1 Aussagenlogik

1.1 Grundbegriffe der Aussagenlogik

atomare/elementare sprachliche Gebilde:

A = „Borussia Dortmund ist deutscher Fußballmeister“

B = „Würzburg ist eine große Stadt“

C = „Würzburg hat über 300.000 Einwohner“

Man nennt diese Gebilde auch *atomare Formeln* oder *Atome*.

Atome können – je nach Interpretation – wahr oder falsch sein.

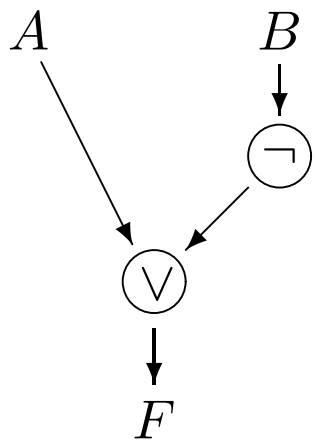
In der Praxis hat man oft eine Standard–Interpretation im Sinne, von der wir in der Logik aber absehen.

Wir bezeichnen alle denkbaren Atome mit Buchstaben $A, B, C, \ldots,$ oder durchnummeriert mit $A_1, A_2, \ldots,$ oder mit allgemeinen Zeichenketten ('Erkältung'), ohne ihnen eine inhaltliche Interpretation zu geben.

Wir interessieren uns dafür, wie sich die Wahrheitswerte von komplexeren Formeln aus den Wahrheitswerten der atomaren Formeln zusammensetzen:

$$F = A \vee \neg B$$

Baumdarstellung:



Wahrheitswerte:

- 0 (falsch),
- 1 (wahr)

4 mögliche Interpretationen:

	A	B	F
I_1	0	0	1
I_2	1	0	1
I_3	0	1	0
I_4	1	1	1

Für jede Belegung I (Interpretation) der beteiligten Atome A und B mit Wahrheitswerten, ergibt sich der Wahrheitswert $I(F)$ für die Formel F.

Definition (Syntax der Aussagenlogik)

Aufbau (komplexer) Formeln

1. Alle atomaren Formeln sind Formeln.
   Die Menge der atomaren Formeln bezeichnen wir mit $\mathbb{A}$.

2. Für alle Formeln F und G sind auch
   - die Konjunktion $(F \wedge G)$ und
   - die Disjunktion $(F \vee G)$

   Formeln. Die Klammen kann man meist weglassen.

3. Für jede Formel F ist auch die Negation $\neg F$ wieder eine Formel.

Die Menge aller Formeln bezeichnen wir mit $\mathbb{F}$. Es gilt $\mathbb{A} \subseteq \mathbb{F}$.

Eine Formel F, die als Teil einer Formel G auftritt, heißt Teilformel von G.

Beispiel (Formeln, Teilformeln)

- $G = (A \vee \neg B)$
  hat die Teilformeln A, B, $\neg B$, G
- $F = ((A \wedge B) \vee (\neg A \wedge \neg B))$
  hat die Teilformeln $A,\ B,\ \neg A,\ \neg B,\ (A \wedge B),\ (\neg A \wedge \neg B),\ F$

Abkürzende Schreibweisen:

$(F_1 \rightarrow F_2)$ für $(\neg F_1 \vee F_2)$ (Implikation)

$(F_1 \leftrightarrow F_2)$ für $((F_1 \wedge F_2) \vee (\neg F_1 \wedge \neg F_2))$ (Äquivalenz)

$(\bigvee_{i=1}^{n} F_i)$ für $(\ldots((F_1 \vee F_2) \vee F_3) \vee \ldots \vee F_n)$

$(\bigwedge_{i=1}^{n} F_i)$ für $(\ldots((F_1 \wedge F_2) \wedge F_3) \wedge \ldots \wedge F_n)$

Wegen der *Kommutativität* und der *Assoziativität* der Junktoren $\wedge$ und $\vee$, kommt es auf die Reihenfolge und Klammerung in den folgenden Formeln nicht an:

$(F_1 \wedge F_2)$ ist gleichwertig zu $(F_2 \wedge F_1)$,
$(F_1 \vee F_2)$ ist gleichwertig zu $(F_2 \vee F_1)$,
$((F_1 \wedge F_2) \wedge F_3)$ ist gleichwertig zu $(F_1 \wedge (F_2 \wedge F_3))$,
$((F_1 \vee F_2) \vee F_3)$ ist gleichwertig zu $(F_1 \vee (F_2 \vee F_3))$.

Es wird sich später zeigen, daß folgende Formeln unter jeder denkbaren Interpretation I wahr sind; man nennt sie *Tautologien*:

$((F_1 \wedge F_2) \leftrightarrow (F_2 \wedge F_1))$,
$((F_1 \vee F_2) \vee F_3) \leftrightarrow (F_1 \vee (F_2 \vee F_3))$,
$(\neg(F_1 \wedge F_2) \leftrightarrow (\neg F_1 \vee \neg F_2))$.

Die letzte Formel entspricht einer der beiden Regeln von *De Morgan*.

Wir schreiben $F_1 \equiv F_2$, falls $I(F_1) = I(F_2)$, für alle passenden Interpretationen I. Dann gilt für $\otimes \in \{\wedge, \vee\}$:

- Kommutatitvität: $F_1 \otimes F_2 \equiv F_2 \otimes F_1$,
- Assoziativität: $(F_1 \otimes F_2) \otimes F_3 \equiv F_1 \otimes (F_2 \otimes F_3)$.

Die beiden arithmetischen Operatoren Multiplikation und Addition sind ebenfalls kommutativ und assoziativ, die Division allerdings nicht:

$$8 \div 4 = 2 \neq 0.5 = 4 \div 8,$$
$$(8 \div 4) \div 2 = 1 \neq 4 = 8 \div (4 \div 2).$$

Es gilt wie in der Arithmetik die Regel “Punkt vor Strich”.
Dabei entspricht die Konjunktion $\wedge$ der Multiplikation (Punkt),
und die Disjunktion $\vee$ entspricht der Addition (Strich). Also gilt:

- $F_1 \wedge F_2 \vee F_3$ entspricht $(F_1 \wedge F_2) \vee F_3$.
- Die andere Bindung $F_1 \wedge (F_2 \vee F_3)$ erreicht man durch Klammerung.

Wahrheitswerte: 0,1

Boolesche Operatoren: $\wedge$, $\vee$, $\neg$

Verknüpfungstafeln:

$\wedge$	0	1
0	0	0
1	0	1

$\vee$	0	1
0	0	1
1	1	1

$\neg$	0	1
	1	0

abgeleitete Verknüpungstafeln:

$F \rightarrow G$	0	1
0	1	1
1	0	1

$F \leftrightarrow G$	0	1
0	1	0
1	0	1

Definition (Semantik der Aussagenlogik)

Eine Interpretation ist eine Abbildung

$$I : D \to \{0, 1\},$$

die zunächst für eine Teilmenge $D \subseteq \mathbb{A}$ der atomaren Formeln definiert ist (Belegung). Sie kann wie folgt zu einer Abbildung

$$I : \mathbb{F} \to \{0, 1\}$$

auf der Menge $\mathbb{F}$ aller Formeln erweitert werden:

1. Für atomare Formeln $A \in D$ ist I bereits definiert.
2. Für Konjunktionen und Disjunktionen gilt

   $$I((F \otimes G)) = I(F) \otimes I(G),$$

   für $\otimes \in \{\wedge, \vee\}$ (Homomorphismus).

D.h.

$$I((F \wedge G)) = \begin{cases} 1, & \text{falls } I(F) = 1 \text{ und } I(G) = 1 \\ 0, & \text{sonst} \end{cases}$$

$$I((F \vee G)) = \begin{cases} 1, & \text{falls } I(F) = 1 \text{ oder } I(G) = 1 \\ 0, & \text{sonst} \end{cases}$$

3. Für Formel $\neg F$ gilt $I(\neg F) = \neg I(F)$.

   D.h.

$$I(\neg F) = \begin{cases} 1 & \text{falls } I(F) = 0 \\ 0, & \text{sonst} \end{cases}$$

Induktive Definition:

„von einfachen Formeln auf komplexere Formeln gehen“

Eine Interpretation I heißt zu einer Formel F *passend*, falls I für alle in F vorkommenden atomaren Formeln A definiert ist (d.h. $A \in D$).

Dabei ist die Erweiterung von I auch auf F definiert, d.h. $I(F) \in \{0, 1\}$. Ansonsten erklären wir I auf F als undefiniert.

Beispiel (Interpretationen)

Zur Formel

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

sind die folgenden beiden Interpretation I und J passend:

I	A	B	C
	1	1	0

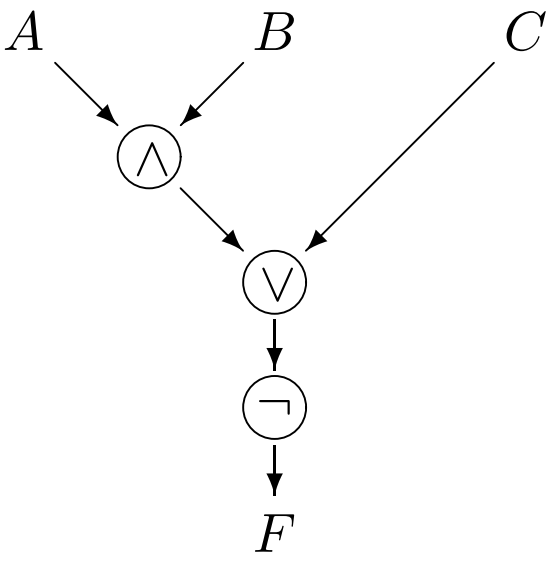
J	A	B	C	D
	1	1	0	1

I und J definieren alle in F vorkommenden Atome, nämlich A, B und C. J definiert sogar noch ein weiteres Atom D.

Formel:

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

Baumdarstellung:

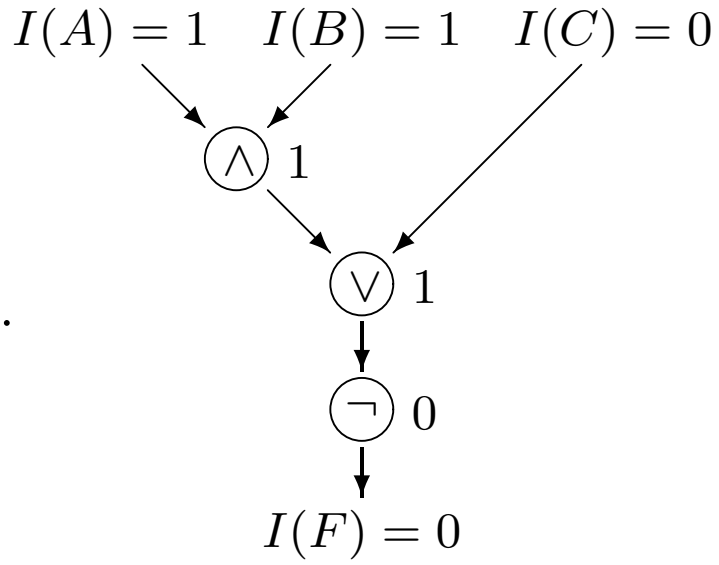


Interpretation I:

I	A	B	C
	1	1	0

Auswertung der Formel:

$$
\begin{aligned}
I(\neg((A \wedge B) \vee C)) \;&=\\
\neg I((A \wedge B) \vee C) \;&=\\
\neg(I(A \wedge B) \vee I(C)) \;&=\\
\neg\underbrace{(\underbrace{(\underbrace{I(A)}_{1} \wedge \underbrace{I(B)}_{1})}_{1} \vee \underbrace{I(C)}_{0})}_{1} \;&=\; 0.
\end{aligned}
$$



Boolesche Funktion

Wenn man eine Formel F als boolesche Funktion f auffaßt, die zu gegebenen Wahrheitswerten $a_i = I(A_i)$ für die in F vorkommenden Atome A_i einen Wahrheitswert berechnet, dann gilt

$$f(a_1, \ldots, a_n) = I(F).$$

Also kann man z.B. für $F = \neg((A \wedge B) \vee C)$ schreiben:

$$\begin{aligned} f(a, b, c) &= \neg((a \wedge b) \vee c), \\ f(1, 1, 0) &= \neg((1 \wedge 1) \vee 0) = 0. \end{aligned}$$

Die Interpretation I wird also zu einer Variablenbelegung $(1, 1, 0)$:

I	A	B	C
	1	1	0

Exkurs: Auswertung in PROLOG

In der Logikprogrammierung werden auch die Rückgabewerte von Berechnungen als Parameter der zugehörigen Prädikate behandelt:

```
/* boolean_and(U, V, W) <-
      W is derived as the boolean conjunction
      of the truth values U and V. */

boolean_and(0, 0, 0).
boolean_and(0, 1, 0).
boolean_and(1, 0, 0).
boolean_and(1, 1, 1).
```

Das Prädikat `boolean_and/3` ist hier als Wahrheitswertetabelle realisiert.

Berechnungsbeispiel:

```
?- boolean_and(1, 0, W).
W = 0 .
```

Man kann die Berechnung sogar umkehren, und nach allen Paaren von Wahrheitswerten `U` und `V` fragen, deren Konjunktion `0` ist:

```
?- boolean_and(U, V, 0).
U = 0, V = 0 ;
U = 0, V = 1 ;
U = 1, V = 0
```

Dabei wird durch die Eingabe von `;` hinter einer Lösung die Berechnung der nächsten Lösung angestoßen.

Sei die Liste `Assignment = [A`$_1$`=v`$_1$`,...,A`$_n$`=v`$_n$`]` eine Zuweisung von Wahrheitswerten `v`$_i$ an die atomaren Formeln `A`$_i$.

Der Wahrheitswert einer Konjunktion `(A,B)` ergibt sich durch Konjunktion der Wahrheitswerte der beiden Glieder `A` und `B`:

```
evaluate_boolean_formula(
      Assignment, (A,B), Value) :-
   evaluate_boolean_formula(Assignment, A, V_A),
   evaluate_boolean_formula(Assignment, B, V_B),
   boolean_and(V_A, V_B, Value).
```

Zunächst werden die Wahrheitswerte `V_A` und `V_B` der beiden Glieder `A` und `B` berechnet, danach deren Konjunktion.

Im DDK ist auch folgende *funktionale* Schreibweise möglich:

```
evaluate_boolean_formula(
      Assignment, (A,B), Value) :-
   Value <= boolean_and(
      evaluate_boolean_formula(Assignment, A),
      evaluate_boolean_formula(Assignment, B) ).
```

Jetzt wird direkt ein verschachtelter Ausdruck ausgewertet, anstelle mittels `evaluate_boolean_formula/3` (unten abgekürzt: $eval$) vorher die Zwischenresultate `V_A` und `V_B` zu erzeugen. Zugehörige Umformung:

$$eval(I, A, V_A),\ eval(I, B, V_B),\ and(V_A, V_B, V)\ \mapsto$$
$$V_A \texttt{<=}\ eval(I, A),\ V_B \texttt{<=}\ eval(I, B),\ V \texttt{<=}\ and(V_A, V_B)\ \mapsto$$
$$V \texttt{<=}\ and(eval(I, A), eval(I, B)).$$

I steht für `Assignment` und V für `Value`.

Der Wahrheitswert einer Disjunktion `(A;B)` ist `1`, falls der Wahrheitswert von mindestens einem der beiden Glieder `A` und `B` gleich `1` ist, sonst ist er `0`:

```
evaluate_boolean_formula(
      Assignment, (A;B), Value) :-
   evaluate_boolean_formula(Assignment, A, V_A),
   evaluate_boolean_formula(Assignment, B, V_B),
   boolean_or(V_A, V_B, Value).

boolean_or(0, 0, 0).
boolean_or(0, 1, 1).
boolean_or(1, 0, 1).
boolean_or(1, 1, 1).
```

Der Wahrheitswert einer Negation `-A` ergibt sich aus dem Wahrheitswert von `A`; für eine atomare Formel `A` wird der Wahrheitswert `Value` mittels `member(A=Value, Assignment)` in `Assignment` nachgeschaut:

```
evaluate_boolean_formula(Assignment, -A, Value) :-
   evaluate_boolean_formula(Assignment, A, V_A),
   boolean_not(V_A, Value).
evaluate_boolean_formula(Assignment, A, Value) :-
   boolean_formula_is_atomic(A),
   member(A=Value, Assignment).

boolean_not(0, 1).
boolean_not(1, 0).
```

Mittels `boolean_formula_is_atomic(A)` wird getestet, ob `A` eine atomare Formel ist, d.h. ohne Konjunktion, Disjunktion bzw. Negation.

Berechnungsbeispiel:

```
?- Formula = -((a,b;c)),
   Assignment = [a=1, b=1, c=0],
   evaluate_boolean_formula(
      Assignment, Formula, Value).
Value = 0
```

Es gilt “Punkt vor Strich”, d.h.

`-((a,b;c))` bedeutet `-((a,b);c)` bzw. $\neg((a \wedge b) \vee c)$.

Die zusätzliche äußere Klammerung in `-((a,b;c))` ist erforderlich.

In der ungeklammerten Formel `-(a,b;c)` faßt PROLOG das `-` als binären Operator mit den zwei Argumenten `a` und `b;c` auf.

Trace in PROLOG:

```
?- trace.
true.

[trace]  ?- eval([a=1, b=1, c=0], -((a,b;c)), Value).

   Call: (7) eval(..., - (a,b;c), _G402) ? creep
   Call: (8) eval(..., (a,b;c), _G573) ? creep
   Call: (9) eval(..., (a,b), _G573) ? skip
   Exit: (9) eval(..., (a,b), 1) ? creep
   Call: (9) eval(..., c, _G579) ? skip
   Exit: (9) eval(..., c, 0) ? creep
   Call: (9) or(1, 0, _G582) ? skip
   Exit: (9) or(1, 0, 1) ? creep
   Exit: (8) eval(..., (a,b;c), 1) ? creep
```

```
   Call: (8) not(1, _G402) ? skip
   Exit: (8) not(1, 0) ? creep
   Exit: (7) eval(..., - (a,b;c), 0) ? creep
Value = 0 .

[trace]  ?-
```

Aus Platzgründen haben wir die Prädikatensymbole abgekürzt und die Variablenbelegung `[a=1, b=1, c=0]` in `eval/3` mittels `...` angedeutet.

Modelle

Definition (Modelle)

Sei F eine Formel und $I : D \to \{0, 1\}$ eine Interpretation.

1. I ist ein *Modell* von (für) F, falls
   - I zu F passend ist und
   - $I(F) = 1$ gilt.

   Schreibweise: $I \models F$.
   Dann sagen wir auch: F *gilt* unter (in, bezüglich) I.

2. Andernfalls ist I *kein Modell* von (für) F.
   Schreibweise: $I \not\models F$.
   Dann sagen wir auch: F gilt unter I nicht.

Beispiel (Modelle)

Die folgende Formel F repräsentiert des exklusive Oder:

$$F \;=\; (A \vee B) \wedge (\neg A \vee \neg B) \;\equiv\; A \oplus B.$$

Interpretationen:

I_1	A	B
	0	1

I_2	A	B
	1	1

I_1 und I_2 sind passend zu F, aber nur I_1 ist ein Modell für F:

$I_1 \models F,\; I_2 \not\models F.$

Es gibt hier noch ein weiteres Modell I_3 für F:

I_3	A	B
	1	0

Man kann alle möglichen passenden Interpretationen über den in F vorkommenden atomaren Formeln in einer Wahrheitswertetafel zusammenstellen:

	A	B	F
I_4	0	0	0
I_1	0	1	1
I_3	1	0	1
I_2	1	1	0

In der Spalte für F wird der Wahrheitswert $I(F)$ angegeben.

I ist ein Modell für F, g.d.w. $I(F) = 1$ ist.

Definition (Erfüllbarkeit, Gültigkeit)

1. Eine Formel F heißt *erfüllbar*, falls F mindestens ein Modell besitzt, andernfalls *unerfüllbar*.

2. Eine Menge M von Formeln heißt *erfüllbar*, falls es eine Interpretation I gibt, welche für alle Formeln $F \in M$ ein Modell ist:

   $$\exists(I: \ D \to \{0,1\}) : \forall F \in M : I \models F$$

   Insbesondere muß I natürlich auch passend sein zu allen $F \in M$.

3. Eine Formel F heißt *gültig* (oder *Tautologie*), falls jede zu F passende Interpretation ein Modell für F ist.

   Dann schreiben wir auch $\models F$; anderenfalls $\not\models F$.

Beispiel (Erfüllbarkeit, Gültigkeit)

1. Die Formel $F = A \vee B$ ist *erfüllbar:*

A	B	F
0	0	0
0	1	1
1	0	1
1	1	1

2. Die Formel $G = A \vee \neg A$ ist *gültig* (Tautologie);
   die Formel $H = \neg G = \neg(A \vee \neg A)$ ist *unerfüllbar:*

A	G	H
0	1	0
1	1	0

Die beiden passenden Interpretationen I_1 und I_2 erfüllen G:
$I_1 \models G,\ I_2 \models G.$

I_1	A
	0

I_2	A
	1

3. Die zu H äquivalente Formel $H' = \neg A \wedge A$ ist ebenfalls unerfüllbar.

4. Die Formel $F = (A \vee B) \wedge (\neg A \vee \neg B)$ zum exklusiven Oder ist erfüllbar, aber keine Tautologie:

A	B	F
0	0	0
0	1	1
1	0	1
1	1	0

Zum Feststellen der Erfüllbarkeit bzw. Gültigkeit einer Formel F genügt es, die endlich vielen Interpretationen, die genau auf den in F vorkommenden atomaren Formeln definiert sind, zu testen.

Falls die Formel n verschiedene atomare Formeln enthällt, so sind dies genau 2^n Interpretationen.

Wahrheitswertetafeln:

$$\begin{aligned} F_1 &= A \to B, \\ F_2 &= (A \wedge (A \to B)) \to B, \\ F_3 &= A \vee \neg A, \\ F_4 &= (A \wedge B) \vee (A \wedge \neg C). \end{aligned}$$

F_2 und F_3 sind offenbar Tautologien.

A	B	F_1	$A \wedge F_1$	F_2
0	0	1	0	1
0	1	1	0	1
1	0	0	0	1
1	1	1	1	1

A	F_3
0	1
1	1

A	B	C	F_4
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

Offenbar gilt: $I(F_1) = I(A \rightarrow B) = 1$, genau dann, wenn $I(A) \leq I(B)$.

Exkurs: Berechnung von Wahrheitswertetafeln in PROLOG

```
?- Formula = -((a,b;c)),
   Atoms = [a,b,c],
   boolean_formula_to_truth_table(Formula, Atoms).
```

a	b	c	F
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

Mittels `boolean_formula_to_models(Formula, Atoms, Models)` werden nur die Modelle ausgegeben.

Satz (Tautologie, Erfüllbarkeit)

Eine Formel F ist genau dann eine *Tautologie*, wenn $\neg F$ *unerfüllbar* ist.

Beweis:

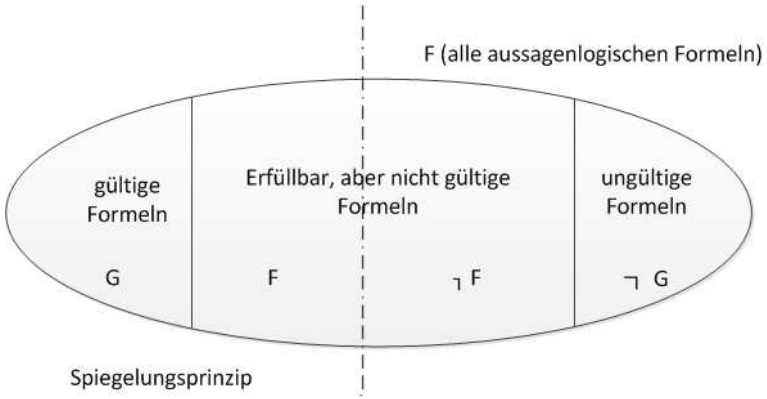
F ist eine Tautologie:

$\Leftrightarrow$ jede zu F passende Interpretation ist ein Modell für F

$\Leftrightarrow$ jede zu $\neg F$ passende Interpretation ist kein Modell für $\neg F$

$\Leftrightarrow$ $\neg F$ besitzt kein Modell

$\Leftrightarrow$ $\neg F$ ist unerfüllbar $\square$



1.2 Äquivalenz und Normalformen

Im folgenden lassen wir – falls möglich – die äußersten Klammern einer Formel meist weg.

Definition ($F \equiv G$)

Zwei Formeln F und G heißen (semantisch) äquivalent, falls für alle Interpretation I,

- die sowohl für F als auch für G passend sind,
- $I(F) = I(G)$ gilt.

Beispiel (Äquivalenz)

1. $A \vee B \equiv B \vee A$ (Kommutativität)
2. $A \vee \neg A \equiv (A \wedge (A \rightarrow B)) \rightarrow B$ (beides Tautologien)

Ersetzbarkeit

Sei $F \equiv F'$, und H eine Formel, in der F als Teilformel auftritt.
Sei H' die Formel, welche man durch Ersetzen aller Vorkommen von F in H durch F' erhält.
Dann gilt $H \equiv H'$.

Beispiel (Ersetzbarkeit)

Aus $F \equiv F'$ folgt

$$F \wedge G \equiv F' \wedge G,$$
$$(F \wedge G) \rightarrow K \equiv (F' \wedge G) \rightarrow K.$$

Satz (Äquivalenz)

Indempotenz	$F \otimes F \equiv F$, für $\otimes \in \{\wedge, \vee\}$,
Absorption	$F \wedge (F \vee G) \equiv F \equiv F \vee (F \wedge G)$,
Kommutativität	$F \otimes G \equiv G \otimes F$, für $\otimes \in \{\wedge, \vee\}$,
Assoziativität	$(F \otimes G) \otimes H \equiv F \otimes (G \otimes H)$, für $\otimes \in \{\wedge, \vee\}$,
Distributivität	$F \wedge (G \vee H) \equiv (F \wedge G) \vee (F \wedge H)$,
	$F \vee (G \wedge H) \equiv (F \vee G) \wedge (F \vee H)$,

Doppelnegation	$\neg(\neg F) \equiv F,$
De Morgan–Regeln	$\neg(F \wedge G) \equiv \neg F \vee \neg G,$
	$\neg(F \vee G) \equiv \neg F \wedge \neg G,$
Resolution	$F \equiv (F \vee G) \wedge (F \vee \neg G),$
Tautologie–Regeln	$F \vee G \equiv F,\ F \wedge G \equiv G,$
	falls F eine Tautologie ist.
Unerfüllbarkeits–Regeln	$F \vee G \equiv G,\ F \wedge G \equiv F,$
	falls F unerfüllbar ist.

Beweis:

Alle Äquivalenzen können leicht mittels Wahrheitstafeln nachgeprüft werden:

F	G	$F \wedge G$	$\neg(F \wedge G)$	$\neg F$	$\neg G$	$\neg F \vee \neg G$	$\neg\neg F$
0	0	0	1	1	1	1	0
0	1	0	1	1	0	1	0
1	0	0	1	0	1	1	1
1	1	1	0	0	0	0	1

Man erkennt:

(DM) $\neg(F \wedge G) \equiv \neg F \vee \neg G$,

(DN) $\neg(\neg F) \equiv F$.

Die duale Aussage von (DM) zeigen wir durch Einsetzen von $\neg F'$ für F und von $\neg G'$ für G und anschließende Doppelnegation.

Aufgrund

- der ersten De Morgan–Regel (DM),
- der Regel für Doppelnegation (DN), und
- der Regel zur Ersetzbarkeit (E)

erhalten wir nun folgendes:

$$
\begin{aligned}
\neg F' \wedge \neg G' \quad & \overset{DN}{\equiv} \quad \neg\neg(\neg F' \wedge \neg G') \\
& \overset{DM}{\equiv} \quad \neg(\underbrace{\neg\neg F'}_{\underset{DN}{\equiv} F'} \vee \underbrace{\neg\neg G'}_{\underset{DN}{\equiv} G'}) \\
& \overset{E}{\equiv} \quad \neg(F' \vee G')
\end{aligned}
$$

u.s.w. $\square$

Bemerkungen:

(i) Das Assoziativgesetz erlaubt es bei mehrgliedrigen Konjunktionen bzw. Disjunktionen auf die Klammerung zu verzichten: z.B. für $\wedge$:

$$A \wedge B \wedge C \wedge D \equiv (A \wedge B) \wedge (C \wedge D) \equiv A \wedge (B \wedge C) \wedge D \equiv$$
$$A \wedge (B \wedge (C \wedge D)) \equiv ((A \wedge B) \wedge C) \wedge D \equiv (A \wedge (B \wedge C)) \wedge D$$

(ii) Aufgrund der Kommutativität wären sogar beliebige Permutationen von A, B, C, D möglich:

$$A \wedge B \wedge C \wedge D \equiv B \wedge D \wedge C \wedge A \equiv \ldots$$

(iii) Die Distributivität gilt hier symmetrisch für $\wedge$ und $\vee$, im Gegensatz zu „ $\cdot$ “ und „$+$“:

$$a \cdot (b + c) = a \cdot b + a \cdot c.$$

Vertauscht mann $\cdot$ und $+$, so gilt die arithmetische Regel nicht.

Definition (Literal)

Sei A eine atomare Formel.

1. Dann sind A (positives Literal) und seine Negation $\neg A$ (negatives Literal) *Literale*.
2. Man nennt die beiden Literale A und $\neg A$ *komplementär*.

Seien die L_i, mit $1 \leq i \leq m$, Literale.

1. Die Konjunktion $\bigwedge_{i=1}^{m} L_i$ ist unerfüllbar,
   falls sie zwei komplementäre Literale enthält.
2. Die *leere Konjunktion* $\square_K = \bigwedge_{i=1}^{0} L_i$ ist eine Tautologie.
3. Die Disjunktion $\bigvee_{i=1}^{m} L_i$ ist eine Tautologie,
   falls sie zwei komplementäre Literale enthält.
4. Die *leere Disjunktion* $\square_D = \bigvee_{i=1}^{0} L_i$ ist unerfüllbar.

Normalformen

Definition (Normalformen)

Gegeben seien Literale L_{ij}, für $1 \leq i \leq m$, $1 \leq j \leq n_i$.

1. Eine Formel F ist in *konjunktiver Normalform (KNF)*, falls sie eine Konjunktion von Disjunktionen ist:

   $$F = \bigwedge_{i=1}^{m} (\bigvee_{j=1}^{n_i} L_{ij}).$$

2. Eine Formel F ist in *disjunktiver Normalform (DNF)*, falls sie eine Disjunktion von Konjunktionen ist:

   $$F = \bigvee_{i=1}^{m} (\bigwedge_{j=1}^{n_i} L_{ij}).$$

Falls F eine Formel über n Atomen ist, so genügt es $n_i \leq n$ zu wählen. Andernfalls sind die Teil–Disjunktionen/Konjunktionen redundant, da sie komplementäre Literale enthalten.

Falls zwei Literale innerhalb derselben

1. Disjunktion $D_i = \bigvee_{j=1}^{n_i} L_{ij}$ komplementär sind, so kann man die Tautologie D_i aus $F = \bigwedge_{i=1}^{m}(\bigvee_{j=1}^{n_i} L_{ij})$ entfernen.
2. Konjunktion $K_i = \bigwedge_{j=1}^{n_i} L_{ij}$ komplementär sind, so kann man die unerfüllbare Teilformel K_i aus $F = \bigvee_{i=1}^{m}(\bigwedge_{j=1}^{n_i} L_{ij})$ entfernen.

In beiden Fällen erhält man eine äquivalente, kürzere Formel F'.

Beispiel (Komplementäre Literale in Normalform)

$$(A \vee B) \wedge (B \vee \neg B) \equiv (A \vee B).$$
$$(A \wedge B) \vee (B \wedge \neg B) \equiv (A \wedge B).$$

Beispiel (Normalformen)

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

A	B	C	F		
0	0	0	1		
$\boxed{0}$	$\boxed{0}$	$\boxed{1}$	1	$\longrightarrow$	in DNF: $\neg A \wedge \neg B \wedge C$
0	1	0	1		
0	1	1	1		
1	0	0	1		
$\boxed{1}$	$\boxed{0}$	$\boxed{1}$	0	$\longrightarrow$	in KNF: $\neg A \vee B \vee \neg C$
1	1	0	0		
1	1	1	0		

KNF:

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

DNF:

$$\begin{aligned} F_D &= (\neg A \wedge \neg B \wedge \neg C) \vee (\neg A \wedge \neg B \wedge C) \vee (\neg A \wedge B \wedge \neg C) \vee \\ &\quad (\neg A \wedge B \wedge C) \vee (A \wedge \neg B \wedge \neg C) \end{aligned}$$

DNF und KNF sind nicht eindeutig:

Durch Umformung von F erhalten wir hier eine weitere, kompaktere zu F äquivalente DNF und KNF:

$$\begin{aligned} F &= \neg(A \wedge (B \vee C)) \equiv \neg A \vee \neg(B \vee C) \\ &\equiv \underbrace{\neg A \vee (\neg B \wedge \neg C)}_{DNF} \equiv \underbrace{(\neg A \vee \neg B) \wedge (\neg A \vee \neg C)}_{KNF} \end{aligned}$$

Die KNF–Darstellung aufgrund der Nullstellen erinnert an die Polynomdarstellung aus der Analysis.

Ein Polynom vom Grad 2 mit den zwei Nullstellen 2 und 3 und dem führenden Koeffizienten 1 hat die Darstellung

$$p(X) \;=\; (X-2)\cdot(X-3).$$

Durch die Produktdarstellung wird erreicht, daß das Polynom genau für die beiden Nullstellen den Wert 0 hat, denn ein Produkt ist genau dann 0, wenn einer der beiden Faktoren 0 ist.

Auch wenn das Polynom keine zwei Nullstellen hat, kann man eine ähnliche Überlegung anstellen.

Eine Analogie zur DNF–Darstellung kann man nicht so leicht ziehen.

Wahrheitswertetabelle $\rightarrow$ Normalformen

Zwei Formeln F und G sind (semantisch) äquivalent genau dann, wenn sie dieselben Wahrheitswertetabellen haben.

Da es nur zwei Wahrheitswerte gibt, genügt es bereits, wenn F und G dieselben Einsstellen haben:

$$I(F) = 1 \iff I(G) = 1.$$

Ebenso genügt es, wenn F und G dieselben Nullstellen haben:

$$I(F) = 0 \iff I(G) = 0.$$

Dies muß für alle zu beiden Formeln passenden Interpretationen I gelten.

Gegeben sei eine Formel F über n Atomen $A_1, \ldots, A_n$.

DNF:

Ist I eine Interpretation mit $I(F) = 1$, so bilden wir eine Konjunktion

$$\alpha_I \;=\; \alpha_I^1 \wedge \alpha_I^2 \wedge \ldots \wedge \alpha_I^n$$

der Länge n mit

$$\alpha_I^i \;=\; \begin{cases} A_i, & \text{falls } I(A_i) = 1 \\ \neg A_i, & \text{falls } I(A_i) = 0 \end{cases}$$

Dann ist F_D eine DNF für F:

$$F_D \;=\; \bigvee_{I:\; I(F)=1} \alpha_I .$$

Für eine Interpretation J ist $J(\alpha_I) = 1$ genau dann, wenn $J = I$ ist.
Also haben F und F_D dieselben Einsstellen, und es gilt $F \equiv F_D$.

Aus den Einsstellen von F bildet man die Konjunktionen α_I der DNF:

$$\begin{aligned}
\alpha_{(0,0,0)} &= \neg A \wedge \neg B \wedge \neg C, \\
\alpha_{(0,0,1)} &= \neg A \wedge \neg B \wedge C, \\
\alpha_{(0,1,0)} &= \neg A \wedge B \wedge \neg C, \\
\alpha_{(0,1,1)} &= \neg A \wedge B \wedge C, \\
\alpha_{(1,0,0)} &= A \wedge \neg B \wedge \neg C.
\end{aligned}$$

Wir repräsentieren hier eine Interpretation I durch ein Tupel von Wahrheitswerten.

DNF:

$$\begin{aligned}
F_D &= \alpha_{(0,0,0)} \vee \alpha_{(0,0,1)} \vee \alpha_{(0,1,0)} \vee \alpha_{(0,1,1)} \vee \alpha_{(1,0,0)} \\
&= (\neg A \wedge \neg B \wedge \neg C) \vee (\neg A \wedge \neg B \wedge C) \vee (\neg A \wedge B \wedge \neg C) \vee \\
&\quad (\neg A \wedge B \wedge C) \vee (A \wedge \neg B \wedge \neg C)
\end{aligned}$$

KNF:

Ist I eine Interpretation mit $I(F) = 0$, so bilden wir eine Disjunktion

$$\beta_I \;=\; \beta_I^1 \vee \beta_I^2 \vee \ldots \vee \beta_I^n$$

der Länge n mit

$$\beta_I^i \;=\; \begin{cases} \neg A_i, & \text{falls } I(A_i) = 1 \\ A_i, & \text{falls } I(A_i) = 0 \end{cases}$$

Dann ist F_K eine KNF für F:

$$F_K \;=\; \bigwedge_{I:\; I(F)=0} \beta_I .$$

Für eine Interpretation J ist $J(\beta_I) = 0$ genau dann, wenn $J = I$ ist.
Also haben F und F_K dieselben Nullstellen, und es gilt $F \equiv F_K$.

Aus den Nullstellen von F bildet man die Disjunktionen β_I der KNF:

$$\begin{aligned}
\beta_{(1,0,1)} &= \neg A \vee B \vee \neg C, \\
\beta_{(1,1,0)} &= \neg A \vee \neg B \vee C, \\
\beta_{(1,1,1)} &= \neg A \vee \neg B \vee \neg C.
\end{aligned}$$

Wir repräsentieren hier eine Interpretation I durch ein Tupel von Wahrheitswerten.

KNF:

$$\begin{aligned}
F_K &= \beta_{(1,0,1)} \wedge \beta_{(1,1,0)} \wedge \beta_{(1,1,1)} \\
&= (\neg A \vee B \vee \neg C) \wedge (\neg A \vee \neg B \vee C) \wedge (\neg A \vee \neg B \vee \neg C)
\end{aligned}$$

Eindeutigkeit

Die DNF/KNF, in der jede Konjunktion/Disjunktion die Länge n hat und jedes Atom A_i genau einmal (als A_i oder als $\neg A_i$) enthält, ist eindeutig – abgesehen von der Reihenfolge der Teilformeln (Kommutativität).

$$
\begin{aligned}
F_D &= (\neg A \wedge \neg B \wedge \neg C) \vee (\neg A \wedge \neg B \wedge C) \vee (\neg A \wedge B \wedge \neg C) \vee \\
&\quad (\neg A \wedge B \wedge C) \vee (A \wedge \neg B \wedge \neg C) \\
&\equiv (\neg A \wedge C \wedge \neg B) \vee (\neg A \wedge \neg B \wedge \neg C) \vee (\neg A \wedge B \wedge \neg C) \vee \\
&\quad (\neg A \wedge B \wedge C) \vee (A \wedge \neg B \wedge \neg C), \\
F_K &= (\neg A \vee B \vee \neg C) \wedge (\neg A \vee \neg B \vee C) \wedge (\neg A \vee \neg B \vee \neg C) \\
&\equiv (\neg A \vee C \vee \neg B) \wedge (\neg A \vee B \vee \neg C) \wedge (\neg A \vee \neg B \vee \neg C).
\end{aligned}
$$

Konjunktionen/Disjunktionen mit mehr als n Atomen müssen eines der Atome entweder doppelt oder komplementär enthalten. Somit können sie verkürzt bzw. ganz weggelassen werden.

Aus einer KNF G_K für eine Formel G kann man eine DNF für die Formel $F = \neg G$ bilden:

$$G_K = \bigwedge_{i=1}^{n} \bigvee_{j=1}^{m_i} L_{ij}$$

Dann erhalten wir aus G_K mittels De Morgan (DM) eine DNF für F:

$$F \;=\; \neg G \equiv \neg G_K \overset{DM}{\equiv} \bigvee_{i=1}^{n} \neg \bigvee_{j=1}^{m_i} L_{ij} \overset{DM}{\equiv} \bigvee_{i=1}^{n} \bigwedge_{j=1}^{m_i} \neg L_{ij} \equiv \bigvee_{i=1}^{n} \bigwedge_{j=1}^{m_i} \overline{L_{ij}}$$

mit $\overline{L_{ij}} \equiv \neg L_{ij}$:

$$\overline{L_{ij}} = \begin{cases} A_k, & \text{falls } L_{ij} = \neg \mathrm{A}_k \\ \neg A_k, & \text{falls } L_{ij} = \mathrm{A}_k \end{cases}$$

Analog erhält man aus einer DNF G_D für G eine KNF für $F = \neg G$.

Für die boolesche Formel

$$F \;=\; \neg(A \vee B)$$

zum NOR ist $G = A \vee B$ bereits in DNF.

Mittels De Morgan (DM) kann man daraus eine KNF für F gewinnen:

$$F \;=\; \neg G \;\overset{DM}{\equiv}\; \neg A \wedge \neg B.$$

Diese kann man aufgrund der Resolutions–Regel noch erweitern, so daß jedes Atom in jeder Disjunktion vorkommt:

- Aus $\neg A$ erhält man durch Erweiterung um B bzw. $\neg B$ die beiden Disjunktionen $\neg A \vee B$ und $\neg A \vee \neg B$.
- Aus $\neg B$ erhält man durch Erweiterung um A bzw. $\neg A$ die beiden Disjunktionen $A \vee \neg B$ und $\neg A \vee \neg B$.

Aufgrund der Idempotenz können wir eine der beiden doppelten Disjunktionen $\neg A \vee \neg B$ eliminieren und erhalten folgende weitere KNF:

$$F \equiv (\neg A \vee B) \wedge (\neg A \vee \neg B) \wedge (A \vee \neg B).$$

Man erkennt, daß diese genau die drei Nullstellen von F widerspiegelt:

A	B	G	F
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

Wir haben also dasselbe Resultat erhalten wie bei der Konstruktion aufgrund der Wahrheitswertetabelle. Es gilt $\beta_{(1,0)} = \neg A \vee B$, $\beta_{(1,1)} = \neg A \vee \neg B$ und $\beta_{(0,1)} = A \vee \neg B$.

Existenz und Konstruktion der Normalformen

Durch die obige Konstruktion aus den Konjunktionen α_I bzw. den Disjunktionen β_I ist die Existenz einer DNF bzw. KNF für jede Formel F eigentlich bereits gezeigt.

Der folgende Satz zeigt dies noch einmal, und zwar durch strukturelle Induktion über den Aufbau der Formel F.

Dabei wird gezeigt, wie man die Normalformen für

1. Negationen,
2. Disjunktionen und
3. Konjunktionen

aus den Normalformen ihrer Bestandteile konstruieren kann.

Satz (Normalformen)

Für jede Formel F gibt es eine äquivalente Formel F' in KNF bzw. DNF.

Beweis: Strukturelle Induktion über den Aufbau der Formel F

Induktionsanfang:
Falls F eine *atomare* Formel ist, so ist F bereits in KNF bzw. in DNF.

Induktionsschritt:

1. Sei $F = \neg G$ eine Negation:

   Nach Induktionsannahme gibt es zu G äquivalente Formeln G_K in KNF und G_D in DNF.

   Wie bereits gezeigt kann man aus G_K / G_D eine zu F äquivalente Formel in DNF / KNF bilden.

2. Sei $F = G \vee H$ eine Disjunktion:

   Nach Induktionsannahme gibt es zu G, H äquivalente Formeln

   - G_K und H_K in KNF und
   - G_D und H_D in DNF

   der folgenden Form:

   $$G_K = \bigwedge_{i=1}^{n} \beta_i, \quad G_D = \bigvee_{i=1}^{m} \alpha_i,$$
   $$H_K = \bigwedge_{j=1}^{n'} \beta'_j, \quad H_D = \bigvee_{j=1}^{m'} \alpha'_j.$$

   Dabei sind

   - β_i, β'_j Disjunktionen und
   - α_i, α'_j Konjunktionen

   von Literalen.

Wir erhalten nach dem Distributivgesetz (D) eine KNF:

$$
\begin{aligned}
F &= G \vee H \equiv G_K \vee H_K \\
&= G_K \vee \bigwedge_{j=1}^{n'} \beta'_j \\
&\overset{D}{\equiv} \bigwedge_{j=1}^{n'} (G_K \vee \beta'_j) \\
&= \bigwedge_{j=1}^{n'} ((\bigwedge_{i=1}^{n} \beta_i) \vee \beta'_j) \\
&\overset{D}{\equiv} \bigwedge_{j=1}^{n'} \bigwedge_{i=1}^{n} (\beta_i \vee \beta'_j)
\end{aligned}
$$

Dies ist eine KNF, da die $\beta_i \vee \beta'_j$ Disjunktionen von Literalen sind.

Diese KNF enthält $m \cdot n$ Disjunktionen von Literalen. In der Praxis kann man diese Darstellung oft vereinfachen.

Wir erhalten durch disjunktive Verkettung eine DNF:

$$\begin{aligned} F &= G \vee H \equiv G_D \vee H_D \\ &= (\bigvee_{i=1}^{m} \alpha_i) \vee (\bigvee_{j=1}^{m'} \alpha'_j) \\ &= \bigvee_{i=1}^{m+m'} \alpha_i \end{aligned}$$

Dabei setzen wir $\alpha_{m+j} = \alpha'_j$, für $1 \leq j \leq m'$.

Dies eine DNF, da die α_i Konjunktionen von Literalen sind.

3. Der Induktionsschritt für eine Konjunktion $F = G \wedge H$ erfolgt analog zur Disjunktion. $\square$

Beispiel (DNF, KNF)

Für die folgende Formel F ist $G = A$ und $H = B \wedge C$:

$$F = A \vee (B \wedge C).$$

F ist bereits in DNF. Nach unserem Schema erhalten wir genau diese DNF durch disjunktive Verkettung der zwei DNFs

- $G_D = \alpha_1 = A$ und
- $H_D = \alpha'_1 = B \wedge C$.

Eine KNF für F erhalten wir nach dem Distributivgesetz durch Kombination der zwei KNFs

- $G_K = \beta_1 = A$ und
- $H_K = \beta'_1 \wedge \beta'_2$, mit $\beta'_1 = B$ und $\beta'_2 = C$.

Die KNF ist $(\beta_1 \vee \beta'_1) \wedge (\beta_1 \vee \beta'_2) = (A \vee B) \wedge (A \vee C)$.

Für die folgende Formel F ist $G = A \vee (D \wedge E)$ und $H = B \wedge C$:

$$F = (A \vee (D \wedge E)) \vee (B \wedge C).$$

Eine DNF für F erhalten wir durch Weglassen der Klammerung um G:

$$A \vee (D \wedge E) \vee (B \wedge C).$$

Eine KNF für F erhalten wir nach dem Distributivgesetz durch Kombination der zwei KNFs

- $G_K = \beta_1 \wedge \beta_2$, mit $\beta_1 = A \vee D$ und $\beta_2 = A \vee E$, und
- $H_K = \beta'_1 \wedge \beta'_2$, mit $\beta'_1 = B$ und $\beta'_2 = C$.

Die KNF ist

$$(\beta_1 \vee \beta'_1) \wedge (\beta_1 \vee \beta'_2) \wedge (\beta_2 \vee \beta'_1) \wedge (\beta_2 \vee \beta'_2) \;=$$
$$(A \vee D \vee B) \wedge (A \vee D \vee C) \wedge (A \vee E \vee B) \wedge (A \vee E \vee C).$$

Exkurs: Berechnung von Normalformen

Aus dem Beweis ergeben sich PROLOG–Regeln zur Berechnung von Normalformen (`knf, dnf`):

```
boolean_normalize(knf, (A;B), Cs) :-
   boolean_normalize(knf, A, As),
   boolean_normalize(knf, B, Bs),
   findall( C,
      ( member(C1, As),
        member(C2, Bs),
        append(C1, C2, C) ),
      Cs ).
boolean_normalize(dnf, (A;B), Cs) :-
   boolean_normalize(dnf, A, As),
   boolean_normalize(dnf, B, Bs),
   append(As, Bs, Cs).
```

```
boolean_normalize(knf, (A,B), Cs) :-
   boolean_normalize(knf, A, As),
   boolean_normalize(knf, B, Bs),
   append(As, Bs, Cs).
boolean_normalize(dnf, (A,B), Cs) :-
   boolean_normalize(dnf, A, As),
   boolean_normalize(dnf, B, Bs),
   findall( C,
      ( member(C1, As),
        member(C2, Bs),
        append(C1, C2, C) ),
      Cs ).
```

Zur Berechnung einer KNF für eine Formel $\neg A$ braucht man eine DNF für die Formel A; analoges gilt für die DNF:

```
boolean_normalize(knf, -A, Cs) :-
   boolean_normalize(dnf, A, Cs_2),
   maplist( literals_to_complements,
      Cs_2, Cs ).
boolean_normalize(dnf, -A, Cs) :-
   boolean_normalize(knf, A, Cs_2),
   maplist( literals_to_complements,
      Cs_2, Cs ).
```

Das Prädikat `literals_to_complements/2` berechnet die Komplemente zu einer Liste von Literalen.

Für atomare Formeln `A` wird eine Liste `[[A]]` bestehend aus einer Einerliste `[A]` erzeugt:

```
boolean_normalize(_, A, [[A]]) :-
   boolean_formula_is_atomic(A),
   !.
```

Man kann jederzeit für weitere Typen von Formeln – wie z.B. Implikation und exklusives Oder – PROLOG–Regeln zur Erzeugung einer KNF / DNF angeben.

Der folgende Test bestimmt den Funktor `F` und die Stelligkeit `N` von `A`. Wenn das Paar `F:N` nicht in der angegebenen Menge liegt, d.h. wenn `A` keine

- Negation (Funktor "`-`" mit Stelligkeit 1),
- Konjunktion (Funktor "`,`" mit Stelligkeit 2) oder
- Disjunktion (Funktor "`;`" mit Stelligkeit 2)

ist, dann ist `A` atomar:

```
boolean_formula_is_atomic(A) :-
   \+ ( functor(A, F, N),
        member(F:N, ['-':1, ',':2, ';':2]) ).
```

Berechnungsbeispiel:

```
?- Formula = -((a,(b;c))),
   boolean_normalize(knf, Formula, Cs),
   boolean_normal_form_to_formula(knf, Cs, F).
Cs = [[-a, -b], [-a, -c]],
F = ((-a;-b), (-a;-c))

?- Formula = -((a,(b;c))),
   boolean_normalize(dnf, Formula, Cs),
   boolean_normal_form_to_formula(dnf, Cs, F).
Cs = [[-a], [-b, -c]],
F = (-a ; -b,-c)
```

Es gilt wieder “Punkt vor Strich”, d.h.

`(-a;-b,-c)` bedeutet `(-a;(-b,-c))` bzw. $\neg a \vee (\neg b \wedge \neg c)$.

Weitere Regeln

Für beliebige aussagenlogische Formeln F, F_1, F_2 und G gilt:

Resolution:

$$(F_1 \vee G) \wedge (F_2 \vee \neg G) \;\rightarrow\; F_1 \vee F_2 \;\text{ ist eine Tautologie,}$$
$$(F \vee G) \wedge (F \vee \neg G) \;\equiv\; F.$$

Implikation:

$$F \vee \neg G \;\equiv\; G \rightarrow F.$$

Moduns Ponens:

$$(G \wedge (G \rightarrow F)) \rightarrow F \;\text{ ist eine Tautologie.}$$

Mithilfe der Resolutions–Regel

$$(F \vee G) \wedge (F \vee \neg G) \;\equiv\; F$$

kann eine KNF vereinfacht werden:

$$\begin{aligned} H_K &= (\neg A \vee B \vee \neg C) \wedge (\neg A \vee \neg B \vee C) \wedge (\neg A \vee \neg B \vee \neg C) \\ &\equiv (\neg A \vee \neg B) \wedge (\neg A \vee \neg C). \end{aligned}$$

Denn wir können folgende Ersetzungen vornehmen:

$$(\neg A \vee B \vee \neg C) \wedge (\neg A \vee \neg B \vee \neg C) \equiv (\neg A \vee \neg C),$$
$$(\neg A \vee \neg B \vee C) \wedge (\neg A \vee \neg B \vee \neg C) \equiv (\neg A \vee \neg B).$$

Die Teilformel $\neg A \vee \neg B \vee \neg C$, die in beiden Ersetzungen verwendet wird, kann man aufgrund der Idempotenz duplizieren.

Aufgrund der Kommutativität und der Assoziativität der Konjunktion $\wedge$ kann man die zu ersetzenden Teilformeln erzeugen.

1.3 Hornformeln

Definition (Hornformeln)

Eine Formel F ist eine Hornformel, falls F in KNF ist und jedes Disjunktionglied in F höchstens ein positives Literal enthält.

nach dem Logiker *Alfred Horn*

Beispiel (Hornformeln)

1. $F \;=\; (\neg A \vee \underline{B} \vee \neg C) \wedge (\neg A \vee \neg B \vee \underline{C}) \wedge (\neg A \vee \neg B \vee \neg C)$
   ist eine Hornformel.

2. $G \;=\; A \vee B$
   ist keine Hornformel.

Aus der Definition der *Implikation* $F \to G \equiv \neg F \vee G$ und den Regeln von De Morgan ergibt sich die Regelschreibweise $A_1 \wedge \ldots \wedge A_n \to B$:

$$\neg A_1 \vee \ldots \vee \neg A_n \vee B \;\equiv$$
$$\neg(A_1 \wedge \ldots \wedge A_n) \vee B \;\equiv\; A_1 \wedge \ldots \wedge A_n \to B.$$

Wir nennen die Konjunktion $A_1 \wedge \ldots \wedge A_n$ den Rumpf und B den Kopf der Regel. Unter der Annahme, daß die Konjunktion stärker bindet als die Implikation, können wir die Klammern um den Regelrumpf weglassen.

Wir setzen $0 = \square_D$ (leere Disjunktion) und $1 = \square_K$ (leere Konjunktion). 0 ist eine unerfüllbare Formel, 1 ist eine Tautologie, und es gilt:

$$\neg A_1 \vee \ldots \vee \neg A_n \;\equiv\; \neg(A_1 \wedge \ldots \wedge A_n) \;\equiv\; A_1 \wedge \ldots \wedge A_n \to 0$$
$$B \;\equiv\; 1 \to B$$

Wir nennen $A_1 \wedge \ldots \wedge A_n \to 0$ eine Integritätsbedingung, $1 \to B$ ein Fakt.

Folglich kann man eine Hornformel als Konjunktion von Implikationen schreiben (*prozedurale Deutung*):

$$\begin{aligned} F &= (\neg A \vee \underline{B} \vee \neg C) \wedge (\neg A \vee \neg B \vee \underline{C}) \wedge (\neg A \vee \neg B \vee \neg C) \\ &\equiv (A \wedge C \rightarrow B) \wedge (A \wedge B \rightarrow C) \wedge (A \wedge B \wedge C \rightarrow 0) \\ H &= A \wedge (\neg A \vee B) \wedge (\neg A \vee \neg C) \\ &\equiv (1 \rightarrow A) \wedge (A \rightarrow B) \wedge (A \wedge C \rightarrow 0). \end{aligned}$$

Wir können eine KNF–Formel

$$F = \beta_1 \wedge \ldots \wedge \beta_n$$

auch als Menge von Disjunktionen auffassen:

$$F = \{\, \beta_1, \ldots, \beta_n \,\}$$

Dann können wir für die Teilformeln $\beta_i \in F$ schreiben.

Interpretationen als Mengen von Atomen

Eine Menge I von atomaren Formeln repräsentiert eine Interpretation I' von F:

$$I'(A) = \begin{cases} 1, & \text{falls } A \in I \\ 0, & \text{falls } A \notin I \end{cases}$$

Beispiel

Die Menge $I = \{ A, C \}$ repräsentiert die folgende Interpretation:

I'	A	B	C	D
	1	0	1	0

I' ist hier nur für eine Auswahl von Atomen, die uns interessieren, angegeben.

I *verletzt* eine Regel

$$A_1 \wedge \ldots \wedge A_n \rightarrow B,$$

falls $\{ A_1, \ldots, A_n \} \subseteq I$ und $B \notin I$.

Die zugehörige Interpretation I' ist noch kein Modell der Regel,
da sie den Regelrumpf erfüllt, nicht aber den Regelkopf.

- Falls $B = 0$ ist (Integritätsbedingung), so kann I nicht zu einer Menge erweitert werden, die ein Modell der Regel repräsentiert.
- Andernfalls repräsentiert die um das Atom B aus dem Regelkopf erweiterte Menge $I \cup \{ B \}$ ein Modell der Regel.

Beispiel

Die obige Interpretation zu $I = \{ A, C \}$ verletzt die Regel $A \rightarrow B$,
da $A \in I$ und $B \notin I$.

Effizienter Erfüllbarkeitstest für Hornformeln F

1. Sei I die Menge aller Kopfatome B zu den Fakten $(1 \rightarrow B) \in F$.
2. Solange sich I verändert:
   - Falls es eine verletzte Integritätsbedingung $(A_1 \wedge \ldots \wedge A_n \rightarrow 0) \in F$ gibt, so ist F unerfüllbar, und der Algorithmus kann stoppen.

     Wir setzen dann $I = \bigtriangledown$.
   - Andernfalls sei J die Menge aller atomaren Formeln B zu den verletzten Regeln $(A_1 \wedge \ldots \wedge A_n \rightarrow B) \in F$.

     Füge J zu I hinzu: $I = I \cup J$.

Falls am Ende $I \neq \bigtriangledown$ gilt, so ist F ist *erfüllbar*, und I repräsentiert ein Modell I' von F.

Beispiel (Erfüllbarkeitstest)

$$\begin{aligned} F &= (\neg A \vee \neg B \vee C) \wedge A \wedge D \wedge (\neg D \vee B) \wedge (\neg C \vee \neg E) \\ &\equiv (A \wedge B \rightarrow C) \wedge (1 \rightarrow A) \wedge (1 \rightarrow D) \wedge (D \rightarrow B) \wedge (C \wedge E \rightarrow 0) \end{aligned}$$

Nach Schritt 1 erhalten wir wegen $(1 \rightarrow A), (1 \rightarrow D) \in F$:

$$I_0 = \{\, A, D \,\}.$$

Nach i Durchläufen von Schritt 2:

$$I_1 = \{\, A, D, B \,\}, \text{ wegen } (D \rightarrow B) \in F,$$
$$I_2 = \{\, A, D, B, C \,\}, \text{ wegen } (A \wedge B \rightarrow C) \in F.$$

Die Regel $(C \wedge E \rightarrow 0)$ feuert als einzige nicht, so daß am Ende I_2 ein Modell I_2' von F repräsentiert:

I_2'	A	B	C	D	E
	1	1	1	1	0

Eigenschaften des Erfüllbarkeitstests

Zu jedem Zeitpunkt des Algorithmus gilt für die aktuelle Menge I:
Für alle Atome $A \in I$ und alle Modelle M von F gilt $M(A) = 1$
(falls es überhaupt Modelle M von F gibt).

Beweis durch vollständige Induktion über die Anzahl N der Schleifendurchläufe:

$N = 0$: Ist $(1 \rightarrow B) \in F$, so muß offenbar $M(B) = 1$ gelten.

$N \rightarrow N + 1$: Seien $A_1, \ldots, A_n \in I$ nach N Schleifendurchläufen.
Dann gilt nach Induktionsannahme für jedes Modell M von F:

- Ist $r = (A_1 \wedge \ldots \wedge A_n \rightarrow B) \in F$ dann gilt auch $M(B) = 1$, sonst würde $M \not\models r$ gelten. Also gilt auch für das im $N + 1$–ten Schleifendurchlauf zu I hinzukommende Atom B: $M(B) = 1$.
- Ist $r = (A_1 \wedge \ldots \wedge A_n \rightarrow 0) \in F$, dann gibt es kein Modell M für F, denn aus $M(A_1) = \ldots = M(A_n) = 1$ folgt $M \not\models r$. $\square$

Falls der Algorithmus mit $I \neq \bigtriangledown$ terminiert, so ist I' ein Modell für F, denn für alle Implikationen $r \in F$ gilt $I' \models r$:

- Fakt $r = (1 \rightarrow B)$:
  Nach Schritt 1 gilt $B \in I$ und somit $I'(B) = 1$, d.h. $I' \models r$.

- Integritätsbedingung $r = (A_1 \wedge \ldots \wedge A_n \rightarrow 0)$:
  Für mindesten ein i gilt $A_i \notin I$,
  sonst hätte der Algorithmus mit $I = \bigtriangledown$ gestoppt.
  Deshalb gilt $I'(A_1 \wedge \ldots \wedge A_n) = 0$, und somit $I' \models r$.

- Regel $r = (A_1 \wedge \ldots \wedge A_n \rightarrow B)$:
  $I'(A_1 \wedge \ldots \wedge A_n) = 1$ ist äquivalent zu $A_1 \in I, \ldots, A_n \in I$.
  Also ist $B \in I$, sonst würde Schritt 2 noch einmal ausgeführt.
  Dies ist wiederum gleichbedeutend mit $I'(B) = 1$.
  Also gilt $I' \models r$. $\square$

Minimale Modelle, kleinstes Modell

Sei M eine Klauselmenge.

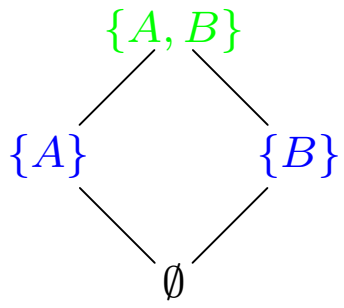
1. Ein Modell I' von M heißt *minimal*, falls die zugehörige Menge $I = \{ A \mid I'(A) = 1 \}$ von wahren Atomen minimal ist: d.h., es gibt kein anderes Modell J' von M mit $J = \{ A \mid J'(A) = 1 \} \subsetneq I$.
2. Ein Modell I' von M heißt *kleinstes* Modell, falls für alle Modelle J' von M gilt: $I \subseteq J$.

Wenn es ein kleinstes Modell gibt, so ist dieses eindeutig; außerdem ist es dann auch das einzige minimale Modell.

Sei F eine Hornformel.

1. Bei Erfüllbarkeit berechnet der Test das kleinste Modell I' für F. Definiert man $0 < 1$, so gilt $I'(A) \leq J'(A)$, für alle Atome A.
2. Falls der Algorithmus mit $I = \bigtriangledown$ stoppt, so gibt es kein Modell für F.

Beispiel (Minimale Modelle, kleinstes Modell)



1. Die Formel

$$G \;=\; A \vee B$$

hat zwei minimale Modelle I_1' und I_2' mit den folgenden zugehörigen Mengen $I_1 = \{\, A \,\}$ und $I_2 = \{\, B \,\}$ von wahren Atomen.

Also hat G kein kleinstes Modell, und somit gibt es keine zu G äquivalente Hornformel.

Daneben hat G noch ein nicht–minimales Modell zu $I_3 = \{\, A, B \,\}$.

2. Die erweiterte Formel

$$H \;=\; (A \vee B) \wedge \neg B$$

ist dagegen äquivalent zur folgenden Hornformel:

$$H \;\equiv\; A \wedge \neg B.$$

Terminierung

Sei n_F die Anzahl der atomaren Formeln in F.

Da in jeder außer der letzten Iteration mindestens ein Atom zu I hinzukommen muß, stoppt der Erfüllbarkeitstest nach höchstens n_F Iterationen.

Satz (Korrektheit des Erfüllbarkeitstests)

1. Falls der Algorithmus mit $I \neq \bigtriangledown$ terminiert, so repräsentiert die berechnete Menge I ein Modell für F, ja sogar das kleinste Modell für F, und F ist erfüllbar.

2. Falls der Algorithmus mit $I = \bigtriangledown$ terminiert, so gibt es kein Modell für F, und F ist unerfüllbar.

Regeln, Fakten und Integritätsbedingungen

1. Regeln

   $$A_1 \wedge \ldots \wedge A_n \rightarrow B$$

   mit $n \geq 1$ und $B \neq 0$ dienen beim Erfüllbarkeitstest zu Erweiterung der Menge I von Atomen.

2. Hornformeln F ohne Fakten

   $$1 \rightarrow B$$

   sind immer erfüllbar, und die berechnete Menge I ist leer:
   $I = \emptyset$, d.h., in I' sind alle Atome falsch.

3. Hornformeln F ohne Integritätsbedingungen

   $$A_1 \wedge \ldots \wedge A_n \rightarrow 0$$

   sind immer erfüllbar.

Exkurs: Logikprogrammierung

Wenn wir in Hornformeln komplexere Atome zulassen, dann sind wir bei einem Spezialfall der Prädikatenlogik, nämlich der Logikprogrammierung.

Die Regel

$$grandparent(X, Z) \leftarrow parent(X, Y) \wedge parent(Y, Z)$$

entspricht der Konjunktion aller Instanzen

$$parent(x, y) \wedge parent(y, z) \rightarrow grandparent(x, z),$$

bei denen die Variablen X, Y, Z durch Personen x, y, z ersetzt wurden.
Ein Beispiel wäre die Instanz

$$parent(\text{'William'}, \text{'Charles'}) \wedge parent(\text{'Charles'}, \text{'Elizabeth'}) \rightarrow grandparent(\text{'William'}, \text{'Elizabeth'}).$$

Zusammen mit den Fakten

$$1 \rightarrow parent(\text{'Elizabeth'}, \text{'George'}),$$
$$1 \rightarrow parent(\text{'Charles'}, \text{'Elizabeth'}),$$
$$1 \rightarrow parent(\text{'William'}, \text{'Charles'}),$$

würde der Erfüllbarkeitstest folgende Mengen ableiten:

$$\begin{aligned} I_0 &= \{\, parent(\text{'Elizabeth'}, \text{'George'}), \\ &\quad\; parent(\text{'Charles'}, \text{'Elizabeth'}), \\ &\quad\; parent(\text{'William'}, \text{'Charles'}) \,\}, \\ I_1 &= I_0 \cup \{\, grandparent(\text{'Charles'}, \text{'George'}), \\ &\quad\; grandparent(\text{'William'}, \text{'Elizabeth'}) \,\}. \end{aligned}$$

Er würde bereits nach der ersten Iteration keine neuen Fakten mehr ableiten.

1.4 Minimale Modelle für Klauselmengen

Eine *Klausel* ist eine Disjunktion von Literalen.
Aus der Definition der *Implikation* $F \to G \equiv \neg F \vee G$ und den Regeln von De Morgan ergibt sich eine Implikationsschreibweise für Klauseln:

$$\begin{aligned} \neg A_1 \vee \ldots \vee \neg A_n \vee B_1 \vee \ldots \vee B_k \;&\equiv \\ \neg\,(A_1 \wedge \ldots \wedge A_n) \vee (B_1 \vee \ldots \vee B_k) \;&\equiv \\ A_1 \wedge \ldots \wedge A_n \to B_1 \vee \ldots \vee B_k \end{aligned}$$

Unter der Annahme, daß auch die Disjunktion stärker bindet als die Implikation, können wir auch die Klammern um den Regelkopf $B_1 \vee \ldots \vee B_k$ weglassen. Dann gilt außerdem:

$$\neg A_1 \vee \ldots \vee \neg A_n \;\equiv\; \neg\,(A_1 \wedge \ldots \wedge A_n) \;\equiv\; A_1 \wedge \ldots \wedge A_n \to 0$$

$$B_1 \vee \ldots \vee B_k \;\equiv\; 1 \to B_1 \vee \ldots \vee B_k$$

Eine KNF–Formel $\beta_1 \wedge \ldots \wedge \beta_n$ kann somit als endliche Klauselmenge $M = \{ \beta_1, \ldots, \beta_n \}$ in Implikationsschreibweise aufgefaßt werden.

1. Man nennt Implikationen der Art

   $$A_1 \wedge \ldots \wedge A_n \rightarrow B_1 \vee \ldots \vee B_k$$

   auch *Regeln*.

2. Regeln mit $n = 0$ Rumpfatomen nennen wir *Fakten*:

   $$1 \rightarrow B_1 \vee \ldots \vee B_k$$

   Falls M keine Fakten enthält, ist M immer erfüllbar, und M hat ein kleinstes Modell I', in dem alle Atome falsch sind.

3. Regeln mit $k = 0$ Kopfatomen nennen wir *Integritätsbedingungen*:

   $$A_1 \wedge \ldots \wedge A_n \rightarrow 0$$

   M ist erfüllbar, falls es keine Integritätsbedingungen enthält.

1 ist die leere Konjunktion, und 0 ist die leere Disjunktion.

Beispiel (Modelle)

1. $M = \{\ c \rightarrow a \vee b\ \}$ enthält keine Fakten.
   Somit hat M ein kleinstes Modell, in dem alle Atome falsch sind.

2. $M = \{\ \mathit{1} \rightarrow a \vee b\ \}$ hat bekanntermaßen zwei minimale Modelle, nämlich zu den Mengen $I_1 = \{\ a\ \}$ und $I_2 = \{\ b\ \}$ von Atomen, sowie ein nicht–minimales Modell zu $I_3 = \{\ a, b\ \}$.

3. $M = \{\ \mathit{1} \rightarrow a, a \rightarrow 0\ \}$ ist unerfüllbar.
   Wegen des Fakts $\mathit{1} \rightarrow a$ müßte eine Modell a wahr machen, was wegen der Integritätsbedingung $a \rightarrow 0$ aber unmöglich ist.

Für viele praktische Anwendungen ist man nur an den minimalen Modellen einer Klauselmenge interessiert.

Man kann den Erfüllbarkeitstest zu einem Algorithmus zur Bestimmung der minimalen Modelle einer endlichen Klauselmenge M erweitern.

Der erweiterte Algorithmus arbeitet nicht mit einer einzigen Menge I von Atomen, sondern mit einer Menge $\mathcal{I}$ solcher Mengen von Atomen.

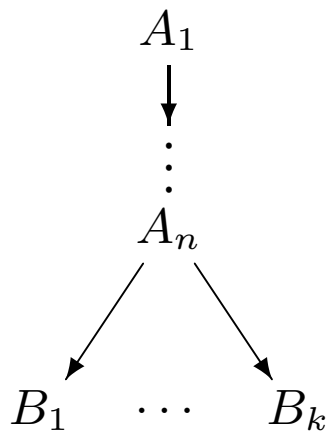
Die Mengen $I \in \mathcal{I}$ werden nach folgender Schlußregel erweitert:

$$\frac{\{ A_1, \ldots, A_n \} \subseteq I, \quad (A_1 \wedge \ldots \wedge A_n \rightarrow B_1 \vee \ldots \vee B_k) \in M}{I \cup \{ B_1 \}, \ldots, I \cup \{ B_k \}}$$

Aus einer Menge I werden also k Mengen $I \cup \{ B_i \}$.

Die Menge $\mathcal{I}$ kann man als Baum veranschaulichen.

1. Jeder Ast entspricht einer Menge $I \in \mathcal{I}$.
2. Falls man alle Atome A_i auf dem Ast finden kann, so wird der Ast um eine Verzweigung erweitert.

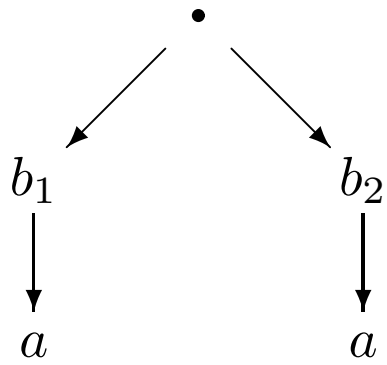


Beispiel (Berechnung der minimalen Modelle)

Für die Klauselmenge

$$M \;=\; \{\; b_1 \rightarrow a,\; b_2 \rightarrow a,\; 1 \rightarrow b_1 \vee b_2 \;\}$$

erhalten wir die folgende Mengenfolge:

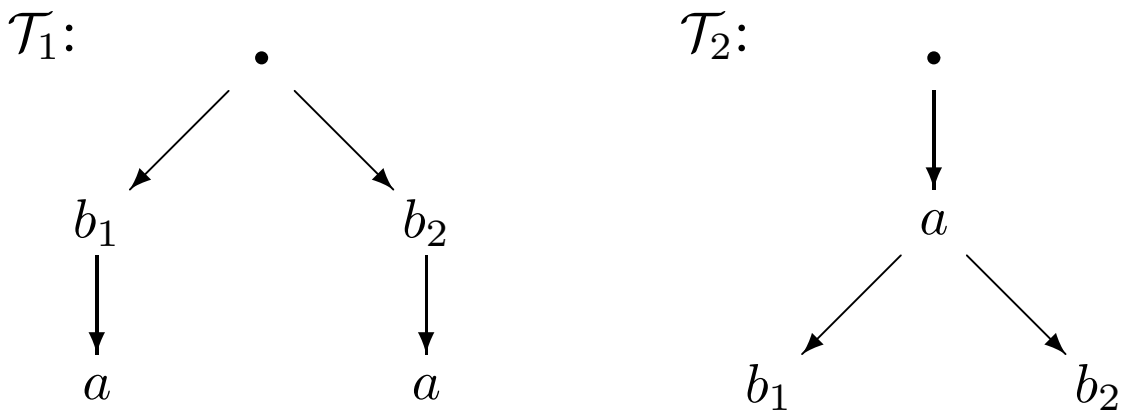


$$\begin{aligned}
\mathcal{I}_0 &= \{\; \emptyset \;\}, \\
\mathcal{I}_1 &= \{\; \{\, b_1 \,\},\; \{\, b_2 \,\} \;\}, \\
\mathcal{I}_n &= \{\; \{\, a, b_1 \,\},\; \{\, a, b_2 \,\} \;\}, \text{ für alle } n \geq 2.
\end{aligned}$$

Zur Erweiterung von $I = \emptyset$ können nur Fakten verwendet werden.
Zuerst wird $I = \emptyset$ wegen des Fakts $1 \rightarrow b_1 \vee b_2$ um b_1 bzw. b_2 erweitert.

Beide Erweiterungen können dann wegen der Regel $b_i \rightarrow a$ jeweils um a erweitert werden.

Der gezeigte Baum $\mathcal{T}_1$ entspricht genau der Berechnung.



Der kompaktere Baum $\mathcal{T}_2$ repräsentiert ebenfalls die Resultatsmenge

$$\mathcal{I}_2 \;=\; \{\, \{\, a, b_1 \,\}, \{\, a, b_2 \,\} \,\},$$

er spiegelt aber nicht die Berechnungsreihenfolge wider.

Definition (Regel: feuert, verletzt)

Sei I eine Menge von Atomen.

1. Eine Regel

   $$A_1 \wedge \ldots \wedge A_n \rightarrow B_1 \vee \ldots \vee B_k$$

   feuert unter I falls $\{ A_1, \ldots, A_n \} \subseteq I$.

2. I *verletzt* eine Regel

   $$A_1 \wedge \ldots \wedge A_n \rightarrow B_1 \vee \ldots \vee B_k,$$

   falls $\{ A_1, \ldots, A_n \} \subseteq I$ und $\{ B_1, \ldots, B_k \} \cap I = \emptyset$.

Falls I eine Regel verletzt, so ist die zugehörige Interpretation noch kein Modell dieser Regel, da sie den Rumpf erfüllt, nicht aber den Kopf.

Wenn man I um eines der Kopfatome B_i erweitert, so verletzt $I \cup \{ B_i \}$ die Regel nicht mehr.

Die Berechnung der minimalen Modelle einer Klauselmenge behandelt Fakten $1 \rightarrow B_1 \vee \ldots \vee B_k$ nicht separat, sondern als Regeln

$$A_1 \wedge \ldots \wedge A_n \rightarrow B_1 \vee \ldots \vee B_k$$

mit $n = 0$ Rumpfatomen.

- Ein Fakt feuert unabhängig von I, da $\{ A_1, \ldots, A_n \} = \emptyset \subseteq I$.
- I verletzt ein Fakt, falls $\{ B_1, \ldots, B_k \} \cap I = \emptyset$.

Während wir beim Erfüllbarkeitstest mit den Kopfatomen der Fakten starten, kann die Berechnung nun mit $\mathcal{I} = \{ \emptyset \}$ starten.

Am Anfang werden dann offensichtlich von $I = \emptyset$ genau die Fakten verletzt; diese feuern im ersten Erweiterungsschritt und erweitern I.

Später müssen die Fakten – genau wie alle anderen Regeln, die schon gefeuert haben – nicht mehr berücksichtigt werden.

Die Berechnung startet mit $\mathcal{I} = \{ \emptyset \}$, d.h. mit einer leeren Menge I in $\mathcal{I}$.

$\mathcal{I}$ wird schrittweise verändert durch Erweiterung bzw. Eliminierung seiner Elemente $I \in \mathcal{I}$.

Zur Erweiterung von I werden alle von I verletzten Regeln $r_1, \ldots, r_m \in M$ herangezogen.
Wenn wir aus jeder Regel r_i ein Kopfatom B_i auswählen, dann repräsentiert die Erweiterung $J = I \cup \{ B_1, \ldots, B_m \}$ ein Modell dieser Regeln.

Sei I^M die Menge all dieser Erweiterungen J:

$$\begin{aligned} I^M \;=\; \{\, & I \cup \{ B_1, \ldots, B_m \} \mid \\ & \text{für alle } 1 \leq i \leq m : \; B_i \text{ ist ein Atom aus dem Kopf von } r_i \,\}. \end{aligned}$$

Falls I keine Regeln verletzt, so ist $I^M = \{ I \}$.

Dann ersetzen wir I in $\mathcal{I}$ durch die Erweiterungen aus I^M.

Beispiel (Berechnung der minimalen Modelle)

Für die Klauselmenge

$$M \;=\; \{\; b \to a_1 \vee a_2,\; b \to c_1 \vee c_2,\; \mathit{1} \to b \;\}$$

erhalten wir

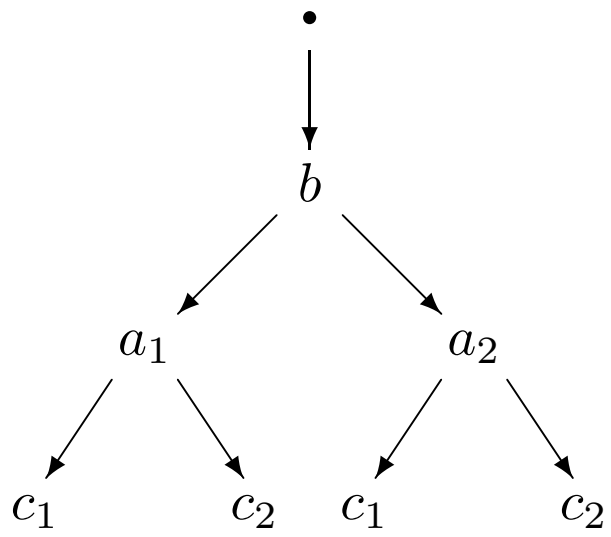
$$\begin{aligned} \emptyset^M &= \{\; \{\, b \,\} \;\}, \\ \{\, b \,\}^M &= \{\; \{\, b, a_1, c_1 \,\},\; \{\, b, a_1, c_2 \,\},\; \{\, b, a_2, c_1 \,\},\; \{\, b, a_2, c_2 \,\} \;\}, \end{aligned}$$

denn

- $I = \emptyset$ verletzt nur das Fakt $\mathit{1} \to b$, und
- $I = \{\, b \,\}$ verletzt die beiden Regeln $b \to a_1 \vee a_2$ und $b \to c_1 \vee c_2$, so daß man 4 Erweiterungen J bilden kann.

Damit erhalten wir die folgende Mengenfolge:

$$
\begin{aligned}
\mathcal{I}_0 &= \{\, \emptyset \,\}, \\
\mathcal{I}_1 &= \{\, \{\, b \,\} \,\}, \\
\mathcal{I}_n &= \{\, \{\, b, a_1, c_1 \,\}, \\
&\quad\ \{\, b, a_1, c_2 \,\}, \\
&\quad\ \{\, b, a_2, c_1 \,\}, \\
&\quad\ \{\, b, a_2, c_2 \,\} \,\}, \text{ für alle } n \geq 2.
\end{aligned}
$$



Zuerst wird $I = \emptyset$ wegen des Fakts *1* $\rightarrow b$ um b erweitert.

Dann wird $I = \{\, b \,\}$ aufgrund der verletzten Regeln $b \rightarrow a_1 \vee a_2$ und $b \rightarrow c_1 \vee c_2$ in einem Schritt zu 4 Mengen erweitert.

Die Mengen $I \in \mathcal{I}_2$ verletzen keine Regeln mehr.

Für die Klauselmenge

$$M \;=\; \{\; \mathit{1} \rightarrow a_1 \vee a_2,\; \mathit{1} \rightarrow c_1 \vee c_2 \;\}$$

erhalten wir

$$\emptyset^M \;=\; \{\; \{\, a_1, c_1 \,\},\; \{\, a_1, c_2 \,\},\; \{\, a_2, c_1 \,\},\; \{\, a_2, c_2 \,\} \;\}.$$

Da Fakten $\mathit{1} \rightarrow B_1 \vee \ldots \vee B_k$ spezielle Regeln

$$A_{\mathit{1}} \wedge \ldots \wedge A_n \rightarrow B_1 \vee \ldots \vee B_k$$

mit $n = 0$ Rumpfatomen sind,

- verletzt $I = \emptyset$ beide Fakten in M,
- so daß man 4 Erweiterungen J bilden kann.

1 steht für die leere Konjunktion, die in jeder Interpretation erfüllt ist.

Ein Berechnungsast zu I kann in zwei Fällen auch wieder absterben:

1. Falls eine Integritätsbedingung $r \in M$ unter I verletzt ist, so stirbt der Ast ab.

   Da r dann eine Regel mit $k = 0$ Kopfatomen ist, kann man kein Kopfatom wählen, um r zu erfüllen. Also gilt $I^M = \emptyset$.

   Wenn wir wieder I durch die Mengen aus I^M ersetzen, dann wird I faktisch eliminiert. Dies entspricht der Setzung $I = \bigtriangledown$ aus dem bekannten Erfüllbarkeitstest.

2. Wir wollen nur die minimalen Modelle berechnen.

   Deswegen kann man zusätzlich noch einen Ast absterben lassen, falls es einen anderen Ast zu einer kleineren Menge $J \subsetneq I$ gibt.

   Wir sagen dann J subsumiert I.

In beiden Fällen wird I aus $\mathcal{I}$ eliminiert.

Definition ($min(\mathcal{I})$)

Sei $\mathcal{I}$ eine Menge von Mengen von Atomen.
Dann enthält $min(\mathcal{I})$ alle $I \in \mathcal{I}$, für die es keine Menge $J \in \mathcal{I}$, mit $J \subsetneq I$, gibt.

Beispiel (Minimale Interpretationen)

Für die Menge

$$\mathcal{I} = \{ \{ a, b \}, \{ a, b, a_2 \}, \{ a, b, c_2 \}, \{ a_2, b, c_2 \} \}$$

gilt $min(\mathcal{I}) = \{ \{ a, b \}, \{ a_2, b, c_2 \} \}$.

Die beiden nicht–minimalen Mengen $\{ a, b, a_2 \}$ und $\{ a, b, c_2 \}$ werden von $\{ a, b \}$ subsumiert und somit eliminiert.

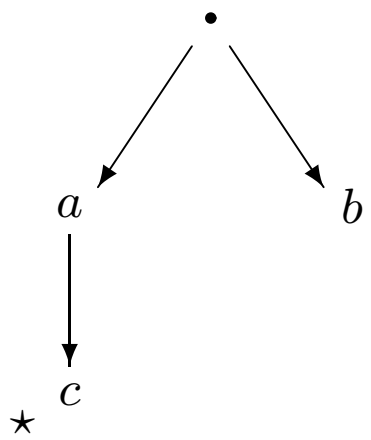
Beispiel (Berechnung der minimalen Modelle)

1. Für die Klauselmenge

$$M \;=\; \{\;\; a \to c,\;\; 1 \to a \vee b,\;\; a \wedge c \to 0 \;\;\}$$

erhalten wir die folgende Mengenfolge:

$$\begin{aligned}
\mathcal{I}_0 &= \{\, \emptyset \,\}, \\
\mathcal{I}_1 &= \{\, \{\, a \,\}, \{\, b \,\} \,\}, \\
\mathcal{I}_2 &= \{\, \{\, a, c \,\}, \{\, b \,\} \,\}, \\
\mathcal{I}_n &= \{\, \{\, b \,\} \,\}, \text{ für alle } n \geq 3.
\end{aligned}$$

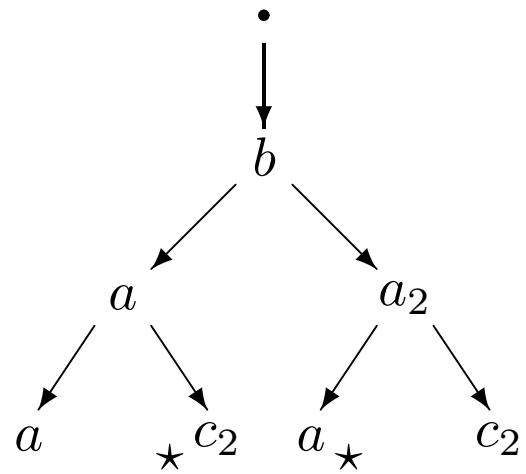


Der Ast zu $\{\, a, c \,\}$ stirbt im dritten Schritt wegen der verletzten Integritätsbedingung $a \wedge c \to 0$ ab.

2. Für die Klauselmenge

$$M \;=\; \{\; b \to a \vee a_2,\; b \to a \vee c_2,\; \mathit{1} \to b \;\}$$

erhalten wir die folgende Mengenfolge:



$$\mathcal{I}_0 \;=\; \{\; \emptyset \;\},$$

$$\mathcal{I}_1 \;=\; \{\; \{\; b \;\}\; \},$$

$$\mathcal{I}_n \;=\; \{\; \{\; b, a \;\},\; \{\; b, a_2, c_2 \;\}\; \},\; \text{für alle } n \geq 2.$$

Zuerst wird $I = \emptyset$ wegen des Fakts $\mathit{1} \to b$ um b erweitert.

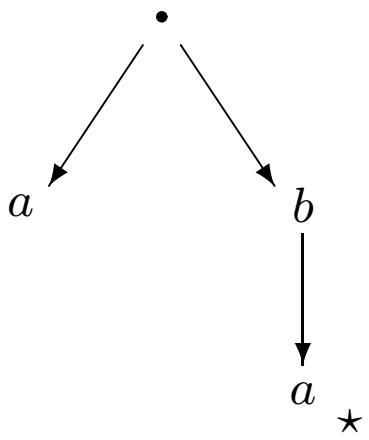
Dann wird $I = \{\; b \;\}$ aufgrund der Regeln $b \to a_1 \vee a_2$ und $b \to c_1 \vee c_2$ in einem Schritt zu 4 Mengen erweitert.

Das doppelte Atom a im linkesten Ast wird auf ein Atom reduziert.
Die beiden nicht–minimalen Mengen $\{\; b, a, a_2 \;\}$ und $\{\; b, a, c_2 \;\}$ werden von $\{\; b, a \;\}$ subsumiert und somit eliminiert.

3. Für die Klauselmenge

$$M \;=\; \{\, b \to a,\; 1 \to a \vee b \,\}$$

erhalten wir die folgende Mengenfolge:



$$\begin{aligned}
\mathcal{I}_0 &= \{\, \emptyset \,\}, \\
\mathcal{I}_1 &= \{\, \{\, a \,\}, \{\, b \,\} \,\}, \\
\mathcal{I}_n &= \{\, \{\, a \,\} \,\}, \text{ für alle } n \geq 2.
\end{aligned}$$

Die Erweiterung $\{\, a, b \,\}$ wird von $\{\, a \,\}$ subsumiert;
sie ist deswegen nicht–minimal, und der entsprechende Ast stirbt ab.

Algorithmus zur Berechnung der minimalen Modelle

Sei M eine Klauselmenge.

1. Sei $\mathcal{I} = \{ \emptyset \}$.
2. Solange sich $\mathcal{I}$ verändert:
   - Ersetze alle $I \in \mathcal{I}$ durch ihre Erweiterungen aus I^M:

     $$\mathcal{I} \mapsto \bigcup_{I \in \mathcal{I}} I^M.$$

     Falls I eine Integritätsbedingung aus M verletzt,
     so wird I ersatzlos eliminiert, denn $I^M = \emptyset$.
   - Eliminiere nicht–minimale Mengen aus $\mathcal{I}$:

     $$\mathcal{I} \mapsto min(\mathcal{I}).$$

Am Ende repräsentiert $\mathcal{I}$ die Menge der minimalen Modelle von M.

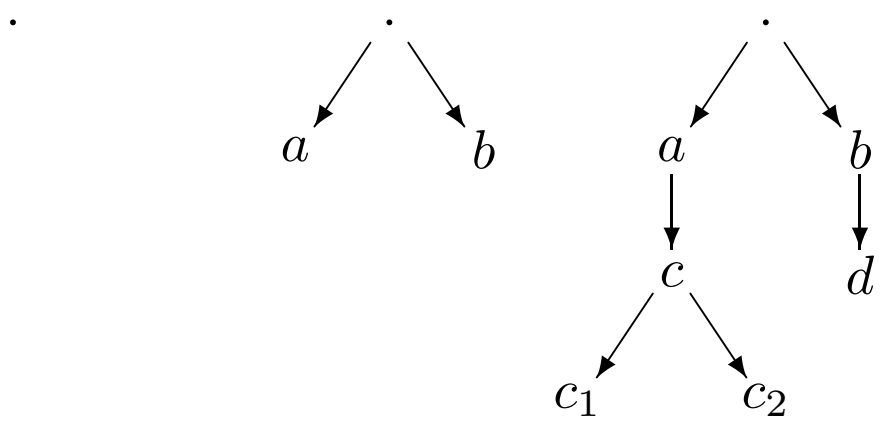
Beispiel (Berechnung der minimalen Modelle)

Für die Klauselmenge

$$M \;=\; \{\; a \to c,\;\; 1 \to a \vee b,\;\; a \to c_1 \vee c_2,\;\; b \to d \;\}$$

erhalten wir die folgenden Mengen und Modellbäume:

$$\begin{aligned}
\mathcal{I}_0 &= \{\, \emptyset \,\}, \\
\mathcal{I}_1 &= \{\, \{\, a \,\}, \{\, b \,\} \,\}, \\
\mathcal{I}_n &= \{\, \{\, a, c, c_1 \,\}, \{\, a, c, c_2 \,\}, \{\, b, d \,\} \,\}, \text{ für alle } n \geq 2.
\end{aligned}$$



Wir wollen uns den Fall $\mathcal{I} \mapsto \bigcup_{I \in \mathcal{I}} I^M$ für $\mathcal{I} = \mathcal{I}_1$ nochmals genauer ansehen.

Für die Klauselmenge $M = \{\, a \to c,\ 1 \to a \vee b,\ a \to c_1 \vee c_2,\ b \to d \,\}$ und die Menge $\mathcal{I} = \{\, \underbrace{\{\, a \,\}}_{I_1}, \underbrace{\{\, b \,\}}_{I_2} \,\}$ von Interpretationen gilt

$$I_1^M = \{\, a \,\}^M = \{\, \{\, a, c, c_1 \,\}, \{\, a, c, c_2 \,\} \,\},$$
$$I_2^M = \{\, b \,\}^M = \{\, \{\, b, d \,\} \,\}.$$

I_1 verletzt die Regeln $a \to c$ und $a \to c_1 \vee c_2$. I_2 verletzt die Regel $b \to d$.

Also erhalten wir

$$\mathcal{I} \mapsto \bigcup_{I \in \mathcal{I}} I^M = I_1^M \cup I_2^M = \{\, \{\, a, c, c_1 \,\}, \{\, a, c, c_2 \,\}, \{\, b, d \,\} \,\}.$$

Es gilt in diesem Fall $min(\mathcal{I}) = \mathcal{I}$.

Der Algorithmus terminiert, da eine verletzte Regel in jedem Berechnungsast nur einmal zur Erweiterung herangezogen werden kann.

Satz (Minimale Modelle einer Klauselmenge)

Sei M eine endliche Klauselmenge.

1. Die berechnete Menge $\mathcal{I}$ repräsentiert die minimalen Modelle von M.
2. Falls der Algorithmus mit $\mathcal{I} = \emptyset$ terminiert, so gibt es kein Modell für M, und M ist somit unerfüllbar.

Für erfüllbare Hornformeln wird das kleinste Modell berechnet.

Auch dieser Algorithmus terminiert nach höchstens n_M Iterationen, wenn n_M die Anzahl der atomaren Formeln in M ist, denn es muß in jeder außer der letzten Iteration mindestens ein Atom zu einem Ast hinzukommen.

1.5 Der Endlichkeitssatz (Kompaktheitssatz)

Satz (Endlichkeit, Kompaktheit)

Eine Formelmenge M ist genau dann *erfüllbar*,
wenn jede endliche Teilmenge von M erfüllbar ist.

Beweis:

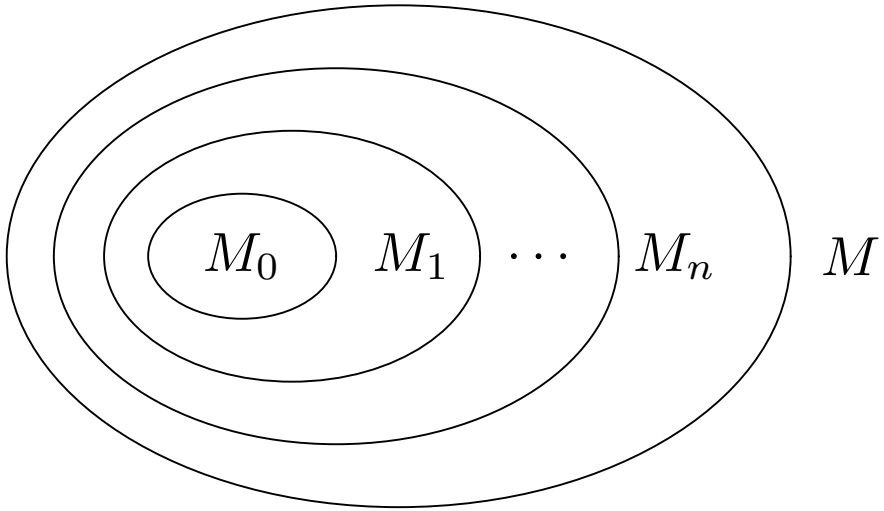
1. Ein Modell für M ist offensichtlich auch ein Modell für jede Teilmenge von M.

2. Nehmen wir also an, daß jede endliche Teilmenge von M erfüllbar ist, also ein Modell besitzt.

   Wir betrachten nun die speziellen Teilmengen M_n von M,
   welche genau die Formeln $F \in M$ enthalten, die nur die atomaren Formeln $A_1, \ldots, A_n$ enthalten.

Dann gilt offenbar:

$$M_1 \subseteq M_2 \subseteq \ldots \subseteq M_n \subseteq M.$$



Die Mengen M_n können i.a. unendlich groß sein.
Falls M unendlich viele Atome enthält, dann gibt es kein n mit $M_n = M$.

Für die endliche Formelmenge

$$M \;=\; \{\, a \vee b,\; \neg a \vee c,\; \neg b \vee d,\; \neg a \vee c_1 \vee c_2 \,\}$$

und die Reihenfolge

$$(A_1, \ldots, A_6) \;=\; (a, b, c, d, c_1, c_2)$$

erhalten wir

$$\begin{aligned}
M_1 &= \emptyset, \\
M_2 &= \{\, a \vee b \,\}, \\
M_3 &= \{\, a \vee b,\; \neg a \vee c \,\}, \\
M_4 &= \{\, a \vee b,\; \neg a \vee c,\; \neg b \vee d \,\}, \\
M_5 &= M_4, \\
M_6 &= \{\, a \vee b,\; \neg a \vee c,\; \neg b \vee d,\; \neg a \vee c_1 \vee c_2 \,\} \;=\; M.
\end{aligned}$$

Da alle Formeln $F \in M_n$ aber nur n Atome enthalten können, gibt es nur maximal 2^{2^n} grundsätzlich verschiedene Formeln

$$F_1, \ldots, F_k \in M_n.$$

Denn es gibt genau $k = 2^{2^n}$ verschiedene Wahrheitswertetafeln:

- jede Tafel hat 2^n Zeilen, und
- jede Auswahl einer Teilmenge von Zeilen mit der Belegung 1 (alle anderen Zeilen haben dann die Belegung 0) ergibt eine andere Tafel.

Jede Formel $F \in M_n$ ist also äquivalent zu einer der Formeln F_i:

$$F \equiv F_i.$$

Deshalb ist jedes Modell für

$$M'_n = \{F_1, \ldots, F_k\}$$

auch ein Modell für M_n.

Da M'_n endlich ist, besitzt M_n nach Voraussetzung ein Modell I_n.

Beispiel:

	A_1	A_2	A_3
I_1	0	$\ldots$	$\ldots$
I_2	0	1	$\ldots$
I_3	1	0	1
$\vdots$	$\vdots$	$\vdots$	$\vdots$

Wegen $M_i \subseteq M_n$, für alle $1 \leq i \leq n$, ist I_n auch ein Modell für M_i:

$$I_n \models M_i, \text{ für alle } 1 \leq i \leq n.$$

Für die endliche Formelmenge

$$M = \{ a \vee b,\ \neg a \vee c,\ \neg b \vee d,\ \neg a \vee c_1 \vee c_2 \}$$

könnte man folgendes erhalten:

	a	b	c	d	c_1	c_2
I_1	0					
I_2	0	1				
I_3	1	0	1			
I_4	1	0	1	0		
I_5	1	0	1	0	0	
I_6	1	0	1	0	1	0

Die fehlenden Einträge der Tabelle können beliebig sein.

Wir wollen nun aus den Modellen I_n der M_n ein Modell I für M konstruieren. Dazu definieren wir stufenweise die Wahrheitswerte $I(A_n)$ der Atome A_n.

Dazu benötigen wir folgende Indexmengen:

$n = 0$: $J_0 = \mathbb{N}_+ = \{1, 2, \ldots\}$,

$n - 1 \to n$: Sei

$$\begin{aligned} J^1_{n-1} &= \{\, j \in J_{n-1} \mid I_j(A_n) = 1 \,\}, \\ J^0_{n-1} &= \{\, j \in J_{n-1} \mid I_j(A_n) = 0 \,\}. \end{aligned}$$

Dann setzen wir

$$J_n = \begin{cases} J^1_{n-1}, & \text{falls } J^1_{n-1} \text{ unendlich} \\ J^0_{n-1}, & \text{sonst} \end{cases}$$

Wenn wir z.B. die fehlenden Tabelleneinträge auf 0 und $I_n = I_6$, für $n \geq 6$, setzen, dann erhalten wir folgende Mengen J_n:

	a	b	c	d	c_1	c_2
I_1	0	0	0	0	0	0
I_2	0	1	0	0	0	0
I_3	1	0	1	0	0	0
I_4	1	0	1	0	0	0
I_5	1	0	1	0	0	0
I_6	1	0	1	0	1	0

$$J_n = \{3, \ldots\}, \text{ für } 1 \leq n \leq 4,$$
$$J_n = \{6, \ldots\}, \text{ für } 5 \leq n.$$

Auf den ersten 4 Atomen sind alle Interpretationen I_n ab $n = 3$ gleich, auf den weiteren Atomen erst ab $n = 6$.

Offenbar gilt

$$J_0 \supseteq J_1 \supseteq \ldots \supseteq J_n.$$

Per Induktion nach n zeigt man, daß alle Indexmengen J_n unendlich sind, denn

- J_0 ist unendlich, und
- es muß immer mindestens eine der beiden Mengen J^1_{n-1} oder J^0_{n-1} unendlich sein.

Für alle I_j mit $j \in J_n$ hat A_n denselben Wahrheitswert:

$$I_j(A_n) = V \in \{0, 1\}, \text{ für alle } j \in J_n,$$

und wir setzen

$$I(A_n) = V.$$

Wir wollen nun zeigen, daß I ein Modell von M ist.

Sei also $F \in M$ eine beliebige Formel.

Dann gilt $F \in M_n$, für ein $n \in \mathbb{N}_+$, denn in F können nur endlich viele Atome $A_{i_1}, \ldots, A_{i_l}$, mit $1 \leq i_1 < i_2 < \ldots < i_l \leq n$, vorkommen.

Aus $M_n \subseteq M_i$, für alle $i \geq n$, folgt nun $F \in M_i$.

Deshalb ist jede Interpretation I_i, mit $i \geq n$, auch ein Modell für F.

Da J_n unendlich ist, gibt es einen Index $i \in J_n$, mit $i \geq n$.
Für diesen Index i gilt:

$$I_i(A_1) \;=\; I(A_1),\; \ldots\;,\; I_i(A_n) \;=\; I(A_n).$$

Denn der Index i liegt in J_n, und damit auch in allen J_m, mit $0 \leq m \leq n$, und es gilt $i \in J_m \Rightarrow I_i(A_m) = I(A_m)$.

Deshalb ist I natürlich auch ein Modell für F, da I auf allen relevanten Atomen $A_1, \ldots, A_n$ mit I_i übereinstimmt. $\square$

Der Endlichkeitssatz gilt nur für Formelmengen M, in denen alle Formeln $F \in M$ endlich sind.

Sei zum Beispiel

$$F_o = A_1 \vee A_2 \vee \ldots = \bigvee_{n=1}^{\infty} A_n$$

die unendliche Disjunktion der Atome A_i, und

$$F_n = \neg A_n, \text{ für } n \in \mathbb{N}_+.$$

Dann hat

$$M = \{ F_n \mid n \in \mathbb{N}_0 \}$$

kein Modell, obwohl jede endliche Teilmenge $M' \subseteq M$ ein Modell besitzt.

Für $M' = \varnothing$ ist dies klar.

Für $M' \neq \varnothing$ gibt es einen maximalen Index n mit $F_n \in M'$.

- Wir setzen $I(A_m) = 0$ für $m \leq n$, und erfüllen damit die Formeln F_m, mit $1 \leq m \leq n$, und
- wir setzen $I(A_m) = 1$ für $m > n$, und erfüllen damit die Formel F_0.

Damit ist I ein Modell für M'.

Die komplette Menge M hat kein Modell, da

- gemäß der Formeln F_m, mit $m \geq 1$, alle Atome A_m falsch sein müssen und
- gemäß der Formel F_0 aber mindestens ein Atom A_m wahr sein muß.

Eine *alternative Beweisidee* für den Endlichkeitssatz baut auf dem Erfüllbarkeits–Algorithmus auf.

Wir starten mit einem leeren Berechnungsbaum, und wir verwenden sukzessive die endlichen Klauselmengen M'_n zur Erweiterung.

- Falls irgendwann alle Äste abgestorben sind, so ist M'_n unerfüllbar.
- Andernfalls bleibt immer mindestens ein Ast am Leben.

Wir können dann wie folgt einen Ast finden, der nicht abstirbt:

- Wir starten in der Wurzel des Berechnungsbaums.
- Wir folgen einem Ast, dessen Teilbaum nicht komplett abstirbt.

Auf diese Weise finden wir eine Menge I von Atomen, so daß die zugehörige Interpretation I' ein Modell für M ist.

1.6 Resolution

Ein *Klausel* K ist eine Disjunktion von Literalen:

$$K = L_1 \vee \ldots \vee L_n.$$

Die Reihenfolge der Literale in K ist unerheblich; wir können K auch als Menge $\{ L_1, \ldots , L_n \}$ von Literalen ansehen.

Deswegen schreiben wir $L_i \in K$, für $1 \leq i \leq n$, und wir bezeichnen mit

$$K \backslash L_i = L_1 \vee \ldots \vee L_{i-1} \vee L_{i+1} \vee \ldots \vee L_n$$

die Klausel, welche man aus K durch *Weglassen* von L_i erhält.

Eine endliche Klauselmenge M entspricht einer Formel F in KNF:

$$F = (L_{1,1} \vee \ldots \vee L_{1,n_1}) \wedge \ldots \wedge (L_{m,1} \vee \ldots \vee L_{m,n_m}).$$

Definition (Resolvente)

Seien K_1 und K_2 Klauseln mit $A \in K_1$ und $\neg A \in K_2$, für ein Atom A. Dann heißt

$$K = (K_1 \backslash A) \vee (K_2 \backslash \neg A)$$

eine *Resolvente* von K_1 und K_2 *nach* A.

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

Spezialfall

Für $K_1 = A \vee K$ und $K_2 = \neg A \vee K$ erhalten wir die Resolvente

$$(K_1 \backslash A) \vee (K_2 \backslash \neg A) = K \vee K = K.$$

Beispiel (Resolvente)

Für die beiden Klauseln

$$\begin{aligned} K_1 &= A \vee B \vee C, \\ K_2 &= \neg A \vee B \vee \neg D \end{aligned}$$

$$\begin{tikzcd} \underline{A} \vee B \vee C \arrow[dr] & & \underline{\neg A} \vee B \vee \neg D \arrow[dl] \\ & B \vee C \vee \neg D & \end{tikzcd}$$

gilt

$$\begin{aligned} K_1 \setminus A &= B \vee C \\ K_2 \setminus \neg A &= B \vee \neg D, \end{aligned}$$

und wir erhalten die Resolvente $K = B \vee C \vee \neg D$.

Nur wenn eine Klausel K ein Atom A und dessen Negation $\neg A$ enthält (z.B. $K = A \vee B \vee \neg A$), dann kann man sie mit sich selbst resolvieren: $K_1 = K_2 = K$. Dann ist K eine Tautologie, und die Resolvente ist wieder K.

Beispiel (Resolvente)

Die beiden möglichen Resolventen aus $K_1 = A \vee \neg B \vee C$ und $K_2 = B \vee \neg C$ sind Tautologien:

$$\begin{tikzcd}
A \vee \underline{\neg B} \vee C \arrow[dr] & & \underline{B} \vee \neg C \arrow[dl] \\
& A \vee C \vee \neg C &
\end{tikzcd}
\qquad
\begin{tikzcd}
A \vee \neg B \vee \underline{C} \arrow[dr] & & B \vee \underline{\neg C} \arrow[dl] \\
& A \vee \neg B \vee B &
\end{tikzcd}$$

$$\begin{tikzcd}
\underline{A} \arrow[dr] & & \underline{\neg A} \arrow[dl] \\
& \square &
\end{tikzcd}$$

Wir bezeichnen hier die *leere Disjunktion* $\square_D$ kurz mit $\square$.

Resolutionslemma

Sei M ein Klauselmenge und $K_1, K_2 \in M$.

Ist K eine Resolvente von K_1 und K_2 (nach einem Atom A), so sind M und $M \cup \{K\}$ äquivalent: $M \equiv M \cup \{K\}$.

Beweis:

Sei I eine zu M – und damit auch zu $M \cup \{K\}$ – passende Interpretation.
Falls $I \models M \cup \{K\}$, dann gilt natürlich auch $I \models M$.
Also nehmen wir nun $I \models M$ an.

Sei $K_1 = A \vee K_1'$ und $K_2 = \neg A \vee K_2'$. Dann gilt $K = K_1' \vee K_2'$.

- Falls $I \models A$, so folgt aus $I \models K_2$ auch $I \models K_2'$.
- Falls $I \models \neg A$, so folgt aus $I \models K_1$ auch $I \models K_1'$.

Also gilt in beiden Fällen $I \models K_1' \vee K_2' = K$.
Deshalb gilt $I \models M \cup \{K\}$. $\square$

Definition (Resolutionsmengen)

Sei M eine Klauselmenge.

$$\begin{aligned} Res(M) \;=\; M \cup \{\, K \mid K \text{ ist eine Resolvente} \\ \text{zweier Klauseln } K_1, K_2 \in M \,\}. \end{aligned}$$

Ferner definieren wir:

$$\begin{aligned} Res^0(M) &= M, \\ Res^{n+1}(M) &= Res(Res^n(M)), \text{ für } n \geq 0, \\ Res^*(M) &= \bigcup_{n \geq 0} Res^n(M). \end{aligned}$$

Offensichtlich gilt die sogenannte Monotonie–Eigenschaft:

$$M \subseteq M' \;\Rightarrow\; Res(M) \subseteq Res(M').$$

Beispiel (Resolutionsmengen)

Für die Klauselmenge

$$M \;=\; \{\, c \vee \neg a,\; a \vee b,\; c_1 \vee c_2 \vee \neg a,\; d \vee \neg b \,\}$$

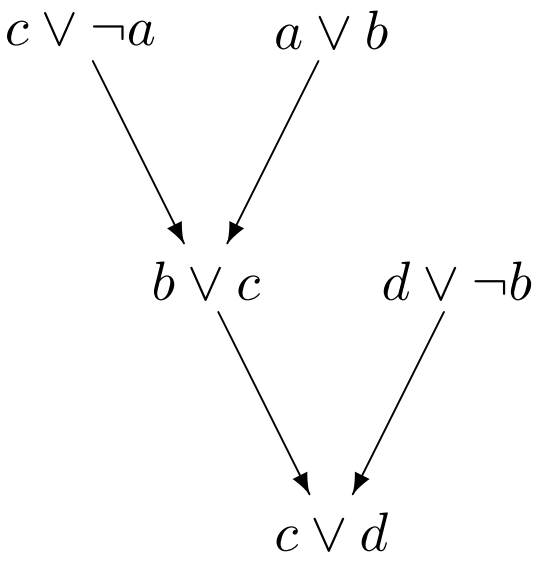
erhalten wir die folgenden Resolutionsmengen:

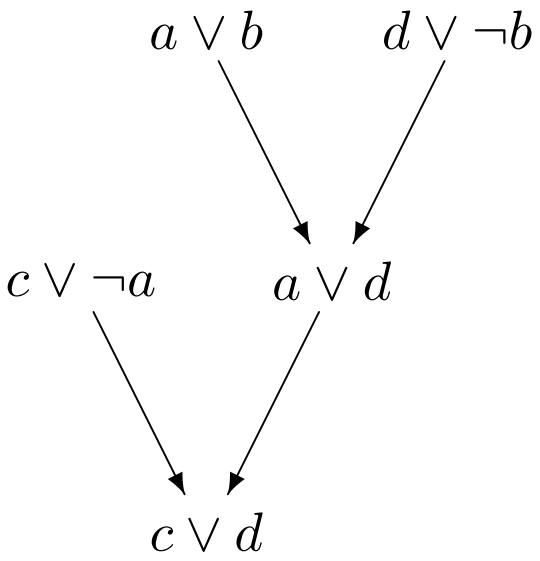
$$\begin{aligned}
Res^0(M) &= M, \\
Res^1(M) &= M \cup \{\, b \vee c,\; b \vee c_1 \vee c_2,\; a \vee d \,\}, \\
Res^2(M) &= Res^1(M) \cup \{\, c \vee d,\; c_1 \vee c_2 \vee d \,\}, \\
Res^3(M) &= Res^2(M).
\end{aligned}$$

1. Iteration:

- Aus $c \vee \neg a$ und $a \vee b$ erhalten wir $b \vee c$.
- Aus $a \vee b$ und $c_1 \vee c_2 \vee \neg a$ erhalten wir $b \vee c_1 \vee c_2$.
- Aus $a \vee b$ und $d \vee \neg b$ erhalten wir $a \vee d$.

$c \vee d$ kann redundant auf zwei unterschiedlichen Wegen abgeleitet werden:





In diesem Fall sind die verwendeten Klauseln aus M (an den Blättern des Baumes) jeweils dieselben. Nur die inneren Knoten der Bäume sind anders.

2. Iteration:

- Aus $c \vee \neg a$ und $a \vee d$ erhalten wir $c \vee d$.
- Aus $c_1 \vee c_2 \vee \neg a$ und $a \vee d$ erhalten wir $c_1 \vee c_2 \vee d$.
- Aus $d \vee \neg b$ und $b \vee c$ erhalten wir ebenfalls $c \vee d$.
- Aus $d \vee \neg b$ und $b \vee c_1 \vee c_2$ erhalten wir ebenfalls $c_1 \vee c_2 \vee d$.

Sowohl $c \vee d$ als auch $c_1 \vee c_2 \vee d$ werden also redundant auf zwei unterschiedlichen Wegen abgeleitet.

In der Praxis versucht man redundante Ableitungen möglichst zu vermeiden.

Für die Klauselmenge

$$M \;=\; \{\, p \vee q,\; \neg p \vee \neg q,\; p \vee \neg q,\; \neg p \vee q \,\}$$

erhalten wir die folgenden Resolutionsmengen:

$$\begin{aligned}
Res^0(M) &= M, \\
Res^1(M) &= M \cup \{\, p,\; q,\; \neg p,\; \neg q,\; p \vee \neg p,\; q \vee \neg q \,\}, \\
Res^2(M) &= Res^1(M) \cup \{\, \Box \,\}, \\
Res^3(M) &= Res^2(M).
\end{aligned}$$

Aus $\neg p \vee \neg q$ und $p \vee \neg q$ erhalten wir z.B. die Resolvente $\neg q$.

In Iteration 2 kann die leere Klausel $\Box$ kann auf zwei unterschiedlichen Wegen abgeleitet werden:

- aus p und $\neg p$ und
- aus q und $\neg q$.

Man kann zeigen:

$$M = Res^0(M) \subseteq Res^1(M) \subseteq \ldots \subseteq Res^n(M) \subseteq Res^*(M).$$

Außerdem gilt:

$$M \equiv Res^n(M) \equiv Res^*(M), \text{ für alle } n \in \mathbb{N}_0,$$

und wegen $Res^0(M) \subseteq Res^1(M)$ auch

$$Res^*(Res(M)) = \bigcup_{n \geq 1} Res^n(M) = \bigcup_{n \geq 0} Res^n(M) = Res^*(M).$$

Resolutionssatz der Aussagenlogik (Widerlegungsvollständigkeit)

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

Beweis:

Korrektheit: Angenommen $\square \in Res^*(M)$. Dann ist $Res^*(M)$ unerfüllbar, und wegen $M \equiv Res^*(M)$ ist auch M unerfüllbar.

Vollständigkeit: Angenommen M ist unerfüllbar.

Aufgrund des Endlichkeitstatzes gibt es bereits eine *endliche* Teilmenge $M' \subseteq M$, welche unerfüllbar ist. Wir können zusätzlich voraussetzen, daß M' keine Tautologien der Form $A \vee \neg A \vee C$ enthält.

Seien $A_1, \ldots, A_n$ alle in M' vorkommenden Atome. Wir zeigen nun per Induktion nach n, daß $\square \in Res^*(M)$.

$n = 0$: Da $M' = \varnothing$ erfüllbar ist, gilt $M' = \{\square\}$ und somit

$$\square \in M' \subseteq Res^*(M') \subseteq Res^*(M).$$

$n \to n+1$: Wir resolvieren Klauselpaare $K_1 \vee A_{n+1}$ und $K_2 \vee \neg A_{n+1}$ aus M' nach A_{n+1} und setzen:

$$\begin{aligned} M_1 &= \{\, K_1 \mid K_1 \vee A_{n+1} \in M' \,\}, \\ M_2 &= \{\, K_2 \mid K_2 \vee \neg A_{n+1} \in M' \,\}. \end{aligned}$$

Sei ferner

$$M_3 = \{\, K_3 \in M' \mid A_{n+1} \text{ kommt nicht in } K_3 \text{ vor} \,\}$$

die Menge aller Klauseln aus M', in denen A_{n+1} werder positiv als A_{n+1} noch negativ als $\neg A_{n+1}$ vorkommt. Dann gilt

$$Res(M') \supseteq M_3 \cup (M_1 \vee M_2) = M'',$$

wobei

$$M_1 \vee M_2 = \{\, K_1 \vee K_2 \mid K_1 \in M_1, K_2 \in M_2 \,\}.$$

Die Menge M'' enthält nur Formeln über $A_1, \ldots, A_n$.

Angenommen M'' ist erfüllbar mit einem Modell I.

- Angenommen $I \not\models M_1$ und $I \not\models M_2$.
  Dann gibt es $K_1 \in M_1$ und $K_2 \in M_2$ mit $I \not\models K_1$ und $I \not\models K_2$, und somit $I \not\models K_1 \vee K_2 \in M_1 \vee M_2 \subseteq M''$, ein Widerspruch.
- Also gilt $I \models M_1$ oder $I \models M_2$ (oder beides).

Wir konstruieren nun ein Modell I' für M':

$$M' = M_3 \cup (\{A_{n+1}\} \vee M_1) \cup (\{\neg A_{n+1}\} \vee M_2).$$

1. Falls $I \models M_1$, so setzen wir:

$$I'(B) = \begin{cases} I(B), & \text{falls } B \in \{A_1, \ldots A_n\} \\ 0, & \text{falls } B = A_{n+1} \end{cases}$$

   - Wegen $I \models M_1$ gilt $I' \models \{A_{n+1}\} \vee M_1$.
   - Wegen $I' \models \neg A_{n+1}$ gilt auch $I' \models \{\neg A_{n+1}\} \vee M_2$.

2. Falls $I \models M_2$, so setzen wir:

$$I'(B) = \begin{cases} I(B), & \text{falls } B \in \{A_1, \ldots, A_n\} \\ 1, & \text{falls } B = A_{n+1} \end{cases}$$

   - Wegen $I \models A_{n+1}$ gilt $I' \models \{A_{n+1}\} \vee M_1$.
   - Wegen $I' \models M_2$ gilt auch $I' \models \{\neg A_{n+1}\} \vee M_2$.

In beiden Fällen gilt wegen $I \models M_3$ auch $I' \models M_3$.

Also gilt in beiden Fällen $I' \models M'$,
im Widerspruch zu der Annahme, daß M' unerfüllbar ist.

Also haben wir gezeigt, daß M'' unerfüllbar ist.

Da M'' nur noch die n Atome $A_1, \ldots, A_n$ enthalten kann, gilt somit nach Induktionsannahme

$$\square \in Res^*(M'').$$

Wegen

$$Res^*(M'') \subseteq Res^*(Res(M')) = Res^*(M') \subseteq Res^*(M),$$

gilt nun also $\square \in Res^*(M)$.

$\square$

Die *Konstruktion* aus dem Beweis zum Resolutionssatz funktioniert für beliebige Klauselmengen M und Atome A:

$$\begin{aligned} M_1 &= \{\, K_1 \mid K_1 \vee A \in M \,\}, \\ M_2 &= \{\, K_2 \mid K_2 \vee \neg A \in M \,\}, \\ M_3 &= \{\, K_3 \in M \mid A \text{ kommt nicht in } K_3 \text{ vor} \,\}. \end{aligned}$$

Die *Transformation*

$$M \mapsto M'' = M_3 \cup (M_1 \vee M_2)$$

erhält die Widerlegungseingenschaften:

$$\begin{aligned} &M \text{ ist unerfüllbar } \Leftrightarrow \ \square \in Res^*(M) \ \Leftrightarrow \\ &M'' \text{ ist unerfüllbar } \Leftrightarrow \ \square \in Res^*(M'') \end{aligned}$$

M'' enthält das Atom A nicht mehr.

Beispiel (Transformation)

Aus der Klauselmenge

$$M \;=\; \{\, p \vee q,\; \neg p \vee \neg q,\; p \vee \neg q,\; \neg p \vee q \,\}$$

erhalten wir für das Atom $A = p$:

$$\begin{aligned} M_1 \;&=\; \{\, q,\; \neg q \,\} \;=\; M_2,\; M_3 \;=\; \emptyset, \\ M'' \;&=\; M_1 \vee M_2 \;=\; \{\, q \vee \neg q,\; q, \neg q \,\} \;\equiv\; \{\, q,\; \neg q \,\}. \end{aligned}$$

Tautologien – wie $q \vee \neg q$ – kann man frühzeitig aus der weiteren Betrachtung ausschließen.

Die nochmalige Transformation für das Atom $A = q$ erzeugt aus M'' nun

$$M'''' \;=\; \{\, \Box \,\}.$$

Also ist M unerfüllbar.

Definition (Deduktion)

Eine *Deduktion* (*Herleitung*, *Beweis*) einer Klausel K aus einer Klauselmenge M ist eine Folge

$$K_1, K_2, \ldots, K_m$$

von Klauseln, mit $K_m = K$, und

$\forall\, 1 \leq i \leq m$:

- $K_i \in M$ oder
- K_i ist die Resolvente aus zwei Klauseln K_{i_1} und K_{i_2}, mit $1 \leq i_1 \leq i_2 \leq i - 1$.

Eine Klauselmenge M ist genau dann unerfüllbar, wenn eine Deduktion der leeren Klausel $K = \square$ aus M existiert.

Beispiel (Deduktion)

Für die Klauselmenge

$$M \;=\; \{\, c \vee \neg a,\; a \vee b,\; c_1 \vee c_2 \vee \neg a,\; d \vee \neg b \,\}$$

erhalten wir z.B. die folgende Deduktion $K_1, \ldots, K_5$ von $K = c \vee d$:

$$\begin{tikzcd}
K_1 = c \vee \neg a \arrow[dr] & & K_2 = a \vee b \arrow[dl] \\
& K_3 = b \vee c \arrow[dr] & & K_4 = d \vee \neg b \arrow[dl] \\
& & K_5 = c \vee d
\end{tikzcd}$$

Beispiel (Deduktion)

$$M = \{\, p \vee q, \neg p \vee \neg q, p \vee \neg q, \neg p \vee q \,\}.$$

Deduktion:

$$\begin{aligned}
K_1 &= p \vee q, \\
K_2 &= p \vee \neg q, \\
K_3 &= p, \\
K_4 &= \neg p \vee \neg q, \\
K_5 &= \neg p \vee q, \\
K_6 &= \neg p, \\
K_7 &= \square.
\end{aligned}$$

Resolutionsgraph:

$$\begin{tikzcd}
K_1 \arrow[dr] & K_2 \arrow[dl] & K_4 \arrow[dr] & K_5 \arrow[dl] \\
& K_3 \arrow[dr] & & K_6 \arrow[dl] \\
& & K_7 = \square &
\end{tikzcd}$$

Definition (Lineare Deduktion)

Eine *lineare Deduktion* (*Herleitung*, *Beweis*) einer Klausel K aus einer Klauselmenge M ist eine Folge

$$K_1, K_2, \ldots, K_{2 \cdot m+1}$$

ungerader Länge von Klauseln, mit $K_{2 \cdot m+1} = K$ und

- $K_1 \in M$ und
- $\forall\, 1 \leq i \leq m$:
  $K_{2 \cdot i+1}$ ist eine Resolvente aus $K_{2 \cdot i-1}$ und einer Klausel $K_{2 \cdot i} \in M$.

K_1 und die Klauseln $K_{2 \cdot i}$ mit geradem Index sind also alle in M,
die anderen Klauseln $K_{2 \cdot i+1}$ mit ungeradem Index sind Resolventen.

$$
\begin{tikzcd}
K_1 \in M \arrow[d] & K_2 \in M \arrow[dl] \\
K_3 \arrow[d, dashed] & \\
K_{2\cdot i-1} \arrow[d] & K_{2\cdot i} \in M \arrow[dl] \\
K_{2\cdot i+1} \arrow[d, dashed] & \\
K_{2\cdot m-1} \arrow[d] & K_{2\cdot m} \in M \arrow[dl] \\
K_{2\cdot m+1} &
\end{tikzcd}
$$

Beispiel (Lineare Deduktion)

1. Für die Horn–Klauselmenge

$$M \;=\; \{\, c \vee \neg a,\; a \vee b,\; c_1 \vee c_2 \vee \neg a,\; d \vee \neg b \,\}$$

   haben wir bereits eine lineare Deduktion $K_1, \ldots, K_5$ von $K = c \vee d$ gesehen.

2. Man kann zeigen, daß es für die Nicht–Horn–Klauselmenge

$$M \;=\; \{\, p \vee q, \neg p \vee \neg q, p \vee \neg q, \neg p \vee q \,\}$$

   keine lineare Deduktion der leeren Klausel $K = \square$ gibt.

   Die oben angegebene Deduktion von $K = \square$ war nicht linear.

Aus dem Erfüllbarkeitstest für Horn–Klauselmengen folgt:

Für alle unerfüllbaren Horn–Klauselmengen M gibt es eine lineare Deduktion

$$K_1, K_2, \ldots, K_{2 \cdot m+1}$$

der leeren Klausel $K = \square$, die mit einer Integritätsbedingung $K_1 \in M$ beginnt. Für diese Deduktion gilt dann:

- Auch alle anderen Klauseln $K_{2 \cdot i+1}$ mit ungeradem Index sind Integritätsbedingungen.
- Alle Klauseln $K_{2 \cdot i}$ mit geradem Index sind Regeln (einschließlich der Fakten) aus M mit einem positiven Atom, dem Kopfatom.

Auch die leere Klausel $K = \square$ kann man als Integritätsbedingung ansehen, da sie keine positiven Atome enthält.

Beispiel (Lineare Deduktion)

Für die unerfüllbare Horn–Klauselmenge

$$M \;=\; \{\, a,\; b,\; \neg a \vee \neg b \,\}$$

gibt es z.B. die folgende lineare Deduktion von $K = \square$:

$$\begin{tikzcd}
K_1 = \neg a \vee \neg b \arrow[d] & K_2 = a \arrow[dl] \\
K_3 = \neg b \arrow[d] & K_4 = b \arrow[dl] \\
K_5 = \square &
\end{tikzcd}$$