

Übung 4: Aussagenlogik II

Diskrete Strukturen im Wintersemester 2013/2014

Markus Kaiser

2. Dezember 2013

Identität $F \wedge \text{true} \equiv F$

$F \vee \text{false} \equiv F$

Dominanz $F \vee \text{true} \equiv \text{true}$

$F \wedge \text{false} \equiv \text{false}$

Idempotenz $F \vee F \equiv F$

$F \wedge F \equiv F$

Doppelte Negation $\neg\neg F \equiv F$

Triviale Tautologie $F \vee \neg F \equiv \text{true}$

Triviale Kontradiktion $F \wedge \neg F \equiv \text{false}$

Kommutativität $F \vee G \equiv G \vee F$
 $F \wedge G \equiv G \wedge F$

Assoziativität $(F \vee G) \vee H \equiv F \vee (G \vee H)$
 $(F \wedge G) \wedge H \equiv F \wedge (G \wedge H)$

Distributivität $F \vee (G \wedge H) \equiv (F \vee G) \wedge (F \vee H)$
 $F \wedge (G \vee H) \equiv (F \wedge G) \vee (F \wedge H)$

De Morgan $\neg(F \wedge G) \equiv \neg F \vee \neg G$
 $\neg(F \vee G) \equiv \neg F \wedge \neg G$

Implikation $F \rightarrow G \equiv \neg F \vee G$

Bikonditional $F \leftrightarrow G \equiv \neg(F \otimes G) [\equiv (F \rightarrow G) \wedge (G \rightarrow F)]$

Definition (Literal)

Ein **Literal** ist eine Variable $v \in V$ oder die Negation $\neg v$ einer Variable.

Definition (Klausel)

Eine **Klausel** verknüpft mehrere Literale mit einem assoziativen Operator.

Beispiel

Seien $a, \neg b, c$ Literale. Dann sind

- $a \wedge \neg b \wedge c$

- $a \vee \neg b \vee c$

Klauseln.

Definition (Disjunktive Normalform)

Eine **DNF-Klausel** ist eine Konjunktion von Literalen L_i .
Eine Formel F , ist in **Disjunktiver Normalform**, wenn sie eine Disjunktion von DNF-Klauseln ist.

$$F := \bigvee_i \bigwedge L_i$$

- Ausnahme: false ist auch in DNF

Beispiel

F ist in DNF.

$$F := \underbrace{(a \wedge b \wedge \neg c)}_{\text{DNF-Klausel}} \vee \underbrace{(\neg b \wedge c)}_{\text{DNF-Klausel}} \vee \underbrace{(\neg a \wedge b \wedge \neg c)}_{\text{DNF-Klausel}}$$

Definition (Konjunktive Normalform)

Eine **KNF-Klausel** ist eine Disjunktion von Literalen L_j .
Eine Formel F , ist in **Konjunktiver Normalform**, wenn sie eine Konjunktion von KNF-Klauseln ist.

$$F := \bigwedge_i \bigvee L_i$$

- Ausnahme: true ist auch in KNF

Beispiel

F ist in KNF.

$$F := \underbrace{(\neg a \vee b)}_{\text{KNF-Klausel}} \wedge \underbrace{(\neg b \vee c)}_{\text{KNF-Klausel}} \wedge \underbrace{(a \vee b \vee \neg c)}_{\text{KNF-Klausel}}$$

- Jede nicht-triviale Formel ist in DNF und KNF umwandelbar
- Durch Äquivalenzumformungen berechenbar (exponentiell groß!)
- Oder: Konstruktion mit Wahrheitstabellen

Normalformen aus Wahrheitstabellen

Gegeben eine Formel F und ihre Wahrheitstabelle

■ DNF

- 1 Betrachte Zeilen mit Eintrag 1
- 2 Bilde **Konjunktion** aus der **Belegung**
- 3 Bilde **Disjunktion** aller erhaltenen Klauseln

■ KNF

- 1 Betrachte Zeilen mit Eintrag 0
- 2 Bilde **Disjunktion** aus der **Negation** der Belegung
- 3 Bilde **Konjunktion** aller erhaltenen Klauseln

Beispiel

Gegeben eine Formel F mit folgender Semantik

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

F dargestellt in

■ DNF

$$(\neg a \wedge b \wedge c) \vee (a \wedge \neg b) \vee (a \wedge b \wedge \neg c)$$

■ KNF

$$(a \vee b) \wedge (\neg a \vee b \vee \neg c) \wedge (\neg a \vee \neg b \vee \neg c)$$

Mengendarstellung der KNF

Eine Formel $F = \bigwedge \bigvee L_i$ in KNF kann in einer Mengendarstellung repräsentiert werden.

- Klauseln werden durch Mengen von Literalen dargestellt

$$\{a, \neg b, c\} \text{ steht für } (a \vee \neg b \vee c)$$

- KNF-Formeln sind Mengen von Klauseln

$$\{\{\neg a\}, \{a, \neg b, c\}\} \text{ steht für } \neg a \wedge (a \vee \neg b \vee c)$$

- \emptyset steht für true, $\{\emptyset\}$ für false

Beispiel

Gegeben $F := (a \vee b) \wedge (\neg a \vee b \vee \neg c) \wedge (\neg a \vee \neg b \vee \neg c)$ in KNF.

$$\{\{a, b\}, \{\neg a, b, \neg c\}, \{\neg a, \neg b, \neg c\}\}$$

Idee

Erzeuge die KNF aus dem Syntaxbaum

- 1 Weise jedem **inneren Knoten** eine Variable zu
- 2 Variablen sind **abhängig** von ihren Kindern
- 3 Berechne **kleine** KNFs und führe diese **zusammen**

$$(x \wedge y) \vee z \equiv$$

$$\wedge (A_V \leftrightarrow A_\wedge \vee z)$$

$$\wedge (A_\wedge \leftrightarrow x \wedge y)$$

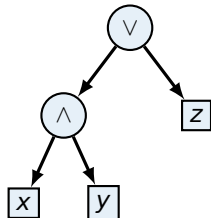
$$\equiv$$

$$\wedge (A_V \vee \neg A_\wedge) \wedge (A_V \vee \neg z)$$

$$\wedge (\neg A_V \vee A_\wedge \vee z)$$

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

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



Idee

Erzeuge die KNF aus dem Syntaxbaum

- 1 Weise jedem **inneren Knoten** eine Variable zu
- 2 Variablen sind **abhängig** von ihren Kindern
- 3 Berechne **kleine KNFs** und führe diese **zusammen**

$$(x \wedge y) \vee z \equiv$$

$$\wedge (A_V \leftrightarrow A_\wedge \vee z)$$

$$\wedge (A_\wedge \leftrightarrow x \wedge y)$$

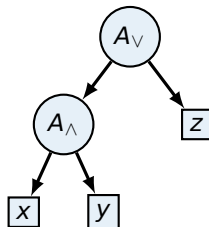
$$\equiv$$

$$\wedge (A_V \vee \neg A_\wedge) \wedge (A_V \vee \neg z)$$

$$\wedge (\neg A_V \vee A_\wedge \vee z)$$

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

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

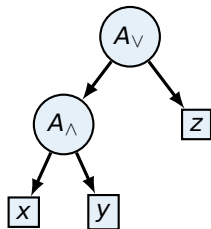


Idee

Erzeuge die KNF aus dem Syntaxbaum

- 1 Weise jedem **inneren Knoten** eine Variable zu
- 2 Variablen sind **abhängig** von ihren Kindern
- 3 Berechne **kleine KNFs** und führe diese **zusammen**

$$\begin{aligned}
 (x \wedge y) \vee z &\equiv A_V \\
 &\wedge (A_V \leftrightarrow A_\wedge \vee z) \\
 &\wedge (A_\wedge \leftrightarrow x \wedge y) \\
 &\equiv \\
 &\wedge (A_V \vee \neg A_\wedge) \wedge (A_V \vee \neg z) \\
 &\quad \wedge (\neg A_V \vee A_\wedge \vee z) \\
 &\wedge (\neg A_\wedge \vee x) \wedge (\neg A_\wedge \vee y) \\
 &\quad \wedge (A_\wedge \vee \neg x \vee \neg y)
 \end{aligned}$$

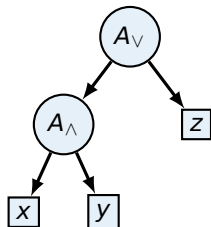


Idee

Erzeuge die KNF aus dem Syntaxbaum

- 1 Weise jedem **inneren Knoten** eine Variable zu
- 2 Variablen sind **abhängig** von ihren Kindern
- 3 Berechne **kleine** KNFs und führe diese **zusammen**

$$\begin{aligned}
 (x \wedge y) \vee z &\equiv A_V \\
 &\wedge (A_V \leftrightarrow A_\wedge \vee z) \\
 &\wedge (A_\wedge \leftrightarrow x \wedge y) \\
 &\equiv A_V \\
 &\wedge (A_V \vee \neg A_\wedge) \wedge (A_V \vee \neg z) \\
 &\quad \wedge (\neg A_V \vee A_\wedge \vee z) \\
 &\wedge (\neg A_\wedge \vee x) \wedge (\neg A_\wedge \vee y) \\
 &\quad \wedge (A_\wedge \vee \neg x \vee \neg y)
 \end{aligned}$$



Definition (DPLL-Belegung)

Sei F eine Formel in KNF und p eine Variable von F .
Dann bezeichnet $F[p \setminus \text{true}]$ die Formel, die entsteht, wenn jedes Vorkommen von p in F durch true ersetzt und vereinfacht wird.

DPLL

Gegeben eine Formel F in KNF

- Wenn $F = \text{true}$ dann antworte **erfüllbar**
- Wenn $F = \text{false}$ dann antworte **unerfüllbar**
- Sonst
 - 1 Wähle eine Variable p in F
 - 2 Prüfe ob $F[p \setminus \text{true}]$ oder $F[p \setminus \text{false}]$ erfüllbar

- Schlaue Wahl der Variable beschleunigt Ausführung
- Wähle Variablen die einzeln stehen (**One-Literal-Rule**)