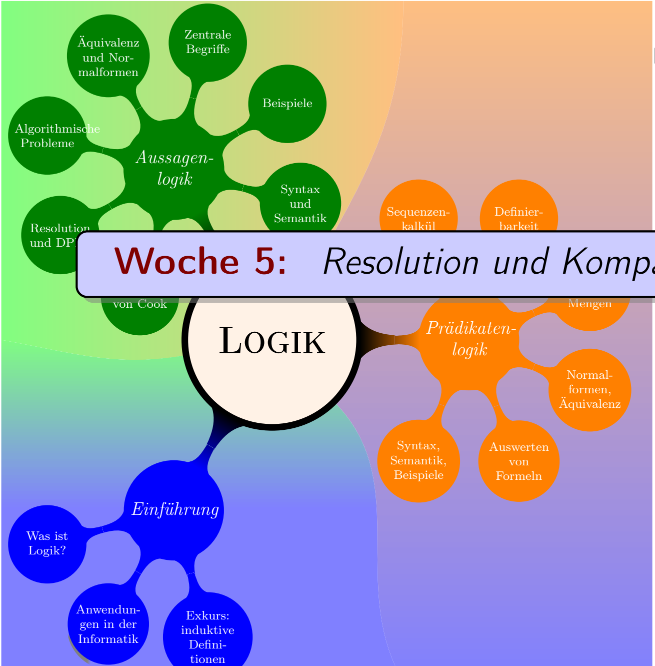
Logik im Wintersemester 2022/2023



Woche 5: *Resolution und Kompaktheit*

Nov.	17	18	19	20	21	22	23	*Einführung*
	24	25	26	27	28	29	30	*Aussagenlogik*
	31	1	2	3	4	5	6	*Normalformen*
	7	8	9	10	11	12	13	*Resolution*
Dez.	14	15	16	17	18	19	20	*Kompaktheit*
	21	22	23	24	25	26	27	*DPLL, Satz von Cook*
			0	1	2	3	4	*Strukturen und FO*
	5	6	7	8	9	10	11	*Prädikatenlogik*
	12	13	14	15	16	17	18	*Komplexität von FO*
Jan.	19	20	21	22	23	24	25	*Weihnachten*
	26	27	28	29	30	31	1	*Neujahr*
	2	3	4	5	6	7	8	*Normalformen*
	9	10	11	12	13	14	15	*Definierbarkeit*
	16	17	18	19	20	21	22	*EF-Spiele*
Feb.	23	24	25	26	27	28	29	*EF-Spiele*
	30	31	1	2	3	4	5	*Sequenzenkalkül AL*
	6	7	8	9	10	11	12	*Sequenzenkalkül FO*
	13	14	15	16	17	18	19	*Ausblick*

Wiederholung

Aussagenlogische Resolution

Die aussagenlogische Resolution ist eine Methode um zu zeigen, dass eine Formel in konjunktiver Normalform nicht erfüllbar ist.

Theorem. Zu jeder Formel φ gibt es eine Formel ψ in KNF, so dass

1. φ ist genau dann erfüllbar, wenn ψ erfüllbar ist.
2. $|\psi| \leq c \cdot |\varphi|$ für eine Konstante $c \in \mathbb{N}$ unabhängig von φ.
3. ψ kann aus φ effizient (in Linearzeit) berechnet werden.

Notation

Eine Formel

$$(\neg X \vee \neg Z) \wedge (X \vee \neg Z) \wedge (\neg Y \vee W) \wedge (Y \vee Z \vee V) \wedge \neg V \wedge (\neg W \vee Z)$$

in KNF schreiben wir als Klauselmenge wie folgt:

$$\{\neg X, \neg Z\}, \quad \{X, \neg Z\}, \quad \{\neg Y, W\}, \quad \{Y, Z, V\}, \quad \{\neg V\}, \quad \{\neg W, Z\}$$

D.h., $\qquad Y \vee Z \vee V \qquad \rightsquigarrow \qquad \{Y, Z, V\}.$

Resolution

Definition. Seien C, C_1, C_2 Klauseln.

C ist eine Resolvente von C_1, C_2, wenn es ein Literal L gibt mit

$$L \in C_1 \text{ und } \overline{L} \in C_2 \text{ und } C = \big(C_1 \setminus \{L\}\big) \cup \big(C_2 \setminus \{\overline{L}\}\big).$$

Wir sagen, dass C_1 und C_2 resolviert werden. Die Menge der Resolventen von C_1 und C_2 bezeichnen wir mit $Res(C_1, C_2)$.

Graphische Darstellung

$$\begin{tikzcd} C_1 \arrow[dr, dash] & & C_2 \arrow[dl, dash] \\ & C & \end{tikzcd}$$

Beispiel

$$\begin{tikzcd} \{\cancel{Y}, Z\} \arrow[dr, dash] & & \{\cancel{\neg Y}, W\} \arrow[dl, dash] \\ & \{Z, W\} & \end{tikzcd}$$

Erinnerung.

Für ein Literal L bezeichnet $\bar{L}$ das duale Literal, d.h. $\bar{X} = \neg X$ und $\overline{\neg X} = X$.

Klausel $C := \{L_1, \dots, L_n\}$ entspricht $\varphi(C) := \bigvee_{i=1}^{n} L_i$

Klauselmenge $\mathcal{C} := \{C_1, \dots, C_n\}$ entspricht $\varphi(\mathcal{C}) := \bigwedge_{i=1}^{n} \varphi(C_i)$.

Aussagenlogische Resolution

Resolvente.
C_1, C_2 Klauseln,

L Literal
mit $L \in C_1$, $\overline{L} \in C_2$.

Resolvente $C = C_1 \setminus \{L\} \cup C_2 \setminus \{\overline{L}\}$.

Lemma.
Sei $\mathcal{C}$ eine Klauselmenge. Seien $C_1, C_2 \in \mathcal{C}$ und $C \in \mathit{Res}(C_1, C_2)$.

Dann gilt $\{C_1, C_2\} \models C$ und $\mathcal{C}$ und $\mathcal{C} \cup \{C\}$ sind äquivalent.

Beweis. Wir zeigen zunächst, dass $\{C_1, C_2\} \models C$.

Zu zeigen: Jede Belegung β mit $\beta \models \{C_1, C_2\}$ erfüllt auch C.

Sei β eine Belegung mit $\beta \models \{C_1, C_2\}$ und L das resolvierte Literal.

Fall 1: $[\![L]\!]^\beta = 1$	**Fall 2:** $[\![L]\!]^\beta = 0$
Es gilt $\beta \models C_2$. Also existiert $L' \in C_2 \setminus \{\overline{L}\}$ mit $[\![L']\!]^\beta = 1$. Da nach Definition $L' \in C$, folgt $\beta \models C$.	Es gilt $\beta \models C_1$. Also existiert $L' \in C_1 \setminus \{L\}$ mit $[\![L']\!]^\beta = 1$. Da nach Definition $L' \in C$, folgt $\beta \models C$.

In beiden Fällen gilt also $\beta \models C$.

Resolutionsableitungen

Resolvente.
C_1, C_2 Klauseln,

L Literal
mit $L \in C_1$, $\overline{L} \in C_2$.

Resolvente $C = C_1 \setminus \{L\} \cup C_2 \setminus \{\overline{L}\}$.

Definition.

1. Eine Resolutionsableitung einer Klausel C aus einer Klauselmenge $\mathcal{C}$ ist eine Sequenz $(C_1, \dots, C_n)$, so dass
   - $C_n = C$ und
   - für alle $1 \leq i \leq n$ gilt

     $C_i \in \mathcal{C}$ oder es gibt $j, k < i$ mit $C_i \in \mathit{Res}(C_j, C_k)$.

   Wir sagen, dass C einen Resolutionsbeweis aus $\mathcal{C}$ hat und schreiben dies als $\mathcal{C} \vdash_R C$.

2. Eine Resolutionswiderlegung einer Klauselmenge $\mathcal{C}$ ist eine Resolutionsableitung der leeren Klausel $\square$.

Beispiel einer Resolutionswiderlegung

$$\varphi := (V_1 \vee V_2) \wedge (\neg V_2 \vee V_n) \wedge (V_2 \vee \neg V_1) \wedge (\neg V_n \vee \neg V_4) \wedge (\neg V_n \vee V_4 \vee \neg V_5) \wedge (V_4 \vee V_5)$$

Klauselform.

$$\begin{tikzcd}
\{V_1, V_2\} \arrow[dr] & & \{\neg V_2, V_n\} \arrow[dl] \arrow[dr] & & \{V_2, \neg V_1\} \arrow[dl] & & \{\neg V_n, \neg V_4\} \arrow[ddr] & & \{\neg V_n, V_4, \neg V_5\} \arrow[dr] & & \{V_4, V_5\} \arrow[dl] \\
& \{V_1, V_n\} \arrow[dr] & & \{V_n, \neg V_1\} \arrow[dl] & & & & & & \{\neg V_n, V_4\} \arrow[dll] & \\
& & \{V_n\} \arrow[drr] & & & & & \{\neg V_n\} \arrow[dlll] & & & \\
& & & & \square & & & & & &
\end{tikzcd}$$

Resolutionsableitung von $\square$.

$$\Big(\{V_1, V_2\},\ \{\neg V_2, V_n\},\ \{V_1, V_n\},\ \{V_2, \neg V_1\},\ \{V_n, \neg V_1\},\ \{V_n\},$$
$$\{V_4, V_5\},\ \{\neg V_n, V_4, \neg V_5\},\ \{\neg V_n, V_4\},\ \{\neg V_n, \neg V_4\},\ \{\neg V_n\},\ \square \Big)$$

Res.wid. von $\mathcal{C}$.
Sequenz $(C_1, \dots, C_n)$:
$C_n = C$ und
für alle $1 \leq i \leq n$ gilt
- $C_i \in \mathcal{C}$ oder
- es gibt $j, k < i$ mit $C_i \in Res(C_j, C_k)$.

Resolvente.
C_1, C_2 Klauseln,
L Literal
mit $L \in C_1$, $\overline{L} \in C_2$.
Resolvente $C = C_1 \setminus \{L\} \cup C_2 \setminus \{\overline{L}\}$.

5.1 Vollständigkeit und Korrektheit der Resolution

Der aussagenlogische Resolutionskalkül

Theorem. Eine Menge $\mathcal{C}$ von Klauseln hat genau dann eine Resolutionswiderlegung, wenn $\mathcal{C}$ unerfüllbar ist. D.h. es gilt

$$\mathcal{C} \vdash_R \Box \quad \text{gdw.} \quad \mathcal{C} \text{ ist unerfüllbar} \quad \text{gdw.} \quad \mathcal{C} \models \bot$$

Der aussagenlogische Resolutionskalkül

Theorem. Eine Menge $\mathcal{C}$ von Klauseln hat genau dann eine Resolutionswiderlegung, wenn $\mathcal{C}$ unerfüllbar ist. D.h. es gilt

$$\mathcal{C} \vdash_R \Box \quad \text{gdw.} \quad \mathcal{C} \text{ ist unerfüllbar} \quad \text{gdw.} \quad \mathcal{C} \models \bot$$

Lemma. (Korrektheit des Resolutionskalküls)
Wenn eine Menge $\mathcal{C}$ von Klauseln eine Resolutionswiderlegung hat, ist $\mathcal{C}$ unerfüllbar.

Lemma. (Vollständigkeit des Resolutionskalküls)
Jede unerfüllbare Klauselmenge $\mathcal{C}$ hat eine Resolutionswiderlegung.

Korrektheit des Resolutionskalküls

Res.wid. von $\mathcal{C}$.
Sequenz $(C_1, \dots, C_n)$:
$C_n = C$ und
für alle $1 \leq i \leq n$ gilt
$C_i \in \mathcal{C}$ oder
es gibt $j, k < i$ mit
$C_i \in Res(C_j, C_k)$.

Lemma. Sei $\mathcal{C}$ eine Klauselmenge und C eine Klausel.

Wenn $\mathcal{C} \vdash_R C$, dann $\mathcal{C} \models C$.

Theorem.
Eine Menge $\mathcal{C}$ von Klauseln hat genau dann eine Resolutionswiderlegung, wenn $\mathcal{C}$ unerfüllbar ist. D.h. es gilt

$$\mathcal{C} \vdash_R \square \quad \text{gdw.} \quad \mathcal{C} \text{ ist unerfüllbar} \quad \text{gdw.} \quad \mathcal{C} \models \bot$$

Korrektheit des Resolutionskalküls

Lemma. Sei $\mathcal{C}$ eine Klauselmenge und C eine Klausel.

Wenn $\mathcal{C} \vdash_R C$, dann $\mathcal{C} \models C$.

Korollar. (Korrektheit des Resolutionskalküls)

Wenn eine Menge $\mathcal{C}$ von Klauseln eine Resolutionswiderlegung hat, dann ist $\mathcal{C}$ unerfüllbar.

Theorem.
Eine Menge $\mathcal{C}$ von Klauseln hat genau dann eine Resolutionswiderlegung, wenn $\mathcal{C}$ unerfüllbar ist. D.h. es gilt

$$\mathcal{C} \vdash_R \Box \quad \text{gdw.} \quad \mathcal{C} \text{ ist unerfüllbar} \quad \text{gdw.} \quad \mathcal{C} \models \bot$$

Res.wid. von $\mathcal{C}$.
Sequenz $(C_1, \dots, C_n)$:
$C_n = C$ und
für alle $1 \leq i \leq n$ gilt
$C_i \in \mathcal{C}$ oder
es gibt $j, k < i$ mit
$C_i \in Res(C_j, C_k)$.

Korrektheit der Resolution

Lemma. Sei $\mathcal{C}$ eine Klauselmenge und C eine Klausel.

Wenn $\mathcal{C} \vdash_R C$, dann $\mathcal{C} \models C$.

Res.wid. von $\mathcal{C}$.
Sequenz $(C_1, \dots, C_n)$:
$C_n = C$ und
für alle $1 \leq i \leq n$ gilt
$C_i \in \mathcal{C}$ oder
es gibt $j, k < i$ mit
$C_i \in Res(C_j, C_k)$.

Resolvente.
C_1, C_2 Klauseln,

L Literal
mit $L \in C_1$, $\overline{L} \in C_2$.

Resolvente $C =$
$C_1 \setminus \{L\} \cup C_2 \setminus \{\overline{L}\}$.

Lemma.
Sei $\mathcal{C}$ Klauselmenge,
$C_1, C_2 \in \mathcal{C}$ und
$C \in Res(C_1, C_2)$.

Dann gilt:
$\{C_1, C_2\} \models C$ und
$\mathcal{C} \equiv \mathcal{C} \cup \{C\}$

Korrektheit der Resolution

Lemma. Sei $\mathcal{C}$ eine Klauselmenge und C eine Klausel.

Wenn $\mathcal{C} \vdash_R C$, dann $\mathcal{C} \models C$.

Beweis. Sei $(C_1, \dots, \underset{=C}{C_n})$ eine Resolutionsableitung von C aus $\mathcal{C}$.

Res.wid. von $\mathcal{C}$. Sequenz $(C_1, \dots, C_n)$: $C_n = C$ und für alle $1 \leq i \leq n$ gilt
- $C_i \in \mathcal{C}$ oder
- es gibt $j, k < i$ mit $C_i \in Res(C_j, C_k)$.

Resolvente. C_1, C_2 Klauseln,

L Literal mit $L \in C_1$, $\overline{L} \in C_2$.

Resolvente $C = C_1 \setminus \{L\} \cup C_2 \setminus \{\overline{L}\}$.

Lemma. Sei $\mathcal{C}$ Klauselmenge, $C_1, C_2 \in \mathcal{C}$ und $C \in Res(C_1, C_2)$.

Dann gilt:
- $\{C_1, C_2\} \models C$ und
- $\mathcal{C} \equiv \mathcal{C} \cup \{C\}$

Korrektheit der Resolution

Lemma. Sei $\mathcal{C}$ eine Klauselmenge und C eine Klausel.

Wenn $\mathcal{C} \vdash_R C$, dann $\mathcal{C} \models C$.

Beweis. Sei $(C_1, \dots, C_n)$ eine Resolutionsableitung von C aus $\mathcal{C}$.

Per Induktion über i zeigen wir, dass $\mathcal{C} \models C_i$ für alle $1 \leq i \leq n$.

Für $i = n$ folgt somit $\mathcal{C} \models C_n = C$.

Res.wid. von $\mathcal{C}$.
Sequenz $(C_1, \dots, C_n)$:
$C_n = C$ und
für alle $1 \leq i \leq n$ gilt
$C_i \in \mathcal{C}$ oder
es gibt $j, k < i$ mit
$C_i \in Res(C_j, C_k)$.

Resolvente.
C_1, C_2 Klauseln,

L Literal
mit $L \in C_1$, $\overline{L} \in C_2$.

Resolvente $C = C_1 \setminus \{L\} \cup C_2 \setminus \{\overline{L}\}$.

Lemma.
Sei $\mathcal{C}$ Klauselmenge,
$C_1, C_2 \in \mathcal{C}$ und
$C \in Res(C_1, C_2)$.

Dann gilt:
$\{C_1, C_2\} \models C$ und
$\mathcal{C} \equiv \mathcal{C} \cup \{C\}$

Korrektheit der Resolution

Lemma. Sei $\mathcal{C}$ eine Klauselmenge und C eine Klausel.

Wenn $\mathcal{C} \vdash_R C$, dann $\mathcal{C} \models C$.

Beweis. Sei $(C_1, \dots, C_n)$ eine Resolutionsableitung von C aus $\mathcal{C}$.

Per Induktion über i zeigen wir, dass $\mathcal{C} \models C_i$ für alle $1 \leq i \leq n$.

Für $i = n$ folgt somit $\mathcal{C} \models C_n = C$.

Induktionsbasis: $i = 1$.

Es gilt $C_1 \in \mathcal{C}$ und somit $\mathcal{C} \models C_1$.

Res.wid. von $\mathcal{C}$.
Sequenz $(C_1, \dots, C_n)$:
$C_n = C$ und
für alle $1 \leq i \leq n$ gilt
$C_i \in \mathcal{C}$ oder
es gibt $j, k < i$ mit
$C_i \in Res(C_j, C_k)$.

Resolvente.
C_1, C_2 Klauseln,

L Literal
mit $L \in C_1$, $\overline{L} \in C_2$.

Resolvente $C =$
$C_1 \setminus \{L\} \cup C_2 \setminus \{\overline{L}\}$.

Lemma.
Sei $\mathcal{C}$ Klauselmenge,
$C_1, C_2 \in \mathcal{C}$ und
$C \in Res(C_1, C_2)$.

Dann gilt:
$\{C_1, C_2\} \models C$ und
$\mathcal{C} \equiv \mathcal{C} \cup \{C\}$

Korrektheit der Resolution

Lemma. Sei $\mathcal{C}$ eine Klauselmenge und C eine Klausel.

Wenn $\mathcal{C} \vdash_R C$, dann $\mathcal{C} \models C$.

Beweis. Sei $(C_1, \dots, C_n)$ eine Resolutionsableitung von C aus $\mathcal{C}$.

Per Induktion über i zeigen wir, dass $\mathcal{C} \models C_i$ für alle $1 \leq i \leq n$.

Für $i = n$ folgt somit $\mathcal{C} \models C_n = C$.

Induktionsbasis: $i = 1$.

Es gilt $C_1 \in \mathcal{C}$ und somit $\mathcal{C} \models C_1$.

Induktionsschritt. Angenommen, die Behauptung gilt für $1, ..., i$.

Res.wid. von $\mathcal{C}$.
Sequenz $(C_1, \dots, C_n)$:
$C_n = C$ und
für alle $1 \leq i \leq n$ gilt
$C_i \in \mathcal{C}$ oder
es gibt $j, k < i$ mit
$C_i \in Res(C_j, C_k)$.

Resolvente.
C_1, C_2 Klauseln,

L Literal
mit $L \in C_1$, $\overline{L} \in C_2$.

Resolvente $C = C_1 \setminus \{L\} \cup C_2 \setminus \{\overline{L}\}$.

Lemma.
Sei $\mathcal{C}$ Klauselmenge,
$C_1, C_2 \in \mathcal{C}$ und
$C \in Res(C_1, C_2)$.

Dann gilt:
$\{C_1, C_2\} \models C$ und
$\mathcal{C} \equiv \mathcal{C} \cup \{C\}$

Korrektheit der Resolution

Lemma. Sei $\mathcal{C}$ eine Klauselmenge und C eine Klausel.

Wenn $\mathcal{C} \vdash_R C$, dann $\mathcal{C} \models C$.

Beweis. Sei $(C_1, \dots, C_n)$ eine Resolutionsableitung von C aus $\mathcal{C}$.

Per Induktion über i zeigen wir, dass $\mathcal{C} \models C_i$ für alle $1 \leq i \leq n$.

Für $i = n$ folgt somit $\mathcal{C} \models C_n = C$.

Induktionsbasis: $i = 1$.

Es gilt $C_1 \in \mathcal{C}$ und somit $\mathcal{C} \models C_1$.

Induktionsschritt. Angenommen, die Behauptung gilt für $1, \dots, i$.

Falls $C_{i+1} \in \mathcal{C}$, so gilt $\mathcal{C} \models C_{i+1}$.

Res.wid. von $\mathcal{C}$.
Sequenz $(C_1, \dots, C_n)$:
$C_n = C$ und
für alle $1 \leq i \leq n$ gilt
$C_i \in \mathcal{C}$ oder
es gibt $j, k < i$ mit
$C_i \in Res(C_j, C_k)$.

Resolvente.
C_1, C_2 Klauseln,

L Literal
mit $L \in C_1$, $\overline{L} \in C_2$.

Resolvente $C = C_1 \setminus \{L\} \cup C_2 \setminus \{\overline{L}\}$.

Lemma.
Sei $\mathcal{C}$ Klauselmenge,
$C_1, C_2 \in \mathcal{C}$ und
$C \in Res(C_1, C_2)$.

Dann gilt:
$\{C_1, C_2\} \models C$ und
$\mathcal{C} \equiv \mathcal{C} \cup \{C\}$

Korrektheit der Resolution

Lemma. Sei $\mathcal{C}$ eine Klauselmenge und C eine Klausel.

Wenn $\mathcal{C} \vdash_R C$, dann $\mathcal{C} \models C$.

Beweis. Sei $(C_1, \ldots, C_n)$ eine Resolutionsableitung von C aus $\mathcal{C}$.

Per Induktion über i zeigen wir, dass $\mathcal{C} \models C_i$ für alle $1 \leq i \leq n$.

Für $i = n$ folgt somit $\mathcal{C} \models C_n = C$.

Induktionsbasis: $i = 1$.

Es gilt $C_1 \in \mathcal{C}$ und somit $\mathcal{C} \models C_1$.

Induktionsschritt. Angenommen, die Behauptung gilt für $1, \ldots, i$.

Falls $C_{i+1} \in \mathcal{C}$, so gilt $\mathcal{C} \models C_{i+1}$.

Anderenfalls gibt es $j, k < i + 1$ mit $C_{i+1} \in Res(C_j, C_k)$.

Res.wid. von $\mathcal{C}$.
Sequenz $(C_1, \ldots, C_n)$:
$C_n = C$ und
für alle $1 \leq i \leq n$ gilt
$C_i \in \mathcal{C}$ oder
es gibt $j, k < i$ mit
$C_i \in Res(C_j, C_k)$.

Resolvente.
C_1, C_2 Klauseln,

L Literal
mit $L \in C_1$, $\overline{L} \in C_2$.

Resolvente $C =$
$C_1 \setminus \{L\} \cup C_2 \setminus \{\overline{L}\}$.

Lemma.
Sei $\mathcal{C}$ Klauselmenge,
$C_1, C_2 \in \mathcal{C}$ und
$C \in Res(C_1, C_2)$.

Dann gilt:
$\{C_1, C_2\} \models C$ und
$\mathcal{C} \equiv \mathcal{C} \cup \{C\}$

Korrektheit der Resolution

Lemma. Sei $\mathcal{C}$ eine Klauselmenge und C eine Klausel.

Wenn $\mathcal{C} \vdash_R C$, dann $\mathcal{C} \models C$.

Beweis. Sei $(C_1, \dots, C_n)$ eine Resolutionsableitung von C aus $\mathcal{C}$.

Per Induktion über i zeigen wir, dass $\mathcal{C} \models C_i$ für alle $1 \leq i \leq n$.

Für $i = n$ folgt somit $\mathcal{C} \models C_n = C$.

Induktionsbasis: $i = 1$.

Es gilt $C_1 \in \mathcal{C}$ und somit $\mathcal{C} \models C_1$.

Induktionsschritt. Angenommen, die Behauptung gilt für $1, ..., i$.

Falls $C_{i+1} \in \mathcal{C}$, so gilt $\mathcal{C} \models C_{i+1}$.

Anderenfalls gibt es $j, k < i + 1$ mit $C_{i+1} \in Res(C_j, C_k)$.

Aus der Induktionsannahme folgt $\mathcal{C} \models C_j$ und $\mathcal{C} \models C_k$.

Nach dem vorherigen Lemma gilt $C_j, C_k \models C_{i+1}$ und somit $\mathcal{C} \models C$. $\square$

Res.wid. von $\mathcal{C}$.
Sequenz $(C_1, \dots, C_n)$:
$C_n = C$ und
für alle $1 \leq i \leq n$ gilt
$C_i \in \mathcal{C}$ oder
es gibt $j, k < i$ mit
$C_i \in Res(C_j, C_k)$.

Resolvente.
C_1, C_2 Klauseln,

L Literal
mit $L \in C_1$, $\overline{L} \in C_2$.

Resolvente $C =$
$C_1 \setminus \{L\} \cup C_2 \setminus \{\overline{L}\}$.

Lemma.
Sei $\mathcal{C}$ Klauselmenge,
$C_1, C_2 \in \mathcal{C}$ und
$C \in Res(C_1, C_2)$.

Dann gilt:
$\{C_1, C_2\} \models C$ und
$\mathcal{C} \equiv \mathcal{C} \cup \{C\}$

Korrektheit des Resolutionskalküls

Lemma. Sei $\mathcal{C}$ eine Klauselmenge und C eine Klausel.

Wenn $\mathcal{C} \vdash_R C$, dann $\mathcal{C} \models C$.

Korollar. (Korrektheit des Resolutionskalküls)

Wenn eine Menge $\mathcal{C}$ von Klauseln eine Resolutionswiderlegung hat, dann ist $\mathcal{C}$ unerfüllbar.

Theorem.
Eine Menge $\mathcal{C}$ von Klauseln hat genau dann eine Resolutionswiderlegung, wenn $\mathcal{C}$ unerfüllbar ist. D.h. es gilt

$$\mathcal{C} \vdash_R \Box \quad \text{gdw.} \quad \mathcal{C} \text{ ist unerfüllbar} \quad \text{gdw.} \quad \mathcal{C} \models \bot$$

Res.wid. von $\mathcal{C}$.
Sequenz $(C_1, \dots, C_n)$:
$C_n = C$ und
für alle $1 \leq i \leq n$ gilt
- $C_i \in \mathcal{C}$ oder
- es gibt $j, k < i$ mit $C_i \in Res(C_j, C_k)$.

Der aussagenlogische Resolutionskalkül

Theorem. Eine Menge $\mathcal{C}$ von Klauseln hat genau dann eine Resolutionswiderlegung, wenn $\mathcal{C}$ unerfüllbar ist. D.h. es gilt

$$\mathcal{C} \vdash_R \square \quad \text{gdw.} \quad \mathcal{C} \text{ ist unerfüllbar} \quad \text{gdw.} \quad \mathcal{C} \models \perp$$

Lemma. (Korrektheit des Resolutionskalküls)
Wenn eine Menge $\mathcal{C}$ von Klauseln eine Resolutionswiderlegung hat, ist $\mathcal{C}$ unerfüllbar.

Lemma. (Vollständigkeit des Resolutionskalküls)
Jede unerfüllbare Klauselmenge $\mathcal{C}$ hat eine Resolutionswiderlegung.

Vollständigkeit und Korrektheit der Resolution

Lemma. (Vollständigkeit des Resolutionskalküls)
Jede unerfüllbare Klauselmenge $\mathcal{C}$ hat eine Resolutionswiderlegung.

Dazu beweisen wir zunächst die folgende Behauptung.

Aus der Behauptung folgt sofort die Vollständigkeit der Resolution für endliche Formelmengen.

Behauptung. *Sei $n \in \mathbb{N}_+$ und sei $\mathcal{C}$ eine unerfüllbare Klauselmenge in den Variablen $\{V_1, \dots, V_{n-1}\}$. Dann hat $\mathcal{C}$ eine Resolutionswiderlegung.*

Beweis der Behauptung

Behauptung. *Sei $n \in \mathbb{N}_+$ und sei $\mathcal{C}$ eine unerfüllbare Klauselmenge in den Variablen $\{V_1, \dots, V_{n-1}\}$. Dann hat $\mathcal{C}$ eine Resolutionswiderlegung.*

Beweis. Der Beweis der Behauptung wird per Induktion über n geführt.

Beweis der Behauptung

Behauptung. *Sei* $n \in \mathbb{N}_+$ *und sei* $\mathcal{C}$ *eine unerfüllbare Klauselmenge in den Variablen* $\{V_1, \ldots, V_{n-1}\}$*. Dann hat* $\mathcal{C}$ *eine Resolutionswiderlegung.*

Beweis. Der Beweis der Behauptung wird per Induktion über n geführt.

Induktionsbasis $n = 1$.

Wir wissen: $\mathcal{C}$ ist unerfüllbar und enthält keine Variablen.

Beweis der Behauptung

Behauptung. *Sei $n \in \mathbb{N}_+$ und sei $\mathcal{C}$ eine unerfüllbare Klauselmenge in den Variablen $\{V_1, \dots, V_{n-1}\}$. Dann hat $\mathcal{C}$ eine Resolutionswiderlegung.*

Beweis. Der Beweis der Behauptung wird per Induktion über n geführt.

Induktionsbasis $n = 1$.

Wir wissen: $\mathcal{C}$ ist unerfüllbar und enthält keine Variablen.

Es gibt nur zwei Klauselmengen ohne Variablen: $\{\}$ und $\{\Box\}$.

Beweis der Behauptung

Behauptung. *Sei $n \in \mathbb{N}_+$ und sei $\mathcal{C}$ eine unerfüllbare Klauselmenge in den Variablen $\{V_1, \ldots, V_{n-1}\}$. Dann hat $\mathcal{C}$ eine Resolutionswiderlegung.*

Beweis. Der Beweis der Behauptung wird per Induktion über n geführt.

Induktionsbasis $n = 1$.

Wir wissen: $\mathcal{C}$ ist unerfüllbar und enthält keine Variablen.

Es gibt nur zwei Klauselmengen ohne Variablen: $\{\}$ und $\{\Box\}$.

Nach Voraussetzung ist $\mathcal{C}$ unerfüllbar.

Also ist $\mathcal{C} := \{\Box\}$ und somit existiert eine Resolutionswiderlegung.

$\Box$

Beispiel einer Resolutionswiderlegung

Induktionsschritt $n \to n+1$.

Sei $\mathcal{C}$ eine unerfüllbare Klauselmenge in den Variablen $\{V_1, \dots, V_n\}$.

$$\mathcal{C} := \{V_1, V_2\}, \quad \{\neg V_2, V_n\}, \quad \{V_2, \neg V_1\}, \quad \{\neg V_n, \neg V_4\}, \quad \{\neg V_n, V_4, \neg V_5\}, \quad \{V_4, V_5\}$$

Beispiel einer Resolutionswiderlegung

Induktionsschritt $n \to n+1$.

Sei $\mathcal{C}$ eine unerfüllbare Klauselmenge in den Variablen $\{V_1, \dots, V_n\}$.

$$\mathcal{C} := \{V_1, V_2\}, \quad \{\neg V_2, V_n\}, \quad \{V_2, \neg V_1\}, \quad \{\neg V_n, \neg V_4\}\ , \quad \{\neg V_n, V_4, \neg V_5\}, \quad \{V_4, V_5\}$$

Resolutionswiderlegung.

$$\begin{tikzcd}
\{V_1, V_2\} \arrow[dr] & & \{\neg V_2, V_n\} \arrow[dl] \arrow[dr] & & \{V_2, \neg V_1\} \arrow[dl] & & \{\neg V_n, \neg V_4\} \arrow[ddr] & & \{\neg V_n, V_4, \neg V_5\} \arrow[dr] & & \{V_4, V_5\} \arrow[dl] \\
& \{V_1, V_n\} \arrow[dr] & & \{V_n, \neg V_1\} \arrow[dl] & & & & & & \{\neg V_n, V_4\} \arrow[dll] & \\
& & \{V_n\} \arrow[drrr] & & & & & \{\neg V_n\} \arrow[dll] & & & \\
& & & & & \square & & & & &
\end{tikzcd}$$

Beispiel einer Resolutionswiderlegung

Induktionsschritt $n \to n+1$.

Sei $\mathcal{C}$ eine unerfüllbare Klauselmenge in den Variablen $\{V_1, \dots, V_n\}$.

$$\mathcal{C} := \{V_1, V_2\}, \quad \{\neg V_2, V_n\}, \quad \{V_2, \neg V_1\}, \quad \{\neg V_n, \neg V_4\}, \quad \{\neg V_n, V_4, \neg V_5\}, \quad \{V_4, V_5\}$$

Resolutionswiderlegung.

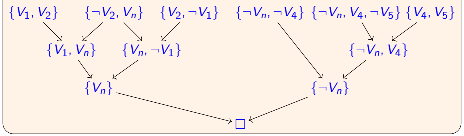
$$\begin{tikzcd}
\{V_1, V_2\} \arrow[dr] & & \{\neg V_2, V_n\} \arrow[dl] \arrow[dr] & & \{V_2, \neg V_1\} \arrow[dl] & & \{\neg V_n, \neg V_4\} \arrow[ddr] & & \{\neg V_n, V_4, \neg V_5\} \arrow[d] & \{V_4, V_5\} \arrow[dl] \\
& \{V_1, V_n\} \arrow[dr] & & \{V_n, \neg V_1\} \arrow[dl] & & & & & \{\neg V_n, V_4\} \arrow[dl] & \\
& & \{V_n\} \arrow[drrr] & & & & & \{\neg V_n\} \arrow[dll] & & \\
& & & & & \Box & & & &
\end{tikzcd}$$

Beweis der Behauptung

Induktionsschritt $n \rightarrow n+1$.

Sei $\mathcal{C}$ eine unerfüllbare Klauselmenge in den Variablen $\{V_1, \ldots, V_n\}$.

Resolutionswiderlegung.



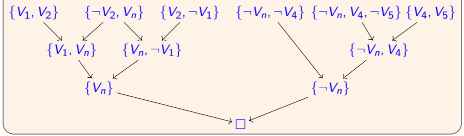
Beweis der Behauptung

Induktionsschritt $n \to n+1$.

Sei $\mathcal{C}$ eine unerfüllbare Klauselmenge in den Variablen $\{V_1, \dots, V_n\}$.

Definiere
$$\mathcal{C}^+ := \{C \setminus \{\neg V_n\} : C \in \mathcal{C} \text{ und } V_n \notin C\} \quad \hat{=} \quad [V_n \mapsto 1]$$
$$\mathcal{C}^- := \{C \setminus \{V_n\} : C \in \mathcal{C} \text{ und } \neg V_n \notin C\}. \quad \hat{=} \quad [V_n \mapsto 0]$$

Resolutionswiderlegung.



Beweis der Behauptung

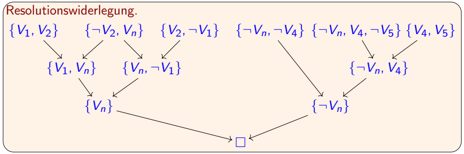
Induktionsschritt $n \to n+1$.

Sei $\mathcal{C}$ eine unerfüllbare Klauselmenge in den Variablen $\{V_1, \dots, V_n\}$.

Definiere
$$\mathcal{C}^+ := \{C \setminus \{\neg V_n\} : C \in \mathcal{C} \text{ und } V_n \notin C\}$$
$$\mathcal{C}^- := \{C \setminus \{V_n\} : C \in \mathcal{C} \text{ und } \neg V_n \notin C\}.$$

D.h., man erhält z.B. $\mathcal{C}^+$ indem

- alle Klauseln, die V_n enthalten entfernt werden und
- $\neg V_n$ aus den anderen entfernt wird.



Beweis der Behauptung

Induktionsschritt $n \to n+1$.

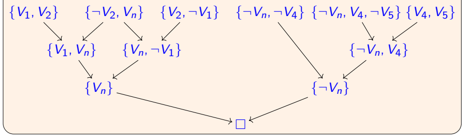
Sei $\mathcal{C}$ eine unerfüllbare Klauselmenge in den Variablen $\{V_1, \dots, V_n\}$.

Definiere
$$\mathcal{C}^+ := \{C \setminus \{\neg V_n\} : C \in \mathcal{C} \text{ und } V_n \notin C\}$$
$$\mathcal{C}^- := \{C \setminus \{V_n\} : C \in \mathcal{C} \text{ und } \neg V_n \notin C\}.$$

D.h., man erhält z.B. $\mathcal{C}^+$ indem

- alle Klauseln, die V_n enthalten entfernt werden und
- $\neg V_n$ aus den anderen entfernt wird.

Resolutionswiderlegung.



Behauptung. $\mathcal{C}^+$ und $\mathcal{C}^-$ sind beide unerfüllbar.

Beweis der Behauptung

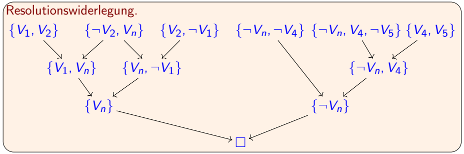
Induktionsschritt $n \to n+1$.

Sei $\mathcal{C}$ eine unerfüllbare Klauselmenge in den Variablen $\{V_1, \dots, V_n\}$.

Definiere
$$\mathcal{C}^+ := \{ C \setminus \{\neg V_n\} : C \in \mathcal{C} \text{ und } V_n \notin C \} \qquad \triangleq \qquad V_n \mapsto 1$$
$$\mathcal{C}^- := \{ C \setminus \{V_n\} : C \in \mathcal{C} \text{ und } \neg V_n \notin C \}.$$

D.h., man erhält z.B. $\mathcal{C}^+$ indem

- alle Klauseln, die V_n enthalten entfernt werden und
- $\neg V_n$ aus den anderen entfernt wird.



Behauptung. $\mathcal{C}^+$ und $\mathcal{C}^-$ sind beide unerfüllbar.

Beweis. Angenommen $\mathcal{C}^+$ wäre erfüllbar, z.B. durch $\beta \models \mathcal{C}^+$.

Dann würde $\beta' := \beta \cup \{V_n \mapsto 1\}$ die Menge $\mathcal{C}$ erfüllen.

Analog für $\mathcal{C}^-$. $\square$

Beispiel einer Resolutionswiderlegung

$\mathcal{C} := \{V_1, V_2\}, \quad \{\neg V_2, V_n\}, \quad \{V_2, \neg V_1\}, \quad \{\neg V_n, \neg V_4\}\ , \quad \{\neg V_n, V_4, \neg V_5\}, \quad \{V_4, V_5\}$

Die Menge $\mathcal{C}^-$.

$\{V_1, V_2\} \quad \{\neg V_2 \quad \} \quad \{V_2, \neg V_1\} \qquad \{V_4, V_5\}$

$\{V_1 \quad \} \quad \{ \quad \neg V_1\}$

$\Box$

Resolutionswiderlegung.

$\{V_1, V_2\} \quad \{\neg V_2, V_n\} \quad \{V_2, \neg V_1\} \quad \{\neg V_n, \neg V_4\} \quad \{\neg V_n, V_4, \neg V_5\} \quad \{V_4, V_5\}$

$\{V_1, V_n\} \quad \{V_n, \neg V_1\} \quad \{\neg V_n, V_4\}$

$\{V_n\} \quad \{\neg V_n\}$

$\Box$

Die Menge $\mathcal{C}^+$.

$\{V_1, V_2\} \quad \{V_2, \neg V_1\} \quad \{ \quad \neg V_4\} \quad \{ \quad V_4, \neg V_5\} \quad \{V_4, V_5\}$

$\{ \quad V_4\}$

$\Box$

Beispiel einer Resolutionswiderlegung

$\mathcal{C} := \{V_1, V_2\}, \quad \{\neg V_2, V_n\}, \quad \{V_2, \neg V_1\}, \quad \{\cancel{\neg V_n}, \neg V_4\}, \quad \{\neg V_n, V_4, \neg V_5\}, \quad \{V_4, V_5\}$

Die Menge $\mathcal{C}^-$.

$\{V_1, V_2\} \quad \{\neg V_2 \quad\} \quad \{V_2, \neg V_1\} \qquad\qquad \{V_4, V_5\}$

$\{V_1 \quad\} \qquad \{\quad \neg V_1\}$

$\square$

Resolutionswiderlegung.

$\{V_1, V_2\} \quad \{\neg V_2, V_n\} \quad \{V_2, \neg V_1\} \quad \{\neg V_n, \neg V_4\} \quad \{\neg V_n, V_4, \neg V_5\} \quad \{V_4, V_5\}$

$\{V_1, V_n\} \quad \{V_n, \neg V_1\} \qquad\qquad \{\neg V_n, V_4\}$

$\{V_n\} \qquad\qquad \{\neg V_n\}$

$\square$

Die Menge $\mathcal{C}^+$.

$\{V_1, V_2\} \qquad \{V_2, \neg V_1\} \quad \{\quad \neg V_4\} \quad \{\quad V_4, \neg V_5\} \quad \{V_4, V_5\}$

$C(C_1, \dots, C_s)$

$\{\quad V_4\}$

$\square$

Beispiel einer Resolutionswiderlegung

$\mathcal{C} := \{V_1, V_2\}, \quad \{\neg V_2, V_n\}, \quad \{V_2, \neg V_1\}, \quad \{\neg V_n, \neg V_4\}, \quad \{\neg V_n, V_4, \neg V_5\}, \quad \{V_4, V_5\}$

Die Menge $\mathcal{C}^-$.

$\{V_1, V_2\} \quad \{\neg V_2, V_n\} \quad \{V_2, \neg V_1\} \quad \{V_4, V_5\}$

$\{V_1, V_n\} \quad \{V_n, \neg V_1\}$

$\{V_n\}$

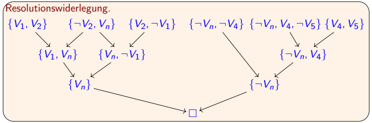
Die Menge $\mathcal{C}^+$.

$\{V_1, V_2\} \quad \{V_2, \neg V_1\} \quad \{\neg V_n, \neg V_4\} \quad \{\neg V_n, V_4, \neg V_5\} \quad \{V_4, V_5\}$

$\{\neg V_n, V_4\}$

$\{\neg V_n\}$

$\square$



Beweis der Behauptung (Forts.)

$n \to n+1$. Sei $\mathcal{C}$ eine unerfüllbare Klauselmenge in den Variablen $\{V_1, \dots, V_n\}$.

$\mathcal{C}^+ := \{C \setminus \{\neg V_n\} : C \in \mathcal{C} \text{ und } V_n \notin C\}$ und $\mathcal{C}^- := \{C \setminus \{V_n\} : C \in \mathcal{C} \text{ und } \neg V_n \notin C\}$.

Schon gezeigt. $\mathcal{C}^+$ und $\mathcal{C}^-$ sind beide unerfüllbar. □

Beweis der Behauptung (Forts.)

$n \to n+1$. Sei $\mathcal{C}$ eine unerfüllbare Klauselmenge in den Variablen $\{V_1, \dots, V_n\}$.

$\mathcal{C}^+ := \{C \setminus \{\neg V_n\} : C \in \mathcal{C}$ und $V_n \notin C\}$ und $\mathcal{C}^- := \{C \setminus \{V_n\} : C \in \mathcal{C}$ und $\neg V_n \notin C\}$.

Schon gezeigt. $\mathcal{C}^+$ und $\mathcal{C}^-$ sind beide unerfüllbar. $\square$

Nach IV ex. Res.-Abl. $(C_1, \dots, C_s)$ und $(D_1, \dots, D_t)$ von $C_s = D_t = \square$ aus $\mathcal{C}^+$ bzw. $\mathcal{C}^-$.

Beweis der Behauptung (Forts.)

$n \to n+1$. Sei $\mathcal{C}$ eine unerfüllbare Klauselmenge in den Variablen $\{V_1, \ldots, V_n\}$.

$\mathcal{C}^+ := \{C \setminus \{\neg V_n\} : C \in \mathcal{C}$ und $V_n \notin C\}$ und $\mathcal{C}^- := \{C \setminus \{V_n\} : C \in \mathcal{C}$ und $\neg V_n \notin C\}$.

Schon gezeigt. $\mathcal{C}^+$ und $\mathcal{C}^-$ sind beide unerfüllbar. $\square$

Nach IV ex. Res.-Abl. $(C_1, \ldots, C_s)$ und $(D_1, \ldots, D_t)$ von $C_s = D_t = \square$ aus $\mathcal{C}^+$ bzw. $\mathcal{C}^-$.

Falls $(C_1, \ldots, C_s)$ schon eine Ableitung von $\square$ aus $\mathcal{C}$ ist, sind wir fertig.

Beweis der Behauptung (Forts.)

$n \to n+1$. Sei $\mathcal{C}$ eine unerfüllbare Klauselmenge in den Variablen $\{V_1, \dots, V_n\}$.

$\mathcal{C}^+ := \{C \setminus \{\neg V_n\} : C \in \mathcal{C}$ und $V_n \notin C\}$ und $\mathcal{C}^- := \{C \setminus \{V_n\} : C \in \mathcal{C}$ und $\neg V_n \notin C\}$.

Schon gezeigt. $\mathcal{C}^+$ und $\mathcal{C}^-$ sind beide unerfüllbar. $\square$

Nach IV ex. Res.-Abl. $(C_1, \dots, C_s)$ und $(D_1, \dots, D_t)$ von $C_s = D_t = \square$ aus $\mathcal{C}^+$ bzw. $\mathcal{C}^-$.

Falls $(C_1, \dots, C_s)$ schon eine Ableitung von $\square$ aus $\mathcal{C}$ ist, sind wir fertig.

Wenn nicht, werden Klauseln C_i benutzt, die aus $\mathcal{C}$ durch Entfernen von $\neg V_n$ entstanden sind, d.h. $C_i \cup \{\neg V_n\} \in \mathcal{C}$.

Beweis der Behauptung (Forts.)

$n \to n+1$. Sei $\mathcal{C}$ eine unerfüllbare Klauselmenge in den Variablen $\{V_1, \dots, V_n\}$.
$\mathcal{C}^+ := \{C \setminus \{\neg V_n\} : C \in \mathcal{C}$ und $V_n \notin C\}$ und $\mathcal{C}^- := \{C \setminus \{V_n\} : C \in \mathcal{C}$ und $\neg V_n \notin C\}$.

Schon gezeigt. $\mathcal{C}^+$ und $\mathcal{C}^-$ sind beide unerfüllbar. $\square$

Nach IV ex. Res.-Abl. $(C_1, \dots, C_s)$ und $(D_1, \dots, D_t)$ von $C_s = D_t = \square$ aus $\mathcal{C}^+$ bzw. $\mathcal{C}^-$.

Falls $(C_1, \dots, C_s)$ schon eine Ableitung von $\square$ aus $\mathcal{C}$ ist, sind wir fertig.

Wenn nicht, werden Klauseln C_i benutzt, die aus $\mathcal{C}$ durch Entfernen von $\neg V_n$ entstanden sind, d.h. $C_i \cup \{\neg V_n\} \in \mathcal{C}$. Fügen wir zu diesen Klauseln und allen Resolventen wieder $\neg V_n$ hinzu, so erhalten wir eine Ableitung $(C_1', \dots, C_s')$ von $\{\neg V_n\}$ aus $\mathcal{C}$.

Beweis der Behauptung (Forts.)

$n \to n+1$. Sei $\mathcal{C}$ eine unerfüllbare Klauselmenge in den Variablen $\{V_1, \dots, V_n\}$.

$\mathcal{C}^+ := \{C \setminus \{\neg V_n\} : C \in \mathcal{C}$ und $V_n \notin C\}$ und $\mathcal{C}^- := \{C \setminus \{V_n\} : C \in \mathcal{C}$ und $\neg V_n \notin C\}$.

Schon gezeigt. $\mathcal{C}^+$ und $\mathcal{C}^-$ sind beide unerfüllbar. $\square$

Nach IV ex. Res.-Abl. $(C_1, \dots, C_s)$ und $(D_1, \dots, D_t)$ von $C_s = D_t = \square$ aus $\mathcal{C}^+$ bzw. $\mathcal{C}^-$.

Falls $(C_1, \dots, C_s)$ schon eine Ableitung von $\square$ aus $\mathcal{C}$ ist, sind wir fertig.

Wenn nicht, werden Klauseln C_i benutzt, die aus $\mathcal{C}$ durch Entfernen von $\neg V_n$ entstanden sind, d.h. $C_i \cup \{\neg V_n\} \in \mathcal{C}$. Fügen wir zu diesen Klauseln und allen Resolventen wieder $\neg V_n$ hinzu, so erhalten wir eine Ableitung $(C'_1, \dots, C'_s)$ von $\{\neg V_n\}$ aus $\mathcal{C}$.

Analog ist entweder $(D_1, \dots, D_t)$ bereits eine Ableitung von $\square$ aus $\mathcal{C}$ oder wir erhalten eine Ableitung $(D'_1, \dots, D'_t)$ von $\{V_n\}$ aus $\mathcal{C}$.

Beweis der Behauptung (Forts.)

$n \to n+1$. Sei $\mathcal{C}$ eine unerfüllbare Klauselmenge in den Variablen $\{V_1, \ldots, V_n\}$.

$\mathcal{C}^+ := \{C \setminus \{\neg V_n\} : C \in \mathcal{C} \text{ und } V_n \notin C\}$ und $\mathcal{C}^- := \{C \setminus \{V_n\} : C \in \mathcal{C} \text{ und } \neg V_n \notin C\}$.

Schon gezeigt. $\mathcal{C}^+$ und $\mathcal{C}^-$ sind beide unerfüllbar. $\square$

Nach IV ex. Res.-Abl. $(C_1, \ldots, C_s)$ und $(D_1, \ldots, D_t)$ von $C_s = D_t = \square$ aus $\mathcal{C}^+$ bzw. $\mathcal{C}^-$.

Falls $(C_1, \ldots, C_s)$ schon eine Ableitung von $\square$ aus $\mathcal{C}$ ist, sind wir fertig.

Wenn nicht, werden Klauseln C_i benutzt, die aus $\mathcal{C}$ durch Entfernen von $\neg V_n$ entstanden sind, d.h. $C_i \cup \{\neg V_n\} \in \mathcal{C}$. Fügen wir zu diesen Klauseln und allen Resolventen wieder $\neg V_n$ hinzu, so erhalten wir eine Ableitung $(C'_1, \ldots, C'_s)$ von $\{\neg V_n\}$ aus $\mathcal{C}$.

Analog ist entweder $(D_1, \ldots, D_t)$ bereits eine Ableitung von $\square$ aus $\mathcal{C}$ oder wir erhalten eine Ableitung $(D'_1, \ldots, D'_t)$ von $\{V_n\}$ aus $\mathcal{C}$.

Ein weiterer Resolutionsschritt auf $\{\neg V_n\}$ und $\{V_n\}$ ergibt dann $\square$.
Also ist $(C'_1, \ldots, C'_s, D'_1, \ldots, D'_t, \square)$ eine Resolutionswiderlegung von $\mathcal{C}$. $\square$

Vollständigkeit des Resolutionskalküls

Behauptung. *Sei $n \in \mathbb{N}_+$ und sei $\mathcal{C}$ eine unerfüllbare Klauselmenge in den Variablen $\{V_1, \dots, V_{n-1}\}$. Dann hat $\mathcal{C}$ eine Resolutionswiderlegung.*

Lemma. (Vollständigkeit des Resolutionskalküls im Endlichen)
Jede unerfüllbare endliche Klauselmenge $\mathcal{C}$ hat eine Resolutionswiderlegung.

Beweis. Sei $\mathcal{C}$ eine unerfüllbare endliche Klauselmenge.

Da $\mathcal{C}$ endlich, enthält $\mathcal{C}$ nur endlich viele Variablen.

O.B.d.A. benutzt $\mathcal{C}$ nur Variablen aus der Menge $\{V_1, \dots, V_n\}$ für ein $n \in \mathbb{N}$.

Der Beweis folgt damit sofort aus der Behauptung. $\square$

Vollständigkeit des Resolutionskalküls

Behauptung. *Sei $n \in \mathbb{N}_+$ und sei $\mathcal{C}$ eine unerfüllbare Klauselmenge in den Variablen $\{V_1, \ldots, V_{n-1}\}$. Dann hat $\mathcal{C}$ eine Resolutionswiderlegung.*

Lemma. (Vollständigkeit des Resolutionskalküls im Endlichen)
Jede unerfüllbare endliche Klauselmenge $\mathcal{C}$ hat eine Resolutionswiderlegung.

Beweis. Sei $\mathcal{C}$ eine unerfüllbare endliche Klauselmenge.

Da $\mathcal{C}$ endlich, enthält $\mathcal{C}$ nur endlich viele Variablen.

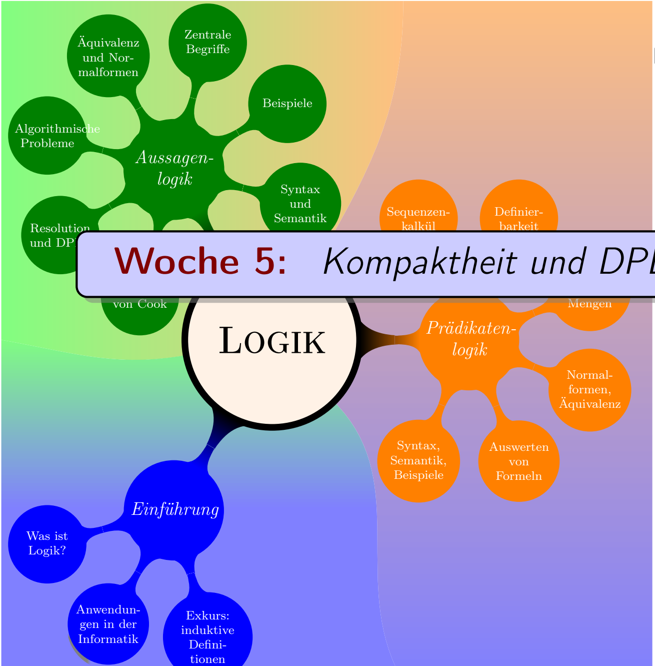
O.B.d.A. benutzt $\mathcal{C}$ nur Variablen aus der Menge $\{V_1, \ldots, V_n\}$ für ein $n \in \mathbb{N}$.

Der Beweis folgt damit sofort aus der Behauptung. $\square$

Vollständigkeit im Unendlichen? Für unendliche Klauselmengen ist die bewiesene Behauptung nicht stark genug.

Hierzu brauchen wir noch den Kompaktheitssatz.

Woche 5: *Kompaktheit und DPLL*



Logik im Wintersemester 2022/2023

Nov.	17	18	19	20	21	22	23	*Einführung*
	24	25	26	27	28	29	30	*Aussagenlogik*
	31	1	2	3	4	5	6	*Normalformen*
	7	8	9	10	11	12	13	*Resolution*
Dez.	14	15	16	17	18	19	20	*Kompaktheit*
	21	22	23	24	25	26	27	*DPLL, Satz von Cook*
	28	29	30	1	2	3	4	*Strukturen und FO*
	5	6	7	8	9	10	11	*Prädikatenlogik*
	12	13	14	15	16	17	18	*Komplexität von FO*
Jan.	19	20	21	22	23	24	25	*Weihnachten*
	26	27	28	29	30	31	1	*Neujahr*
	2	3	4	5	6	7	8	*Normalformen*
	9	10	11	12	13	14	15	*Definierbarkeit*
	16	17	18	19	20	21	22	*EF-Spiele*
Feb.	23	24	25	26	27	28	29	*EF-Spiele*
	30	31	1	2	3	4	5	*Sequenzenkalkül AL*
	6	7	8	9	10	11	12	*Sequenzenkalkül FO*
	13	14	15	16	17	18	19	*Ausblick*

5.1 Der Kompaktheitssatz

Der Kompaktheitssatz der Aussagenlogik

Theorem. (Kompaktheits- oder Endlichkeitssatz)

Sei $\Phi \subseteq \mathrm{AL}$ eine Formelmenge und $\psi \in \mathrm{AL}$ eine Formel.

1. Φ ist erfüllbar gdw. jede endliche Teilmenge $\Phi' \subseteq \Phi$ erfüllbar ist.
2. $\Phi \models \psi$ gdw. eine endliche Teilmenge $\Phi_0 \subseteq \Phi$ existiert mit $\Phi_0 \models \psi$.

Erinnerung.

Eine Menge Φ aussagenlogischer Formeln ist erfüllbar, wenn es eine Belegung β gibt, die zu allen $\varphi \in \Phi$ passt und alle $\varphi \in \Phi$ erfüllt.

Wir schreiben $\beta \models \Phi$.

Der Kompaktheitssatz der Aussagenlogik

Theorem. (Kompaktheits- oder Endlichkeitssatz)

Sei $\Phi \subseteq \mathsf{AL}$ eine Formelmenge und $\psi \in \mathsf{AL}$ eine Formel.

1. Φ ist erfüllbar gdw. jede endliche Teilmenge $\Phi' \subseteq \Phi$ erfüllbar ist.
2. $\Phi \models \psi$ gdw. eine endliche Teilmenge $\Phi_0 \subseteq \Phi$ existiert mit $\Phi_0 \models \psi$.

Bemerkung. Der Satz kann auch für überabzählbare Variablenmengen und somit überabzählbare Formelmengen bewiesen werden.

Erinnerung.

Eine Menge Φ aussagenlogischer Formeln ist erfüllbar, wenn es eine Belegung β gibt, die zu allen $\varphi \in \Phi$ passt und alle $\varphi \in \Phi$ erfüllt.

Wir schreiben $\beta \models \Phi$.

Beweis des Satzes

Wir werden zunächst den ersten Teil des Satzes beweisen, d.h.

Behauptung. Eine Menge $\Phi \subseteq \mathsf{AL}$ ist genau dann erfüllbar, wenn jede endliche Teilmenge $\Phi' \subseteq \Phi$ erfüllbar ist.

Kompaktheitssatz

Sei $\Phi \subseteq \mathsf{AL}$, $\psi \in \mathsf{AL}$.

1. Φ erfüllbar gdw. alle endlichen $\Phi' \subseteq \Phi$ erfüllbar.
2. $\Phi \models \psi$ gdw. es ex. endl. $\Phi_0 \subseteq \Phi$ mit $\Phi_0 \models \psi$.

Beweis des Satzes

Wir werden zunächst den ersten Teil des Satzes beweisen, d.h.

Behauptung. Eine Menge $\Phi \subseteq \mathrm{AL}$ ist genau dann erfüllbar, wenn jede endliche Teilmenge $\Phi' \subseteq \Phi$ erfüllbar ist.

Vorüberlegungen.

1. Offenbar ist die Behauptung trivial, wenn Φ bereits endlich ist.

Sei Φ also eine unendliche Menge aussagenlogischer Formeln.

Kompaktheitssatz

Sei $\Phi \subseteq \mathrm{AL}$, $\psi \in \mathrm{AL}$.

1. Φ erfüllbar gdw. alle endlichen $\Phi' \subseteq \Phi$ erfüllbar.
2. $\Phi \models \psi$ gdw. es ex. endl. $\Phi_0 \subseteq \Phi$ mit $\Phi_0 \models \psi$.

Beweis des Satzes

Wir werden zunächst den ersten Teil des Satzes beweisen, d.h.

Behauptung. Eine Menge $\Phi \subseteq \mathsf{AL}$ ist genau dann erfüllbar, wenn jede endliche Teilmenge $\Phi' \subseteq \Phi$ erfüllbar ist.

Vorüberlegungen.

1. Offenbar ist die Behauptung trivial, wenn Φ bereits endlich ist.

Sei Φ also eine unendliche Menge aussagenlogischer Formeln.

2. O.B.d.A. nehmen wir an, dass es keine zwei verschiedenen Formeln $\psi, \psi' \in \Phi$ gibt, so dass $\psi \equiv \psi'$.

   Denn, angenommen, es gäbe solche Formeln. Dann ist Φ genau dann erfüllbar, wenn $\Phi \setminus \{\psi'\}$ erfüllbar ist.

Kompaktheitssatz

Sei $\Phi \subseteq \mathsf{AL}$, $\psi \in \mathsf{AL}$.

1. Φ erfüllbar gdw. alle endlichen $\Phi' \subseteq \Phi$ erfüllbar.
2. $\Phi \models \psi$ gdw. es ex. endl. $\Phi_0 \subseteq \Phi$ mit $\Phi_0 \models \psi$.

Beweis des Satzes

Wir werden zunächst den ersten Teil des Satzes beweisen, d.h.

Behauptung. Eine Menge $\Phi \subseteq \text{AL}$ ist genau dann erfüllbar, wenn jede endliche Teilmenge $\Phi' \subseteq \Phi$ erfüllbar ist.

Vorüberlegungen.

1. Offenbar ist die Behauptung trivial, wenn Φ bereits endlich ist.

   Sei Φ also eine unendliche Menge aussagenlogischer Formeln.

2. O.B.d.A. nehmen wir an, dass es keine zwei verschiedenen Formeln $\psi, \psi' \in \Phi$ gibt, so dass $\psi \equiv \psi'$.

   Denn, angenommen, es gäbe solche Formeln. Dann ist Φ genau dann erfüllbar, wenn $\Phi \setminus \{\psi'\}$ erfüllbar ist.

3. Es reicht also, den Beweis für unendliche Formelmengen zu zeigen, in denen alle Formeln paarweise nicht äquivalent sind.

Kompaktheitssatz

Sei $\Phi \subseteq \text{AL}$, $\psi \in \text{AL}$.

1. Φ erfüllbar gdw. alle endlichen $\Phi' \subseteq \Phi$ erfüllbar.
2. $\Phi \models \psi$ gdw. es ex. endl. $\Phi_0 \subseteq \Phi$ mit $\Phi_0 \models \psi$.

Beweis des Satzes: Hinrichtung

Behauptung. Eine Menge $\Phi \subseteq \mathsf{AL}$ ist genau dann erfüllbar, wenn jede endliche Teilmenge $\Phi' \subseteq \Phi$ erfüllbar ist.

Hinrichtung.

Wir nehmen an, dass Φ erfüllbar ist.

Dann existiert eine Belegung β, die jede Formel in Φ erfüllt.

Also erfüllt β auch jede endliche Teilmenge von Φ.

Es folgt, dass jede endliche Teilmenge von Φ erfüllbar ist. $\dashv$

Kompaktheitssatz

Sei $\Phi \subseteq \mathsf{AL}$, $\psi \in \mathsf{AL}$.

1. Φ erfüllbar gdw. alle endlichen $\Phi' \subseteq \Phi$ erfüllbar.
2. $\Phi \models \psi$ gdw. es ex. endl. $\Phi_0 \subseteq \Phi$ mit $\Phi_0 \models \psi$.

Beweis des Satzes: Rückrichtung

Behauptung. Eine Menge $\Phi \subseteq \mathrm{AL}$ ist genau dann erfüllbar, wenn jede endliche Teilmenge $\Phi' \subseteq \Phi$ erfüllbar ist.

Rückrichtung. Angenommen, alle endlichen Teilmengen $\Phi' \subseteq \Phi$ sind erfüllbar.

Seien $X_1, X_2, \dots$ die in Formeln in Φ vorkommenden Aussagenvariablen.

Für all $n \geq 0$ definieren wir

$$\Phi_n := \big\{\varphi \in \Phi : \mathrm{var}(\varphi) \subseteq \{X_1, \dots, X_n\}\big\}.$$

Φ_n enthält also alle Formeln aus Φ in denen nur die Variablen $X_1, \dots, X_n$ vorkommen (es müssen aber nicht alle vorkommen).

Beweis des Satzes: Rückrichtung

Behauptung. Eine Menge $\Phi \subseteq \mathrm{AL}$ ist genau dann erfüllbar, wenn jede endliche Teilmenge $\Phi' \subseteq \Phi$ erfüllbar ist.

Rückrichtung. Angenommen, alle endlichen Teilmengen $\Phi' \subseteq \Phi$ sind erfüllbar.

Seien $X_1, X_2, \dots$ die in Formeln in Φ vorkommenden Aussagenvariablen.

Für all $n \geq 0$ definieren wir

$$\Phi_n := \{\varphi \in \Phi : \mathrm{var}(\varphi) \subseteq \{X_1, \dots, X_n\}\}.$$

Φ_n enthält also alle Formeln aus Φ in denen nur die Variablen $X_1, \dots, X_n$ vorkommen (es müssen aber nicht alle vorkommen).

Beobachtung. Für alle i gilt $\Phi_i \subseteq \Phi_{i+1} \subseteq \Phi$

$\Phi_0 \subseteq \Phi_1 \subseteq ... \subseteq \Phi$.

Beweis des Satzes: Rückrichtung

$\Phi_n := \{\varphi \in \Phi : \mathrm{var}(\varphi) \subseteq \{X_1, \dots, X_n\}\}$

Rückrichtung.
Wenn alle endl. $\Phi' \subseteq \Phi$ erfüllbar, dann Φ erfüllbar.

Bereits gesehen.

- Es gibt $\leq 2^{2^n}$ paarweise nicht-äquivalente Formeln in n Variablen.
- Da Φ keine paarweise äquivalenten Formeln enthält, gilt $|\Phi_n| \leq 2^{2^n}$.

Nach Voraussetzung gibt es also für jedes $n \geq 0$ eine Belegung β_n mit $\beta_n \models \Phi_n$ und somit auch $\beta_n \models \Phi_i$ für alle $i \leq n$.

Sei $I_0 := \{\beta_n : n \geq 0\}$.

Wir konstruieren induktiv eine Belegung $\alpha : \{X_1, \dots\} \to \{0, 1\}$ und Mengen $I_n \subseteq I_0$, so dass für alle n folgende Eigenschaft $(\star)$ gilt:

Eigenschaft $(\star)$.

- I_n ist unendlich, $\beta_0, \dots, \beta_{n-1} \notin I_n$ und
- für alle $\beta, \beta' \in I_n$ und $j \leq n$ gilt $\beta(X_j) = \beta'(X_j) = \alpha(X_j)$.

$\beta_1(X_1) = 0$

β_1

$\Phi_0 \subseteq \Phi_1 \subseteq \Phi_2 \dots$

$\beta_0 \quad \beta_2 \quad \beta_2$

$\beta \not\models \Phi_3$

$\beta_2(X_1) = 1$

Beweis des Kompaktheitssatzes

	X_1	X_2	X_3	X_4	X_5	X_6	X_7	...
β_1	0	1						
β_2	0	1	1	1	0	0	1	...
β_3	1	1	1	1	0	0	1	...
β_4	0	1	1	1	0	0	1	...
β_5	0	1	1	0	0	0	1	...
β_6	0	0	1	1	0	0	1	...
β_7	0	1	1	1	0	0	1	...
β_8	0	1	1	0	0	0	1	...
β_9	1	0	1	1	0	0	1	...
β_{10}	0	1	0	0	0	0	1	...
β_{11}	0	1	1	0	0	0	1	...
$\vdots$	...	...	...	...	...	...	...	...
α								

Konstr. $\alpha : \{X_1, ...\} \to \{0, 1\}$ und $I_n \subseteq I_0$, $n \geq 1$:

Für alle n gilt Eigenschaft $(*)$:

- I_n ist unendlich, $\beta_1, \dots, \beta_{n-1} \notin I_n$ und
- für alle $\beta, \beta' \in I_n$ und $j \leq n$ gilt
  $\beta(X_j) = \beta'(X_j) = \alpha(X_j)$.

Beweis des Kompaktheitssatzes

	X_1	X_2	X_3	X_4	X_5	X_6	X_7	...
β_1	0	1						
β_2	0	1	1	1	0	0	1	...
β_3	1	1	1	1	0	0	1	...
β_4	0	1	1	1	0	0	1	...
β_5	0	1	1	0	0	0	1	...
β_6	0	0	1	1	0	0	1	...
β_7	0	1	1	1	0	0	1	...
β_8	0	1	1	0	0	0	1	...
β_9	1	0	1	1	0	0	1	...
β_{10}	0	1	0	0	0	0	1	...
β_{11}	0	1	1	0	0	0	1	...
$\vdots$	...	...	...	...	...	...	...	...
α								

Konstr. $\alpha : \{X_1, ...\} \to \{0, 1\}$ und $I_n \subseteq I_0$, $n \geq 1$:

Für alle n gilt Eigenschaft $(*)$:

- I_n ist unendlich, $\beta_1, \dots, \beta_{n-1} \notin I_n$ und
- für alle $\beta, \beta' \in I_n$ und $j \leq n$ gilt
  $\beta(X_j) = \beta'(X_j) = \alpha(X_j)$.

Induktionsbasis $n = 1$.

Da Φ I_0 unendlich, existiert ein $t \in \{0, 1\}$ so dass $\beta_n(X_1) = t$ für unendlich viele $\beta_n \in I_0$.

Setze

$\alpha(X_1) := t$ und

$I_1 := \{\beta \in I_0 : \beta(X_1) = t \text{ und } \beta \neq \beta_1\}$.

Offenbar ist $(\star)$ erfüllt.

Beweis des Kompaktheitssatzes

	X_1	X_2	X_3	X_4	X_5	X_6	X_7	$\dots$
~~β_1~~	~~0~~	~~1~~						
β_2	0	1	1	1	0	0	1	$\dots$
~~β_3~~	~~1~~	~~1~~	~~1~~	~~1~~	~~0~~	~~0~~	~~1~~	~~$\dots$~~
β_4	0	1	1	1	0	0	1	$\dots$
β_5	0	1	1	0	0	0	1	$\dots$
β_6	0	0	1	1	0	0	1	$\dots$
β_7	0	1	1	1	0	0	1	$\dots$
β_8	0	1	1	0	0	0	1	$\dots$
~~β_9~~	~~1~~	~~0~~	~~1~~	~~1~~	~~0~~	~~0~~	~~1~~	~~$\dots$~~
β_{10}	0	1	0	0	0	0	1	$\dots$
β_{11}	0	1	1	0	0	0	1	$\dots$
$\vdots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$
α	0							

$(X_3 \lor X_4) \land (- -)$

Konstr. $\alpha : \{X_1, ...\} \to \{0,1\}$ und $I_n \subseteq I_0$, $n \geq 1$:

Für alle n gilt Eigenschaft $(*)$:

- I_n ist unendlich, $\beta_1, \dots, \beta_{n-1} \notin I_n$ und
- für alle $\beta, \beta' \in I_n$ und $j \leq n$ gilt
  $\beta(X_j) = \beta'(X_j) = \alpha(X_j)$.

Induktionsbasis $n = 1$.

Da Φ unendlich, existiert ein $t \in \{0,1\}$ so dass $\beta_n(X_1) = t$ für unendlich viele $\beta_n \in I_0$.

Setze

$\alpha(X_1) := t$ und

$I_1 := \{\beta \in I_0 : \beta(X_1) = t \text{ und } \beta \neq \beta_1\}$.

Offenbar ist $(\star)$ erfüllt.

$\varphi \in \Phi_3$

$\varphi = X_3 \land (X_4 \lor \dots)$

Beweis des Kompaktheitssatzes

	X_1	X_2	X_3	X_4	X_5	X_6	X_7	...
~~β_1~~	~~0~~	~~1~~						
β_2	0	1	1	1	0	0	1	...
~~β_3~~	~~1~~	~~1~~	~~1~~	~~1~~	~~0~~	~~0~~	~~1~~	~~...~~
β_4	0	1	1	1	0	0	1	...
β_5	0	1	1	0	0	0	1	...
β_6	0	0	1	1	0	0	1	...
β_7	0	1	1	1	0	0	1	...
β_8	0	1	1	0	0	0	1	...
~~β_9~~	~~1~~	~~0~~	~~1~~	~~1~~	~~0~~	~~0~~	~~1~~	~~...~~
β_{10}	0	1	0	0	0	0	1	...
β_{11}	0	1	1	0	0	0	1	...
$\vdots$	...	...	...	...	...	...	...	...
α	0							

Konstr. $\alpha : \{X_1, ...\} \to \{0, 1\}$ und $I_n \subseteq I_0$, $n \geq 1$:

Für alle n gilt Eigenschaft $(*)$:

- I_n ist unendlich, $\beta_1, \dots, \beta_{n-1} \notin I_n$ und
- für alle $\beta, \beta' \in I_n$ und $j \leq n$ gilt
  $\beta(X_j) = \beta'(X_j) = \alpha(X_j)$.

Induktionsvoraussetzung. Für $i < n$ seien $I_i, \alpha(X_i)$ schon konstruiert so dass $(\star)$ gilt.

Beweis des Kompaktheitssatzes

	X_1	X_2	X_3	X_4	X_5	X_6	X_7	...
~~β_1~~	~~0~~	~~1~~						
β_2	0	1	1	1	0	0	1	...
~~β_3~~	~~1~~	~~1~~	~~1~~	~~1~~	~~0~~	~~0~~	~~1~~	~~...~~
β_4	0	1	1	1	0	0	1	...
β_5	0	1	1	0	0	0	1	...
β_6	0	0	1	1	0	0	1	...
β_7	0	1	1	1	0	0	1	...
β_8	0	1	1	0	0	0	1	...
~~β_9~~	~~1~~	~~0~~	~~1~~	~~1~~	~~0~~	~~0~~	~~1~~	~~...~~
β_{10}	0	1	0	0	0	0	1	...
β_{11}	0	1	1	0	0	0	1	...
$\vdots$	...	...	...	...	...	...	...	...
α	0							

Konstr. $\alpha : \{X_1, ...\} \to \{0,1\}$ und $I_n \subseteq I_0$, $n \geq 1$:

Für alle n gilt Eigenschaft $(*)$:

- I_n ist unendlich, $\beta_1, \dots, \beta_{n-1} \notin I_n$ und
- für alle $\beta, \beta' \in I_n$ und $j \leq n$ gilt
  $\beta(X_j) = \beta'(X_j) = \alpha(X_j)$.

Induktionsvoraussetzung. Für $i < n$ seien $I_i, \alpha(X_i)$ schon konstruiert so dass $(\star)$ gilt.

Induktionsschritt. Da wegen $(*)$ I_{n-1} unendlich ist, gibt es ein $t \in \{0,1\}$, so dass $\beta(X_n) = t$ für unendlich viele $\beta \in I_{n-1}$.

Setze

$\alpha(X_n) := t$ und

$I_n := \{\beta \in I_{n-1} : \beta(X_n) = t \text{ und } \beta \neq \beta_{n-1}\}$.

Offenbar ist $(\star)$ erfüllt.

Beweis des Kompaktheitssatzes

	X_1	X_2	X_3	X_4	X_5	X_6	X_7	$\dots$
~~β_1~~	~~0~~	~~1~~						
~~β_2~~	~~0~~	~~1~~	~~1~~	~~1~~	~~0~~	~~0~~	~~1~~	~~$\dots$~~
~~β_3~~	~~1~~	~~1~~	~~1~~	~~1~~	~~0~~	~~0~~	~~1~~	~~$\dots$~~
β_4	0	1	1	1	0	0	1	$\dots$
β_5	0	1	1	0	0	0	1	$\dots$
~~β_6~~	~~0~~	~~0~~	~~1~~	~~1~~	~~0~~	~~0~~	~~1~~	~~$\dots$~~
β_7	0	1	1	1	0	0	1	$\dots$
β_8	0	1	1	0	0	0	1	$\dots$
~~β_9~~	~~1~~	~~0~~	~~1~~	~~1~~	~~0~~	~~0~~	~~1~~	~~$\dots$~~
β_{10}	0	1	0	0	0	0	1	$\dots$
β_{11}	0	1	1	0	0	0	1	$\dots$
$\vdots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$
α	0	1						$\dots$

Konstr. $\alpha : \{X_1, \dots\} \to \{0,1\}$ und $I_n \subseteq I_0$, $n \geq 1$:

Für alle n gilt Eigenschaft $(*)$:

- I_n ist unendlich, $\beta_1, \dots, \beta_{n-1} \notin I_n$ und
- für alle $\beta, \beta' \in I_n$ und $j \leq n$ gilt
  $\beta(X_j) = \beta'(X_j) = \alpha(X_j)$.

Induktionsvoraussetzung. Für $i < n$ seien $I_i, \alpha(X_i)$ schon konstruiert so dass $(\star)$ gilt.

Induktionsschritt. Da wegen $(*)$ I_{n-1} unendlich ist, gibt es ein $t \in \{0,1\}$, so dass $\beta(X_n) = t$ für unendlich viele $\beta \in I_{n-1}$.

Setze

$\alpha(X_n) := t$ und

$I_n := \{\beta \in I_{n-1} : \beta(X_n) = t \text{ und } \beta \neq \beta_{n-1}\}$.

Offenbar ist $(\star)$ erfüllt.

Beweis des Kompaktheitssatzes

	X_1	X_2	X_3	X_4	X_5	X_6	X_7	...
~~β_1~~	~~0~~	~~1~~						
~~β_2~~	~~0~~	~~1~~	~~1~~	~~1~~	~~0~~	~~0~~	~~1~~	~~...~~
~~β_3~~	~~1~~	~~1~~	~~1~~	~~1~~	~~0~~	~~0~~	~~1~~	~~...~~
~~β_4~~	~~0~~	~~1~~	~~1~~	~~1~~	~~0~~	~~0~~	~~1~~	~~...~~
β_5	0	1	1	0	0	0	1	...
~~β_6~~	~~0~~	~~0~~	~~1~~	~~1~~	~~0~~	~~0~~	~~1~~	~~...~~
β_7	0	1	1	1	0	0	1	...
β_8	0	1	1	0	0	0	1	...
~~β_9~~	~~1~~	~~0~~	~~1~~	~~1~~	~~0~~	~~0~~	~~1~~	~~...~~
~~β_{10}~~	~~0~~	~~1~~	~~0~~	~~0~~	~~0~~	~~0~~	~~1~~	~~...~~
β_{11}	0	1	1	0	0	0	1	...
$\vdots$	...	...	...	...	...	...	...	...
α	0	1	1					...

Konstr. $\alpha : \{X_1, ...\} \rightarrow \{0, 1\}$ und $I_n \subseteq I_0$, $n \geq 1$:

Für alle n gilt Eigenschaft $(*)$:

- I_n ist unendlich, $\beta_1, \ldots, \beta_{n-1} \notin I_n$ und
- für alle $\beta, \beta' \in I_n$ und $j \leq n$ gilt
  $\beta(X_j) = \beta'(X_j) = \alpha(X_j)$.

Induktionsvoraussetzung. Für $i < n$ seien $I_i, \alpha(X_i)$ schon konstruiert so dass $(\star)$ gilt.

Induktionsschritt. Da wegen $(*)$ I_{n-1} unendlich ist, gibt es ein $t \in \{0, 1\}$, so dass $\beta(X_n) = t$ für unendlich viele $\beta \in I_{n-1}$.

Setze

$\alpha(X_n) := t$ und

$I_n := \{\beta \in I_{n-1} : \beta(X_n) = t$ und $\beta \neq \beta_{n-1}\}$.

Offenbar ist $(\star)$ erfüllt.

Beweis des Kompaktheitssatzes

	X_1	X_2	X_3	X_4	X_5	X_6	X_7	$\dots$
$\cancel{\beta_1}$	~~0~~	~~1~~						
$\cancel{\beta_2}$	~~0~~	~~1~~	~~1~~	~~1~~	~~0~~	~~0~~	~~1~~	~~$\dots$~~
$\cancel{\beta_3}$	~~1~~	~~1~~	~~1~~	~~1~~	~~0~~	~~0~~	~~1~~	~~$\dots$~~
$\cancel{\beta_4}$	~~0~~	~~1~~	~~1~~	~~1~~	~~0~~	~~0~~	~~1~~	~~$\dots$~~
$\cancel{\beta_5}$	~~0~~	~~1~~	~~1~~	~~0~~	~~0~~	~~0~~	~~1~~	~~$\dots$~~
$\cancel{\beta_6}$	~~0~~	~~0~~	~~1~~	~~1~~	~~0~~	~~0~~	~~1~~	~~$\dots$~~
$\cancel{\beta_7}$	~~0~~	~~1~~	~~1~~	~~1~~	~~0~~	~~0~~	~~1~~	~~$\dots$~~
β_8	0	1	1	0	0	0	1	$\dots$
$\cancel{\beta_9}$	~~1~~	~~0~~	~~1~~	~~1~~	~~0~~	~~0~~	~~1~~	~~$\dots$~~
$\cancel{\beta_{10}}$	~~0~~	~~1~~	~~0~~	~~0~~	~~0~~	~~0~~	~~1~~	~~$\dots$~~
β_{11}	0	1	1	0	0	0	1	$\dots$
$\vdots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$
α	0	1	1	0				$\dots$

Konstr. $\alpha : \{X_1, \dots\} \to \{0, 1\}$ und $I_n \subseteq I_0$, $n \geq 1$:

Für alle n gilt Eigenschaft $(*)$:

- I_n ist unendlich, $\beta_1, \dots, \beta_{n-1} \notin I_n$ und
- für alle $\beta, \beta' \in I_n$ und $j \leq n$ gilt
  $\beta(X_j) = \beta'(X_j) = \alpha(X_j)$.

Induktionsvoraussetzung. Für $i < n$ seien $I_i, \alpha(X_i)$ schon konstruiert so dass $(\star)$ gilt.

Induktionsschritt. Da wegen $(*)$ I_{n-1} unendlich ist, gibt es ein $t \in \{0, 1\}$, so dass $\beta(X_n) = t$ für unendlich viele $\beta \in I_{n-1}$.

Setze

$\alpha(X_n) := t$ und

$I_n := \{\beta \in I_{n-1} : \beta(X_n) = t$ und $\beta \neq \beta_{n-1}\}$.

Offenbar ist $(\star)$ erfüllt.

Beweis des Kompaktheitssatzes

$\Phi_1 \subseteq \Phi_2 \subseteq \cdots \Phi_i \subseteq \Phi_{i+1}$:

	X_1	X_2	X_3	X_4	X_5	X_6	X_7	...
~~β_1~~	~~0~~	~~1~~						
~~β_2~~	~~0~~	~~1~~	~~1~~	~~1~~	~~0~~	~~0~~	~~1~~	~~...~~
~~β_3~~	~~1~~	~~1~~	~~1~~	~~1~~	~~0~~	~~0~~	~~1~~	~~...~~
~~β_4~~	~~0~~	~~1~~	~~1~~	~~1~~	~~0~~	~~0~~	~~1~~	~~...~~
~~β_5~~	~~0~~	~~1~~	~~1~~	~~0~~	~~0~~	~~0~~	~~1~~	~~...~~
~~β_6~~	~~0~~	~~0~~	~~1~~	~~1~~	~~0~~	~~0~~	~~1~~	~~...~~
~~β_7~~	~~0~~	~~1~~	~~1~~	~~1~~	~~0~~	~~0~~	~~1~~	~~...~~
β_8	0	1	1	0	0	0	1	...
~~β_9~~	~~1~~	~~0~~	~~1~~	~~1~~	~~0~~	~~0~~	~~1~~	~~...~~
~~β_{10}~~	~~0~~	~~1~~	~~0~~	~~0~~	~~0~~	~~0~~	~~1~~	~~...~~
β_{11}	0	1	1	0	0	0	1	...
$\vdots$	...	...	...	...	...	...	...	...
α	0	1	1	0	0			...

$\varphi = X_3 \wedge (\neg X_4 \vee X_6)$

Konstr. $\alpha : \{X_1, \ldots\} \to \{0, 1\}$ und $I_n \subseteq I_0$, $n \geq 1$:

Für alle n gilt Eigenschaft $(*)$:

- I_n ist unendlich, $\beta_1, \ldots, \beta_{n-1} \notin I_n$ und
- für alle $\beta, \beta' \in I_n$ und $j \leq n$ gilt
  $\beta(X_j) = \beta'(X_j) = \alpha(X_j)$.

Induktionsvoraussetzung. Für $i < n$ seien $I_i, \alpha(X_i)$ schon konstruiert so dass $(\star)$ gilt.

Induktionsschritt. Da wegen $(*)$ I_{n-1} unendlich ist, gibt es ein $t \in \{0, 1\}$, so dass $\beta(X_n) = t$ für unendlich viele $\beta \in I_{n-1}$.

Setze

$\alpha(X_n) := t$ und

$I_n := \{\beta \in I_{n-1} : \beta(X_n) = t$ und $\beta \neq \beta_{n-1}\}$.

Offenbar ist $(\star)$ erfüllt.

Beweis des Satzes

Konstr. $\alpha : \{X_1, \dots\} \to \{0, 1\}$ und $I_n \subseteq I_0$, $n \geq 1$:

Für alle n gilt Eigenschaft $(*)$:

- I_n ist unendlich, $\beta_1, \dots, \beta_{n-1} \notin I_n$ und
- für alle $\beta, \beta' \in I_n$ und $j \leq n$ gilt
  $\beta(X_j) = \beta'(X_j) = \alpha(X_j)$.

Behauptung. $\alpha \models \Phi$.

Sei $\varphi \in \Phi$.

Beweis des Satzes

Konstr. $\alpha : \{X_1, ...\} \to \{0, 1\}$ und $I_n \subseteq I_0$, $n \geq 1$:

Für alle n gilt Eigenschaft $(*)$:

- I_n ist unendlich, $\beta_1, \dots, \beta_{n-1} \notin I_n$ und
- für alle $\beta, \beta' \in I_n$ und $j \leq n$ gilt
  $\beta(X_j) = \beta'(X_j) = \alpha(X_j)$.

Behauptung. $\alpha \models \Phi$.

Sei $\varphi \in \Phi$.

Da φ nur endlich viele Variablen enthält, ist $\varphi \in \Phi_n$ für ein n.

Es gilt also $\beta_i \models \varphi$ für alle $i \geq n$, insb. also $\beta \models \varphi$ für alle $\beta \in I_n$.

Beweis des Satzes

Konstr. $\alpha : \{X_1, ...\} \to \{0, 1\}$ und $I_n \subseteq I_0$, $n \geq 1$:

Für alle n gilt Eigenschaft $(*)$:

- I_n ist unendlich, $\beta_1, \dots, \beta_{n-1} \notin I_n$ und
- für alle $\beta, \cancel{\beta' \in} I_n$ und $j \leq n$ gilt
  $\beta(X_j) \cancel{= \beta'(X_j)} = \alpha(X_j)$.

weg.

Behauptung. $\alpha \models \Phi$.

Sei $\varphi \in \Phi$.

Da φ nur endlich viele Variablen enthält, ist $\varphi \in \Phi_n$ für ein n.

Es gilt also $\beta_i \models \varphi$ für alle $i \geq n$, insb. also $\beta \models \varphi$ für alle $\beta \in I_n$.

Sei $\beta \in I_n$. So ein β existiert, da wegen $(*)$ $I_n \neq \emptyset$.

Da nach $(*)$ für alle $i \leq n$ gilt $\alpha(X_i) = \beta(X_i)$ und $\beta \models \varphi$, folgt also $\alpha \models \varphi$. $\square$

Der Kompaktheitssatz der Aussagenlogik

Theorem. (Kompaktheits- oder Endlichkeitssatz)

Sei $\Phi \subseteq \mathsf{AL}$ eine Formelmenge und $\psi \in \mathsf{AL}$ eine Formel.

1. Φ ist erfüllbar gdw. jede endliche Teilmenge $\Phi' \subseteq \Phi$ erfüllbar ist.
2. $\Phi \models \psi$ gdw. eine endliche Teilmenge $\Phi_0 \subseteq \Phi$ existiert mit $\Phi_0 \models \psi$.

Der Kompaktheitssatz der Aussagenlogik

Theorem. (Kompaktheits- oder Endlichkeitssatz)

Sei $\Phi \subseteq \mathsf{AL}$ eine Formelmenge und $\psi \in \mathsf{AL}$ eine Formel.

1. Φ ist erfüllbar gdw. jede endliche Teilmenge $\Phi' \subseteq \Phi$ erfüllbar ist.
2. $\Phi \models \psi$ gdw. eine endliche Teilmenge $\Phi_0 \subseteq \Phi$ existiert mit $\Phi_0 \models \psi$.

Beweis von Teil 2. aus Teil 1.

$\Phi_0 \cap$

Offenbar gilt $\Phi \models \psi$ genau dann, wenn $\Phi \cup \{\neg\psi\}$ unerfüllbar ist.

Dies ist aber nach Teil 1. genau dann der Fall, wenn bereits eine endliche Teilmenge Φ_0 unerfüllbar ist.

- Ist $\neg\psi \in \Phi_0$, so gilt also $\Phi_0 \setminus \{\neg\psi\} \models \psi$.
- Anderenfalls ist $\Phi_0 \subseteq \Phi$, und da Φ_0 unerfüllbar ist, folgt $\Phi_0 \models \psi$.

Die Umkehrung ist trivial. □

Anwendung in der Resolution

Vollständigkeit des Resolutionskalküls

Behauptung. *Sei $n \in \mathbb{N}$ und sei $\mathcal{C}$ eine unerfüllbare Klauselmenge in den Variablen $\{V_1, \ldots, V_{n-1}\}$. Dann hat $\mathcal{C}$ eine Resolutionswiderlegung.*

Lemma. (Vollständigkeit des Resolutionskalküls)
Jede unerfüllbare Klauselmenge $\mathcal{C}$ hat eine Resolutionswiderlegung.

Beweis. Sei $\mathcal{C}$ eine unerfüllbare Klauselmenge.

1. Ist $\mathcal{C}$ endlich, dann enthält sie nur endlich viele Variablen und der Beweis folgt sofort aus der Behauptung.

2. Ist $\mathcal{C}$ unendlich, dann folgt aus dem Kompaktheitssatz, dass bereits eine endliche Teilmenge $\mathcal{C}' \subseteq \mathcal{C}$ unerfüllbar ist.
Also hat $\mathcal{C}'$ eine Resolutionswiderlegung.
Diese ist aber auch eine Resolutionswiderlegung von $\mathcal{C}$. $\square$

Vollständigkeit und Korrektheit des Resolutionskalküls

Theorem. Eine Menge $\mathcal{C}$ von Klauseln hat genau dann eine Resolutionswiderlegung, wenn $\mathcal{C}$ unerfüllbar ist.

- Der Resolutionskalkül ist also eine vollständige Methode, um Unerfüllbarkeit (und damit auch Erfüllbarkeit) nachzuweisen.
- Trotz seiner abstrakten Formulierung hat das Erfüllbarkeitsproblem der Aussagenlogik wichtige algorithmische Anwendungen in der Informatik.
- Ein effizientes Verfahren zur Lösung des Problems hätte daher weitreichende Auswirkungen.
- Wir werden aber als nächstes zeigen, dass ein solches Verfahren vermutlich nicht existieren kann, da das Problem NP-vollständig ist.

$\cancel{(x_1 \lor x_{15})} \land (x_2 \lor \neg x_{25}) \ldots \neg x_1$

$x_1 = 0$

$\neg x_t \lor \neg x_1$

x_1

$x_1 = 1$

5.2 Der DPLL Algorithmus

$x_2 = 0$

$x_2 = 1$

Der DPLL Algorithmus

Wir werden als nächstes einen Algorithmus kennen lernen, um Formeln auf Unerfüllbarkeit zu testen.

Der DPLL-Algorithmus ist benannt nach seinen Erfindern, Davis, Putnam, Logemann, Loveland.

Der Algorithmus kombiniert backtracking mit Einheitsresolution.

Varianten des DPLL-Algorithmus bilden die Basis der meisten aktuellen SAT-Löser, wie z. B. BerkMin, zChaff, etc.

DPLL Algorithmen

Der Algorithmus arbeitet auf Formeln $\varphi := \bigwedge_{i=1}^{n} \bigvee_{j=1}^{n_i} L_{i,j}$ in KNF.

Repr. φ als Klauselmenge: $\Phi := \{\{L_{1,1}, \dots, L_{1,n_1}\}, \dots, \{L_{n,1}, \dots, L_{n,n_n}\}\}$.

Basis Algorithmus DPLL(Φ, β)

Eingabe. Klauselmenge $\Phi := \{\{L_{1,1}, \dots, L_{1,n_1}\}, \dots, \{L_{n,1}, \dots, L_{n,n_n}\}\}$
Partielle Belegung β mit *def*$(\beta) \subseteq$ var(Φ).

Ausgabe. $\beta' \supseteq \beta$ mit $\beta' \models \Phi$ oder unerfüllbar, wenn kein β' existiert.

Algorithmus. Wenn $[\![\Phi]\!]^{\beta} = 1$, gib β zurück.
Wenn $[\![\Phi]\!]^{\beta} = 0$, gib unerfüllbar zurück.

Sonst (d.h. wenn der Wert von Φ unter β unbestimmt ist)

reduce(β, Φ);

branch(β); Wähle unbelegte Variable X aus var(Φ) und Wahrheitswert t.

$\beta' :=$ DPLL$(\Phi, \beta[X \mapsto t])$.

wenn $\beta' \neq$ unerfüllbar, gibt β' zurück

sonst gib DPLL$(\Phi, \beta[X \mapsto 1 - t])$ zurück.

Einheitsresolution

Die Funktion reduce(β, Φ) dient der Reduktion der Formel nach jedem Schritt.

$\{L_i\}$ $\{\neg L_i, X, Y\}$

Meistens wird nur Einheitsresolution verwendet.

$\{X, Y\}$

Unit Clause Propagation.

Enthält Φ Klausel $C = \{\overbrace{L_1, \dots, L_k}^{X}\}$ in der alle bis auf 1 Literal L_i durch β belegt werden, so muss jede erfüllende Belegung $\beta' \supseteq \beta$ das Literal L_i mit 1 belegen.

Ist $L_i = X$, können wir also direkt $[X \mapsto 1]$ zu β hinzufügen.

Ist $L_i = \neg X$, fügen wir $[X \mapsto 0]$ zu β hinzu.

Dies wird solange wiederholt, bis es keine Einer-Klauseln mehr gibt.

Basis DPLL(Φ, β)
Ausgabe.
$\beta' \supseteq \beta$ mit $\beta' \models \Phi$ oder unerf.
Algorithmus.
$[\![\Phi]\!]^\beta = 1 \rightsquigarrow$ return β.
$[\![\Phi]\!]^\beta = 0 \rightsquigarrow$ return unerf.

Sonst
reduce(β, Φ);
branch(β):
Wähle $X \in \text{var}(\Phi)$, $t \in \{0, 1\}$.
$\beta' :=$ DPLL$(\Phi, \beta[X \mapsto t])$.
wenn $\beta' \models \Phi$, *return* β'
sonst
return DPLL$(\Phi, \beta[X \mapsto 1 - t])$

Branching

Reduce. Die Funktion reduce(β, Φ); dient der Reduktion der Formel nach jedem Schritt.

Branch. Die Funktion branch(β, Φ); wählt die nächste Variable aus.

Hier gibt es verschiedenste Heuristiken.

dynamic largest individual sum (DLIS). Wähle Literal L das am häufigsten vorkommt und setze es auf 1.

dynamic largest combined sum (DLCS). Wähle Variable, die am häufigsten vorkommt.

Maximal occurrence in minimal clauses (MOM). Wähle Variable die am häufigsten in kürzesten Klauseln vorkommt.

Basis DPLL(Φ, β)
Ausgabe.
$\beta' \supseteq \beta$ mit $\beta' \models \Phi$ oder unerf.
Algorithmus.
$[\![\Phi]\!]^\beta = 1 \rightsquigarrow$ return β.
$[\![\Phi]\!]^\beta = 0 \rightsquigarrow$ return unerf.

Sonst

reduce(β, Φ);

branch(β):
  Wähle $X \in \text{var}(\Phi)$, $t \in \{0, 1\}$.

$\beta' := \text{DPLL}(\Phi, \beta[X \mapsto t])$.

wenn $\beta' \models \Phi$, *return* β'

sonst
  return DPLL$(\Phi, \beta[X \mapsto 1 - t])$

Beispiel für den DPLL-Algorithmus

$$\{U, \neg X\}, \{\neg U, \neg X\}, \{\neg W, Y\}, \{W, X\}, \{X, Z\}, \{\neg Y, \neg Z\}$$

Beispiel für den DPLL-Algorithmus

$$\{U, \neg X\}, \{\neg U, \neg X\}, \{\neg W, Y\}, \{W, X\}, \{X, Z\}, \{\neg Y, \neg Z\}$$

$$X$$

Beispiel für den DPLL-Algorithmus

$$\{U, \neg X\}, \{\neg U, \neg X\}, \{\neg W, Y\}, \{W, X\}, \{X, Z\}, \{\neg Y, \neg Z\}$$

X

$\xrightarrow{1}$

$$\{U\}, \{\neg U\}, \{\neg W, Y\}, \{\neg Y, \neg Z\}$$

Beispiel für den DPLL-Algorithmus

$$\{U, \neg X\}, \{\neg U, \neg X\}, \{\neg W, Y\}, \{W, X\}, \{X, Z\}, \{\neg Y, \neg Z\}$$

X

1

$$\{U\}, \{\neg U\}, \{\neg W, Y\}, \{\neg Y, \neg Z\}$$

U

Beispiel für den DPLL-Algorithmus

$$\{U, \neg X\}, \{\neg U, \neg X\}, \{\neg W, Y\}, \{W, X\}, \{X, Z\}, \{\neg Y, \neg Z\}$$

$$\Big\downarrow \begin{smallmatrix} X \\ 1 \end{smallmatrix}$$

$$\{U\}, \{\neg U\}, \{\neg W, Y\}, \{\neg Y, \neg Z\}$$

$$\Big\downarrow \begin{smallmatrix} U \\ 1 \end{smallmatrix}$$

$$\Box, \dots$$

Beispiel für den DPLL-Algorithmus

$$\{U, \neg X\}, \{\neg U, \neg X\}, \{\neg W, Y\}, \{W, X\}, \{X, Z\}, \{\neg Y, \neg Z\}$$

$$X \xrightarrow{1}$$

$$\{U\}, \{\neg U\}, \{\neg W, Y\}, \{\neg Y, \neg Z\}$$

$$U$$

$$\xleftarrow{1} \quad \xrightarrow{0}$$

$$\Box, \dots \qquad \Box, \dots$$

Beispiel für den DPLL-Algorithmus

$$\{U, \neg X\}, \{\neg U, \neg X\}, \{\neg W, Y\}, \{W, X\}, \{X, Z\}, \{\neg Y, \neg Z\}$$

X

1 | 0

$$\{U\}, \{\neg U\}, \{\neg W, Y\}, \{\neg Y, \neg Z\} \qquad \{\neg W, Y\}, \{W\}, \{Z\}, \{\neg Y, \neg Z\}$$

U

1 | 0

$$\square, \ldots \qquad \square, \ldots$$

Beispiel für den DPLL-Algorithmus

$$\{U, \neg X\}, \{\neg U, \neg X\}, \{\neg W, Y\}, \{W, X\}, \{X, Z\}, \{\neg Y, \neg Z\}$$

X

1: $\{U\}, \{\neg U\}, \{\neg W, Y\}, \{\neg Y, \neg Z\}$

- U
  - 1: $\Box, \ldots$
  - 0: $\Box, \ldots$

0: $\{\neg W, Y\}, \{W\}, \{Z\}, \{\neg Y, \neg Z\}$

- W

Beispiel für den DPLL-Algorithmus

$$\{U, \neg X\}, \{\neg U, \neg X\}, \{\neg W, Y\}, \{W, X\}, \{X, Z\}, \{\neg Y, \neg Z\}$$

X

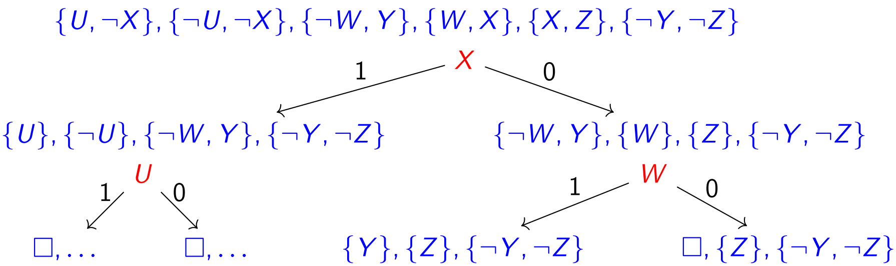
$X \xrightarrow{1}$ $\{U\}, \{\neg U\}, \{\neg W, Y\}, \{\neg Y, \neg Z\}$

- $U \xrightarrow{1}$ $\Box, \ldots$
- $U \xrightarrow{0}$ $\Box, \ldots$

$X \xrightarrow{0}$ $\{\neg W, Y\}, \{W\}, \{Z\}, \{\neg Y, \neg Z\}$

- $W \xrightarrow{0}$ $\Box, \{Z\}, \{\neg Y, \neg Z\}$

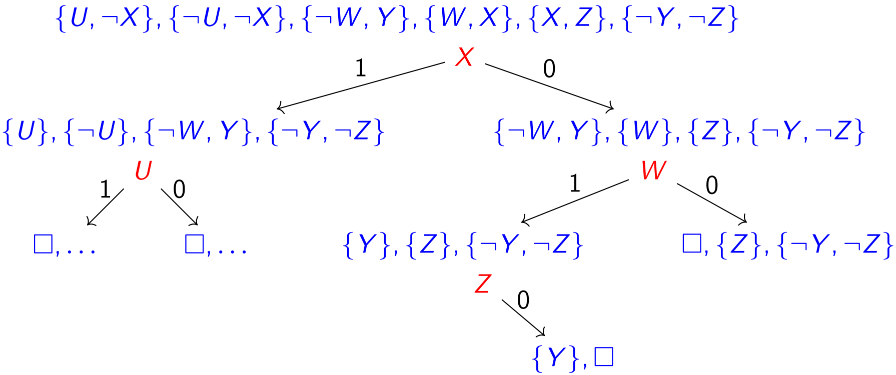
Beispiel für den DPLL-Algorithmus



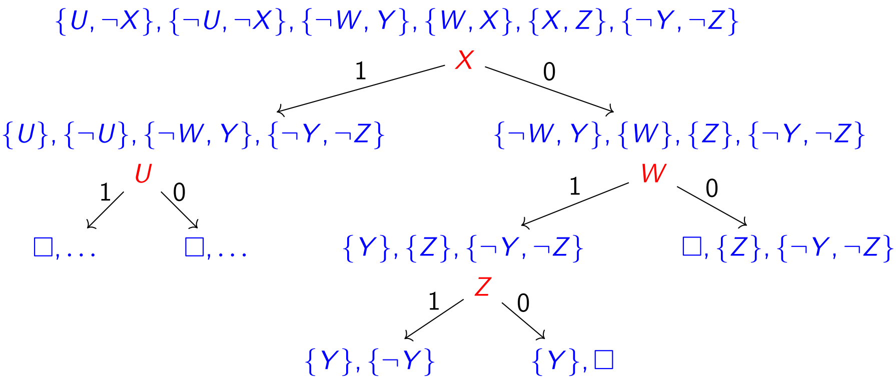
Beispiel für den DPLL-Algorithmus

$$
\begin{tikzcd}
 & & \begin{array}{c}\{U,\neg X\},\{\neg U,\neg X\},\{\neg W,Y\},\{W,X\},\{X,Z\},\{\neg Y,\neg Z\}\\ X\end{array} \arrow[dl, "1"'] \arrow[dr, "0"] & & \\
 & \begin{array}{c}\{U\},\{\neg U\},\{\neg W,Y\},\{\neg Y,\neg Z\}\\ U\end{array} \arrow[dl, "1"'] \arrow[d, "0"] & & \begin{array}{c}\{\neg W,Y\},\{W\},\{Z\},\{\neg Y,\neg Z\}\\ W\end{array} \arrow[d, "1"'] \arrow[dr, "0"] & \\
\square,\dots & \square,\dots & & \begin{array}{c}\{Y\},\{Z\},\{\neg Y,\neg Z\}\\ Z\end{array} & \square,\{Z\},\{\neg Y,\neg Z\}
\end{tikzcd}
$$

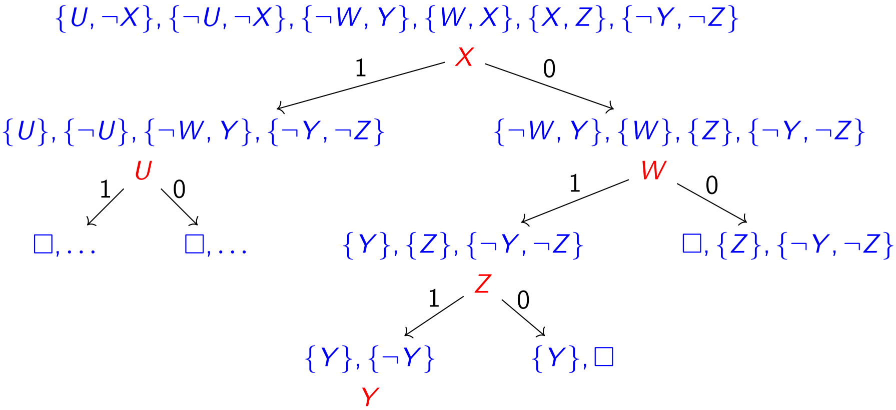
Beispiel für den DPLL-Algorithmus



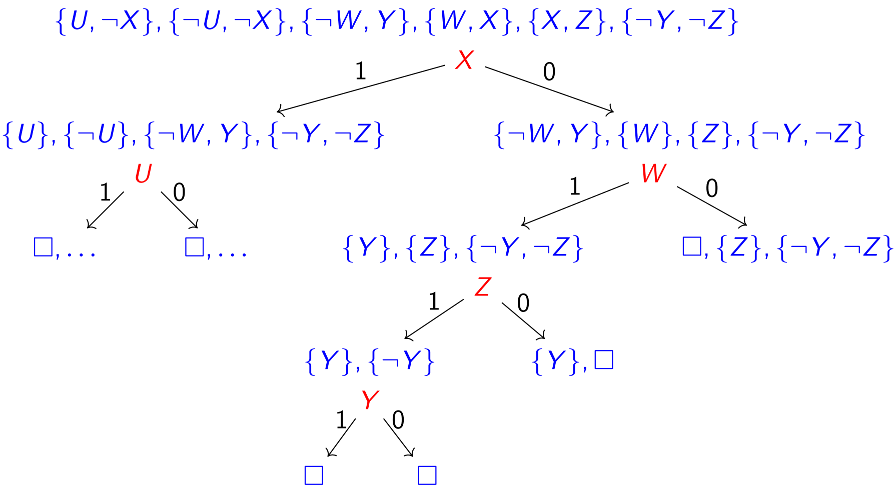
Beispiel für den DPLL-Algorithmus



Beispiel für den DPLL-Algorithmus



Beispiel für den DPLL-Algorithmus



Der DPLL-Algorithmus

In praktischen Implementierungen des Algorithmus' werden verschiedene Optimierungen verwendet.

Auswahlregel: Welches Literal wird beim Verzweigen gewählt

Conflict Analysis: Bei einem Backtracking Schritt wird der Grund der Unerfüllbarkeit (Conflict) analysiert und intelligenter zurück gesprungen.

Clause Learning: Aus der Konfliktanalyse werden neue Klauseln generiert, die zur Formel hinzugenommen werden.

Dies soll verhindern, dass in den gleichen Konflikt hineingelaufen wird.

Random restarts: Bisweilen wird ein DPLL-Lauf abgebrochen und neu angefangen. Die gelernten Klauseln bleiben erhalten.

DPLL vs. Resolution

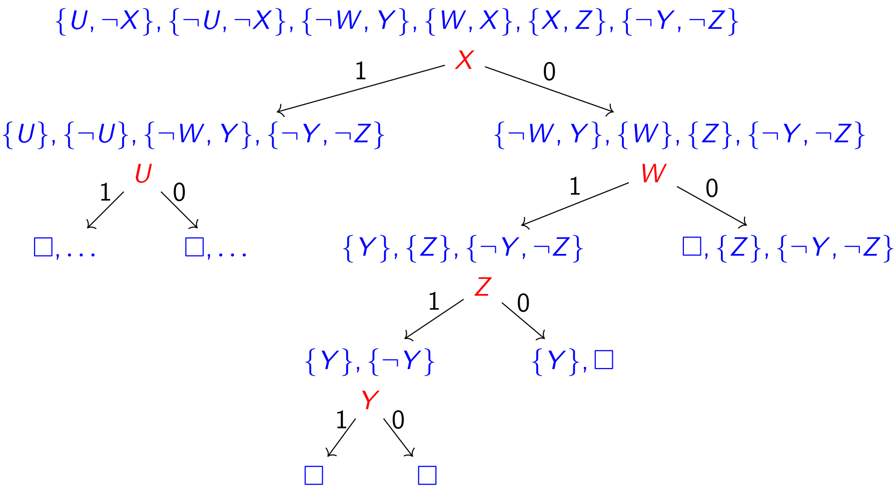
Es besteht ein enger Zusammenhang zwischen Resolutionswiderlegungen und Widerlegungen einer Formel durch den DPLL-Algorithmus.

Dazu stellt man einen Lauf des DPLL-Algorithmus als Entscheidungsbaum dar.

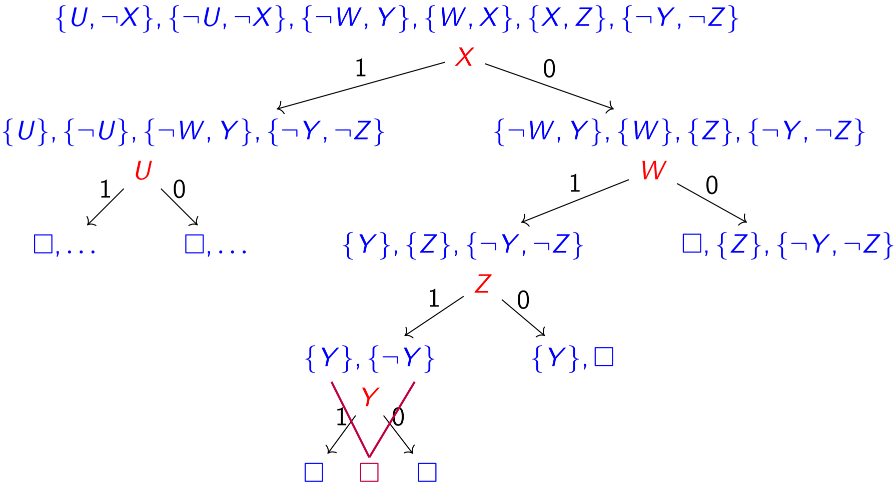
Hat der Baum die Höhe h, so gibt es auch eine (baumartige) Resolutionswiderlegung gleicher Höhe.

Der DPLL-Algorithmus ist also nicht "schneller" als die Resolution.

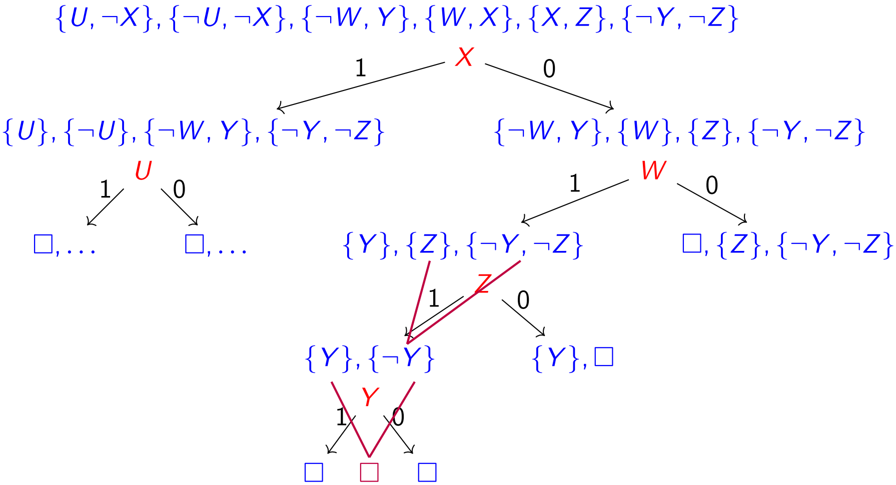
Beispiel für den DPLL-Algorithmus



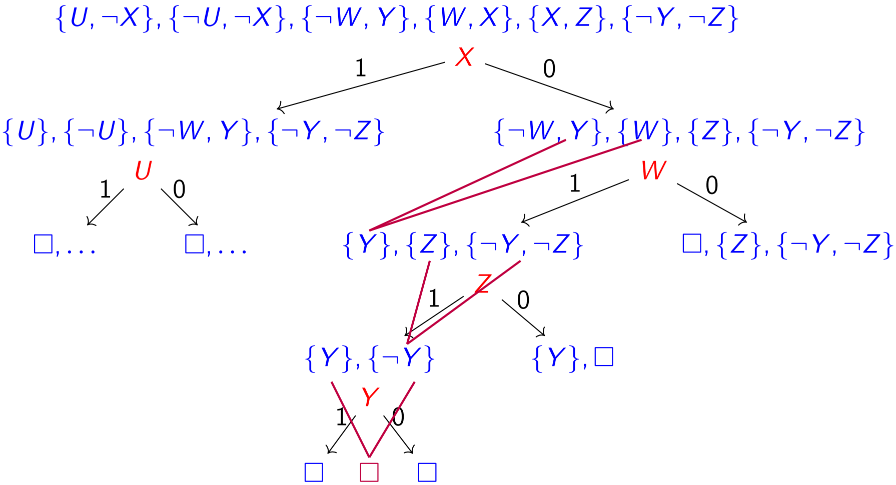
Beispiel für den DPLL-Algorithmus



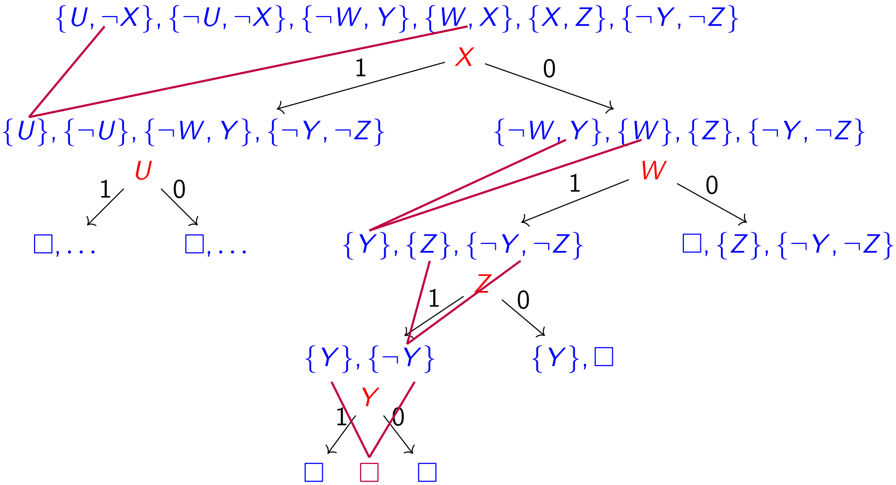
Beispiel für den DPLL-Algorithmus



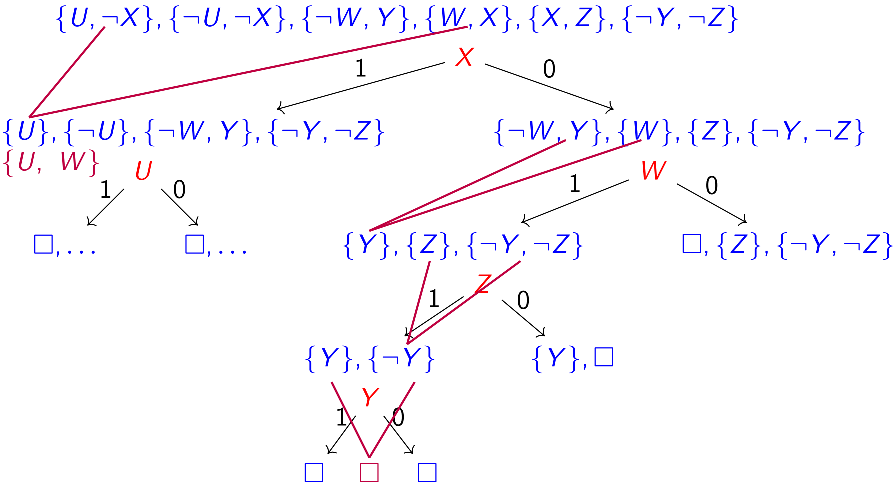
Beispiel für den DPLL-Algorithmus



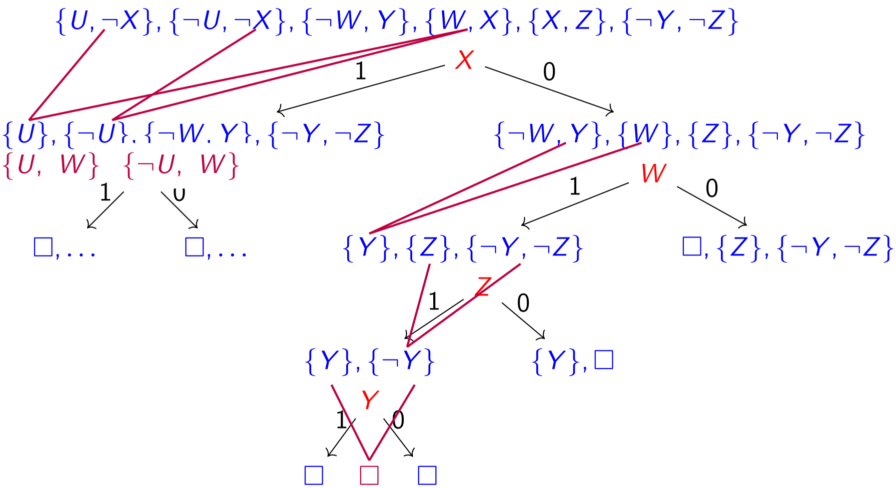
Beispiel für den DPLL-Algorithmus



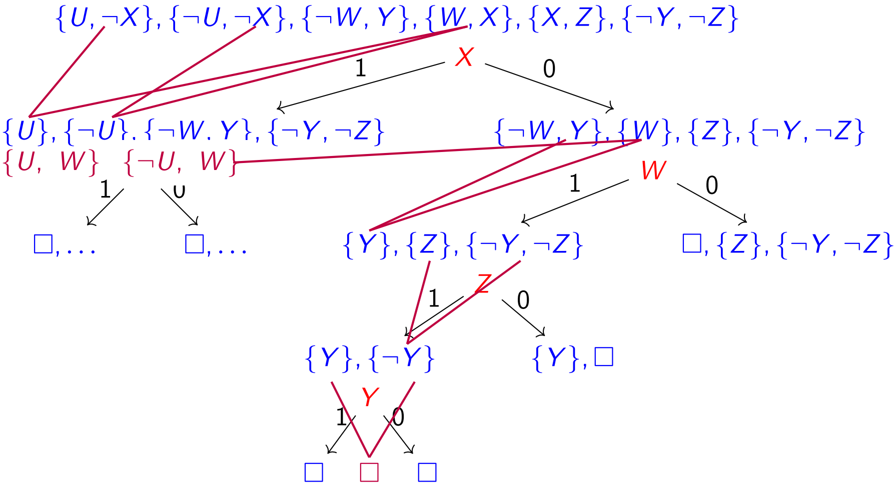
Beispiel für den DPLL-Algorithmus



Beispiel für den DPLL-Algorithmus



Beispiel für den DPLL-Algorithmus



Beispiel für den DPLL-Algorithmus