Resolutionsalgorithmus (Erfüllbarkeitstest). In Algorithmus 9 (Seite 173) ist ein naives Verfahren für den Nachweis der (Un-)Erfüllbarkeit einer KNF-Formel angegeben, das auf unseren bisherigen Ergebnissen beruht. Die Terminierung des Verfahrens folgt aus Lemma 4.43 (Seite 166); die Korrektheit der Antworten aus dem Resolutionssatz (Satz 4.53 Seite 169).

Algorithmus 9 Resolutionsalgorithmus

(*	Eingabe:	KNF-Formel α	*)
(*	Aufgabe:	prüft, ob α erfüllbar ist	*)

$\alpha_0 = \alpha$;
$i := 0$; (* Berechne die Mengen $\alpha_i = Res^i(\alpha)$, $i = 1, 2, 3, \ldots$ *)
REPEAT
 $i := i + 1$;
 $\alpha_i := \alpha_{i-1} \cup Res(\alpha_{i-1})$
UNTIL $\sqcup \in \alpha_i$ oder $\alpha_i = \alpha_{i-1}$;
IF $\sqcup \in \alpha_i$
 THEN return “nein, α ist unerfüllbar”
 ELSE return “ja, α ist erfüllbar”
FI

Vollständigkeit versus Widerlegungsvollständigkeit. Die Vollständigkeitsaussage des Resolutionskalküls in Satz 4.53 bezieht sich lediglich auf die Herleitbarkeit der leeren Klausel aus unerfüllbaren Klauselmengen. Es gilt jedoch *nicht*, dass alle logisch folgerbaren Klauseln herleitbar sein müssen. Die Umkehrung des zweiten Teils des Resolventenlemmas gilt also *nicht* für beliebige Klauseln. In diesem Sinn ist der Resolutionskalkül *unvollständig*. Ein Beispiel, das die Unvollständigkeit belegt, ist die unerfüllbare KNF-Formel $\alpha = y \wedge \neg y \wedge x$. Die Einheitsklausel $\{\neg x\}$ ist zwar eine logische Folgerung aus α, jedoch ist $\{\neg x\}$ nicht aus α durch Resolution herleitbar. Dennoch gilt die Vollständigkeit der Resolution als Kalkül für logische Folgerbarkeit in folgendem Sinn:

Satz 4.55 (Resolutionsherleitbarkeit von Klauseln). *Für jede nicht-tautologische KNF-Formel α und jede nicht-tautologische Klausel λ gilt:*

$$\alpha \Vdash \lambda \quad \textit{gdw} \quad \lambda' \in Res^*(\alpha) \textit{ für eine Klausel } \lambda' \textit{ mit } \lambda' \subseteq \lambda \qquad \text{(ohne Beweis)}$$

Mit $\lambda = \sqcup$ ergibt sich erneut die Widerlegungskorrektheit und -vollständigkeit, die im Resolutionssatz nachgewiesen wurde. Eine weitere Konsequenz aus Satz 4.55 ist die Vollständigkeit der Resolution als Kalkül für die Folgerbarkeit von Einheitsklauseln aus erfüllbaren KNF-Formeln. Ist nämlich α eine erfüllbare KNF-Formel und L ein Literal, so gilt aufgrund der Aussage von Satz 4.55:

$$\alpha \Vdash L \quad \text{gdw} \quad L \in Res^*(\alpha).$$

Resolutionsalgorithmus für logische Folgerungen. Soll anstelle der Erfüllbarkeit von α geprüft werden, ob eine Klausel λ logisch aus α folgt (also ob $\alpha \Vdash \lambda$), so ist das Schleifenabbruchkriterium im Resolutionsalgorithmus (Algorithmus 9) wie folgt zu ändern:

"**UNTIL** es gibt eine Klausel $\lambda' \in \alpha_i$ mit $\lambda' \subseteq \lambda$ oder $\alpha_i = \alpha_{i-1}$;"

Alternativ kann man den Resolutionsalgorithmus unverändert lassen und ihn mit der Klauselmenge bestehend aus den Klauseln von α und den Einheitsklauseln $\overline{L_1}, \ldots, \overline{L_k}$ starten, wobei $L_1, \ldots, L_k$ die Literale von λ sind. Wir erinnern daran, dass $\alpha \Vdash \lambda$ genau dann gilt, wenn $\alpha \wedge \neg\lambda$ unerfüllbar ist. Ist nun $\lambda = L_1 \vee \ldots \vee L_k$, so ist

$$\begin{aligned} \alpha \wedge \neg\lambda &= \alpha \wedge \neg(L_1 \vee \ldots \vee L_k) \\ &\equiv \alpha \wedge \neg L_1 \wedge \ldots \wedge \neg L_k \\ &\equiv \alpha \wedge \overline{L_1} \wedge \ldots \wedge \overline{L_k} \end{aligned}$$

Die Formel $\alpha \wedge \neg\lambda$ kann daher als KNF-Formel mit der induzierten Klauselmenge

$$\cup \left\{ \{\overline{L_1}\}, \ldots, \{\overline{L_k}\} \right\}$$

aufgefasst werden. Diese Vorgehensweise hat den Nachteil, dass man die Anzahl an Klauseln und damit die Anzahl an potentiellen Resolventen erhöht.

Die einfache Formulierung des Resolutionsalgorithmus bildet alle möglichen Resolventen und ist daher hoffnungslos ineffizient, da es im Allgemeinen sehr viele Resolventen gibt. Falls n Aussagensymbole in der Eingabeformel α vorkommen, so wächst die maximale Anzahl an Schleifendurchläufen des Resolutionsalgorithmus exponentiell in n. Ein Beispiel für Formeln, in denen der Resolutionsalgorithmus tatsächlich exponentiell viele Schleifendurchläufe benötigt, sind die *Pigeonhole-Formeln*, die bereits in Beispiel 4.4 auf Seite 120 betrachtet wurden. Diese beschreiben das unlösbare Problem, n Tauben auf $n-1$ Löcher zu verteilen, so dass jede Taube in einem Loch sitzt, aber keine zwei Tauben sich in demselben Loch befinden:

$$\alpha_n = \bigwedge_{1 \leqslant i < n} \; \bigwedge_{1 \leqslant j < k \leqslant n} \underbrace{(\neg x_{i,j} \vee \neg x_{i,k})}_{\substack{\text{Taube } j \text{ oder Taube } k \\ \text{ist nicht in Loch } i}} \wedge \bigwedge_{1 \leqslant j \leqslant n} \underbrace{(x_{1,j} \vee \ldots \vee x_{n-1,j})}_{\substack{\text{Taube } j \text{ sitzt} \\ \text{in einem Loch}}}$$

Man kann zeigen, dass es eine Konstante $c > 1$ gibt, so dass die kürzeste Resolutionswiderlegung für α_n mindestens c^n Resolutionsschritte benötigt. Wir verzichten auf den Nachweis dieser Aussage.

Resolutionsalgorithmus für 2KNF. Trotz des schlechten worst-case-Verhaltens gibt es mehrere Teilklassen von KNF-Formeln, für die der Resolutionsalgorithmus (selbst in der naiven Version) recht effizient ist. Dazu zählt die Klasse der 2KNF-Formeln (d.h., KNF-Formeln mit höchstens zwei Literalen pro Klausel), für die nachgewiesen werden kann, dass selbst im schlimmsten Fall nur quadratisch viele Iterationen ausgeführt werden. Beachte, dass erstens Resolventen von 2KNF-Formeln stets höchstens zwei Literale haben und dass zweitens die Anzahl an Klauseln mit maximal zwei Literalen über einer n-elementigen Atommenge $\{x_1, \ldots, x_n\}$ durch

$$2n \cdot (2n-1) + 2n + 1 = \mathcal{O}\left(n^2\right)$$

beschränkt ist.Diese Schranke ergibt sich wie folgt: der Summand 1 steht für die leere Klausel, der Summand $2n$ für die Anzahl an einelementigen Klauseln, und der Summand $2n * (2n-1)$ steht für die Anzahl an zweielementigen Klauseln.

N-Resolution und andere Resolutionsstrategien

In vielen Fällen ist es nicht so dramatisch wie im Falle der Pigeonhole-Formeln, da oftmals "relativ kurze" Widerlegungen unerfüllbarer Formeln existieren, obwohl es "sehr viele" Resolventen gibt. Für praktische Anwendungen der Resolutionsidee sind daher Strategien von großer Bedeutung, welche die Anzahl an relevanten Resolventen einschränken und gezielt nach Resolventen suchen, die tatsächlich zu einer Widerlegung beitragen können.

N- und P-Resolution zählen zu den Resolutionsstrategien, welche die Anzahl an relevanten Resolventen einschränken, die für den Nachweis der Unerfüllbarkeit einer Klauselmenge benötigt werden. Wir beginnen mit Erläuterungen der N-Resolution, in der nur solche Resolutionsschritte zugelassen werden, in denen eine der Elternklauseln aus lauter negativen Literalen besteht. Nicht-leere Klauseln, welche nur aus negativen Literalen bestehen, werden auch *negative Klauseln* genannt.

$$\begin{tikzcd}
\{x, L_1, \ldots, L_k\} \arrow[dr] & & \{\neg x, \neg y_1, \ldots, \neg y_m\} \arrow[dl] \\
& \{L_1, \ldots, L_k, \neg y_1, \ldots, \neg y_m\} &
\end{tikzcd}$$

Abbildung 40: Schema der N-Resolution

Zunächst machen wir uns intuitiv klar, warum die Beschränkung auf derartige N-Resolventen sinnvoll ist. Der letzte Resolutionsschritt einer Widerlegung (der Länge $\geqslant 1$) resolviert stets zwei zueinander komplementäre Einheitsklauseln $\{x\}$ und $\{\neg x\}$. Insbesondere besteht eine der Elternklauseln, nämlich $\{\neg x\}$, nur aus negativen Literalen. Die Klausel $\{\neg x\}$ kann nur durch Resolution

- *entweder* einer Klausel der Form $\{\neg y, \neg x\}$ und einer der beiden Klauseln $\{y\}$ oder $\{y, \neg x\}$
- *oder* einer negativen Einheitsklausel $\{\neg y\}$ und der Klausel $\{y, \neg x\}$

entstanden sein. In jedem Fall ist eine negative Klausel beteiligt. Die positive Einheitsklausel $\{x\}$ kann zwar zunächst durch die Resolvierung zweier nicht-negativer Klauseln, etwa $\{\neg z, x\}$ und entweder $\{z, x\}$, entstanden sein; jedoch kann man durch eine vorangestellte Resolvierung mit der negativen Einheitsklausel $\{\neg x\}$ die Elternklausel $\{\neg z, x\}$ zu $\{\neg z\}$ verkürzen.

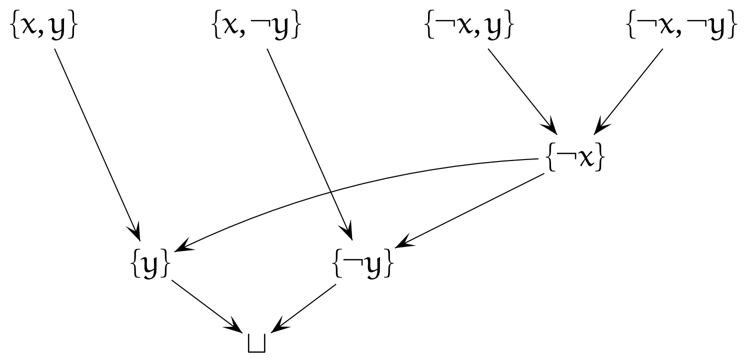
Diese Überlegungen motivieren die sogenannte *N-Resolution*, bei der ausschließlich solche Resolventen zugelassen sind, welche aus einer beliebigen Klausel und einer negativen Klausel gebildet werden können. Das allgemeine Schema ist in Abbildung 40 angegeben, wobei $L_1, \ldots, L_k$ beliebige Literale mit $x \notin \{L_1, \ldots, L_k\}$ und $y_1, \ldots, y_m$ Aussagensymbole mit $x \notin \{y_1, \ldots, y_m\}$ sind.

Definition 4.56 (N-Herleitung). Eine N-Herleitung von λ aus einer KNF-Formel α bezeichnet eine Herleitung

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

von λ aus α, so dass $\kappa_1, \ldots, \kappa_m$ negative Klauseln sind. ■

Andere Begriffe und Bezeichnungen, die für die gewöhnliche Resolution eingeführt wurden, sind analog definiert. Der N-Resolutionsabschluss von α besteht aus allen Klauseln, für die es eine N-Herleitung aus α gibt. Er wird mit $NRes^*(\alpha)$ bezeichnet. Eine N-Widerlegung für α bezeichnet eine N-Herleitung der leeren Klausel aus α.



Die Abbildung oben zeigt eine N-Widerlegung für die unerfüllbare KNF-Formel

$$(x \vee y) \wedge (x \vee \neg y) \wedge (\neg x \vee y) \wedge (\neg x \vee \neg y).$$

Tatsächlich schränkt die N-Resolution oftmals die Anzahl an möglichen Resolventen erheblich ein. Liegt z.B. die KNF-Formel

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

vor, so können unter der gewöhnlichen Resolution sieben Resolventen gebildet werden. Die erste Klausel kann mit der zweiten (auf zwei Arten), vierten und fünften Klausel resolviert werden. Weiter können Resolventen aus der zweiten und dritten Klausel, aus der dritten und vierten und aus der vierten und fünften Klausel generiert werden. Man erhält:

$$\begin{aligned} Res^1(\alpha) \;&=\; \alpha \cup \\ &\big\{ \{x, z, \neg z\}, \{x, y, \neg y\}, \{x, \neg y, w\}, \{x, \neg y, \neg w\}, \{x, \neg z, \neg w\}\ \{x, \neg y, \neg z\}, \{x, \neg z\} \big\} \end{aligned}$$

Nun können weitere Resolventen, z.B. aus $\{x, z, \neg z\}$ und der ersten, zweiten, dritten und vierten Klausel von α sowie aus $\{x, z, \neg z\}$ und $\{x, \neg z, \neg w\}$, gebildet werden. Mit der N-Resolution ist dagegen nur die Resolventenbildung zwischen der ersten und fünften Klausel sowie der vierten und der fünften zulässig, da nur die fünfte Klausel $\neg w \vee \neg z$ aus lauter negativen Literalen besteht und diese nur mit der ersten und vierten Klausel resolvierbar ist. Man erhält die beiden N-Resolventen $\{x, \neg y, \neg w\}$ und $\{x, \neg z\}$. Weitere N-Resolventen können nicht gebildet werden, da diese beiden Klauseln das positive Literal x enthalten. Aus der Widerlegungsvollständigkeit der N-Resolution, (siehe Satz 4.57 unten) ergibt sich damit die Erfüllbarkeit von α mit der Bildung von nur zwei N-Resolventen.

Die Korrektheit der N-Resolution als Kalkül für den Nachweis der Unerfüllbarkeit ist klar, da aus der Existenz einer N-Widerlegung die Aussage $\sqcup \in Res^*(\alpha)$ folgt. Der Resolutionssatz impliziert daher die Unerfüllbarkeit von α. Interessanterweise ist die N-Resolution auch vollständig für den Nachweis der Unerfüllbarkeit. Wir zitieren das Ergebnis ohne Beweis:

Satz 4.57 (Korrektheit und Widerlegungsvollständigkeit der N-Resolution). *Sei α eine KNF-Formel. Dann gilt: α ist genau dann unerfüllbar, wenn es eine N-Widerlegung für α gibt. (ohne Beweis)*

Bemerkung 4.58. Die Widerlegungsvollständigkeit der N-Resolution belegt nochmals die Erfüllbarkeit von KNF-Formeln, in denen jede Klausel wenigstens ein positives Literal besitzt. Siehe Teil (a) von Lemma 4.23 auf Seite 147. Aus den Klauseln solcher KNF-Formeln können keine N-Resolventen gebildet werden. Daher ist eine N-Widerlegung ausgeschlossen. ■

Der Nachteil der N-Resolution ist, dass N-Widerlegungen oftmals länger als kürzeste Widerlegungen sind. Wir erwähnen ohne Beweis, dass es unerfüllbare KNF-Formeln α_n gibt, deren kürzeste Widerlegungen polynomielle Länge haben, während alle N-Widerlegungen mindestens exponentiell lang sind.

P-Resolution. In analoger Weise ist die P-Resolution definiert, bei der nur Resolventen aus einer beliebigen Klausel und einer Klausel mit nur positiven Literalen gebildet werden dürfen. Die Begriffe P-Herleitung und P-Widerlegung sind wie für die N-Resolution definiert. Die Korrektheit und Widerlegungsvollständigkeit der P-Resolution kann wie für die N-Resolution beweisen werden. Eine gegebene KNF-Formel α ist also genau dann unerfüllbar, wenn es eine P-Widerlegung für α gibt. Ein Beispiel für eine P-Widerlegung für die unerfüllbare KNF-Formel $(x \vee y) \wedge (x \vee \neg y) \wedge (\neg x \vee y) \wedge (\neg x \vee \neg y)$ ist in Abbildung 41 angegeben.

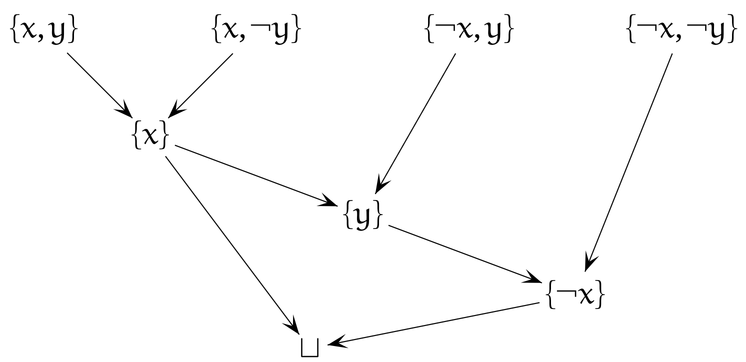


Abbildung 41: Beispiel für eine P-Widerlegung

Lineare Resolution. Die N-Resolution schränkt zwar die Anzahl an Resolventen ein, so dass unter Einsatz der N-Strategie oftmals eine deutliche Verbesserung im Resolutions-Algorithmus zu verzeichnen ist; jedoch können die zu N-Widerlegungen gehörenden Graphen sehr komplexe Struktur haben. Eine gezielte Suche nach Widerlegungen ist unter der Einschränkung der N-Resolution nicht wirklich gewährleistet. Analoges gilt für die P-Resolution.

Lineare Resolution löst sich von dem Schema des Resolutions-Algorithmus und sucht stattdessen gezielt nach Herleitungen der leeren Klausel. Die Idee besteht darin, eine “lineare” Folge von Klauseln $\lambda_1, \ldots, \lambda_m$ zu geniereren, so dass die i-te Klausel λ_i ein Resolvent von λ_{i-1} und einer Klausel τ_i in $\alpha \cup \{\lambda_1, \ldots, \lambda_{i-2}\}$ ist. τ_i wird in diesem Kontext auch *Seitenklausel* genannt. Sei λ eine Klausel und α eine KNF-Formel. Eine Herleitung von λ aus α wird *linear* genannt, wenn sie die Gestalt

$$\langle \lambda_0, \tau_1, \lambda_1 \rangle \ \langle \lambda_1, \tau_2, \lambda_2 \rangle \ \ldots \ \langle \lambda_{m-1}, \tau_m, \lambda_m \rangle, \quad \text{wobei } \lambda_m = \lambda$$

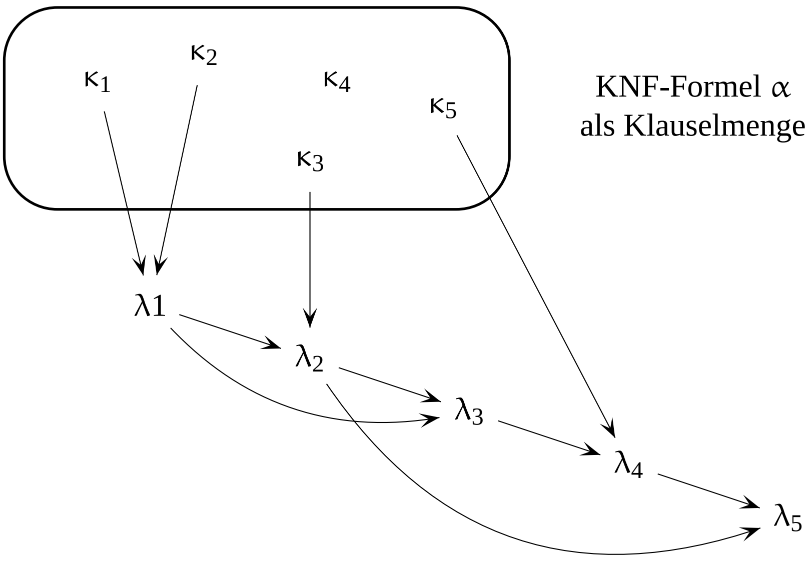


Abbildung 42: Schema einer linearen Resolution

hat. Eine lineare Widerlegung für α bezeichnet eine lineare Herleitung der leeren Klausel $\sqcup$ aus α. Die in Abbildung 41 auf Seite 177 angegebene P-Widerlegung ist in der Tat zugleich eine lineare Widerlegung. Lineare Widerlegungen sind zwar leichter lesbar, jedoch sind sie oftmals wesentlich länger (im schlimmsten Fall exponentiell länger) als kürzeste gewöhnliche Widerlegungen. Wir erwähnen ohne Beweis, dass jede Widerlegung im Wesentlichen durch Vertauschen von Resolutionsschritten linearisiert werden kann, woraus sich die Widerlegungsvollständigkeit der linearen Resolution ergibt. Die Korrektheit ist aufgrund des Resolutionssatzes klar.

Input-Resolution. Ein Spezialfall linearer Resolution ist die sogenannte Input-Resolution, in der in jedem Resolutionsschritt eine der Klauseln κ der zugrundegelegten KNF-Formel α als Elternklausel eingesetzt werden muss. Eine Input-Herleitung aus einer KNF-Formel α ist eine lineare Herleitung

$$\langle \kappa_0, \kappa_1, \lambda_1 \rangle \ \langle \lambda_1, \kappa_2, \lambda_2 \rangle \ \ldots \ \langle \lambda_{m-1}, \kappa_m, \lambda_m \rangle$$

aus α, so dass $\kappa_0, \kappa_1, \ldots, \kappa_m$ Klauseln von α sind. Input-Widerlegungen sind Input-Herleitungen der leeren Klausel. Am Beispiel der unerfüllbaren KNF-Formel α mit der induzierten Klauselmenge

$$\big\{\{x,y\},\{\neg x,y\},\{x,\neg y\},\{\neg x,\neg y\}\big\}$$

kann man sich klarmachen, dass die Input-Resolution *nicht widerlegungsvollständig* ist, da in jedem Resolutionsschritt, in dem eine Klausel λ mit einer Klausel κ in α resolviert wird, eine Klausel mit mindestens einem Literal entsteht. Man beachte, dass alle Klauseln in α zweielementig sind. Resolvierung mit einer Klausel κ in α als Elternklausel liefert daher eine Klausel, die wenigstens eines der Literale von κ enthält. Die leere Klausel kann daher nicht durch Input-Herleitungen aus α generiert werden. Dennoch spielt die Input-Resolution eine wichtige Rolle für die Logikprogrammierung, da sie – wie wir im nächsten Abschnitt sehen werden – für die Klasse der Hornformeln widerlegungsvollständig ist.

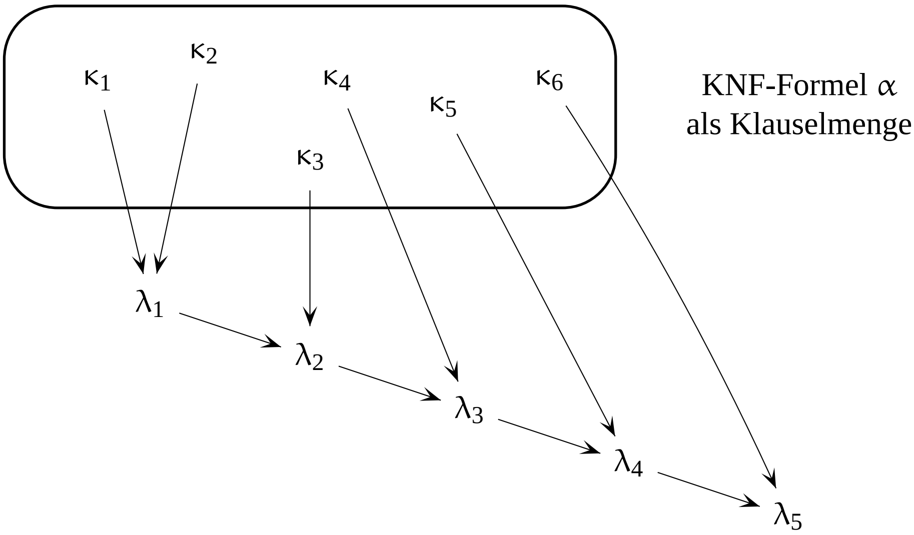


Abbildung 43: Schema einer Input-Resolution

Resolution für Hornformeln. Im Kontext der Logikprogrammierung spielt eine spezielle Resolutionsstrategie für Hornformeln eine wichtige Rolle, die auf einer Variante der linearen Resolution beruht. Hierauf gehen wir nicht ein, sondern beenden diesen Abschnitt mit ein paar einfachen Bemerkungen zur Resolution für Hornformeln.

Lemma 4.59. *Der Resolvent zweier Hornklauseln ist wieder eine Hornklausel.*

Beweis. Seien κ und τ zwei Hornklauseln und λ ein Resolvent von κ und τ; etwa

$$\lambda \;=\; \kappa \setminus \{x\} \;\cup\; \tau \setminus \{\neg x\}$$

wobei $x \in \kappa$ und $\neg x \in \tau$. Die Elternklausel τ enthält höchstens ein positives Literal, während κ außer x kein weiteres positives Literal enthält. Daher enthält auch λ höchstens ein positives Literal. $\square$

Z.B. haben die Klauseln $x \wedge y \to z \;\equiv\; \neg x \vee \neg y \vee z$ und $z \wedge w \to \mathit{false} \;\equiv\; \neg z \vee \neg w$ den Resolventen

$$x \wedge y \wedge w \to \mathit{false} \;\equiv\; \neg x \vee \neg y \vee \neg w.$$

Die Resolventenbildung für Hornklauseln ist sehr eingeschränkt. Je zwei Zielklauseln (also Hornklauseln, die kein positives Literal enthalten) haben keine Resolventen. Ebenso wenig können Resolventen aus zwei Fakten (positiven Einheitsklauseln) gebildet werden. Zwei Regeln haben nur dann einen Resolventen, wenn der Kopf der einen Regel im Rumpf der anderen Regel vorkommt. Z.B. haben

$$x \wedge y \to v \equiv \neg x \vee \neg y \vee v \quad \text{und} \quad x \wedge z \to w \equiv \neg x \vee \neg z \vee w$$

keinen Resolventen.

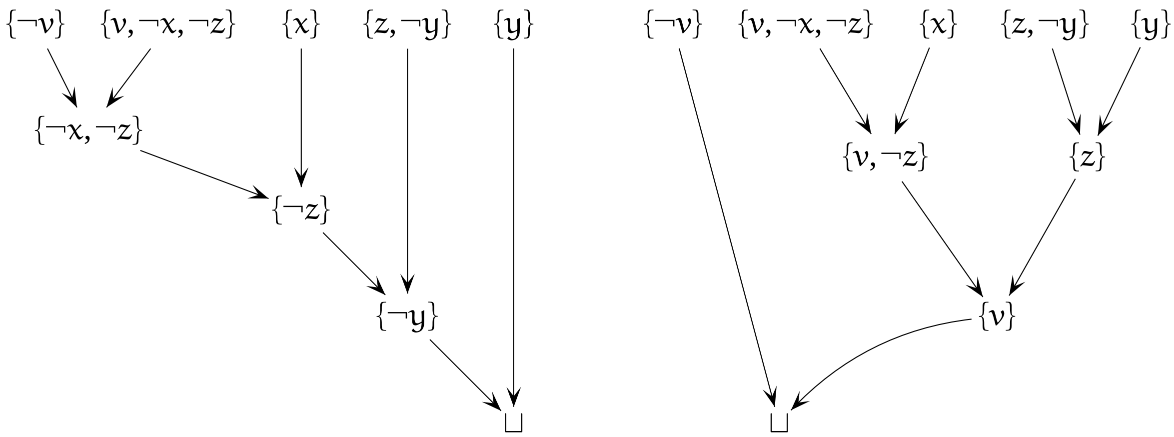
Resolutionsstrategien für Hornformeln. Da die N-Resolution für alle Klauselmengen widerlegungsvollständig ist, trifft dies natürlich auch auf Hornformeln zu. Für unerfüllbare Hornformeln kann daher stets eine Widerlegung angegeben werden, so dass in jedem Resolutionsschritt eine Zielklausel als Elternklausel eingesetzt wird. Resolventenbildungen zwischen zwei definiten Programmklauseln (Fakten oder Regeln) müssen also nicht betrachtet werden.

Aus der Widerlegungsvollständigleit der P-Resolution ergibt sich, dass unerfüllbare Hornformeln stets eine Widerlegung haben, in der in jedem Resolutionsschritt ein Faktum als Elternklausel eingesetzt wird.

Als Beispiel betrachten wir die unerfüllbare Hornformel

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

Das Bild unten zeigt links eine N-Widerlegung und rechs eine P-Widerlegung für α. In diesem Beispiel haben die N- und P-Widerlegung für α also völlig unterschiedliche Struktur.



Ist eine der Elternklauseln eine Zielklausel und die andere Elternklausel entweder eine Regel oder ein Faktum, so enthält der Resolvent kein positives Literal. Daher sind N-Resolventen, die aus den Klauseln einer Hornformel gebildet werden, stets Zielklauseln. Jede N-Herleitung induziert daher eine Input-Herleitung.[18] Aus der Widerlegungsvollständigkeit der N-Resolution folgt daher die Widerlegungsvollständigkeit der Input-Resolution für Hornformeln.

The End

[18]Zunächst müssen N-Herleitungen nicht linear sein. Liegt eine N-Herleitung aus einer Hornformel vor, so kann diese durch Streichen von Resolutionsschritten, die für die hergeleitete Hornklausel irrelevant sind, in eine lineare N-Herleitung mit derselben hergeleiteten Klausel überführt werden. Diese ist dann zugleich eine Input-Herleitung.