

# Maschinelles Beweisen

Vorlesung mit Übungen

WS 2002/2003

Prof. F. v. Henke

# Organisatorisches

Vorlesung: Mo 10 – 12

Mi 10 – 12

Übungen finden während der Vorlesungszeiten statt,  
in unregelmäßigem Rythmus

Betreuer: Holger Pfeifer, Marko Luther

Sprechstunde: nach der Vorlesung, oder nach Vereinbarung

vhenke@informatik.uni-ulm.de

pfeifer@informatik.uni-ulm.de

**Vorlesungsunterlagen:** Kopien der Vorlesungsfolien werden über die Webseite zur Vorlesung verfügbar gemacht (Zugang über die Webseite der Abt. KI).

**Voraussetzungen:** Logik (wie für Vordiplom Informatik)

*Einführung in die KI* wünschenswert, aber nicht notwendig

# Einordnung der Lehrveranstaltung

Grundlegende Vorlesung für das Gebiet der maschinell unterstützten Deduktion

Grundlage für *formale Methoden* in der Informatik

Folgeveranstaltungen

- weiterführende Vorlesungen, wie
  - *Software-Verifikation*
  - *Modellierung und Analyse eingebetter Systeme*

Bemerkung: Die Lehrveranstaltung *Maschinelles Beweisen* wird in diesem Semester parallel zur KI-Einführung angeboten.

- Seminare, Praktika , Bachelor-, Master-, Diplomarbeiten
- Mitarbeit in Forschungsprojekten der Abteilung
- Tätigkeit als studentische Hilfskraft

Die Abteilung KI hat seit vielen Jahren einen Forschungsschwerpunkt im Gebiet des maschinellen Beweisens und dessen Anwendungen.

# 0. Vorbemerkungen

Gegenstand der Vorlesung:

Methoden des logischen Schließens,  
deren logische Grundlagen und maschinelle Unterstützung,  
verschiedene Anwendungen bei der formalen Modellierung und Analyse

“Maschinelle Unterstützung” reicht von maschineller Überprüfung manuell oder mit Hilfe eines “Beweis-Editors” eingegebener Beweise (“proof checking”) über interaktive Konstruktion von Beweisen bis zu vollautomatischer Beweisgenerierung.

Maschinelle Unterstützung ist unverzichtbar für umfangreiche Beweise.

Theoretische Grundlagen werden knapp und zusammenfassend bereitgestellt aber Betonung liegt mehr auf methodischen und praktischen Aspekten  
“angewandte Logik”, d.h. keine reine Logik-Vorlesung

Die Bedeutung des Gebiets der Inferenzsysteme bzw. des Maschinellen Beweisens als wissenschaftliche Disziplin zeigt sich z.B. durch die Existenz

- eigener Zeitschriften (z.B. Journal of Automated Reasoning)
- eigener internationaler Konferenzen  
z.B. CADE – *Conference on Automated Deduction*,  
CAV – *Computer-Aided Verification*
- eigener wissenschaftliche Vereinigungen (z.B. AAR)
- eines eigenen wissenschaftlichen Preises

## Warum beschäftigt man sich mit Inferenzsystemen und maschinellem Beweisen?

- Modellierung und Formalisierung menschlichen Schließens - seit Aristoteles
- Ableitung mathematischer Theoreme: “mechanisierte Mathematik” (QED-Projekt, Mizar-Projekt)
- In der Künstlichen Intelligenz: viele KI-Systeme bauen auf einer Form von Deduktion auf bzw. enthalten einen Modul für maschinelle Deduktion
- Logisches Programmieren
- Deduktive Datenbanken
- Deduktive Programm-Synthese (“automatisches Programmieren”)

## Warum beschäftigt man sich mit Inferenzsystemen und maschinellem Beweisen? (Forts.)

- Modellierung und Analyse von Systemen und System-Komponenten – sowohl Software wie Hardware:
  - Spezifikation, Validierung, Konsistenz-Nachweis für Implementierungen (Verifikation)
  - Nachweis kritischer Eigenschaften: z.B. Sicherheit (*safety, security*), Fehlertoleranz

Dies ist gegenwärtig das praktisch wichtigste Anwendungsgebiet für maschinelles Beweisen, als zentrale Komponente von formalen Methoden der Software- und Systementwicklung

“It is reasonable to expect that the relationship between computation and mathematical logic will be as fruitful in the next century as that between physics and analysis in the past.”

J. McCarthy, 1963

# Inhalt der Vorlesung

Verschiedene Ansätze des maschinellen Beweisens werden behandelt: unterschiedliche Logiken, Beweisstile, Systeme

- Grundlegende Begriffe formaler deduktiver Systeme (Axiom, Theorem, Beweis, Ableitbarkeit, Theorie, usw.)

Zunächst am Beispiel der Aussagenlogik

- Gleichungslogik, Gleichungskalküle und Termersetzung  
Anwendungen: Mathematische Theorien, algebraische Spezifikation abstrakter Datentypen, funktionale Programmierung
- Prädikatenlogik erster Stufe
  - Unifikation, Resolution (zusammenfassende Wiederholung, sofern notwendig)
  - Tableau-Methoden, natürliche Deduktion
  - getypte Logik
- Induktion: Beweise über rekursive Funktionen und induktiv definierte (abstrakte) Datenstrukturen

- Deduktion in Logik höherer Ordnung
- Entscheidbare Theorien und Entscheidungsprozeduren
- Temporallogik(en)
- Methoden der Modellüberprüfung (*model checking*),  
*Binary Decision Diagrams* (BDDs)
- Praktischer Umgang mit Beweiser-Systemen;  
insbesondere
  - PVS für getypte Prädikatenlogik, Logik höherer Ordnung, Induktion, abstrakte Datentypen
- Überblick über weitere wichtige Systeme zur Unterstützung des maschinellen Beweisens
- Anwendungen: Modellierung und Beweise in den Bereichen Software-Spezifikation, formale Verifikation, Hardware, Compiler, andere Systeme

Reihenfolge der Themen ist (noch) flexibel.

# Ziele der Vorlesung

Vertrautheit mit den behandelten Grundbegriffen und Logiken und Verständnis für deren Mechanisierung (maschinelle Unterstützung)

Vertrautheit mit wichtigen Anwendungsgebieten

Entwicklung der Fähigkeit, einfache Anwendungsprobleme formal anzugehen und zu lösen

Grundlage für das Verstehen weiterführender Literatur (an der Front der Forschung)

## Literatur:

für die “klassischen” Gebiete insbesondere

K.H. Bläsius, H.-J. Bürckert (Hrsg.), *Deduktionssysteme*. Automatisierung des logischen Denkens. 2. Auflage, Oldenbourg Verlag 1992.

D. Hofbauer, R.-D. Kutsche, *Grundlagen des maschinellen Beweisens*. Vieweg 1989.

Viele Logik-Bücher

S. auch: Semesterapparat

aber: es gibt kein Lehrbuch, daß die in der Vorlesung behandelten Themen hinreichend darstellt.

Material zu den zu benutzenden Beweisern wird bereitgestellt.

# 1. Grundlagen - Aussagenlogik

Wichtige Konzepte und Begriffe in Logiken:

- *Syntax* (Signatur, Formel, Term, . . . ):  
Festlegung, welche syntaktischen Gebilde als Formeln (Aussagen, "Sätze", *sentences*) einer Logik akzeptiert werden.
- *Semantik*: Festlegung, wann eine Formel als *wahr* (bzw. *falsch*) angesehen wird.
- *Modell*: mathematische Struktur, bezüglich der die Wahrheit einer Aussage definiert werden kann;  
Interpretation von Formeln
- *Folgerbarkeit*: wann/wie auf die Wahrheit einer Aussage aus der Wahrheit anderer Aussagen geschlossen werden kann.
- *Ableitung, Schlußregel, Beweis, Ableitbarkeit*

- Axiom, Theorem, Theorie
- (Un-)Erfüllbarkeit, (Un-)Gültigkeit
- (Un-)Entscheidbarkeit

diskutiert zunächst am Beispiel der *Aussagenlogik (Propositional Logic)*:

Aussagenlogik formalisiert das Umgehen mit einfachen “Aussagen” (*propositions*), denen ein Wahrheitswert (wahr oder falsch) zugeordnet wird.

# Aussagenlogik: Syntax

Die *Formeln* der Aussagenlogik werden auf folgende Weise gebildet.

- Das Vokabular besteht aus
  - Konstanten  $W$  und  $F$  (“wahr” und “falsch”)
  - Aussagen-Symbolen  $A, B, C, \dots$  (Elemente einer abzählbaren Menge)
  - den logischen Verknüpfungen (Junktoren, *connectives*):
    - einstellig:  $\neg$  (Negation)
    - zweistellig:  $\wedge$  (Konjunktion),  $\vee$  (Disjunktion),  $\Rightarrow$  (Implikation),  $\Leftrightarrow$  (Äquivalenz)(Die Notation für Verknüpfungen ist nicht einheitlich.)

## Aussagenlogik: Syntax (2)

- Die Menge der *aussagenlogischen Formeln* oder *Aussagen* (*propositional formulae*, *propositions*) wird definiert durch:
  1. Jede Konstante und jedes Aussagen-Symbol ist eine Formel.
  2. Für beliebige Formeln  $X, Y$  sind  $\neg X, X \wedge Y$  usw. (für alle Verknüpfungen) Formeln.
  3. Alle Formeln werden nach 1. oder 2. gebildet.
- Die o.a. Reihenfolge der Verknüpfungen gibt auch die Präzedenz an (absteigend von links nach rechts). Sofern für Eindeutigkeit notwendig, werden Klammern zur Gruppierung benutzt:  
z.B.  $(A \vee B) \wedge C$

Bemerkung: Eine induktive Definition wie die der Menge von Formeln ist typisch für die Art der Beschreibung von Syntax.

# Aussagenlogik: Semantik

Eine aussagenlogische Formel hat für sich allein keine Bedeutung. Sie erhält eine Bedeutung dadurch, daß den aussagenlogischen Symbolen, die in ihr vorkommen, Wahrheitswerte zugewiesen werden.

Eine *Interpretation* von aussagenlogische Formeln ist eine Zuordnung von Wahrheitswerten zu Formeln. Formal:

- Die Grundlage einer Interpretation ist eine Abbildung der Menge der Symbole in die Menge der Wahrheitswerte  $\{\mathbf{W}, \mathbf{F}\}$ .
- Den Konstanten  $W$  und  $F$  wird immer der Wert  $\mathbf{W}$  bzw.  $\mathbf{F}$  zugeordnet.
- Die Bedeutung einer Verknüpfung  $c$  ist eine bestimmte Funktion  $f_c$  auf Wahrheitswerten; die Funktion kann (und wird typischerweise) in der Form einer *Wahrheitstafel* (oder -tabelle) angegeben werden.

Mit dieser Semantik der Verknüpfungen wird eine Interpretation von Symbolen eindeutig erweitert zu einer Interpretation von Formeln: unter einer Interpretation erhält jede (aussagenlogische) Formel einen Wahrheitswert.

Eine Interpretation, unter der eine Aussage wahr ist, definiert ein *Modell* für die Aussage.

Eine Interpretation ist ein Modell für eine Menge von Aussagen, wenn sie Modell jeder Aussage in der Menge ist.

Das umgangssprachliche “ $X$  ist wahr” bedeutet genauer:  $X$  hat den Wahrheitswert **W** (unter einer gegebenen Interpretation).

## Aussagenlogik: Semantik (2)

Verschiedene Interpretationen können einer Formel unterschiedliche Wahrheitswerte zuordnen; sie beschreiben verschiedene “mögliche Welten”. Intendierte Bedeutungen von Aussagen sind aber für den Kalkül irrelevant; er berücksichtigt nur die formalen Wahrheitswerte.

Eine Formel heißt

- *gültig* (*valid*) oder *Tautologie*, wenn sie für jede Interpretation wahr ist (z.B.  $P \vee \neg P$ ).
- *erfüllbar* (*satisfiable*), wenn es eine Interpretation (ein Modell) gibt, die sie wahr macht (z.B.  $P \wedge \neg Q$ ).
- *unerfüllbar* oder *widersprüchlich*, wenn sie für keine Interpretation wahr ist (z.B.  $P \wedge \neg P$ ).

**Satz:** Es ist entscheidbar, ob eine Formel (a) eine Tautologie ist, (b) erfüllbar ist.

Beweis: Ein Entscheidungsalgorithmus ist z.B. gegeben durch das Aufstellen einer Wahrheitstafel für alle  $2^n$  möglichen Interpretationen ( $n$  die Anzahl der verschiedenen in der Formel vorkommenden Aussagen-Symbole).

*Semantische Folgerung (entailment):* Für eine Aussagenmenge  $S$  und eine Aussage  $X$  heißt  $X$  semantische Folgerung von  $S$ , wenn jede Interpretation, die  $S$  wahr macht, auch  $X$  wahr macht.

Notation:  $S \models X$  soll bedeuten “ $X$  folgt (semantisch) aus  $S$ ”

“ $X$  gültig” ist gleichbedeutend mit “ $\models X$ ,”  
d.h.  $X$  folgt semantisch aus der leeren Menge von Formeln.

# Aussagenlogik: Gesetze

Die folgenden Gesetze gelten für aussagenlogische Formeln.

- Kommutativität:

$$X \wedge Y \Leftrightarrow Y \wedge X$$

$$X \vee Y \Leftrightarrow Y \vee X$$

$$(X \Leftrightarrow Y) \Leftrightarrow (Y \Leftrightarrow X)$$

- Assoziativität:

$$(X \wedge Y) \wedge Z \Leftrightarrow X \wedge (Y \wedge Z)$$

$$(X \vee Y) \vee Z \Leftrightarrow X \vee (Y \vee Z)$$

- Distributivität:

$$X \wedge (Y \vee Z) \Leftrightarrow (X \wedge Y) \vee (X \wedge Z)$$

$$X \vee (Y \wedge Z) \Leftrightarrow (X \vee Y) \wedge (X \vee Z)$$

- deMorgan'sche Regeln:

$$\neg(X \wedge Y) \Leftrightarrow (\neg X \vee \neg Y)$$

$$\neg(X \vee Y) \Leftrightarrow (\neg X \wedge \neg Y)$$

- Umformungen:

$$\neg\neg X \Leftrightarrow X$$

$$X \Rightarrow Y \Leftrightarrow \neg X \vee Y$$

...

Die logischen Äquivalenzen können auch als “Gleichungen” interpretiert und im Sinne eines Gleichungskalküls als Ersetzungsregeln benutzt werden.

~> Thema “Termersetzung”

# Aussagenlogik: Normalformen

Verschiedene Interpretationen können einer Formel unterschiedliche Wahrheitswerte zuordnen; sie beschreiben verschiedene “mögliche Welten”. Intendierte Bedeutungen von Aussagen sind aber für den Kalkül irrelevant; er berücksichtigt nur die formalen Wahrheitswerte.

*Literale:*

- Symbole  $A, B, \dots$  (*positive* Literale)
- negierte Symbole  $\neg A, \neg B, \dots$  (*negative* Literale)

Die Menge der Literale ist eine Teilmenge der Formeln.

Für ein Literal  $L$  soll  $\bar{L}$  seine negierte Form bezeichnen, also  $\overline{\bar{A}} := \neg A$  und  $\overline{\neg A} := A$ .

*Konjunktive Normalform* (KNF): Formel der Form Konjunktion von Disjunktionen von Literalen, z.B.

$$(L_{11} \vee \dots \vee L_{1n_1}) \wedge \dots \wedge (L_{m1} \vee \dots \vee L_{mn_m})$$

*Disjunktive Normalform* (DNF): Formel der Form Disjunktion von Konjunktionen von Literalen, z.B.

$$(L_{11} \wedge \dots \wedge L_{1n_1}) \vee \dots \vee (L_{m1} \wedge \dots \wedge L_{mn_m})$$

**Satz:** Jede aussagenlogische Formel kann in äquivalente Formeln in konjunktiver bzw. disjunktiver Normalform umgewandelt werden.

*Klausel:* Disjunktion von Literalen, häufig als Menge geschrieben.

Eine Formel in KNF kann als Menge von Klauseln dargestellt werden.

# Axiomatische Systeme

Ein axiomatisches System (Hilbert-System, formaler Kalkül) wird charakterisiert durch

- eine Menge von wohlgeformten Formeln (bzw. eine Vorschrift, z.B. eine Grammatik, die deren Form bestimmt)
- eine Menge von *Axiomen*, eine Teilmenge der Menge von Formeln, die als gültig angesehen werden
- eine Menge von Schluß- oder Ableitungsregeln, die eine Menge von Formeln, den Prämissen, eine Formel, die Schlußfolgerung, zuordnet; geschrieben i.a. als

$$\frac{F_1, \dots, F_n}{F_{n+1}}$$

Eine *Ableitung* aus einer Menge  $S$  von Formeln ist eine endliche Folge  $F_1, \dots, F_n$  von Formeln, so daß jede Formel entweder ein Axiom (genauer: eine Instanz eines Axiomenschemas), ein Element von  $S$ , oder die Schlußfolgerung der Anwendung einer Ableitungsregel, angewandt auf vorhergehende Formeln als Prämissen, ist.

## Axiomatische Systeme (2)

Ein *Beweis* ist eine Ableitung aus der leeren Menge von Formeln.

Eine Formel  $F$  ist *ableitbar* aus  $S$  in einem axiomatischen System, wenn es die letzte Formel einer Ableitung aus  $S$  ist.

Eine Formel  $F$  ist ein *Theorem*, wenn es die letzte Formel eines Beweises ist.

Ableitbarkeit wird üblicherweise durch das Symbol  $\vdash$  (engl. *turnstile*) bezeichnet:

$S \vdash F$       $F$  ist aus (der Menge von Formeln)  $S$  ableitbar.  
 $\vdash F$       $F$  ist ein Theorem.

## Axiomatische Systeme (3)

**Deduktionstheorem:** Eine Formel  $G$  ist aus  $S \cup \{F\}$  ableitbar genau dann, wenn  $F \Rightarrow G$  aus  $S$  ableitbar ist:

$$S \cup \{F\} \vdash G \text{ g.d.w. } S \vdash F \Rightarrow G$$

Der Beweis setzt eine minimale Stärke des axiomatischen Systems voraus.

Bemerkung: Das Theorem ist ein *Metatheorem*: es sagt etwas über ein axiomatisches System aus.

# Axiomatische Systeme (4)

## Adäquatheit eines Kalküls:

- *Korrektheit*: nur folgerbare Formeln sind ableitbar  
 $\vdash F$  impliziert  $\models F$
- *Vollständigkeit*: jede folgerbare Formel ist ableitbar  
 $\models F$  impliziert  $\vdash F$

# Axiomatisierung der Aussagenlogik

Ein axiomatisches System für Aussagenlogik ist gegeben durch:

Axiome: für beliebige Formeln  $X, Y, Z$  (d.h. es handelt sich um Axiomen-Schemata)

$$X \Rightarrow (Y \Rightarrow X)$$

$$(X \Rightarrow (Y \Rightarrow Z)) \Rightarrow ((X \Rightarrow Y) \Rightarrow (X \Rightarrow Z))$$

$$(\neg X \Rightarrow \neg Y) \Rightarrow ((\neg X \Rightarrow Y) \Rightarrow X)$$

Bemerkung: Jedes der Axiome ist eine Tautologie.

Schlußregel: *modus ponens*

$$\frac{X \quad X \Rightarrow Y}{Y}$$

## Axiomatisierung der Aussagenlogik (2)

**Satz:** Das angegebene axiomatische System ist korrekt und vollständig.

Das Axiomensystem ist minimal: keines der Axiome kann fortgelassen werden. Beachte, daß die anderen logischen Verknüpfungen mit Hilfe von  $\neg$  und  $\Rightarrow$  ausgedrückt werden können.

Bemerkung: Das Deduktionstheorem für Aussagenlogik läßt sich aus den ersten beiden Axiomen und der Schlußregel ableiten.

# Beweise im Aussagenkalkül

In der Aussagenlogik ist man meistens daran interessiert, Tautologien abzuleiten/zu beweisen.

Beweismethoden:

1. Aufstellen einer Wahrheitstafel
2. Ableiten durch äquivalente Umformung (mit Hilfe der o.a. Regeln)
3. Beweis aus den Axiomen mit Hilfe der Schlußregeln
4. Wang-Algorithmus
5. Resolutionsverfahren
6. Tableaux-Verfahren
7. OBDD's (*'Ordered Binary Decision Diagrams'*)

8. und mehr . . .

Einige dieser Verfahren werden im Laufe der Vorlesung behandelt.

*Beispiel* eines Beweises aus Axiomen mit der Schlußregel:

für die Formel  $P \Rightarrow P$

- (1)  $(P \Rightarrow ((P \Rightarrow P) \Rightarrow P)) \Rightarrow ((P \Rightarrow (P \Rightarrow P)) \Rightarrow (P \Rightarrow P))$
- (2)  $(P \Rightarrow ((P \Rightarrow P) \Rightarrow P))$
- (3)  $((P \Rightarrow (P \Rightarrow P)) \Rightarrow (P \Rightarrow P))$
- (4)  $(P \Rightarrow (P \Rightarrow P))$
- (5)  $(P \Rightarrow P)$

## Beispiel-Beweis mit Wahrheitstafel:

$$(*) \quad [P \Rightarrow (Q \Rightarrow R)] \Leftrightarrow [(P \wedge Q) \Rightarrow R]$$

$P$	$Q$	$R$	$Q \Rightarrow R$	$P \Rightarrow (Q \Rightarrow R)$	$P \wedge Q$	$P \wedge Q \Rightarrow R$	(*)
$W$	$W$	$W$	$W$	$W$	$W$	$W$	$W$
$W$	$W$	$F$	$F$	$F$	$W$	$F$	$W$
$W$	$F$	$W$	$W$	$W$	$F$	$W$	$W$
$W$	$F$	$F$	$W$	$W$	$F$	$W$	$W$
$F$	$W$	$W$	$W$	$W$	$F$	$W$	$W$
$F$	$W$	$F$	$F$	$W$	$F$	$W$	$W$
$F$	$F$	$W$	$W$	$W$	$F$	$W$	$W$
$F$	$F$	$F$	$W$	$W$	$F$	$W$	$W$

# Wang-Algorithmus

(H. Wang 1960) – eine Beweismethode für Aussagenlogik, die im allgemeinen schneller ist als das Aufstellen einer Wahrheitstafel

Voraussetzung: Zu beweisende Aussage wird formuliert in Form einer *Sequenz* (engl. *sequent*): Prämissen links vom Ableitungssymbols  $\vdash$ , die abzuleitende Schlußfolgerung rechts von  $\vdash$ .

Beispiel:

$$P \Rightarrow Q, \quad Q \Rightarrow R, \quad \neg R \vdash \neg P$$

Die allgemeine Form einer Sequenz,

$$P_1, P_2, \dots, P_m \vdash Q_1, \dots, Q_n$$

ist (informell) zu interpretieren als

$$P_1 \wedge P_2 \wedge \dots \wedge P_m \Rightarrow Q_1 \vee \dots \vee Q_n$$

## Wang-Algorithmus (2)

Der Wang-Algorithmus besteht darin, die Sequenz schrittweise nach den folgenden Regeln so umzuformen, daß einfachere Sequenzen entstehen.

1. Eine negierte Aussage  $\neg X$  wird gestrichen und  $X$  auf der anderen Seite von  $\vdash$  hinzugefügt.
2. Für eine Konjunktion  $X \wedge Y$  auf der linken Seite wird  $\wedge$  durch ein Komma ersetzt. Für eine Disjunktion  $X \vee Y$  auf der rechten Seite wird  $\vee$  durch ein Komma ersetzt.
3. Wenn eine Aussage auf der linken Seite die Form  $X \vee Y$  hat, wird die Sequenz durch zwei neue Sequenzen ersetzt, in denen die Disjunktion durch  $X$  bzw.  $Y$  ersetzt ist.
4. Wenn eine Aussage auf der rechten Seite die Form  $X \wedge Y$  hat, wird die Sequenz durch zwei neue Sequenzen ersetzt, in denen die Konjunktion durch  $X$  bzw.  $Y$  ersetzt ist.

5. Eine Implikation  $X \Rightarrow Y$  wird durch  $\neg X \vee Y$  ersetzt, d.h. Implikationen werden eliminiert.
6. (Terminierungsregel 1): Eine Sequenz wird als bewiesen angesehen, wenn eine Aussage  $X$  sowohl auf der linken wie auf der rechten Seite von  $\vdash$  auftritt. (Man beachte, daß  $X$  auch zusammengesetzt sein kann, also kein Symbol sein muß.) Eine solche Sequenz wird *Axiom* genannt.  
Die ursprüngliche Sequenz ist bewiesen, wenn alle aus ihr abgeleiteten Sequenzen bewiesen sind.
7. (Terminierungsregel 2): Eine Sequenz ist nicht als gültig nachgewiesen, wenn alle Formeln in ihr individuelle Aussagensymbole sind und kein Symbol sowohl auf der linken wie der rechten Seite von  $\vdash$  auftritt. Wenn eine solche Sequenz gefunden worden ist, terminiert der Algorithmus, und die ursprüngliche Schlußfolgerung folgt nicht aus den Prämissen.

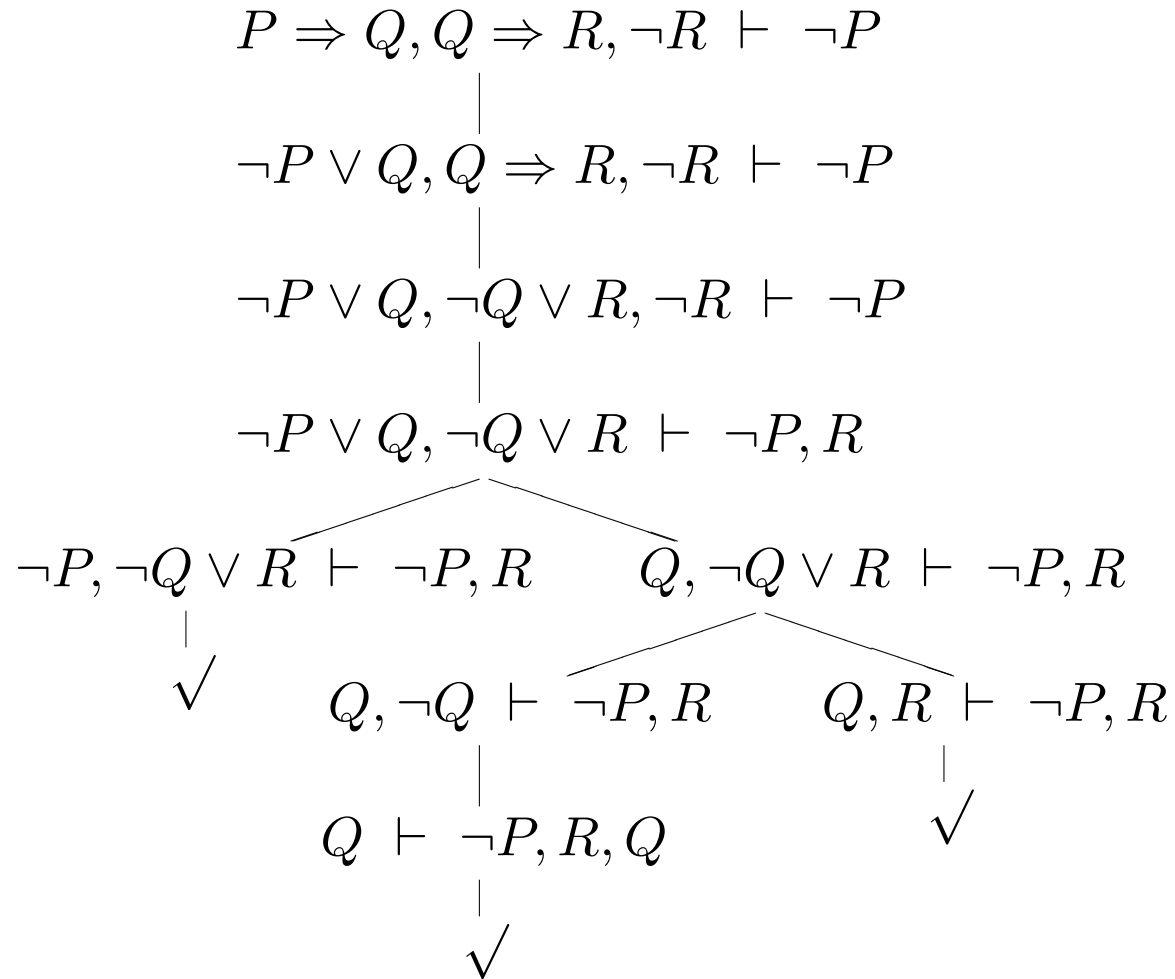
# Wang-Algorithmus: Beispiel

Beispiel eines Beweises mit dem Wang-Algorithmus für die o.a. Sequenz:

- |                                                              |                                                                           |
|--------------------------------------------------------------|---------------------------------------------------------------------------|
| (1) $P \Rightarrow Q, Q \Rightarrow R, \neg R \vdash \neg P$ | Ausgangs-Sequenz                                                          |
| (2) $\neg P \vee Q, \neg Q \vee R, \neg R \vdash \neg P$     | $2 \times \Rightarrow$ -Regel                                             |
| (3) $\neg P \vee Q, \neg Q \vee R \vdash \neg P, R$          | $\neg$ -Regel                                                             |
| (4) $\neg P, \neg Q \vee R \vdash \neg P, R$                 | aus (3) mit $\vee$ -Regel<br>$\Rightarrow$ (4) ist Axiom ( $\checkmark$ ) |
| (5) $Q, \neg Q \vee R \vdash \neg P, R$                      | auch aus (3) mit $\vee$ -Regel                                            |
| (6) $Q, \neg Q \vdash \neg P, R$                             | aus (5) mit $\vee$ -Regel                                                 |
| (7) $Q, R \vdash \neg P, R$                                  | auch aus (5) mit $\vee$ -Regel: $\checkmark$                              |
| (8) $Q \vdash \neg P, R, Q$                                  | aus (6) mit $\neg$ -Regel: $\checkmark$                                   |

alle Sequenzen auf Axiome reduziert:  $\rightsquigarrow$  fertig

## Beispiel-Beweis als (Und-)Baum:



## **Korrektheit** des Wang-Algorithmus?

Der Algorithmus terminiert immer mit einer eindeutigen Lösung: In jedem Schritt wird eine Verknüpfungsoperation eliminiert, wodurch die Sequenz verkürzt wird, selbst wenn neue Sequenzen generiert werden.

Die Reihenfolge, in der die Regeln angewendet werden, hat keinen Einfluß auf die Terminierung und das Resultat, wohl aber auf die Länge der Ableitung.

# Grundbegriffe: Zusammenfassung

- Formalisierung der Syntax der Sprache einer Logik
- Formalisierung der Semantik: Modelltheorie
- Ableitbarkeit (auf syntaktischen Strukturen)
- Folgerbarkeit (als semantische/modelltheoretischer Begriff)
- Ableitbarkeit und Folgerbarkeit sollen nach Möglichkeit zusammenfallen (wie bei der Aussagenlogik)
- Im Idealfall ist die Logik (oder eine Teilmenge) entscheidbar (oder wenigstens semi-entscheidbar), so daß mit Formeln “gerechnet” werden kann.  
Ein Ziel der maschinellen Deduktion ist es, dieses Rechnen möglichst effektiv und effizient zu machen.