlauben einen schnellen Test, schränken die möglichen Formeln aber ein.
- Wir gehen nun anders vor ...

Motivation

Der Sinn von **Ableitungen**:

- Aufgrund **syntaktischer Eigenschaften** von Formeln/Formelmengen auf **semantische Eigenschaften schließen** (statt Wahrheitswerteverläufe zu erstellen und zu prüfen).

Dies ist vor allem auch deswegen wichtig, da für komplexere Logiken (wie die Prädikatenlogik) i.A. die Tabellen nicht mehr komplett aufgebaut werden können!

Motivation

Der Sinn von **Ableitungen**:

- Aufgrund **syntaktischer Eigenschaften** von Formeln/Formelmengen auf **semantische Eigenschaften schließen** (statt Wahrheitswerteverläufe zu erstellen und zu prüfen).

Dies ist vor allem auch deswegen wichtig, da für komplexere Logiken (wie die Prädikatenlogik) i.A. die Tabellen nicht mehr komplett aufgebaut werden können!

Motivation

Der Sinn von **Ableitungen**:

- Aufgrund **syntaktischer Eigenschaften** von Formeln/Formelmengen auf **semantische Eigenschaften schließen** (statt Wahrheitswerteverläufe zu erstellen und zu prüfen).

Dies ist vor allem auch deswegen wichtig, da für komplexere Logiken (wie die Prädikatenlogik) i.A. die Tabellen nicht mehr komplett aufgebaut werden können!

Beispiel

Unabhängig von F ist

- $F \vee \neg F$ eine Tautolgoie
- $F \wedge \neg F$ eine Kontradiktion

Dies kann man allein am **syntaktischen Muster** erkennen!

Ähnlich kann man schlussfolgern (besser: ableiten):

- Wenn F und $F \Rightarrow G$ als wahr angenommen werden
- dann muss auch G wahr sein

(Man kann sich das semantisch klar machen und dann immer wenn das Muster F und $F \Rightarrow G$ vorliegt G **ableiten**.)

Beispiel

Unabhängig von F ist

- $F \vee \neg F$ eine Tautolgoie
- $F \wedge \neg F$ eine Kontradiktion

Dies kann man allein am **syntaktischen Muster** erkennen!

Ähnlich kann man schlussfolgern (besser: ableiten):

- Wenn F und $F \Rightarrow G$ als wahr angenommen werden
- dann muss auch G wahr sein

(Man kann sich das semantisch klar machen und dann immer wenn das Muster F und $F \Rightarrow G$ vorliegt G **ableiten**.)

Inferenzregeln

Konjunktionseinführung (KE):	$\frac{F, G}{F \land G}$	
Konjunktionslöschung (KL):	$\frac{F \land G}{F}$	$\frac{F \land G}{G}$
Disjunktionseinführung (DE):	$\frac{F}{F \lor G}$	$\frac{G}{F \lor G}$
Disjunktiver Syllogismus (DS):	$\frac{\neg G, F \lor G}{F}$	$\frac{\neg F, F \lor G}{G}$
Modus Ponens (MP):	$\frac{F, F \Rightarrow G}{G}$	
Modus Tollens (MT):	$\frac{\neg G, F \Rightarrow G}{\neg F}$	
Hypothetischer Syllogismus (HS):	$\frac{F \Rightarrow G, G \Rightarrow H}{F \Rightarrow H}$	
Konstruktives Dilemma (KD):	$\frac{F \Rightarrow G, H \Rightarrow I, F \lor H}{G \lor I}$	

Motivation

Mit den Regeln von eben kann ein Kalkül eingeführt werden und dann (rein syntaktisch) Ableitungen definiert werden. Wir wollen diese Möglichkeit hier nicht weiter ausführen, aber wir wollen nun die **Resolution** betrachten, ein **spezielles Ableitungsverfahren**, das mit KNFs arbeitet.

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

Auf die Idee zur Resolution kann man kommen, wenn man einige der eben dargestellten Regeln betrachtet ...

Motivation

Mit den Regeln von eben kann ein Kalkül eingeführt werden und dann (rein syntaktisch) Ableitungen definiert werden. Wir wollen diese Möglichkeit hier nicht weiter ausführen, aber wir wollen nun die **Resolution** betrachten, ein **spezielles Ableitungsverfahren**, das mit KNFs arbeitet.

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

Auf die Idee zur Resolution kann man kommen, wenn man einige der eben dargestellten Regeln betrachtet ...

Motivation

Mit den Regeln von eben kann ein Kalkül eingeführt werden und dann (rein syntaktisch) Ableitungen definiert werden. Wir wollen diese Möglichkeit hier nicht weiter ausführen, aber wir wollen nun die **Resolution** betrachten, ein **spezielles Ableitungsverfahren**, das mit KNFs arbeitet.

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

Auf die Idee zur Resolution kann man kommen, wenn man einige der eben dargestellten Regeln betrachtet ...

Motivation

Mit den Regeln von eben kann ein Kalkül eingeführt werden und dann (rein syntaktisch) Ableitungen definiert werden. Wir wollen diese Möglichkeit hier nicht weiter ausführen, aber wir wollen nun die **Resolution** betrachten, ein **spezielles Ableitungsverfahren**, das mit KNFs arbeitet.

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

Auf die Idee zur Resolution kann man kommen, wenn man einige der eben dargestellten Regeln betrachtet ...

Motivation

- Disjunktiver Syllogismus 1: $\frac{\neg G, F \vee G}{F}$
- Disjunktiver Syllogismus 2: $\frac{\neg F, F \vee G}{G}$
- Modus Ponens: $\frac{F, F \Rightarrow G}{G} \ldots \frac{F, \neg F \vee G}{G}$
- Modus Tollens: $\frac{\neg G, F \Rightarrow G}{\neg F} \ldots \frac{\neg G, \neg F \vee G}{\neg F}$
- Hypothetischer Syllogismus: $\frac{F \Rightarrow G, G \Rightarrow H}{F \Rightarrow H} \ldots \frac{\neg F \vee G, \neg G \vee H}{\neg F \vee H}$

Motivation

- Disjunktiver Syllogismus 1: $\frac{\neg G, F \vee G}{F}$
- Disjunktiver Syllogismus 2: $\frac{\neg F, F \vee G}{G}$
- Modus Ponens: $\frac{F, F \Rightarrow G}{G} \ldots . \frac{F, \neg F \vee G}{G}$
- Modus Tollens: $\frac{\neg G, F \Rightarrow G}{\neg F} \ldots . \frac{\neg G, \neg F \vee G}{\neg F}$
- Hypothetischer Syllogismus: $\frac{F \Rightarrow G, G \Rightarrow H}{F \Rightarrow H} \ldots . \frac{\neg F \vee G, \neg G \vee H}{\neg F \vee H}$

Es werden Pärchen komplementäre Literale entfernt. Dies kann man verallgemeinern um zu einem korrekten Ableitungsverfahren zu gelangen.

Vorausschau

Bei der Resolution

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

Sneak Peek

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

Vorausschau

Bei der Resolution

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

Sneak Peek

$$\begin{tikzcd} \neg B \lor C \lor A \arrow[d] & A \lor B \lor D \arrow[dl] \\ C \lor A \lor A \lor D & \end{tikzcd}$$

Mengendarstellung

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

- Ist $K = \vee_{i=1}^{m} L_i$ eine Klausel, dann ist $\mathbf{K} = \{L_1, \dots, L_m\}$ die Mengendarstellung von K.
- Ist $F = (\wedge_{i=1}^{n}(\vee_{j=1}^{m_i} L_{i,j}))$ 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

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

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

Mengendarstellung

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

- Ist $K = \vee_{i=1}^{m} L_i$ eine Klausel, dann ist $\mathbf{K} = \{L_1, \ldots, L_m\}$ die Mengendarstellung von K.
- Ist $F = (\wedge_{i=1}^{n}(\vee_{j=1}^{m_i} L_{i,j}))$ eine Formel in KNF, dann ist $\mathbf{F} = \{\{L_{1,1}, \ldots, L_{1,m_1}\}, \ldots, \{L_{n,1}, \ldots, L_{n,m_n}\}\}$ 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

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

- Ist $K = \vee_{i=1}^{m} L_i$ eine Klausel, dann ist $\mathbf{K} = \{L_1, \ldots, L_m\}$ die Mengendarstellung von K.
- Ist $F = (\wedge_{i=1}^{n} (\vee_{j=1}^{m_i} L_{i,j}))$ eine Formel in KNF, dann ist $\mathbf{F} = \{\{L_{1,1}, \ldots, L_{1,m_1}\}, \ldots, \{L_{n,1}, \ldots, L_{n,m_n}\}\}$ 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

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

- Ist $K = \vee_{i=1}^{m} L_i$ eine Klausel, dann ist $\mathbf{K} = \{L_1, \dots, L_m\}$ die Mengendarstellung von K.
- Ist $F = (\wedge_{i=1}^{n}(\vee_{j=1}^{m_i} L_{i,j}))$ 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.

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.

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 $\overline{L} \in K_2$. Dann heißt die (Literal-)Menge $R = (K_1 - \{L\}) \cup (K_2 - \{\overline{L}\})$ **Resolvente** von K_1 und K_2 (bzgl. L).

Dabei ist $\overline{L} = A$, falls $L = \neg A$ und $\overline{L} = \neg A$, 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 $\overline{L} \in K_2$. Dann heißt die (Literal-)Menge $R = (K_1 - \{L\}) \cup (K_2 - \{\overline{L}\})$ **Resolvente** von K_1 und K_2 (bzgl. L).

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

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

Die Resolutionsregel

Definition (Resolvente)

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

Dabei ist $\overline{L} = A$, falls $L = \neg A$ und $\overline{L} = \neg A$, 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 $\overline{L} \in K_2$. Dann heißt die (Literal-)Menge $R = (K_1 - \{L\}) \cup (K_2 - \{\overline{L}\})$ **Resolvente** von K_1 und K_2 (bzgl. L).

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

$$
\begin{tikzcd}
K_1 \arrow[dr] & & K_2 \arrow[dl] \\
& R &
\end{tikzcd}
$$

Die Resolutionsregel

Definition (Resolvente)

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

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

$$\begin{tikzcd}
K_1 \arrow[dr] & & K_2 \arrow[dl] \\
& R &
\end{tikzcd}$$

Die Resolutionsregel

Definition (Resolvente)

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

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

$$\begin{tikzcd}
K_1 \arrow[dr] & & K_2 \arrow[dl] \\
& R &
\end{tikzcd}$$

Beispiele

- $$\begin{tikzcd}
\{\neg B, C\} \arrow[d] & \{\neg C\} \arrow[dl] \\
\{\neg B\} &
\end{tikzcd}$$

- $$\begin{tikzcd}
\{\neg B, C, A\} \arrow[d] & \{A, D, B\} \arrow[dl] \\
\{C, A, D\} &
\end{tikzcd}$$

- $$\begin{tikzcd}
\{C\} \arrow[d] & \{\neg C\} \arrow[dl] \\
\square &
\end{tikzcd}$$

Beispiele

- $$\begin{tikzcd} \{\neg B, C\} \arrow[d] & \{\neg C\} \arrow[dl] \\ \{\neg B\} & \end{tikzcd}$$

- $$\begin{tikzcd} \{\neg B, C, A\} \arrow[d] & \{A, D, B\} \arrow[dl] \\ \{C, A, D\} & \end{tikzcd}$$

- $$\begin{tikzcd} \{C\} \arrow[d] & \{\neg C\} \arrow[dl] \\ \square & \end{tikzcd}$$

Beispiele

- $$\begin{tikzcd} \{\neg B, C\} \arrow[d] & \{\neg C\} \arrow[dl] \\ \{\neg B\} & \end{tikzcd}$$

- $$\begin{tikzcd} \{\neg B, C, A\} \arrow[d] & \{A, D, B\} \arrow[dl] \\ \{C, A, D\} & \end{tikzcd}$$

- $$\begin{tikzcd} \{C\} \arrow[d] & \{\neg C\} \arrow[dl] \\ \square & \end{tikzcd}$$

Beispiele

Wichtige Anmerkung

Das unten stehende auf keinen Fall!

$$\begin{tikzcd} \{\neg B, C\} \arrow[d] & \{B, \neg C\} \arrow[dl] \\ \square & \end{tikzcd}$$

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 auf keinen Fall!

$$\begin{tikzcd}
\{\neg B, C\} \arrow[d] & \{B, \neg C\} \arrow[dl] \\
\Box &
\end{tikzcd}$$

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!

$$\{\neg B, B\}$$
$$\downarrow$$
$$\square$$

Wichtige Anmerkung

Zur Resolventenbildung werden zwei Klauseln benötigt!

Beispiele

Wichtige Anmerkung

Das unten stehende auch nicht!

$$\{\neg B, B\}$$
$$\downarrow$$
$$\square$$

Wichtige Anmerkung

Zur Resolventenbildung werden zwei Klauseln benötigt!

Resolventenmengen

Ähnlich wie bei den Ableitungen, führen wir auch hier mehrschrittige Resolventenbildungen ein:

Definition

Sei F eine Formel in KNF dargestellt als Klauselmenge.

$$\begin{aligned} Res(F) &:= F \cup \{R \mid R \text{ ist Resolvente zweier Klauseln aus } F\} \\ Res^0(F) &:= F \\ Res^{n+1}(F) &:= Res(Res^n(F)) \\ Res^*(F) &:= \bigcup_{i \geq 0} Res^i(F) \end{aligned}$$

Resolventenmengen

Ähnlich wie bei den Ableitungen, führen wir auch hier mehrschrittige Resolventenbildungen ein:

Definition

Sei F eine Formel in KNF dargestellt als Klauselmenge.

$$
\begin{aligned}
Res(F) &:= F \cup \{R \mid R \text{ ist Resolvente zweier Klauseln aus } F\} \\
Res^0(F) &:= F \\
Res^{n+1}(F) &:= Res(Res^n(F)) \\
Res^*(F) &:= \bigcup_{i \geq 0} Res^i(F)
\end{aligned}
$$

Fragen

Sei $F = \{\{A, \neg B\}, \{\neg A, B, \neg C\}, \{A, B\}\}$. Welche Klausel ist **nicht** in $Res^*(F)$?

1. $\{A\}$
2. $\{B, \neg C\}$
3. $\{\neg C\}$
4. $\{A, \neg C\}$
5. $\{A, \neg A, \neg C\}$

Fragen

Wie ist die Resolvente R zweier Klauseln K_1, K_2 definiert?

1. $R = (K_1 - \{L\}) \cap (K_2 - \{\overline{L}\})$
2. $R = (K_1 - \{L\}) \cup (K_2 - \{\overline{L}\})$
3. $R = (K_1 - \{L, \overline{L}\}) \cup (K_2 - \{L, \overline{L}\})$
4. $R = (K_1 \cup K_2) - \{L \cup \overline{L}\})$
5. $R = (K_1 \cap K_2) - \{L \cup \overline{L}\})$

Fragen

Sei $F = \{\{A, \neg B, C\}, \{\neg C\}, \{B, C\}, \{\neg A, C\}\}$. Welche Klausel ist **nicht** in $Res^*(F)$?

1. $\{A, C\}$
2. $\{A, \neg C\}$
3. $\{A, \neg B\}$
4. $\{\neg A\}$
5. $\square$

Zur Nachbereitung

Zur Nachbereitung

1. 3.
2. 2. (wobei noch $L \in K_1$ und $\overline{L} \in K_2$ sein muss)
3. 2.

Resolutionssatz (Vorschau)

Gleich zeigen wir:

Satz (Resolutionssatz)

Eine Klauselmenge F ist unerfüllbar genau dann, wenn $\square \in Res^*(F)$ gilt.

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

Korrektheit:

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

Vollständigkeit:

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

Beispiel

Beispiel

Wir wollen die folgende Formel auf Erfüllbarkeit testen:

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

Als Klauselmenge:

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

$$\begin{tikzcd}
\{A\} \arrow[dr] & \{\neg A, B\} \arrow[d] & \{\neg B, C\} \arrow[d] & \{\neg C\} \arrow[dl] \\
 & \{B\} \arrow[d] & \{\neg B\} \arrow[dl] & \\
 & \square & &
\end{tikzcd}$$

Beispiel

Beispiel

Wir wollen die folgende Formel auf Erfüllbarkeit testen:

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

Als Klauselmenge:

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

$$\begin{tikzcd}
\{A\} \arrow[dr] & \{\neg A, B\} \arrow[d] & \{\neg B, C\} \arrow[d] & \{\neg C\} \arrow[dl] \\
& \{B\} \arrow[d] & \{\neg B\} \arrow[dl] & \\
& \square & &
\end{tikzcd}$$

Beispiel

Beispiel

Wir wollen die folgende Formel auf Erfüllbarkeit testen:

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

Als Klauselmenge:

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

$$\begin{tikzcd}
\{A\} \arrow[dr] & \{\neg A, B\} \arrow[d] & \{\neg B, C\} \arrow[d] & \{\neg C\} \arrow[dl] \\
& \{B\} \arrow[d] & \{\neg B\} \arrow[dl] & \\
& \square & &
\end{tikzcd}$$

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 $K = \vee_{i=1}^{m} L_i$ eine Klausel, dann ist $\mathbf{K} = \{L_1, \ldots, L_m\}$ die Mengendarstellung von K.
- Ist $F = (\wedge_{i=1}^{n} (\vee_{j=1}^{m_i} L_{i,j}))$ eine Formel in KNF, dann ist $\mathbf{F} = \{\{L_{1,1}, \ldots, L_{1,m_1}\}, \ldots, \{L_{n,1}, \ldots, L_{n,m_n}\}\}$ 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 Formel in KNF als Klauselmenge 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 Klauselmenge. Dann gilt

$$F \equiv Res^1(F) \equiv Res^2(F) \equiv \ldots \equiv Res^n(F) \equiv \ldots \equiv Res^*(F)$$

Satz (Resolutionssatz)

Eine Klauselmenge F ist unerfüllbar genau dann, wenn $\square \in Res^*(F)$ gilt.

Resolutionslemma

Satz (Resolutionslemma)

Sei F eine Formel 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 $R = (K_1 - \{L\}) \cup (K_2 - \{\overline{L}\})$ mit $K_1, K_2 \in F$ und $L \in K_1$, $\overline{L} \in K_2$. Nun gibt es zwei Fälle:

1. $\mathcal{A} \models L$. Dann folgt $\mathcal{A} \models K_2 - \{\overline{L}\}$ aus $\mathcal{A} \models K_2$ und $\mathcal{A} \not\models \overline{L}$.
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 $\mathcal{A} \models F \cup \{R\}$. □

Resolutionslemma

Satz (Resolutionslemma)

Sei F eine Formel 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 $R = (K_1 - \{L\}) \cup (K_2 - \{\overline{L}\})$ mit $K_1, K_2 \in F$ und $L \in K_1$, $\overline{L} \in K_2$. Nun gibt es zwei Fälle:

1. $\mathcal{A} \models L$. Dann folgt $\mathcal{A} \models K_2 - \{\overline{L}\}$ aus $\mathcal{A} \models K_2$ und $\mathcal{A} \not\models \overline{L}$.
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 $\mathcal{A} \models F \cup \{R\}$. $\square$

Resolutionslemma

Satz (Resolutionslemma)

Sei F eine Formel 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 $R = (K_1 - \{L\}) \cup (K_2 - \{\overline{L}\})$ mit $K_1, K_2 \in F$ und $L \in K_1$, $\overline{L} \in K_2$. Nun gibt es zwei Fälle:

1. $\mathcal{A} \models L$. Dann folgt $\mathcal{A} \models K_2 - \{\overline{L}\}$ aus $\mathcal{A} \models K_2$ und $\mathcal{A} \not\models \overline{L}$.
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 $\mathcal{A} \models F \cup \{R\}$. $\square$

Resolutionslemma

Satz (Resolutionslemma)

Sei F eine Formel 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 $R = (K_1 - \{L\}) \cup (K_2 - \{\overline{L}\})$ mit $K_1, K_2 \in F$ und $L \in K_1$, $\overline{L} \in K_2$. Nun gibt es zwei Fälle:

1. $\mathcal{A} \models L$. Dann folgt $\mathcal{A} \models K_2 - \{\overline{L}\}$ aus $\mathcal{A} \models K_2$ und $\mathcal{A} \not\models \overline{L}$.
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 $\mathcal{A} \models F \cup \{R\}$. □

Resolutionslemma

Satz (Resolutionslemma)

Sei F eine Formel 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 $R = (K_1 - \{L\}) \cup (K_2 - \{\overline{L}\})$ mit $K_1, K_2 \in F$ und $L \in K_1$, $\overline{L} \in K_2$. Nun gibt es zwei Fälle:

1. $\mathcal{A} \models L$. Dann folgt $\mathcal{A} \models K_2 - \{\overline{L}\}$ aus $\mathcal{A} \models K_2$ und $\mathcal{A} \not\models \overline{L}$.
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 $\mathcal{A} \models F \cup \{R\}$. □

Resolutionslemma

Satz (Resolutionslemma)

Sei F eine Formel 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 $R = (K_1 - \{L\}) \cup (K_2 - \{\overline{L}\})$ mit $K_1, K_2 \in F$ und $L \in K_1$, $\overline{L} \in K_2$. Nun gibt es zwei Fälle:

1. $\mathcal{A} \models L$. Dann folgt $\mathcal{A} \models K_2 - \{\overline{L}\}$ aus $\mathcal{A} \models K_2$ und $\mathcal{A} \not\models \overline{L}$.
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 $\mathcal{A} \models F \cup \{R\}$. □

Resolutionslemma

Satz (Resolutionslemma)

Sei F eine Formel 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 $R = (K_1 - \{L\}) \cup (K_2 - \{\overline{L}\})$ mit $K_1, K_2 \in F$ und $L \in K_1$, $\overline{L} \in K_2$. Nun gibt es zwei Fälle:

1. $\mathcal{A} \models L$. Dann folgt $\mathcal{A} \models K_2 - \{\overline{L}\}$ aus $\mathcal{A} \models K_2$ und $\mathcal{A} \not\models \overline{L}$.
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 $\mathcal{A} \models F \cup \{R\}$. □

Resolutionslemma

Satz (Resolutionslemma)

Sei F eine Formel 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 $R = (K_1 - \{L\}) \cup (K_2 - \{\overline{L}\})$ mit $K_1, K_2 \in F$ und $L \in K_1$, $\overline{L} \in K_2$. Nun gibt es zwei Fälle:

1. $\mathcal{A} \models L$. Dann folgt $\mathcal{A} \models K_2 - \{\overline{L}\}$ aus $\mathcal{A} \models K_2$ und $\mathcal{A} \not\models \overline{L}$.
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 $\mathcal{A} \models F \cup \{R\}$. □

Ein Lemma/Korollar

Lemma

Sei F eine Formel in KNF als Klauselmenge. Dann gilt

$$F \equiv Res^1(F) \equiv Res^2(F) \equiv \ldots \equiv Res^n(F) \equiv \ldots \equiv Res^*(F)$$

Beweis.

Folgt sofort aus dem Resolutionslemma von eben ... □

Der Resolutionssatz

Satz (Resolutionssatz)

Eine Klauselmenge F ist unerfüllbar genau dann, wenn $\square \in Res^*(F)$ gilt.

Beweis

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

Der Resolutionssatz

Satz (Resolutionssatz)

Eine Klauselmenge F ist unerfüllbar genau dann, wenn $\square \in Res^*(F)$ gilt.

Beweis

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

Dies zeigt die *Korrektheit* der Resolution.

Der Resolutionssatz

Satz (Resolutionssatz)

Eine Klauselmenge F ist unerfüllbar genau dann, wenn $\square \in Res^*(F)$ gilt.

Beweis

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

Dies zeigt die *Korrektheit* der Resolution.

Der Resolutionssatz

Satz (Resolutionssatz)

Eine Klauselmenge F ist unerfüllbar genau dann, wenn $\square \in Res^*(F)$ gilt.

Beweis

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

Dies zeigt die *Korrektheit* der Resolution.

Der Resolutionssatz

Satz (Resolutionssatz)

Eine Klauselmenge F ist unerfüllbar genau dann, wenn $\square \in Res^*(F)$ gilt.

Beweis

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

Dies zeigt die *Korrektheit* der Resolution.

Der Resolutionssatz

Satz (Resolutionssatz)

Eine Klauselmenge F ist unerfüllbar genau dann, wenn $\square \in Res^*(F)$ gilt.

Beweis

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

Fortsetzung des Beweises

Beweis

Sei nun F als unerfüllbar angenommen. Wir wollen $\square \in Res^*(F)$ 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 $F \subseteq Res^*(F)$ ist $\square \in F \subseteq Res^*(F)$.

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

Induktionsschritt. Sei F eine unerfüllbare Formelmenge, die die atomaren Formeln $A_1, \ldots, A_{n+1}$ 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

Sei nun F als unerfüllbar angenommen. Wir wollen $\square \in Res^*(F)$ 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 $F \subseteq Res^*(F)$ ist $\square \in F \subseteq Res^*(F)$.

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

Induktionsschritt. Sei F eine unerfüllbare Formelmenge, die die atomaren Formeln $A_1, \ldots, A_{n+1}$ 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

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

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

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

Induktionsschritt. Sei F eine unerfüllbare Formelmenge, die die atomaren Formeln $A_1, \ldots, A_{n+1}$ 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

Sei nun F als unerfüllbar angenommen. Wir wollen $\square \in Res^*(F)$ 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 $F \subseteq Res^*(F)$ ist $\square \in F \subseteq Res^*(F)$.

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

Induktionsschritt. Sei F eine unerfüllbare Formelmenge, die die atomaren Formeln $A_1, \ldots, A_{n+1}$ 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

Sei nun F als unerfüllbar angenommen. Wir wollen $\square \in Res^*(F)$ 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 $F \subseteq Res^*(F)$ ist $\square \in F \subseteq Res^*(F)$.

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

Induktionsschritt. Sei F eine unerfüllbare Formelmenge, die die atomaren Formeln $A_1, \ldots, A_{n+1}$ 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 $B \in \{A_1, \ldots, A_n\}$ 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 $\square \in Res^*(F_0)$ und $\square \in Res^*(F_1)$ nach Induktionsannahme.

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 $B \in \{A_1, \ldots, A_n\}$ 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 $\square \in Res^*(F_0)$ und $\square \in Res^*(F_1)$ nach Induktionsannahme.

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 $B \in \{A_1, \ldots, A_n\}$ 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 $\square \in Res^*(F_0)$ und $\square \in Res^*(F_1)$ nach Induktionsannahme.

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 $B \in \{A_1, \ldots, A_n\}$ 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 $\square \in Res^*(F_0)$ und $\square \in Res^*(F_1)$ nach Induktionsannahme.

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 $B \in \{A_1, \ldots, A_n\}$ 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 $\square \in Res^*(F_0)$ und $\square \in Res^*(F_1)$ nach Induktionsannahme.

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 $B \in \{A_1, \ldots, A_n\}$ 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 $\square \in Res^*(F_0)$ und $\square \in Res^*(F_1)$ nach Induktionsannahme.

Fortsetzung des Beweises

Beweis

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

- $K_m = \square$ und für $i = 1, \ldots, m$ 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 $K'_1, \ldots, K'_t$.

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

Zwei Anmerkungen:

1. Gilt hier bereits $\square \in Res^*(F)$, so sind wir bereits fertig! Dies tritt auf, wenn in keiner der Klauseln $K_1, \ldots, K_m$ 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 $K_1, \ldots, K_m$ nicht auf!

Fortsetzung des Beweises

Beweis

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

- $K_m = \Box$ und für $i = 1, \ldots, m$ 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 $K'_1, \ldots, K'_t$.
Wir stellen nun durch Wiedereinführung von A_{n+1} in die Klauseln $K_1, \ldots, K_m$ dort, wo es gestrichen wurde, die ursprünglichen Klauseln wieder her und erhalten so eine Folge von Klauseln, die $\{A_{n+1}\} \in Res^*(F)$ bezeugt.
Zwei Anmerkungen:

1. Gilt hier bereits $\Box \in Res^*(F)$, so sind wir bereits fertig! Dies tritt auf, wenn in keiner der Klauseln $K_1, \ldots, K_m$ 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 $K_1, \ldots, K_m$ nicht auf!

Fortsetzung des Beweises

Beweis

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

- $K_m = \Box$ und für $i = 1, \ldots, m$ 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 $K'_1, \ldots, K'_t$.
Wir stellen nun durch Wiedereinführung von A_{n+1} in die Klauseln $K_1, \ldots, K_m$ dort, wo es gestrichen wurde, die ursprünglichen Klauseln wieder her und erhalten so eine Folge von Klauseln, die $\{A_{n+1}\} \in Res^*(F)$ bezeugt.

Zwei Anmerkungen:

1. Gilt hier bereits $\Box \in Res^*(F)$, so sind wir bereits fertig! Dies tritt auf, wenn in keiner der Klauseln $K_1, \ldots, K_m$ 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 $K_1, \ldots, K_m$ nicht auf!

Fortsetzung des Beweises

Beweis

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

- $K_m = \Box$ und für $i = 1, \ldots, m$ 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 $K'_1, \ldots, K'_t$.
Wir stellen nun durch Wiedereinführung von A_{n+1} in die Klauseln $K_1, \ldots, K_m$ dort, wo es gestrichen wurde, die ursprünglichen Klauseln wieder her und erhalten so eine Folge von Klauseln, die $\{A_{n+1}\} \in Res^*(F)$ bezeugt.
Zwei Anmerkungen:

1. Gilt hier bereits $\Box \in Res^*(F)$, so sind wir bereits fertig! Dies tritt auf, wenn in keiner der Klauseln $K_1, \ldots, K_m$ 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 $K_1, \ldots, K_m$ 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 $\{\neg A_{n+1}\} \in Res^*(F)$ bezeugt (Mit analogen Anmerkungen wie eben.)

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

Dies zeigt die *Vollständigkeit* der Resolution und schließt den Beweis des Resolutionssatzes ab. $\square$

Fortsetzung des Beweises

Beweis.

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

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

Dies zeigt die *Vollständigkeit* der Resolution und schließt den Beweis des Resolutionssatzes ab. $\square$

Fortsetzung des Beweises

Beweis.

Ebenso erhält man aus den $K'_1, \ldots, K'_t$ durch Wiedereinführen von $\neg A_{n+1}$ dort, wo es gelöscht wurde eine Folge, die $\{\neg A_{n+1}\} \in Res^*(F)$ bezeugt (Mit analogen Anmerkungen wie eben.)
Insg. haben wir damit $\{A_{n+1}\} \in Res^*(F)$ und $\{\neg A_{n+1}\} \in Res^*(F)$. Die Resolvente dieser beiden Klauseln bezeugt dann $\square \in Res^*(F)$.
Dies zeigt die *Vollständigkeit* der Resolution und schließt den Beweis des Resolutionssatzes ab. $\square$

Fortsetzung des Beweises

Beweis.

Ebenso erhält man aus den $K'_1, \ldots, K'_t$ durch Wiedereinführen von $\neg A_{n+1}$ dort, wo es gelöscht wurde eine Folge, die $\{\neg A_{n+1}\} \in \mathit{Res}^*(F)$ bezeugt (Mit analogen Anmerkungen wie eben.)
Insg. haben wir damit $\{A_{n+1}\} \in \mathit{Res}^*(F)$ und $\{\neg A_{n+1}\} \in \mathit{Res}^*(F)$. Die Resolvente dieser beiden Klauseln bezeugt dann $\square \in \mathit{Res}^*(F)$.
Dies zeigt die *Vollständigkeit* der Resolution und schließt den Beweis des Resolutionssatzes ab. $\square$

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.

Fragen

Der Resolutionssatz besagte: Eine Klauselmenge F ist unerfüllbar genau dann, wenn $\Box \in Res^*(F)$ gilt. Welche Richtung war dabei die Korrektheit?

1. Wenn $\Box \in Res^*(F)$, dann ist F unerfüllbar.
2. Wenn F unerfüllbar ist, dann ist $\Box \in Res^*(F)$.

Fragen

Das Resolutionslemma besagte: Sei F eine Formel in KNF und R eine Resolvente zweier Klauseln K_1 und K_2 in F. Dann sind F und $F \cup \{R\}$ äquivalent. An welcher Stelle im Beweis des Resolutionssatzes wird dies gebraucht?

1. Beim Korrektheitsbeweis
2. Beim Vollständigkeitsbeweis
3. Wird an dieser Stelle nicht benötigt

Fragen

Beim Beweis der Vollständigkeit wird im Resolutionssatz eine Induktion gemacht. Worüber?

1. Strukturelle Induktion über den Formelaufbau der Formel F
2. Strukturelle Induktion über den Formelaufbau der Resolvente R
3. Vollständige Induktion über die Länge der Formel F
4. Vollständige Induktion über die Anzahl der Quantoren in F
5. Vollständige Induktion über die Anzahl der Aussagensymbole in F

Zur Nachbereitung

Zur Nachbereitung

1. 1.
2. 1.
3. 5. (Genauer: Die Anzahl der *verschiedenen* Aussagensymbole/atomaren Formeln in F)

Resolutionsalgorithmus

Ein Algorithmus zur Resolutionsbildung:

Algorithmus 1 Resolutionsalgorithmus

Input: Eine Formel F in KNF
Bilde zu F eine Klauselmenge (auch F genannt)
repeat
    $G := F$
    $F := Res(F)$
until $(\square \in F)$ **or** $(\mathrm{F} = \mathrm{G})$
if $\square \in F$ **then**
    return F ist unerfüllbar
else
    return F ist erfüllbar
end if

Termination

Der Resolutionsalgorithmus terminiert aufgrund des folgenden Satzes:

Satz

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

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

Beweis.

Enthalte F die n atomaren Formeln $A_1, \ldots, A_n$. 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 $A_1, \ldots, A_n$ aufgebaut sein können und die in $Res^*(F)$ vorkommen können.
Da in einem Schritt von $Res^i(F)$ zu $Res^{i+1}(F)$ mindestens eine Klausel hinzukommen muss, muss $k \leq 4^n$ gelten. □

Termination

Der Resolutionsalgorithmus terminiert aufgrund des folgenden Satzes:

Satz

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

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

Beweis.

Enthalte F die n atomaren Formeln $A_1, \ldots, A_n$. 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 $A_1, \ldots, A_n$ aufgebaut sein können und die in $Res^*(F)$ vorkommen können.
Da in einem Schritt von $Res^i(F)$ zu $Res^{i+1}(F)$ mindestens eine Klausel hinzukommen muss, muss $k \leq 4^n$ gelten. □

Termination

Der Resolutionsalgorithmus terminiert aufgrund des folgenden Satzes:

Satz

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

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

Beweis.

Enthalte F die n atomaren Formeln $A_1, \ldots, A_n$. 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 $A_1, \ldots, A_n$ aufgebaut sein können und die in $Res^*(F)$ vorkommen können.
Da in einem Schritt von $Res^i(F)$ zu $Res^{i+1}(F)$ mindestens eine Klausel hinzukommen muss, muss $k \leq 4^n$ gelten. □

Termination

Der Resolutionsalgorithmus terminiert aufgrund des folgenden Satzes:

Satz

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

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

Beweis.

Enthalte F die n atomaren Formeln $A_1, \ldots, A_n$. 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 $A_1, \ldots, A_n$ aufgebaut sein können und die in $Res^*(F)$ vorkommen können.
Da in einem Schritt von $Res^i(F)$ zu $Res^{i+1}(F)$ mindestens eine Klausel hinzukommen muss, muss $k \leq 4^n$ gelten. □

Termination

Der Resolutionsalgorithmus terminiert aufgrund des folgenden Satzes:

Satz

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

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

Beweis.

Enthalte F die n atomaren Formeln $A_1, \ldots, A_n$. 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 $A_1, \ldots, A_n$ aufgebaut sein können und die in $Res^*(F)$ vorkommen können.

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

Termination

Der Resolutionsalgorithmus terminiert aufgrund des folgenden Satzes:

Satz

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

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

Beweis.

Enthalte F die n atomaren Formeln $A_1, \ldots, A_n$. 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 $A_1, \ldots, A_n$ aufgebaut sein können und die in $Res^*(F)$ vorkommen können.

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

Termination

Der Resolutionsalgorithmus terminiert aufgrund des folgenden Satzes:

Satz

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

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

Beweis.

Enthalte F die n atomaren Formeln $A_1, \ldots, A_n$. 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 $A_1, \ldots, A_n$ aufgebaut sein können und die in $Res^*(F)$ vorkommen können.
Da in einem Schritt von $Res^i(F)$ zu $Res^{i+1}(F)$ mindestens eine Klausel hinzukommen muss, muss $k \leq 4^n$ gelten. □

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 Wahrheitstafeln 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.

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 Wahrheitstafeln 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).

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).

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.)

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.)

Wiederholung

Dies schließt die Aussagenlogik zunächst ab.

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 Äquivalenzumforungen
  - mittels Wahrheitstafeln

Wiederholung

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

- Hornformeln kennengelernt, die eine *Einschränkung* der Aussagenlogik sind, dafür aber einen effizienten (Un-)Erfüllbarkeitstest erlauben und
- Resolution (spezielles Ableitungsverfahren) kennengelernt.

Literaturhinweis

Alle obigen Themen werden im Buch *Logik für Informatiker* von Uwe Schöning, an dem sich dieser Vorlesungsteil orientiert, behandelt.