

*Einführung in die Informatik I*  
*Aussagenlogik*

Prof. Bernd Brügge, Ph.D  
Technische Universität München

Wintersemester 2000/2001

20 November 2000

# *Überblick über den Vorlesungsblock*

- ❖ Rubrik Sonstiges
- ❖ **Wiederholung einiger Folien der letzten Vorlesung (mit Folie # in dieser Vorlesung gekennzeichnet):**
  - Boolesche Algebra
  - Interpretation
  - Semantische Äquivalenz
- ❖ Beweistheorie und Modelltheorie
- ❖ Syntax und Semantik
- ❖ **Definition der Aussagenlogik:** Eine Boolesche Algebra (Signatur, Operationen, Gesetze) und 3 Metaregeln: Modus Ponens, Tertium Non Datur und Gleichheitsregel
- ❖ Viele Beispiele

## *Rubrik Sonstiges*

- ❖ Bitte keine Papierflieger (Schwalben) während der Vorlesung
- ❖ Einlegung einer Pause nach 45 Minuten:
  - Fragen und Antworten
- ❖ Zusätzliche Literatur für diese Vorlesung:
  - **David Gries, Fred B. Schneider** "A logical approach to discrete math", Springer Verlag, ISBN 0-387-94115-0
  - **Geoffrey Hunter**, "Metalogic: An introduction to the metatheory of standard first order logic, University of California Press, ISBN 0-520-02356-0
  - **Douglas Hofstadter**, "Gödel, Escher, Bach: An Eternal Golden Braid", Basic Books, New York 1979, ISBN 0-465-02685-0.  
(PQ- Logik)
  - **Roland Backhouse**: "Program construction and Verification", Prentice Hall, 1986. (Superman Beispiel)

## *Bevor wir weitergehen: Warum machen wir überhaupt Formale Systeme?*

- ❖ Die **Hauptaufgabe** bei der Erstellung von Informatik-Systemen ist die Modellierung, d.h.
  - die Abbildung von Gegenständen aus der Realität auf ein geeignetes Informatik-System.
- ❖ Das **Ziel** ist es ein Programm schreiben - d.h. einen Algorithmus, der dann von einem Rechner exekutierbar ist, und für bestimmte Eingaben bestimmte Ausgaben errechnet.
- ❖ Die Eingabe, die Darstellung des Algorithmus und die Ausgabe müssen *ohne Bezug auf irgend eine Interpretation maschinell verarbeitbar* sein.
  - Genau das macht eine formale Sprache.
  - Das ist sogar die Essenz einer formalen Sprache: Sie ist vollständig ohne Bezug auf eine Bedeutung definiert.

# *Anwendung formaler Sprachen*

- ❖ Eine **formale Sprache** ist definiert durch die Menge aller wohlgeformten Terme, wir haben sie auch "syntaktisch korrekte Terme" genannt.
- ❖ Um einem Rechner ein Programm exekutieren zu lassen, schreibe ich es in einer formalen Sprache, und zwar so, dass das Programm ein syntaktisch korrekter Term der Sprache ist.
  - Das heißt, ich möchte keine Fehlermeldungen vom Compiler haben.
- ❖ Natürlich schreibe ich das Programm nicht einfach durch Aneinanderreihen von Buchstaben! Das Programm soll auch eine Bedeutung haben! Damit kommen wir zum Begriff der Interpretation:
  - Eine Interpretation eines korrekten Terms einer formalen Sprache ist eine Belegung der Symbole, die in den Termen auftauchen, durch Werte aus der Anwendungsdomäne.

# *Formale Logik: Beweistheorie und Modelltheorie*

- ❖ In der **Formalen Logik** betrachten wir formale Systeme unter 2 wichtigen Aspekten:
  - **Beweistheorie:** Der Teil der formalen Sprachen, in dem man Ableitungen durchführen kann, *ohne* sich um die Bedeutung der Symbole zu kümmern. Wichtige Themen:
    - ◆ Wie muss ein formaler Beweis aufgebaut sein?  
Was sind die ableitbaren Formeln (auch Theoreme genannt) eines Systems?
  - **Modelltheorie:** Die Theorie der Interpretation von formalen Sprachen. Wichtige Themen:
    - ◆ Interpretationen, Konsistenz, Inkonsistenz
    - ◆ Vollständigkeit, Korrektheit
- ❖ Wichtig: Die formale Logik definiert den Begriff Modell anders als die Informatik!

# *Zwei Definitionen des Begriffs Modells*

## ❖ **Definition Modell (in der Logik):**

- Die Interpretation einer wohldefinierten Formel, die den Wert L (d.h. „wahr“) ergibt.

## ❖ **Definition Modell (in der Informatik, siehe erste Vorlesung):**

- Die Menge aller
  - ◆ Begriffe von (real existierenden oder nur gedachten) Dingen,
  - ◆ Begriffe von Personen,
  - ◆ Begriffe von Abläufen in der Zeit, und
  - ◆ Beziehungen zwischen diesen Begriffen.

# *Warum müssen wir Boolesche Algebra lernen?*

- ❖ Kenntnis der Logik und der Booleschen Algebra sind für den Informatiker sehr wichtig:
  - Algorithmen (Informatik) können durch Regelsysteme (Logik) beschrieben werden.
  - Die formalen Ableitungssysteme einer Logik sind ebenfalls Regelsysteme.
  - Systeme logischer Formeln können als Beschreibungen von Algorithmen gesehen werden (Programmiersprache Prolog, Logikprogrammierung).
- ❖ **Programmierung:** Boolesche Ausdrücke sind Bestandteil vieler Programmiersprachen (z.B. in bedingten Anweisungen)
- ❖ **Detaillierter Entwurf:** Vor und nach dem Aufruf einer Operation müssen bestimmte Bedingungen gelten, die wir durch Boolesche Gleichungen beschreiben können. (Wir nennen das den **Kontrakt** einer Schnittstelle).

# *Syntax und Semantik*

- ❖ Analog zu Beweistheorie und Modelltheorie in der Logik unterscheiden wir auch Syntax und Semantik bei den formalen Sprachen.
- ❖ **Definition Syntax:** Alles, was mit formalen Sprachen und formalen Systemen zu tun hat, ohne dass wir irgendwann mal eine Interpretation brauchen.
- ❖ **Definition Semantik:** Alles, was mit der Interpretation von formalen Sprachen und formalen Systemen zu tun hat.
- ❖ Beispiel 1:
  - **Frage:** Ist es eine syntaktische oder semantische Eigenschaft einer wohldefinierten Formel, ob sie eine natürliche Zahl bezeichnet?
  - **Antwort:** Eine semantische Eigenschaft. Der letzte Teil des Satzes kann formuliert werden als "ob sie so interpretiert werden kann, dass sie eine natürliche Zahl darstellt".

## *Noch zwei Beispiele*

### ❖ **Beispiel 2:**

- **Frage:** Ist es eine syntaktische oder semantische Eigenschaft einer wohldefinierten Formel, dass sie wahr ist?
- **Antwort:** Eine semantische Eigenschaft. Der letzte Teil des Satz kann umformuliert werden in "*dass sie als wahr interpretiert werden kann.*"

### ❖ **Beispiel 3:**

- **Frage:** Ist es eine syntaktische oder semantische Eigenschaft einer wohldefinierten Formel, dass sie aus 7 Zeichen und Operatoren besteht?
- **Antwort:** Eine syntaktische Eigenschaft.

## *Kleine Wiederholung zur Syntax*

- ❖ Die Menge aller wohldefinierten Terme wird durch ihren Grammatik-Erfinder (also durch Sie!) definiert. Was müssen Sie dabei beachten?
  - Zunächst einmal müssen Sie eine Menge von Symbolen definieren (das Alphabet der Sprache).
  - Dann müssen Sie eine Anzahl von Transformationsregeln schreiben, die dann bestimmen, welche Sequenzen von Symbolen von diesem Alphabet syntaktische korrekte Terme sind.
    - ◆ Die korrekten Terme werden in der Logik auch wohldefinierte Formeln genannt (in Englisch: well-formed formulas, oder wff) sind.
- ❖ Das ganze muss ohne irgendeine Referenz zu einer Interpretation geschehen, sonst definieren Sie keine formale Sprache.

# Beispiele und Gegenbeispiele

- ❖ **Beispiel 1:** Gegeben sei das Alphabet  $T = \{\square, \diamond\}$ . Gegeben sei die Menge  $W$  aller Terme, die mit  $\square$  beginnen.
  - **Frage:** Ist  $W$  eine formale Sprache?
  - **Antwort:** Ja. (Welche Regeln müssen Sie erfinden?)
- ❖ **Beispiel 2:** Gegeben sei das Alphabet  $T = \{a, b, c, d, e, f, g\}$  und die Menge  $X$  aller endlichen Zeichenketten über  $T$ , die kein deutsches Wort sind.
  - **Frage:** Ist  $X$  eine formale Sprache?
  - **Antwort:** Nein, denn um zu entscheiden, ob ein gegebenes Eingabewort zur Sprache gehört, muss man wissen, ob es ein deutsches Wort ist oder nicht.
    - ◆ Wie weiss man beispielsweise, ob "Vorlesung" oder "sdfvolr" zu  $X$  gehören? Man muss wissen, dass "sdfvolr" keine Bedeutung in der deutschen Sprache hat.

# *Formale Logik*

- ❖ Formale Logik ist die Theorie der Formulierung von formalen Systemen und ihrer Interpretationen. Beweistheorie und Modelltheorie sind Teilgebiete der formalen Logik.
- ❖ Die Gegenstände der Formalen Logik sind "Formale Sprachen", "Formale Systeme", und "Interpretationen". Das Ziel der formalen Logik ist, möglichst viel Wissen über Wahrheiten über diese Objekte anzusammeln. Wichtige Begriffe aus der formalen Logik sind:
  - ◆ **Beweis:** Theoreme werden rein durch symbolische Manipulation gefunden.
  - ◆ **Vollständigkeit:** Alle Wahrheiten eines Systems sind beweisbar
  - ◆ **Entscheidbarkeit:** Für jede Aussage ist es möglich, zu beweisen, ob sie wahr oder falsch ist.
  - ◆ **Konsistenz:** Falsches darf nicht beweisbar sein

## *Historische Notizen*

- ❖ **1700:** Gottfried Wilhelm **Leibniz** träumt von einer allgemeinen Methode, in der Beweise in der Mathematik zur reinen Symbolmanipulation reduziert werden.
- ❖ **1854:** George **Boole** ("The Laws of Thought") schafft dafür die Grundlage mit der Booleschen Algebra.
- ❖ **1900:** David **Hilbert** schlug vor, die gesamte Mathematik als formales System aufzubauen, in dem alle Theoreme durch Textersetzung bewiesen werden können. Er meinte, die Mathematik muss *vollständig*, *konsistent* und *entscheidbar* sein:
  - Alle Wahrheiten können bewiesen werden, falsches ist nicht beweisbar, jede Aussage ist entweder wahr oder falsch.
- ❖ **1929:** Kurt **Gödel** zeigt, dass das nicht möglich ist:
  - Jedes formale System, das arithmetische Ausdrücke enthält, ist entweder unvollständig oder inkonsistent.
  - Es gibt arithmetische Aussagen, die nicht entscheidbar sind.

# *Zielsprache vs Metasprache*

- ❖ Um in der formalen Logik über formale Sprachen sprechen zu können, brauchen eine zweite Sprache, um diese Sprachen beschreiben zu können.
- ❖ Formale Sprachen heissen auch Zielsprachen.
- ❖ **Definition Metasprache:** Die Sprache, die wir benutzen, um eine Zielsprache zu beschreiben, ist die Metasprache.
- ❖ Wir benutzen Deutsch und einige spezielle Symbole (z.B.  $\mathbb{R}$ ,  $\mathbb{N}$ ,  $\mathbb{C}$ , ...) als unsere Metasprache.
  - Die Sprache der Booleschen Ausdrücke ist unsere Zielsprache.
  - Wenn wir Metaregeln definieren, z.B. Regeln, was ableitbar ist und was nicht, dann benutzen wir die Metasprache.

# *Wahrheitsfunktionen*

- ❖ **Definition Wahrheitswert:** Wir bezeichnen **L** und **O** als Wahrheitswerte.
- ❖ **Definition Wahrheitsfunktion (auch Boolesche Funktion genannt):** Eine Funktion, deren Argumente aus Wahrheitswerten bestehen, und deren Bildbereich die Menge aller Wahrheitswerte ist.

# Abbildungen zwischen Wahrheitswerten (Folie 05.19)

Diese Abbildungen heißen Wahrheitsfunktionen oder auch Boolesche Funktionen

$$\neg : \mathbb{B} \rightarrow \mathbb{B}$$

$$\neg 0 = 1$$

$$\neg 1 = 0$$

$$\vee : \mathbb{B} \times \mathbb{B} \rightarrow \mathbb{B}$$

$$0 \vee 0 = 0$$

$$0 \vee 1 = 1$$

$$1 \vee 0 = 1$$

$$1 \vee 1 = 1$$

$$\wedge : \mathbb{B} \times \mathbb{B} \rightarrow \mathbb{B}$$

$$0 \wedge 0 = 0$$

$$0 \wedge 1 = 0$$

$$1 \wedge 0 = 0$$

$$1 \wedge 1 = 1$$

$$\Rightarrow : \mathbb{B} \times \mathbb{B} \rightarrow \mathbb{B}$$

$$0 \Rightarrow 0 = 1$$

$$0 \Rightarrow 1 = 1$$

$$1 \Rightarrow 0 = 0$$

$$1 \Rightarrow 1 = 1$$

$$\Leftrightarrow : \mathbb{B} \times \mathbb{B} \rightarrow \mathbb{B}$$

$$0 \Leftrightarrow 0 = 1$$

$$0 \Leftrightarrow 1 = 0$$

$$1 \Leftrightarrow 0 = 0$$

$$1 \Leftrightarrow 1 = 1$$

## *Interpretation von Booleschen Termen (Folie 05.20)*

**Definition:** Eine Abbildung  $\beta: ID \rightarrow \mathbb{B}$ , die jedem Element aus  $ID$  einen Wahrheitswert zuordnet, heißt **Boolesche Belegung**.

❖ **Beispiel einer Belegung  $\beta$  für  $ID = \{a, b, c, x, y, z\}$ :**

- $\beta [a] = L$
- $\beta [b] = O$
- $\beta [c] = L$
- $\beta [x] = L$
- $\beta [y] = L$
- $\beta [z] = L$

❖ **Beispiel einer anderen Belegung  $\beta_1$  für  $ID = \{a, b, c, x, y, z\}$ :**

- $\beta_1 [a] = O$
- $\beta_1 [b] = O$
- $\beta_1 [c] = L$
- $\beta_1 [x] = O$
- $\beta_1 [y] = O$
- $\beta_1 [z] = O$

- Die Menge aller Booleschen Belegungen heißt  $ENV$ .  $\beta, \beta_1 \in ENV$ .
- Für eine gegebene Belegung  $\beta$  lässt sich nicht nur den Elementen aus  $ID$ , sondern jedem Booleschen Term ein Wahrheitswert zuordnen.
- Wir definieren dafür die Interpretation  $I_\beta$ .

## Interpretation $I_f$ von Booleschen Termen (Folie 05.21)

- ❖  $I_\beta[\mathbf{True}] = L$                        $I_\beta[\mathbf{False}] = 0$
- ❖  $I_\beta[x] = \beta(x)$  d.h. für alle Elemente aus ID ist der Wahrheitswert der Wert der sich aus der Booleschen Belegung  $\beta$  ergibt.
  - Für unsere Beispiel Belegung  $\beta$ :  $I_\beta[a] = \beta[a] = L$
- ❖  $I_\beta[\neg t] = \boxed{\neg (I_\beta[t])}$ 
  - d.h. wir finden den Wahrheitswert von  $\neg t$ , indem wir den Wahrheitswert von Term  $t$  finden, und ihn dann negieren (mit der Wahrheitstafel für **Negation** auf Folie 19).
- ❖  $I_\beta[t1 \vee t2] = I_\beta[t1] \vee I_\beta[t2]$ 
  - d.h. wir finden den Wahrheitswert von  $t1 \vee t2$ , indem wir die Wahrheitswerte von Term  $t1$  und  $t2$  finden, und sie dann mit Hilfe der Wahrheitstafel für **Or** verknüpfen.
- ❖  $I_\beta[t1 \wedge t2] = I_\beta[t1] \wedge I_\beta[t2]$
- ❖  $I_\beta[t1 \Rightarrow t2] = I_\beta[t1] \Rightarrow I_\beta[t2]$
- ❖  $I_\beta[t1 \Leftrightarrow t2] = I_\beta[t1] \Leftrightarrow I_\beta[t2]$



# *Bemerkungen zu den binären Booleschen Funktionen*

- ❖ Insgesamt gibt es 16 verschiedene Möglichkeiten für die Verknüpfung von 2 booleschen Argumenten. Für viele gibt es natürlichsprachliche Interpretationen. Die wichtigsten sind:
- ❖ **Disjunktion  $\vee$** : Wir lesen  $x \vee y$  als "x oder y", denn es ist wahr, wenn x oder y wahr sind oder wenn beide wahr sind.
- ❖ **Konjunktion  $\wedge$** : Wir lesen  $x \wedge y$  als "x und y", denn es ist nur dann wahr, wenn sowohl x als auch y wahr sind.
- ❖ **Implikation  $\Rightarrow$**  : Wir lesen  $x \Rightarrow y$  als "wenn x dann y" oder "x impliziert y". Wichtig:  $x \Rightarrow y$  ist wahr, wenn x falsch ist. Dies ist eventuell schwierig zu verstehen:
  - "Wenn Andreas Harrer 3 m groß ist, dann kann Bernd Brügge an der Decke vom AudiMax spazieren". Wahr? Falsch?
  - Wahr, denn Herr Harrer ist nicht 3 Meter groß.
- ❖ **Äquivalenz  $\Leftrightarrow$**  : Wir lesen  $x \Leftrightarrow y$  als "x ist gleich y" oder "x ist äquivalent zu y".
- ❖ **Nand, Nor**: Lesen wir als "Not and" und "Not or". Werden im Übungsbetrieb behandelt.

# *Modellierung von Aussagen mit Booleschen Ausdrücken*

- ❖ Wir hatten eine Aussage definiert als eine Nachricht, die nur als wahr oder falsch interpretiert werden kann.
- ❖ Beispiel:
  - Henry VIII hatte einen Sohn und Cleopatra hatte zwei.
- ❖ Wir schauen jetzt mal, wie wir diese Aussage in einen booleschen Ausdruck verwandeln können.
- ❖ Eine triviale Methode ist es, einfach einen Identifikator (auch boolesche Variable genannt) zu kreieren, der diese Aussage darstellt.
  - $p$ : Henry VIII hatte einen Sohn und Cleopatra hatte zwei.
- ❖ Wir stellen aber fest, dass die Aussage in Wirklichkeit zwei Teilaussagen enthält:
  - $x$ : Henry VIII hatte einen Sohn
  - $y$ : Cleopatra hatte zwei Söhne
- ❖ Damit können wir das Beispiel präziser formulieren:  $x \wedge y$

# *Übersetzung von natürlichsprachlichen Aussagen in Boolesche Ausdrücke (Analyse)*

Um eine natürlichsprachliche Aussage  $p$  in einen booleschen Ausdruck zu übersetzen:

1. Finde alle Unteraussagen, und bezeichne sie durch boolesche Variable  $x, y, z, \dots$
2. Ersetze die Unteraussagen durch ihre korrespondierenden booleschen Variablen.
3. Ersetze folgende Worte durch boolesche Operatoren:

und	wird zu	$\wedge$
oder	wird zu	$\vee$
nicht	wird zu	$\neg$
wenn $p$ dann $q$	wird zu	$\Rightarrow$

4. Schritte 1, 2 und 3 ergeben einen booleschen Ausdruck.

## *Analyse Beispiele*

- ❖  $x$ : Henry VIII hatte einen Sohn     $y$ : Cleopatra hatte zwei Söhne
- ❖  $z$ : Ich esse meinen Hut                     $w$ : 1 ist eine Primzahl
- ❖ **Problemstellung: Henry VIII hatte einen Sohn oder ich esse meinen Hut**
- ❖ *Analyse:  $x \vee z$*
- ❖ **Problemstellung: Henry VIII hatte einen Sohn und 1 ist nicht eine Primzahl**
  - Hier müssen wir noch "1 ist nicht eine Primzahl" umformulieren in "Es ist nicht der Fall, dass 1 eine Primzahl ist". Dann können wir dafür  $\neg w$  schreiben und das ergibt:
- ❖ *Analyse:  $x \wedge \neg w$*
- ❖ **Problemstellung: Wenn 1 eine Primzahl ist und Cleopatra zwei Söhne hatte, dann esse ich meinen Hut**
- ❖ *Analyse:  $w \wedge y \Rightarrow z$*

# Übersetzung von "Oder"

- ❖ Das Wort "oder" wird in der deutschen Sprache in 2 Arten benutzt:
  - **Inklusives Oder:** Ich trage eine schwarze Jacke oder ein Mikrofon.
    - ◆ Das "oder" ist inklusiv, denn ich kann beides tragen
  - **Exklusives Oder:** Ich werde meine Ferien entweder in Tirol oder in Timbuktu verbringen.
    - ◆ Das "oder" ist exklusiv, denn ich kann nicht an beiden Plätzen gleichzeitig sein.
- ❖ Das inklusive Oder in "b oder c" wird als  $b \vee c$  übersetzt.
- ❖ Das exklusive Oder in "b oder c" wird als  $b \Leftrightarrow \neg c$  übersetzt.
  - Die Aussage ist wahr, wenn genau einer der Teilaussagen b oder c wahr ist, entweder b oder c.

# Übersetzung von "Wenn...dann"

- ❖ Aussagen der Form "Wenn b dann c" oder "Wenn b, c" werden gewöhnlich als  $b \Rightarrow c$  übersetzt.
- ❖ Beispiel: "*Wenn Du Deinen Spinat nicht isst, dann verhaue ich Dich.*"
  - Wir bekommen die Unteraussagen und boolesche Variablen:
    - ◆ **x**: Du isst Deinen Spinat.
    - ◆ **y**: Ich verhaue Dich.
- ❖ Die Übersetzung der Aussage ergibt dann  $\neg x \Rightarrow y$
- ❖ Die Implikation ist in den Problembeschreibungen oft versteckt.  
Beispiel:
  - "*Jeder Name im Studentenverzeichnis ist auch im TUM Telefonbuch*"ist eigentlich eine Implikation:
  - "*Wenn ein Name im Studentenverzeichnis erscheint, dann ist er auch im TUM Telefonbuch*"

# *Übersetzung von Äquivalenz*

- ❖ Die Heuristik "Wenn...dann..." Phrasen in Aussagen mit der Implikation zu übersetzen, ist nicht immer richtig.
- ❖ Einige "Wenn ...dann..." Phrasen sollten eigentlich eher als Äquivalenzen übersetzt werden. Beispiel
- ❖ Problembeschreibung: "Wenn 2 Seiten eines Dreiecks gleich sind, dann ist das Dreieck gleichschenkelig."
  - t: 2 Seiten eines Dreieckes sind gleich
  - s: Ein Dreieck ist gleichschenkelig
  
  - Übersetzung:  $t \Leftrightarrow s$

## *Ein komplizierteres Beispiel: Superman existiert nicht*

- ❖ Wenn Superman in der Lage und willig ist, Übel zu verhindern, dann würde er es tun.
  - ❖ Wenn Superman nicht in der Lage ist, Übel zu verhindern, dann wäre er impotent.
  - ❖ Wenn Superman nicht willig wäre, Übel zu verhindern, dann wäre er böartig.
  - ❖ Superman verhindert keine Übel.
  - ❖ Wenn Superman existiert, dann ist er weder impotent noch böartig.
  - ❖ Deswegen existiert Superman nicht.
- ❖ **a: Superman ist in der Lage, Übel zu verhindern.**
  - ❖ **w: Superman ist willig, Übel zu verhindern.**
  - ❖ **i: Superman ist impotent**
  - ❖ **m: Superman ist böartig**
  - ❖ **p: Superman verhindert Übel.**
  - ❖ **e: Superman existiert.**

# Übersetzung in einen Booleschen Ausdruck

**Satz 1:** Wenn Superman in der Lage und willig ist, Übel zu verhindern, dann würde er es tun.

**Satz 2:** Wenn Superman nicht in der Lage ist, Übel zu verhindern, dann wäre er impotent.

**Satz 3:** Wenn Superman nicht willig wäre, Übel zu verhindern, dann wäre er böseartig.

**Satz 4:** Superman verhindert keine Übel.

**Satz 5:** Wenn Superman existiert, dann ist er weder impotent noch böseartig.

**Satz 6:** Deswegen existiert Superman nicht.

**a:** Superman ist in der Lage, Übel zu verhindern.

**w:** Superman ist willig, Übel zu verhindern.

**i:** Superman ist impotent

**b:** Superman ist böseartig

**p:** Superman verhindert Übel.

**e:** Superman existiert.

Satz 1 ergibt:

$$A0: a \wedge w \Rightarrow p$$

Satz 2 und Satz 3 ergeben:

$$A1: (\neg a \Rightarrow i) \wedge (\neg w \Rightarrow b)$$

Satz 4 ergibt:  $A2: \neg p$

Satz 5 ergibt:

$$A3: e \Rightarrow \neg i \wedge \neg b$$

Satz 6 ergibt:

$$A0 \wedge A1 \wedge A2 \wedge A3 \Rightarrow \neg e$$

## *Boolescher Ausdruck für Superman*

❖ Teilausdrücke:

$$A0: a \wedge w \Rightarrow p$$

$$A1: (\neg a \Rightarrow i) \wedge (\neg w \Rightarrow b)$$

$$A2: \neg p$$

$$A3: e \Rightarrow \neg i \wedge \neg b$$

$$A4: A0 \wedge A1 \wedge A2 \wedge A3 \Rightarrow \neg e$$

❖  $(a \wedge w \Rightarrow p) \wedge ((\neg a \Rightarrow i) \wedge (\neg w \Rightarrow b)) \wedge (\neg p) \wedge (e \Rightarrow \neg i \wedge \neg b) \Rightarrow \neg e$

❖ Wenn ein Ausdruck für alle Booleschen Belegungen wahr ist, dann nennen wir ihn allgemeingültig. Ist **A4** allgemeingültig?

❖ Um das herauszufinden, müssen wir  $2^6 = 64$  Belegungen testen.

–Zeitaufwendig.

–Wir können aber auch versuchen, mit Hilfe von Booleschen Gesetzen den Ausdruck zu vereinfachen.

# *Semantische Äquivalenz Boolescher Terme*

## *(Folie 05.25)*

- ❖ **Wiederholung der Definition der semantischen Äquivalenz:**
- ❖ Zwei boolesche Terme  $t_1, t_2$  heissen **semantisch äquivalent**, falls sie für alle möglichen Belegungen  $\beta \in ENV$  bezüglich der früher definierten Interpretation  $I_\beta$  gleich sind:
  - ◆  $I_\beta[t_1] = I_\beta[t_2]$
- ❖ Da bei semantischer Äquivalenz die Gleichheit unabhängig von der Belegung ist, schreibt man auch
  - ◆  $I[t_1] = I[t_2]$
- ❖ Mit dem Begriff der semantischen Äquivalenz können wir die Begriffe Allgemeingültigkeit und Tautologie definieren.

# *Erfüllbarkeit, Allgemeingültigkeit, Tautologie*

- ❖ **Definition Erfüllbarkeit (satisfiability):** Ein boolescher Ausdruck  $P$  mit einer Belegung  $\beta$  ist **erfüllt** (satisfied), wenn sein Wert  $L$  unter dieser Belegung ist.
  - Ein Ausdruck ist nicht erfüllbar, wenn es keine Belegung gibt, für die der Ausdruck den Wert  $L$  hat.
  - Beispiel:  $y \wedge \neg y$  ist nicht erfüllbar.
- ❖ **Definition Allgemeingültigkeit (validity):** Ein boolescher Ausdruck  $P$  ist **allgemeingültig**, wenn er für alle Belegungen wahr ist, das heißt, alle Interpretationen ergeben den Wert  $L$ .
- ❖ **Definition Tautologie (tautology):** Ein allgemeingültiger boolescher Ausdruck heißt **Tautologie**.
  - Alle Boolesche Gesetze auf der nächsten Folie sind Tautologien.

# *Erfüllbarkeit, Allgemeingültigkeit, Tautologie*

❖ **Definition Erfüllbarkeit (satisfiability):** Ein boolescher Ausdruck  $P$  mit einer Belegung  $\beta$  ist **erfüllt** (satisfied), wenn sein Wert **L** unter dieser Belegung ist.

- Ein Ausdruck ist nicht erfüllbar, wenn es keine Belegung gibt, für die der Ausdruck den Wert **L** hat.
- Beispiel:  $y \wedge \neg y$  ist nicht erfüllbar.

$\wedge$ :		
x	y	$x \wedge y$
O	O	O
O	L	O
L	O	O
L	L	L

## 4. Schritt: Definition von Booleschen Gesetzen (Folie 05.26)

(1) $\text{True} \Leftrightarrow (x \vee \neg x)$	Gesetz für True
(2) $\text{True} \Leftrightarrow \neg \text{False}$	Gesetz für False
(3) $\neg \neg x \Leftrightarrow x$	Involutionsgesetz
(4) $x \wedge y \Leftrightarrow y \wedge x$	Kommutativgesetz
(5) $(x \wedge y) \wedge z \Leftrightarrow x \wedge (y \wedge z)$	Assoziationsgesetz
(6) $x \wedge x \Leftrightarrow x$	Idempotenzgesetz
(7) $x \vee (y \wedge \neg y) \Leftrightarrow x$	Neutralitätsgesetz
(8) $x \wedge (y \vee x) \Leftrightarrow x$	Absorptionsgesetz
(9) $x \wedge (y \vee z) \Leftrightarrow (x \wedge y) \vee (x \wedge z)$	Distributivgesetz
(10) $\neg(x \wedge y) \Leftrightarrow (\neg x) \vee (\neg y)$	DeMorgan's Gesetz

**Bemerkung:**  $a=b$  wird genauso evaluiert wie  $a \Leftrightarrow b$ ,  
solange a und b boolesche Ausdrücke sind (Gries, S. 29)

## *Definition Boolesche Algebra*

- ❖ **Definition:** Eine Struktur  $K = (\mathbb{B}, \Sigma, Q)$  mit
  - der Trägermenge  $\mathbb{B} = \{O, L\}$ ,
  - der Signatur  $\Sigma = \{\text{False}, \text{True}, \neg, \wedge, \vee, \Rightarrow, \Leftrightarrow\}$ ,
  - wobei jede Operation  $f \in \Sigma$  durch die Wahrheitstafeln auf Folie 17 definiert ist
  - und den Gesetzen  $Q$  auf Folie 34ist ein konkretes Exemplar einer Booleschen Algebra.

# *Wie finden wir semantische Äquivalenz?*

*(Folie 05.29)*

## ❖ **Mit Wertetafeln:**

- Durchprobieren aller Belegungen  $f$  für alle möglichen Werte der Booleschen Terme.
- Dies ist sehr zeitaufwendig bei komplizierten Termen
  - ◆ Bei  $n$  Identifikatoren müssen wir  $2^n$  Belegungen durchprobieren.

## ❖ **Durch Termvereinfachung:**

1. Wir *vereinfachen die Terme* durch Anwendung der Booleschen Gesetze:
  - ◆ Hierzu betrachten wir die Booleschen Gesetze als Regeln eines Textersetzungssystems.
2. Wir wenden die sogenannte *Substitutionsregel* an:
  - ◆ Diese Regel gestattet es, freie Identifikatoren durch Terme zu ersetzen.

# Semantische Äquivalenz mit Wertetafeln am Beispiel des Absorptionsgesetzes (Folie 05.30) 11/20/00

Wir müssen beweisen, dass  $x \wedge (x \vee y) = x$  semantisch äquivalent sind

<b>Wahrheitstabelle</b>	$\beta(x)$	$\beta(y)$	$I_\beta[x \vee y]$	$I_\beta[x \wedge (x \vee y)]$
	O	O	O	O
	O	L	L	O
	L	O	L	L
	L	L	L	L

Unter Weglassung der Belegung  $\beta$  und Interpretation  $I_\beta$ :

<b>Wahrheitstabelle</b>	x	y	$x \vee y$	$x \wedge (x \vee y)$
	O	O	O	O
	O	L	L	O
	L	O	L	L
	L	L	L	L

## *Noch eine Bemerkung zu Belegungen*

❖ **Gegeben sei die Boolesche Variable:**

– **a: Der Mond besteht aus grünem Käse**

❖ Ist **a** wahr oder falsch? Die Wahrheit von **a** hängt von der Anwendungsdomäne ab. 2 Beispiele:

**1. Navigationssystem für eine Mondfähre. Wir wissen, dass a falsch ist, d.h. es gibt nur eine Belegung:**

–  $\beta [a] = \mathbf{0}$

**2. Anwendungsdomäne: Videospiele für Kinder**

❖ **Mögliche Belegungen für  $ID = \{a\}$**

–  $\beta [a] = \mathbf{L}$

–  $\beta [a] = \mathbf{0}$

❖ Die erste Belegung wäre sinnvoll für Anfänger auf Spielniveau 1, die zweite Belegung für Experten auf Spielniveau 10.

## *Noch eine Bemerkung zu Belegungen*

### ❖ Gegeben sei die Boolesche Variable:

– **a: Der Mond besteht aus grünem Käse**

❖ Ist **a** wahr oder falsch? Die Wahrheit von **a** hängt von der Anwendungsdomäne ab. 2 Beispiele:

**1. Navigationssystem für eine Mondfähre. Wir sind auf dem Mond, d.h. es gibt nur eine Belegung:**

–  $\beta [a] = 0$

**2. Anwendungsdomäne: Videospiele für Kinder**

❖ **Mögliche Belegungen für  $ID = \{a\}$**

–  $\beta [a] = L$

–  $\beta [a] = 0$

❖ Die erste Belegung wäre sinnvoll für Anfänger auf Spielniveau 1, die zweite Belegung für Experten auf Spielniveau 10.

Wenn wir für bestimmte Aussagen die Belegung konstant halten können, reduziert sich der Rechenaufwand bei der Interpretation von komplizierten Booleschen Formeln.

# Äquivalenz und Implikation sind Abkürzungen (Folie 05.28)

- ❖ Die Operatoren  $\Leftrightarrow$  (Äquivalenz) und  $\Rightarrow$  (Implikation) sind selbst Abkürzungen. Wir definieren folgende Gleichheiten:

$$- t1 \Rightarrow t2 = \neg t1 \vee t2$$

$$- t1 \Leftrightarrow t2 = (t1 \wedge t2) \vee (\neg t1 \wedge \neg t2)$$

- ❖ Wir können noch einen Schritt weitergehen. Der Operator  $\wedge$  (And) kann vollständig mit den Operatoren  $\neg$  (Negation) und  $\vee$  (Or) geschrieben werden.

$$- t1 \wedge t2 = \neg(\neg t1 \vee \neg t2)$$

- ❖ Der Operator  $\vee$  (Or) kann vollständig mit den Operatoren  $\neg$  (Negation) und  $\wedge$  (And) geschrieben werden.

$$- t1 \vee t2 = \neg(\neg t1 \wedge \neg t2)$$

- ❖ Wir sagen auch,  $\vee$  und  $\wedge$  verhalten sich dual zueinander.
- ❖ Beweis von  $t1 \Rightarrow t2 = \neg t1 \vee t2$  und  $t1 \vee t2 = \neg(\neg t1 \wedge \neg t2)$  mit Wertetafeln auf den nächsten zwei Folien.

# Semantische Äquivalenz von $t1 \Rightarrow t2 = \neg t1 \vee t2$

Die Belegung  $\beta$  und Interpretation  $I_\beta$  sind nicht gezeigt

t1	t2	$t1 \Rightarrow t2$	$\neg t1$	$\neg t1 \vee t2$
O	O	L	L	L
O	L	L	L	L
L	O	O	O	O
L	L	L	O	L

# Semantische Äquivalenz von $t1 \vee t2 = \neg (\neg t1 \wedge \neg t2)$

Die Belegung  $\beta$  und Interpretation  $I_\beta$  sind wieder nicht gezeigt

t1	t2	$t1 \vee t2$	$\neg t1 \wedge \neg t2$	$\neg (\neg t1 \wedge \neg t2)$
O	O	O	L	O
O	L	L	O	L
L	O	L	O	L
L	L	L	O	L

# *Semantische Äquivalenz durch Vereinfachen von Booleschen Termen*

❖ Eingabe:  $y \wedge \neg y$

❖ Ableitung:

$$y \wedge \neg y \Rightarrow \neg y \wedge y \quad (\text{Kommutativgesetz})$$

$$\Rightarrow \neg y \wedge \neg \neg y \quad (\text{Involutionsgesetz})$$

$$\Rightarrow \neg (y \vee \neg y) \quad (\text{DeMorgan})$$

$$\Rightarrow \neg (\text{true}) \quad (\text{Gesetz für True})$$

$$\Rightarrow \text{false} \quad (\text{Gesetz für False})$$

❖ Also:  $y \wedge \neg y \stackrel{*}{\Rightarrow} \text{false}$ .

❖ Beweistheoretisch arbeiten wir syntaktisch, d.h. die Ableitung geschieht nur unter Anwendung von Booleschen Gesetzen.

❖ Modelltheoretisch können wir verschiedene Belegungen für  $y \wedge \neg y$  und **false** ausprobieren. Wir sehen dann, dass die Ableitung  $y \wedge \neg y \stackrel{*}{\Rightarrow} \text{false}$  bei jeder Belegung den Wert **0** ergibt, das heißt, die Aussage  $y \wedge \neg y$  ist nicht erfüllbar.

## *Maschinelles Beweisen (Folie 05.32)*

- ❖ Was wir eben gemacht haben, ist aus gegebenen Booleschen Gleichungen durch Anwendungen von Tautologien andere Boolesche Gleichungen zu gewinnen.
  - Dies ist die zentrale Idee hinter Theorembeweisern, d.h. dem maschinellen Erstellen von Beweisen.
- ❖ Rein syntaktisch haben wir gezeigt, dass wir  $y \wedge \neg y \stackrel{*}{\Rightarrow} \text{false}$  ableiten können.
- ❖ Durch Anwendung der Definition der Interpretation für Boolesche Terme (Folie 23) können wir dann zeigen, dass  $y \wedge \neg y$  nicht erfüllbar ist, das heißt, es gilt  $y \wedge \neg y = \text{false}$ .
- ❖ Im folgenden formalisieren wir das maschinelle Beweisen mit Metaregeln der Aussagenlogik, insbesondere mit der sogenannten Modus Ponens Regel und dem Gleichheitsgesetz.
- ❖ Die **Aussagenlogik** ist eine spezielle Form der Logik. Zunächst definieren wir deshalb den allgemeinen Begriff der Logik.

# *Die Logik als Formales System*

- ❖ **Definition Logik:** Eine Logik ist eine Menge von Regeln, definiert durch
  - eine Menge von Symbolen (Signatur)
  - eine Menge von Formeln, die aus diesen Symbolen konstruiert werden können (wohldefinierte Terme)
  - Eine Menge von ausgezeichneten Formeln (Axiome)
  - Eine Menge von Inferenzregeln
- ❖ Die Inferenzregeln sind die Metaregeln der Logik. Sie werden manchmal auch Ableitungsregeln genannt.
- ❖ Die Menge aller Formeln, die aus den Axiomen und Inferenzregeln erzeugbar sind, ist die Menge aller Theoreme.
- ❖ Die Menge aller Theoreme ist die formale Sprache der Logik.

# *Beweis und Theorem*

## ❖ **Definition Beweis:**

- Ein Beweis ist eine endliche Zeichenkette von wohldefinierten Formeln (syntaktisch korrekten Termen), wobei jede Formel entweder ein Axiom ist oder durch Anwendung einer Inferenzregel abgeleitet worden ist.

## ❖ **Definition Theorem:**

- Ein Theorem einer Logik ist entweder ein Axiom oder eine Formel, die mit den Inferenzregeln aus den Axiomen und/oder aus bereits bewiesenen Theoremen abgeleitet wird.

## *Beispiel einer Logik: Die PQ-L Logik*

- ❖ Zeichenvorrat T: {P, Q, -}
- ❖ Formeln: Alle Zeichenketten der Form aPbQc, wobei a, b und c endliche Folgen von 0 oder mehr '-' sind. Beispiel: .--P----Q--
- ❖ Axiome:
  - A0: -P-Q--
  - A1: --P-Q---
- ❖ Inferenzregel:
  - aPbQc, dPeQf
  - adPbeQcf

(Beispiel aus Hofstadter: Gödel, Escher und Bach, 1979)

## *Notation von Inferenzregeln (Ableitungsregeln)*

- ❖ Für die Ableitung von Formeln benutzen wir zwei Notationen: das Zeichen  $\vdash$  oder eine graphische Notation
- ❖ Wenn  $H = \{t_1, t_2, \dots, t_n\}$  eine Menge von Theoremen ist, und  $t$  aus  $H$  ableitbar ist, dann schreiben wir

$$H \vdash t$$

– oder

$$\{t_1, t_2, \dots, t_n\} \vdash t$$

– oder

$$\frac{t_1, t_2, \dots, t_n}{t}$$

# *Interpretation einer Logik*

- ❖ Die Sprache einer Logik wird *vollständig syntaktisch definiert*.
  - Die Formeln einer Logik bestimmen in einer Anwendungsdomäne, was wahre und falsche Aussagen sind.
- ❖ Eine Interpretation ist die Abbildung von der rein syntaktischen Repräsentation auf solche Information.
- ❖ Beispiel einer Interpretation für die PQ-L Logik:
  - **Additions-Gleichheit Interpretation:** Jede Formel  $aPbQc$  wird auf  $\#a + \#b = \#c$  abgebildet, wobei  $\#x \in \mathbb{N}$  die Zahl der Bindestriche ("-") im Wort  $x$  zählt.
  - Die Formel  $-PQ-$  bedeutet also  $1+0=1$ , was wahr ist in der Anwendungsdomäne der natürlichen Zahlen  $\mathbb{C}$ .
  - Die Formel  $-P--Q---$  bedeutet also  $1+2 = 3$  was wahr ist.
  - $-P-Q-$  wird als  $1+1=1$  interpretiert, ist also falsch.
  - Die Axiome  $A0: -P-Q--$  und  $A1: --P-Q---$  werden als  $1+1=2$  und  $2+1=3$  interpretiert. Sie sind wahr, weil sie Axiome sind

# *Verallgemeinerung der Definitionen Belegung und Interpretation von Termen*

- ❖ Bisher haben wir nur Boolesche Belegungen betrachtet.
- ❖ Gegeben sei eine Algebra  $A$  mit Elementaroperanden  $ID$  und Signatur  $\Sigma$  und der Trägermenge  $S$ .
- ❖ **Definition Boolesche Belegung:** Eine Abbildung  $\beta: ID \rightarrow \mathbb{B}$ , die jedem Element aus  $ID$  einen Wahrheitswert zuordnet.
- ❖ **Definition Belegung:** Eine Abbildung  $\beta: ID \rightarrow S$ , die jedem Element von  $ID$  einen Wert aus  $S$  zuordnet.
- ❖ **Definition Interpretation:** Für jede Belegung  $f$  ist die Interpretation  $I_\beta[t]$  eines Terms  $t$  mit freien Variablen (Identifikatoren) aus  $ID$ :
  - Für alle Elemente  $x \in ID$ :  $I_\beta [x] = \beta(x)$
  - Für alle  $f \in \Sigma$ :  $I_\beta [f(t_1, \dots, t_n)] = f(I_\beta [t_1], \dots, I_\beta [t_n])$
- ❖ Damit haben wir **Belegung** und **Interpretation** auf beliebige Trägermengen und beliebige Funktionen verallgemeinert.

# Interpretationen der PQ-L Logik

- ❖ Elementaroperanden  $ID = \{-\}$
- ❖  $\Sigma = \{P, Q\}$
- ❖ Trägermenge:  $\mathbb{N}$
- ❖ Axiome:  $\{A0: -P-Q--, A1: --P-Q---\}$
- ❖ **Belegung:**  $\beta(a) = \#x \in \mathbb{N}$ , wobei  $\#x$  die Anzahl der "-" in der Zeichensequenz  $a$  ist.
- ❖ **Additions-Gleichheit Interpretation:** Die Formel  $aPbQc$  wird als  $\#a + \#b = \#c$  interpretiert und ist wahr, wenn  $\#a + \#b = \#c$  in der Anwendungsdomäne der natürlichen Zahlen gilt.
  - Die Axiome  $A0: -P-Q--$  und  $A1: --P-Q---$  werden als  $1+1=2$  und  $2+1=3$  interpretiert und sind wahr, denn Axiome sind immer wahr.
  - Die Formel  $-PQ-$  bedeutet  $1+0=1$ , was wahr in der Anwendungsdomäne der natürlichen Zahlen  $\mathbb{C}$  ist.
  - Die Formel  $-P--Q---$  bedeutet  $1+2 = 3$ , was wahr ist.
  - $-P-Q-$  wird als  $1+1=1$  interpretiert, ist also falsch.

## *Eine zweite Interpretation für die PQ-L Logik*

- ❖ Weil eine Logik ein rein syntaktisch definiertes formales System ist, kann es mehr als eine Interpretation haben.
- ❖ Beispiel einer zweiten Interpretation für die PQ-L Logik:
- ❖ **Additions-Ungleichheit Interpretation:**
  - Eine Formel  $aPbQc$  ist wahr, wenn  $\#a + \#b \leq \#c$  in der Anwendungsdomäne der natürlichen Zahlen gilt, wobei  $\#x$  wieder die Anzahl der Bindestriche ("-") in  $x$  zählt.
- ❖ Die Interpretationen **Additions-Gleichheit** und **Additions-Ungleichheit** sind nicht identisch!
- ❖ Als Gegenbeispiel nehmen wir die Formel **-P-Q---**
  - Die Additions-Gleichheit Interpretation bildet **-P-Q---** auf falsch ab, denn  $1+1=3$  ist falsch.
  - Die Additions-Ungleichheit Interpretation bildet **-P-Q---** auf wahr ab, denn  $1+1 \leq 3$  ist wahr.

# *Korrektheit und Vollständigkeit einer Logik*

- ❖ Der Zusammenhang zwischen der rein syntaktischen Definition einer Logik und ihrer Interpretation ist sehr wichtig.
  - Eine Interpretation ist ein Modell für eine Logik, wenn jedes Theorem unter dieser Interpretation wahr ist.
- ❖ Beziehungen zwischen Logik und möglichen Interpretationen:
  - **Definition:** Eine Logik ist **korrekt** (sound) wenn jedes Theorem allgemeingültig (valid) ist.
  - **Definition:** Eine Logik ist **vollständig** (complete) wenn jede allgemeingültige Formel auch ein Theorem ist.
- ❖ Die *PQL-Logik* ist *korrekt* bezüglich der Additions-Ungleichheit Interpretation, aber *nicht vollständig*.
  - Gegenbeispiel: Die allgemeingültige Formel **-PQ-** ( $1+0=1$ ) ist kein Theorem, d.h. die Formel ist in der PQ-L Logik nicht ableitbar. Axiome:  $\{A0: \text{-P-Q--}, A1: \text{--P-Q---}\}$

# *Definition der Aussagenlogik*

- ❖ **Symbole:** Der Zeichenvorrat  $T$  der Booleschen Algebra
- ❖ **Formeln:** Alle syntaktisch korrekten Terme der Booleschen Algebra
- ❖ **Axiome:** Formeln der Booleschen Algebra (üblicherweise eine Teilmenge der Formeln von Folie 50)
- ❖ **Inferenzregeln:**
  - **1. Modus Ponens:** Wenn  $x$  wahr ist, und wenn  $x \Rightarrow y$  gilt, dann ist  $y$  wahr.
  - **2. Tertium non datur:** Entweder  $t$  oder  $\neg t$  ist wahr.
  - **3. Gleichheitsgesetz:** Wenn zwei Ausdrücke  $X$  und  $Y$  semantisch gleich sind, dann kann man  $Y$  aus  $X$  sowie  $X$  aus  $Y$  ableiten.
  - Beispiel fürs Gleichheitsgesetz:
    - ◆  $y \wedge \neg y = \text{false}$
    - ◆ Damit kann ich ableiten  $y \wedge \neg y \vdash \text{false}$  sowie  $\text{false} \vdash y \wedge \neg y$

## *Schreibweise von Modus Ponens*

- ❖ Die Modus Ponens Regel schreiben wir als:

$\{x, x \Rightarrow y\} \vdash y$  oder als

$$\frac{x, x \Rightarrow y}{y}$$

- ❖ Die Aussagen  $x$  und  $x \Rightarrow y$  bezeichnet man auch als die Prämissen,  $y$  nennt man die Konsequenz.
- ❖ Die Modus Ponens Regel formalisiert das Konzept des mathematischen Beweises.
  - Aus einer Menge von gültig angenommenen Aussagen, den Axiomen, werden weitere gültige Aussagen nach festgelegten Regeln abgeleitet.

## *Beispiel der Anwendung von Modus Ponens*

x: Der Mond ist aus grünem Käse

y: Ich esse einen Besen

- ❖  $x \Rightarrow y$ : Wenn der Mond aus grünem Käse ist, dann esse ich einen Besen
- ❖  $\{x, x \Rightarrow y\} \vdash y$
- ❖ Prämisse: x ist wahr. (Applikationsdomäne: Videospiele)
- ❖ Konsequenz: y ist wahr.

## *Eine wichtige Metaregel: Dualitätsprinzip (05.27)*

- ❖ Für Boolesche Gesetze gilt das Dualitätsprinzip:
  - Ein Boolesches Gesetz gilt auch wenn wir **True** mit **False**, und  $\wedge$  mit  $\vee$  vertauschen.
- ❖ Das heißt, immer wenn wir ein boolesches Gesetz anwenden können, können wir auch das duale Gesetz anwenden.

❖ Beispiele:

$$x \vee (y \wedge \neg y) = x \quad \text{(Neutralitätsgesetz)}$$

$$x \wedge (y \vee \neg y) = x \quad \text{(Anwendung des Dualitätsprinzips)}$$

$$\neg(x \wedge y) = (\neg x) \vee (\neg y) \quad \text{(De Morgan's Gesetz)}$$

$$\neg(x \vee y) = (\neg x) \wedge (\neg y) \quad \text{(Anwendung des Dualitätsprinzips)}$$

# *Zusammenfassung*

- ❖ **Boolesche Algebra:** Eine Menge von Symbolen, eine Menge von Booleschen Funktionen und Boolesche Gesetze
- ❖ **Belegung:** Interpretation von Aussagen
- ❖ **Definitionen:** Erfüllbarkeit, Allgemeingültigkeit, Tautologie
- ❖ **Ermittlung der Bedeutung von Aussagen:**
  - Vereinfachung durch Anwendung Boolescher Gesetze
  - Durch Wahrheitstabellen
- ❖ **Aussagenlogik:** Boolesche Algebra + Axiome + 3 Metaregeln:
  - Modus Ponens, Tertium Non Datur, Gleichheitsregel.
- ❖ **Beweis:** Eine Ableitung eines Theorems.
- ❖ **Modell (Definition der Logiker):** Eine Belegung einer Booleschen Formel, die wahr ergibt.
- ❖ **Formale Logik** besteht aus Beweistheorie und Modelltheorie
  - **Syntax:** Textuelle Manipulation von Zeichenketten
  - **Semantik:** Bedeutung von Aussage