

Vorlesung

– Automatisches Beweisen –

Prof. Dr. Wolfgang Kuechlin

Dipl.-Inform., Dr. sc. techn. (ETH)

**Arbeitsbereich Symbolisches Rechnen
Wilhelm-Schickard-Institut für Informatik
Fakultät für Informations- und Kognitionswissenschaften**

Universität Tübingen

**Steinbeis Transferzentrum
Objekt- und Internet-Technologien (OIT)**

Wolfgang.Kuechlin@uni-tuebingen.de

<http://www-sr.informatik.uni-tuebingen.de>



Vorlesung Wintersemester 2008:

Aussagenlogik

Prof. Dr. W. Küchlin



Logischer Formalismus

- **Syntax:** Wie werden Formeln gebildet?
 - Typisch: nach bestimmten Regeln gebildete Zeichenreihen
- **Semantik:** Was ist die Bedeutung einer Formel?
 - Wie kann der Wahrheitswert einer Formel ausgerechnet werden?
- **Kalkül:** Nach welchen Regeln können aus wahren Formeln weitere wahre Formeln gebildet werden?
 - Inferenzregeln



Aussagenlogik: Syntax

- Bestandteile von Formeln:
 - **Aussagenvariablen:** (x, y, z, \dots) : Platzhalter für beliebige (atomare) Aussagen, wie z.B. „5 ist eine Primzahl“, „ $2 > 3$ “
 - **Konstanten:** \perp [false] und \top [true] (oder auch f, t)
 - **Junktoren:** mindestens \neg [„nicht“], \wedge [„und“], \vee [„oder“]
- Nach Bedarf weitere, z.B.: $\Rightarrow, \Leftrightarrow, \oplus$
- Zur Bildung komplexer Formeln, die zusammengesetzte Aussagen repräsentieren.
- **Hilfssymbole:** (Klammern)

- **Formal: Formeln = (bestimmte) Zeichenreihen**



Aussagenlogik: Syntax (2)

- Aussagenlogische Formeln (induktiv definiert):
 - Alle Aussagenvariablen sind Formeln
 - f und t sind Formeln
 - Falls F und G Formeln, so auch $(\neg F)$, $(F \wedge G)$, $(F \vee G)$,
 $(F \Rightarrow G)$, $(F \Leftrightarrow G)$, $(F \oplus G)$
 - Nichts sonst ist eine Formel

➤ Beispiele:

- $((\neg(\neg x)) \vee (y \wedge z)) \vee f$
- $((x \wedge \neg(y \Rightarrow z)) \oplus x)$

➤ Präzedenz: (abnehmende Bindungsstärke)

- $\neg, \wedge, \vee, \Rightarrow, \Leftrightarrow, \oplus$



Aussagenlogik: Begriffe und Symbole

- $\mathcal{B} = \{0, 1\}$: Menge der (booleschen) Wahrheitswerte (0: falsch, 1: wahr)
- $\mathcal{V} = \{x_0, \dots, x_n, \dots, x, y, z, \dots\}$: Menge der Aussagenvariablen
- $\mathcal{L} = \Phi_0 \cup \neg\Phi_0 = \mathcal{V} \cup \{\neg x \mid x \in \mathcal{V}\}$: Menge der Literale
- $\text{Var}(F)$: Menge der in F vorkommenden Variablen
 - Z.B:

$$\text{Var}(((x \wedge \neg(y \Rightarrow z)) \oplus x)) = \{x, y, z\}$$



Semantik von Formeln (allgemein)

- Wie wird die Bedeutung einer Formel bestimmt?
- Bilde die Menge der neuen (unbekannten) Formeln \mathcal{F} ab in eine Menge bekannter (berechenbarer) Formeln \mathcal{B}
- Semantische Abbildung (Interpretation v , Bedeutungsfunktion β , meaning function μ)

$$\beta : \mathcal{F} \rightarrow \mathcal{B}$$

- Problem: endliche Definition dieser Abbildung
- Lösung: induktive Definition über Formelaufbau in \mathcal{F}
 - definiere Abbildung der Elementarsymbole
 - definiere Abbildung der Operatoren
 - Induktion



Aussagenlogik: Semantik

- Wie wird der Wahrheitsgehalt einer Formel bestimmt?
- Notwendig: Annahme über Wahrheitswerte der atomaren Aussagen (Aussagenvariablen)
 - Variablenbelegung: $\beta_0 : \text{Var}(F) \rightarrow \mathcal{B}$
 - Z.B: $\{x \mapsto 0, y \mapsto 1, z \mapsto 1\}$
- Berechnung des Wahrheitswertes einer Formel über Interpretation β



Aussagenlogik: Semantik (2)

- Variablenbelegung β_0 gegeben.
- Interpretation $\beta: \{\text{Formeln}\} \rightarrow \mathcal{B}$ rekursiv definiert:

$\beta(x) = \beta_0(x)$ für Aussagenvariable x

$$\beta(f) = 0$$

$$\beta(t) = 1$$

$$\beta(\neg F) = 1 - \beta(F)$$

$$\beta(F \vee G) = \max(\beta(F), \beta(G))$$

$$\beta(F \wedge G) = \min(\beta(F), \beta(G))$$

$$\beta(F \Rightarrow G) = \beta(\neg F \vee G)$$

$$\beta(F \Leftrightarrow G) = \beta((F \Rightarrow G) \wedge (G \Rightarrow F))$$

$$\beta(F \oplus G) = \beta(\neg(F \Leftrightarrow G))$$



Aussagenlogik: Semantik (3)

- Eine Formel F heißt **erfüllbar**, wenn es eine Variablenbelegung $\beta_0 : \text{Var}(F) \rightarrow \mathcal{B}$ gibt, so dass $\beta(F)=1$. β wird dann auch Modell von F genannt (i.Z. $\beta \models F$).
- Eine Formel F heißt **(allgemein-)gültig** (i.Z. $\models F$), falls für alle $\beta_0 : \text{Var}(F) \rightarrow \mathcal{B}$ gilt, dass $\beta(F)=1$.
- Berechnung der Erfüllbarkeit z.B. unter Zuhilfenahme von Wahrheitstabellen



Wahrheitstabelle

$$F = ((x \wedge \neg(y \Rightarrow z)) \oplus x)$$

x	y	z	$y \Rightarrow z$	$\neg(y \Rightarrow z)$	$x \wedge \neg(y \Rightarrow z)$	$(x \wedge \neg(y \Rightarrow z)) \oplus x$
0	0	0	1	0	0	0
0	0	1	1	0	0	0
0	1	0	0	1	0	0
0	1	1	1	0	0	0
1	0	0	1	0	0	1
1	0	1	1	0	0	1
1	1	0	0	1	1	0
1	1	1	1	0	0	1

Also ist F erfüllbar, aber nicht allgemeingültig.

Problem: n Variablen führen zu 2^n Zeilen in der Tabelle, daher nicht für große Formeln geeignet.



semantischer Folgerungsbegriff

- Eine Formel F folgt (semantisch) für eine Interpretation β aus einer Formelmenge \mathcal{M} , i.Z. $\mathcal{M} \models_{\beta} F$, falls $\beta(\mathcal{M})=1$ schon $\beta(F)=1$ impliziert.
 - Jede Belegung, die \mathcal{M} wahr macht, macht auch F wahr.
- Gilt $\mathcal{M} \models_{\beta} F$ für alle Interpretationen β , so folgt F (semantisch) aus \mathcal{M} , i.Z. $\mathcal{M} \models F$.
- Schreibweisen:
 - $G \models F$ für $\{G\} \models F$
 - $\models F$ für $\{\} \models F$



Äquivalenzumformungen

Für $F \equiv (F \Leftrightarrow G)$ schreiben wir auch $F \equiv G$.

- Achtung: \equiv ist ein Meta-Symbol, kein Operator der A-Logik
- \equiv bezeichnet eine Äquivalenzrelation (\rightarrow nachprüfen!)

➤ Distributivgesetze:

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

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

➤ Absorptionsgesetze:

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

➤ DeMorgansche Gesetze:

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

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

➤ Kommutativität und Assoziativität von \wedge , \vee .

.. und weitere



Boolesche Algebra

$[B; \wedge, \vee, ', 0, 1]$ ist ein **Boolesche Algebra**, wenn

1. \wedge und \vee sind assoziativ und kommutativ
2. es gelten die Distributivgesetze
3. $x \wedge 0 = 0$, $x \wedge 1 = x \quad \forall x$
4. $x \vee 0 = x$, $x \vee 1 = 1 \quad \forall x$
5. Zu jedem x existiert genau ein x' mit $x \wedge x' = 0$ und $x \vee x' = 1$.

Aus diesen Axiomen folgen (durch *equational reasoning*) weitere Gleichungen, z.B. die Absorptionsgesetze und die De Morgan'schen Regeln.



Beweise in der Aussagenlogik

Nun gilt:

- Mit der Äquivalenzrelation \equiv bilden die Formeln der Aussagenlogik eine Boolesche Algebra (\rightarrow nachprüfen!)
- Also folgen die üblichen Gesetze (de Morgan etc) durch equational reasoning.

Also stehen jetzt 2 Beweisverfahren für $F \equiv G$ zur Verfügung:

- per Definition von \equiv beweise $\models (F \Leftrightarrow G)$ über Belegungen
- Führe Gleichheitsbeweise (*equational reasoning*) in der Booleschen Algebra (Umformungen von F zu F' und/oder G zu G' die beide identisch gleich sind, also $F' = G'$).

Problem: keines der Verfahren ist effizient!



Dualität von Formeln

F heißt **dual** zu G, falls F aus G durch Vertauschen der Junktoren \wedge und \vee und der Symbole f und t entsteht.

Schreibweise: F^δ für die zu F duale Formel

Satz: Falls $F \equiv G$, so auch $F^\delta \equiv G^\delta$



Deduktionstheorem

Sei \mathcal{M} eine Menge von Formeln und seien F und G Formeln. Dann gilt:

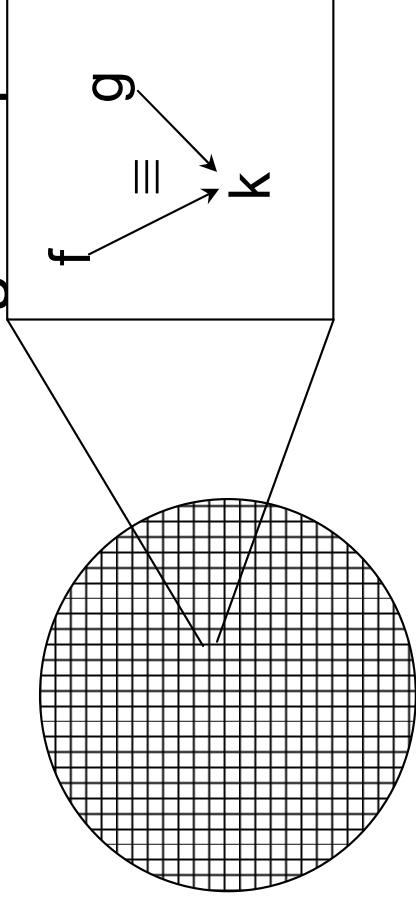
$\mathcal{M} \cup \{ F \} \models G$ impliziert $\mathcal{M} \models (F \Rightarrow G)$

Mithilfe des Deduktionstheorems lässt sich das Folgerungsproblem letztlich auf das Tautologieproblem zurückführen.



Normalformen

- Allgemeine Bedeutung von Normalformen:
 - Einheitliche Darstellung (bestimmtes Aussehen)
 - Einfachere Datenstrukturen
 - Einfachere Algorithmen
- Unterscheidung zur kanonischen Form:
 - kan. Form ist eindeutiger Repräsentant pro Äquivalenzklasse



Normalformen: vereinfachte Formel

➤ F heißt **vereinfacht**, wenn $F = \neg f$ oder $F = f$ oder wenn f in F nicht vorkommt (bzw. wenn $F = \top$ oder $F = \perp$ oder \top , \perp nicht in F vorkommen).

➤ **Beispiel:** $\neg (x \wedge f) \Rightarrow \neg y$ ist nicht vereinfacht

➤ **Algorithmus:** Wende folgende (Reduktions-)Regeln so lange wie möglich an:

- $F \vee f \Rightarrow F$ $F \vee t \Rightarrow t$
- $f \vee F \Rightarrow F$ $t \vee F \Rightarrow t$
- $F \wedge f \Rightarrow f$ $F \wedge t \Rightarrow F$
- $f \wedge F \Rightarrow f$ $t \wedge F \Rightarrow F$



Normalformen: Negationsnormalform (NNF)

- F ist in NNF: Negationen kommen nur direkt vor Variablen oder vor f vor
- **Beispiel:** $\neg(x \vee y)$ ist nicht in NNF, $\neg x \wedge \neg y$ ist in NNF
- **Algorithmus:**
 - Ausgehend von vereinfachter Form
 - Wende folgende Regeln so lange wie möglich an:
 - $\neg(F \wedge G) \Rightarrow \neg F \vee \neg G$
 - $\neg(F \vee G) \Rightarrow \neg F \wedge \neg G$
 - $\neg\neg F \Rightarrow F$



Normalformen: Konjunktive Normalform (CNF)

- F ist in CNF: F ist Konjunktion von Klauseln
- Klausel (clause): Disjunktion von Literalen
 - Horn-Klausel: hat höchstens ein positives Literal.
- Beispiel: $(x \vee y \vee \neg z)$ ist eine Klausel, $(\neg z \vee y)$ ist Horn-K.
- Beispiel: $(x \vee y \vee \neg z) \wedge (\neg z \vee y) \wedge x$ ist in CNF
- Auch CNF ist immer noch keine kanonische Form
 - $(x \vee \neg y) \wedge (x \vee y) \equiv x$. Beide Seiten sind in verschiedener CNF.

➤ Algorithmus:

- Ausgehend von NNF
- Wende folgende Regeln so lange wie möglich an
 - $F \vee (G \wedge H) \Rightarrow (F \vee G) \wedge (F \vee H)$
 - $(G \wedge H) \vee F \Rightarrow (G \vee F) \wedge (H \vee F)$



Normalformen: CNF - Darstellung

- Operatoren in CNF durch Form bestimmt:
 - ∨ immer in Klauseln, ∧ immer zwischen Klauseln

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

- Vereinfachung: Mengenschreibweise

$$F = \{\{x, \neg y\}, \{x, \neg z\}, \{z, \neg y\}, \{z, \neg x\}\}$$

- Operatoren weggelassen
- Formel = Menge von Klauseln
- Klausel = Menge von Literalen
- CNF zählt Menge von Constraints auf, die simultan erfüllt sein müssen, damit die beschriebene Funktion =1 wird.

- **Anmerkung:** DNF ist dual zu CNF

- DNF zählt die 1-Stellen der Funktion auf



Normalformen: CNF-Transformation - Komplexität

- Problem: Anwendung des Distributivgesetzes kann die Größe der Formel (annähernd) verdoppeln
- Im schlimmsten Fall: CNF(F) exponentiell größer als F
- Beispiel: $(x_{11} \wedge x_{12}) \vee \dots \vee (x_{n1} \wedge x_{n2})$.
 - CNF enthält 2^n Klauseln mit je n Variablen (\rightarrow Induktion)
- Lässt sich dieses Wachstum vermeiden?



CNF: Tseitin-Transformation

- Umformung von F in eine erfüllbarkeits-äquivalente Formel F^*
- $F \cong F^*$ iff [F ist erfüllbar gdw. F^* ist erfüllbar].
- Beispiel: $x \cong y$, aber $x \neq y$
- Idee: führe neue (Hilfs-)Variable für Konjunktion ein
- Verfahren von Tseitin:
 - $F \vee (G \wedge H) \cong (F \vee x) \wedge (G \vee \neg x) \wedge (H \vee \neg x)$
 - $(G \wedge H) \vee F \cong (F \vee x) \wedge (G \vee \neg x) \wedge (H \vee \neg x)$
- Ergebnis: keine Verdoppelung von F , kein exponentielles Wachstum bei CNF-Transformation



CNF: Tseitin-Transformation

➤ Begründung des Verfahrens von Tseitin:

- $F \vee (G \wedge H) \stackrel{\cong}{\equiv} (F \vee x) \wedge (G \vee \neg x) \wedge (H \vee \neg x)$
- $(G \wedge H) \vee F \stackrel{\cong}{\equiv} (F \vee x) \wedge (G \vee \neg x) \wedge (H \vee \neg x)$

➤ Beweis der Erfüllbarkeits-Äquivalenz

- Bemerkung: $(F \vee x) \wedge (G \vee \neg x) \wedge (H \vee \neg x) \equiv (F \vee x) \wedge (x \Rightarrow G) \wedge (x \Rightarrow H)$
- “ \rightarrow ”: Falls $\beta(F \vee (G \wedge H)) = 1$, dann $\beta(F) = 1$ oder $\beta(G \wedge H) = 1$
 - Falls $\beta(F) = 1$, dann existiert auch Erweiterung von β zu β^x mit $\beta^x(x) = 0$, also $\beta^x(F) = 1$ und $\beta^x(x \Rightarrow G) = 1$ und $\beta^x(x \Rightarrow H) = 1$, egal wie G, H von β bewertet sind
 - Falls $\beta(F) = 0$, dann folgt $\beta(G \wedge H) = 1$. Also existiert Erweiterung β^x von β mit $\beta^x(x) = 1$, also $\beta^x(F \vee x) = 1$ und $\beta^x(x \Rightarrow G) = 1$ und $\beta^x(x \Rightarrow H) = 1$.
- “ \leftarrow ”: $\beta^x((F \vee x) \wedge (x \Rightarrow G) \wedge (x \Rightarrow H)) = 1$.
 - Falls $\beta^x(x) = 1$, dann auch $\beta^x(G) = 1$ und $\beta^x(H) = 1$, also $\beta^x(F \vee (G \wedge H)) = 1$.
 - Falls $\beta^x(x) = 0$, dann auch $\beta^x(F) = 1$, also $\beta^x(F \vee (G \wedge H)) = 1$.



CNF: Tseitin-Transformation (2)

➤ Nach einer Tseitin-Transformation kann sich die Anzahl erfüllender Belegungen erhöhen (und zwar um den Faktor 2 für jede neue Tseitin-Hilfsvariable).

- Erinnerung Tseitin: $F \vee (G \wedge H) \equiv (F \vee x) \wedge (x \Rightarrow G) \wedge (x \Rightarrow H)$
- Falls $\beta(F)=1$ und $\beta(G \wedge H)=1$, dann $\beta^x((F \vee x) \wedge (x \Rightarrow G) \wedge (x \Rightarrow H))=1$ sowohl für β^x mit $\beta^x(x)=0$ als auch für β^x mit $\beta^x(x)=1$
 - in diesem Fall führt β zu 2 erfüllenden Belegungen, sonst nur zu einer.
- Umgekehrt: Falls es im transformierten System zwei erfüllende Belegungen $\beta^{x=1}$ und $\beta^{x=0}$ gibt, die sich nur auf einer Tseitin-Variable x unterscheiden, dann gibt es im Ursprungssystem dazu nur eine erfüllende Belegung
 - denn auf jeder Ursprungsvariablen v gilt $\beta^{x=1}(v) = \beta^{x=0}(v) = \beta(v)$.
 - Für die gemeinsame Restriktion β gilt in diesem Fall $\beta(F)=1$ und $\beta(G \wedge H)=1$, denn aus $\beta^{x=0}(F \vee x)$ folgt $\beta(F)=1$ und aus $\beta^{x=1}((x \Rightarrow G) \wedge (x \Rightarrow H))=1$ folgt $\beta(G \wedge H)=1$.

