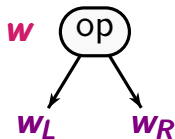


Sei  $T$  der Syntaxbaum der PNF-Formel  $\alpha_{PNF}$  mit Wurzel  $v_0$  und  $In(T)$  = Menge der inneren Knoten

$$\alpha_{3KNF} \stackrel{\text{def}}{=} v_0 \wedge \bigwedge_{w \in In(T)} \sigma_w$$

für jeden inneren Knoten:



$\sigma_w$  3KNF-Formel mit 3 Klauseln

$$\sigma_w \equiv w \leftrightarrow w_L \text{ op } w_R$$

**Korrektheit:**

$\alpha_{PNF}$  erfüllbar gdw  $\alpha_{3KNF}$  erfüllbar

1. Schritt: erstelle eine zu  $\alpha$  äquivalente PNF-Formel  $\alpha_{\text{PNF}}$ , in der *true* und *false* nicht vorkommen

$$\text{Kosten: } |\alpha_{\text{PNF}}| = \mathcal{O}(|\alpha|)$$

2. Schritt: konstruiere den Syntaxbaum  $T$  von  $\alpha_{\text{PNF}}$

$$\text{Kosten: Größe des Syntaxbaums} \\ |T| = \mathcal{O}(|\alpha_{\text{PNF}}|) = \mathcal{O}(|\alpha|)$$

3. Schritt: erstelle anhand des Syntaxbaums eine zu  $\alpha_{\text{PNF}}$  erfüllbarkeitsäquivalente 3KNF-Formel  $\alpha_{\text{3KNF}}$

$$\text{Kosten: } |\alpha_{\text{3KNF}}| = \mathcal{O}(|T|) = \mathcal{O}(|\alpha|)$$

fasse Klauseln als Mengen von Literalen auf

$$L_1 \vee L_2 \vee \dots \vee L_k \rightsquigarrow \{L_1, L_2, \dots, L_k\}$$

$$x \vee \neg y \vee z \rightsquigarrow \{x, \neg y, z\}$$

$$z \rightsquigarrow \{z\}$$

$$\mathit{false} = \perp \rightsquigarrow \emptyset$$

fasse KNF-Formeln als Mengen von Klauseln auf

$$\kappa_1 \wedge \kappa_2 \wedge \dots \wedge \kappa_m \rightsquigarrow \{\kappa_1, \kappa_2, \dots, \kappa_m\}$$

$$(x \vee \neg y) \wedge y \wedge (\neg x \vee z) \rightsquigarrow \{\{x, \neg y\}, \{y\}, \{\neg x, z\}\}$$

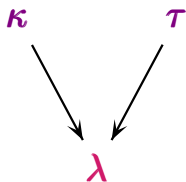
$$\mathit{true} \rightsquigarrow \emptyset$$

Seien  $\kappa$ ,  $\tau$  zwei Klauseln und  $L$  ein Literal mit  $L \in \kappa$  und  $\bar{L} \in \tau$ . Dann wird die Klausel

$$\lambda = \kappa \setminus \{L\} \cup \tau \setminus \{\bar{L}\}$$

ein Resolvent von  $\kappa$  und  $\tau$  genannt.

$\kappa$  und  $\tau$  heißen Elternklauseln von  $\lambda$ . Schreibweise:



Elternklauseln

Resolvent

Seien  $\kappa$ ,  $\tau$  zwei Klauseln und  $L$  ein Literal mit  $L \in \kappa$  und  $\bar{L} \in \tau$ . Dann wird die Klausel

$$\lambda = \kappa \setminus \{L\} \cup \tau \setminus \{\bar{L}\}$$

ein Resolvent von  $\kappa$  und  $\tau$  genannt.

Beispiel:  $\kappa = x \vee \neg y \vee z$        $\tau = \neg x \vee w$

$\{x, \neg y, z\}$                        $\{\neg x, w\}$

Klausel  $\neg y \vee z \vee w$

Ist  $\lambda$  Resolvent der Klauseln  $\kappa$  und  $\tau$ , so gilt:

$$\{\kappa, \tau\} \Vdash \lambda$$

Beweis: Sei  $L$  ein Literal mit  $L \in \kappa$  und  $\bar{L} \in \tau$  und

$$\lambda = \kappa \setminus \{L\} \cup \tau \setminus \{\bar{L}\}$$

Sei  $I$  ein Modell für  $\kappa$  und  $\tau$ . Zeige  $\lambda^I = 1$ .

1. Fall:  $L^I = 1$ . Wegen  $\tau^I = 1$  gibt es ein Literal  $L_0 \in \tau \setminus \{\bar{L}\}$  mit  $L_0^I = 1$ .

Wegen  $L_0 \in \lambda$  gilt  $\lambda^I = 1$ .

2. Fall:  $L^I = 0$ . Wegen  $\kappa^I = 1$  gibt es ein Literal ...

Seien  $\alpha = \kappa_1 \wedge \kappa_2 \wedge \dots \wedge \kappa_m$  eine KNF-Formel und sei  $\lambda$  ein Resolvent zweier Klauseln  $\kappa_i$  und  $\kappa_j$  von  $\alpha$ .  
Dann gilt:

$$\alpha \Vdash \lambda \quad \text{und} \quad \alpha \wedge \lambda \equiv \alpha$$

Wegen  $\{\kappa_i, \kappa_j\} \Vdash \lambda$  gilt:  $\alpha = \{\kappa_1, \kappa_2, \dots, \kappa_m\} \Vdash \lambda$

Wegen  $\alpha \Vdash \lambda$  gilt:

$\alpha$  und  $\alpha \wedge \lambda$  haben dieselbe Modelle.

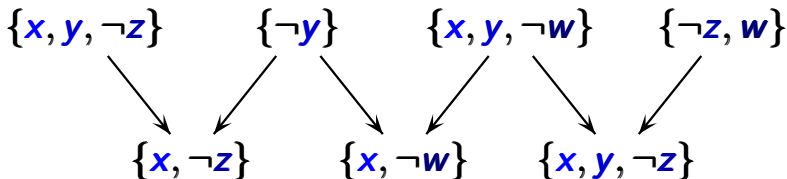
Daher gilt  $\alpha \wedge \lambda \equiv \alpha$ .

Sei  $\alpha$  eine KNF-Formel, aufgefasst als Klauselmenge.

$$\text{Res}(\alpha) = \{ \lambda : \lambda \text{ ist Resolvent zweier Klauseln in } \alpha \}$$

Beispiel:

$$\alpha = (x \vee y \vee \neg z) \wedge \neg y \wedge (x \vee y \vee \neg w) \wedge (\neg z \vee w)$$



$$\text{Res}(\alpha) = \{ \{x, \neg z\}, \{x, \neg w\}, \{x, y, \neg z\} \}$$

Sei  $\alpha$  eine KNF-Formel, aufgefasst als Klauselmenge.

$$Res(\alpha) = \{ \lambda : \lambda \text{ ist Resolvent zweier Klauseln in } \alpha \}$$

Resolutionsabschluss von  $\alpha$ :

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

wobei

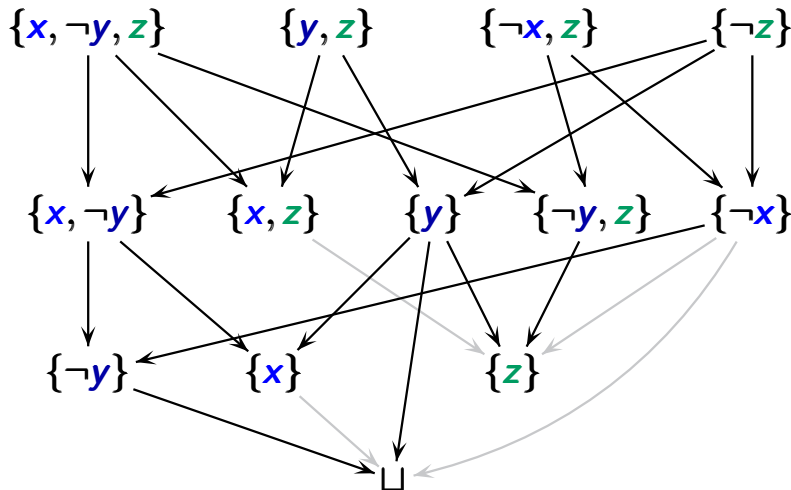
$$Res^0(\alpha) = \alpha$$

$$Res^{i+1}(\alpha) = Res^i(\alpha) \cup Res(Res^i(\alpha))$$

# Beispiel: Resolutionsabschluss

101

Sei  $\alpha = (x \vee \neg y \vee z) \wedge (y \vee z) \wedge (\neg x \vee z) \wedge \neg z$



Sei  $\alpha$  eine KNF-Formel mit den Atomen  $x_1, \dots, x_n$ .

Resolutionsabschluss von  $\alpha$ :

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

Es gilt:

$$Res^0(\alpha) \subseteq Res^1(\alpha) \subseteq Res^2(\alpha) \subseteq \dots \subseteq 2^{\mathcal{L}},$$

wobei  $\mathcal{L} = \{x_1, \dots, x_n, \neg x_1, \dots, \neg x_n\}$

endliche Menge  
von Literalen

Es gibt ein  $k \geq 0$  mit

$$Res^*(\alpha) = Res^k(\alpha) = Res^{k+1}(\alpha) = Res^{k+2}(\alpha) = \dots$$

Sei  $\alpha$  eine KNF-Formel.

1. Teil: Für alle Klauseln  $\lambda \in Res(\alpha)$  gilt:

$$\alpha \Vdash \lambda$$

2. Teil: Für alle Klauseln  $\lambda \in Res^*(\alpha)$  gilt:

$$\alpha \Vdash \lambda$$

Alle Resolventen zweier Klauseln von  $\alpha$  sind logische Folgerungen aus  $\alpha$ .

Daher sind alle Klauseln im Resolutionsabschluss logische Folgerungen aus  $\alpha$ .

Sei  $\alpha$  eine KNF-Formel und  $\lambda$  eine Klausel.

Eine Resolutionsherleitung von  $\lambda$  aus  $\alpha$  ist eine Folge

$$\langle \kappa_1, \tau_1, \lambda_1 \rangle \langle \kappa_2, \tau_2, \lambda_2 \rangle \dots \langle \kappa_m, \tau_m, \lambda_m \rangle$$

von Klauseltripeln, so dass  $\lambda_m = \lambda$  und für  $1 \leq i \leq m$ :

- $\lambda_i$  ist Resolvent von  $\kappa_i$  und  $\tau_i$
- $\kappa_i, \tau_i \in \alpha \cup \{\lambda_1, \lambda_2, \dots, \lambda_{i-1}\}$

$m$  wird Länge der Herleitung genannt.

$\lambda$  heißt dann herleitbar aus  $\alpha$ .

Sei  $\alpha$  eine KNF-Formel mit den Atomen  $x_1, \dots, x_n$ .

Resolutionsabschluss von  $\alpha$ :

$$Res^*(\alpha) = \bigcup_{i \geq 0} Res^i(\alpha) = Res^k(\alpha) \quad \text{für ein } k \leq 4^n$$

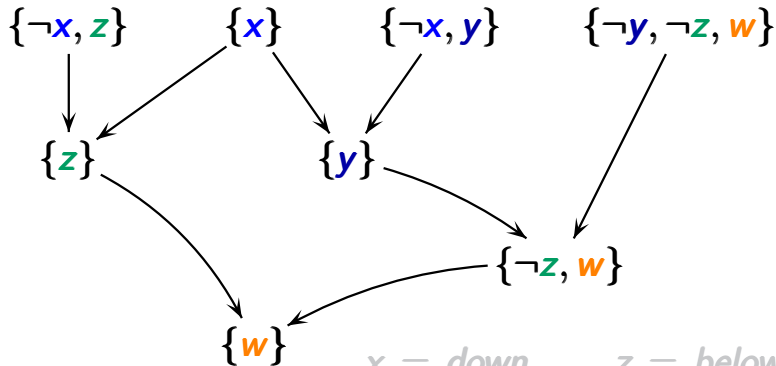
Für alle Klauseln  $\lambda$  gilt:

$$\lambda \in Res^*(\alpha) \quad \text{gdw} \quad \left\{ \begin{array}{l} \text{es gibt eine Herleitung} \\ \text{von } \lambda \text{ aus } \alpha \end{array} \right.$$

Insbesondere: ist  $\lambda$  aus  $\alpha$  herleitbar, so gilt  $\alpha \Vdash \lambda$

ist  $\perp$  aus  $\alpha$  herleitbar, so ist  $\alpha$  unerfüllbar.

$$\begin{aligned}\alpha &= (x \rightarrow z) \wedge (x \rightarrow y) \wedge (y \wedge z \rightarrow w) \wedge x \\ &\equiv (\neg x \vee z) \wedge (\neg x \vee y) \wedge (\neg y \vee \neg z \vee w) \wedge x\end{aligned}$$



$x = \text{down}$

$z = \text{below}$

$y = \text{up}$

$w = \text{banana}$

Sei  $\alpha$  eine KNF-Formel.

Eine Resolutionswiderlegung für  $\alpha$  ist eine Herleitung der leeren Klausel aus  $\alpha$

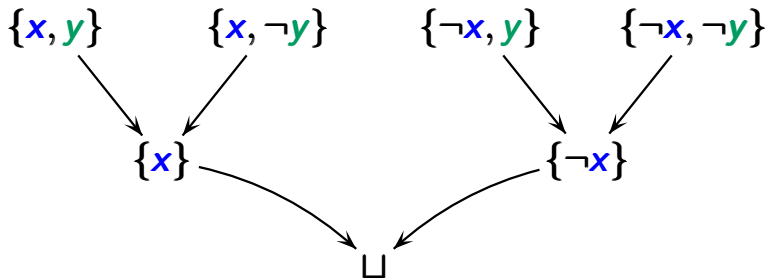
$$\langle \kappa_1, \tau_1, \lambda_1 \rangle \dots \langle \kappa_{m-1}, \tau_{m-1}, \lambda_{m-1} \rangle \langle \kappa_m, \tau_m, \sqcup \rangle$$

↑  
leere Klausel

bereits gezeigt: falls  $\alpha$  eine Widerlegung besitzt,  
dann ist  $\alpha$  unerfüllbar.

$$\alpha = (x \vee y) \wedge (x \vee \neg y) \wedge (\neg x \vee y) \wedge (\neg x \vee \neg y)$$

unerfüllbare KNF-Formel



Für jede KNF-Formel  $\alpha$  gilt:

$\alpha$  unerfüllbar    gdw     $\perp \in Res^*(\alpha)$

Beweis von “ $\Leftarrow$ ”: folgt aus dem Resolventenlemma

Gilt  $\perp = \mathbf{false} \in Res^*(\alpha)$ , so gilt  $\alpha \Vdash \mathbf{false}$ .

Also ist  $\alpha$  unerfüllbar.

Beweis von “ $\Rightarrow$ ”: durch Induktion nach der Anzahl  $n$   
der in  $\alpha$  vorkommenden Atome

$\perp = \mathbf{false}$     leere Klausel

$Res^*(\alpha)$     Resolutionsabschluss von  $\alpha$

Sei  $\alpha$  eine KNF-Formel mit  $n+1$  Atomen, wobei  $n \geq 0$ .  
Sei  $x$  ein Atom, das in  $\alpha$  vorkommt. In den KNF-Formeln

$$\alpha_0 = \alpha[x/0] \quad \text{und} \quad \alpha_1 = \alpha[x/1]$$

kommen jeweils höchstens  $n$  Atome vor. Mit  $\alpha$  sind auch  $\alpha_0$  und  $\alpha_1$  unerfüllbar. Nach Induktionsvoraussetzung gilt:

$$\perp \in Res^*(\alpha_0) \quad \text{und} \quad \perp \in Res^*(\alpha_1)$$

Betrachte nun Widerlegungen für  $\alpha_0$  und  $\alpha_1$  und

- lifte die Widerlegung für  $\alpha_0$  zu einer Herleitung von  $\perp$  oder  $x$  aus  $\alpha$
- lifte die Widerlegung für  $\alpha_1$  zu einer Herleitung von  $\perp$  oder  $\neg x$  aus  $\alpha$

Hieraus folgt:

$$\perp \in Res^*(\alpha)$$

gegeben: Widerlegung für  $\alpha_0$  der Länge  $\geq 1$

$$\langle \kappa_1, \tau_1, \lambda_1 \rangle \quad \langle \kappa_2, \tau_2, \lambda_2 \rangle \quad \dots \quad \langle \kappa_m, \tau_m, \lambda_m \rangle$$

wobei  $\kappa_i, \tau_i \in \alpha_0 \cup \{\lambda_1, \dots, \lambda_{i-1}\}$

$$\lambda_m = \perp$$

Lifting zu einer Herleitung aus  $\alpha$ :

$$\langle \kappa'_1, \tau'_1, \lambda'_1 \rangle \quad \langle \kappa'_2, \tau'_2, \lambda'_2 \rangle \quad \dots \quad \langle \kappa'_m, \tau'_m, \lambda'_m \rangle$$

Induktive Definition von  $\kappa'_i$  und  $\tau'_i$

- Ist  $\kappa_i \in \alpha$ , so ist  $\kappa'_i = \kappa_i$ .
- Ist  $\kappa_i = \kappa \setminus \{x\}$ , wobei  $\kappa \in \alpha$ , so ist  $\kappa'_i = \kappa$ .
- Ist  $\kappa_i = \lambda_j$ , so ist  $\kappa'_i = \lambda'_j$ .  $\leftarrow$  Resolvent aus  $\kappa'_j$  und  $\tau'_j$

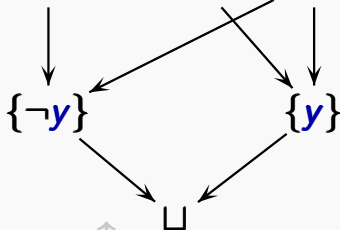
analoge  
Definition für  $\tau'_i$

$$\alpha = (x \vee \neg y \vee z) \wedge (y \vee z) \wedge (\neg x \vee z) \wedge \neg z$$

$$\alpha_0 = \alpha[x/0]$$

$$(\neg y \vee z) \wedge (y \vee z) \wedge \neg z$$

$$\{\neg y, z\} \quad \{y, z\} \quad \{\neg z\}$$



Wiedereinfügen von  $x$

$$\alpha_1 = \alpha[x/1]$$

$$(y \vee z) \wedge z \wedge \neg z$$

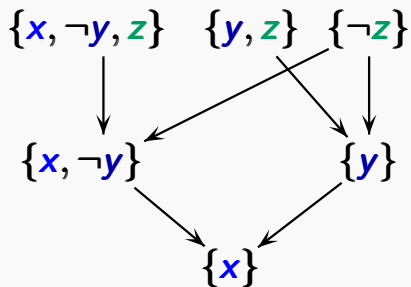
$$\{y, z\} \quad \{z\} \quad \{\neg z\}$$



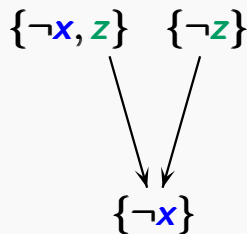
Wiedereinfügen von  $\neg x$

$$\alpha = (x \vee \neg y \vee z) \wedge (y \vee z) \wedge (\neg x \vee z) \wedge \neg z$$

Herleitung von  $\{x\}$   
aus  $\alpha$

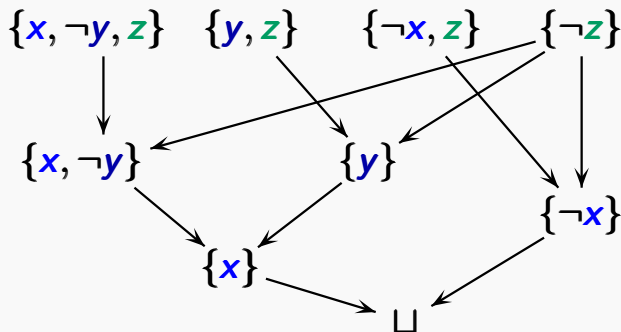


Herleitung von  $\{\neg x\}$   
aus  $\alpha$



$$\alpha = (x \vee \neg y \vee z) \wedge (y \vee z) \wedge (\neg x \vee z) \wedge \neg z$$

Kombination der Herleitungen von  $\{x\}$  und  $\{\neg x\}$



Widerlegung  
für  $\alpha$