

# Formale Systeme

*Vorlesung:* Winfried Kurth

*Übung:* Aleksi Tavkhelidze

Lehrstuhl Computergrafik und ökologische Informatik

Büsgenweg 4, Raum 0.116 bzw. 0.125 (1. Stock)

Tel. 39-9715

wk<at>informatik.uni-goettingen.de

<http://www.uni-goettingen.de/de/72781.html>

*Hausaufgaben zur Übung:* Bearbeitung der Übungsblätter ist freiwillig. Die Lösungen werden in der darauffolgenden Übung besprochen. Der Stoff der Übungen ist prüfungsrelevant.

**Hinweise zur Lehrveranstaltung und Web-Links:**

[http://www.uni-forst.gwdg.de/~wkurth/fs22\\_home.htm](http://www.uni-forst.gwdg.de/~wkurth/fs22_home.htm)

Quellenangaben unter

[http://www.uni-forst.gwdg.de/~wkurth/fs10\\_lit.htm](http://www.uni-forst.gwdg.de/~wkurth/fs10_lit.htm)

*Inhaltsübersicht:*

1. Logik
2. Relationen und Inferenz
3. Algebra und Begriffsverbände
4. Regelbasierte Systeme
5. Modelle für Nebenläufigkeit

*Prüfung:* Abschlussklausur (90 Min.)

# 1. Logik

Geschichte:

erste Ansätze bei Aristoteles, Leibniz

Gottlob Frege (Ende 19. Jh.)

um 1900 Grundlagenkrise der Mathematik, löste verstärktes Interesse an Logik aus

1920 David Hilbert: "Hilbertsches Programm" – Forderung, die Mathematik als widerspruchsfrei nachzuweisen

1930 Kurt Gödel: Unvollständigkeitssätze

1963 Paul Cohen: Unabhängigkeit des Auswahlaxioms und der Kontinuumshypothese

Zielsetzung der math. Logik: Untersuchung der Grundlagen der Mathematik

Voraussetzung: es gibt ein "mathematisches Universum" – *Hintergrundmathematik* (z.B. Mengen, Relationen, Funktionen, Beweismethoden...)

in der Logik zu entwerfende formale Mathematik: *Objektmathematik* – eine Imitation der Hintergrundmathematik, so formuliert, dass eine Maschine sie verstehen und mathematisch argumentieren kann.

Anwendungen: automatisches Beweisen von Theoremen, Wissensrepräsentation, Schlussfolgern auf Grundlage einer Wissensbasis, Auffinden von Gesetzmäßigkeiten in Daten, Auffinden von Widersprüchen, Modelle von Systemen

elementarste Variante der Logik:

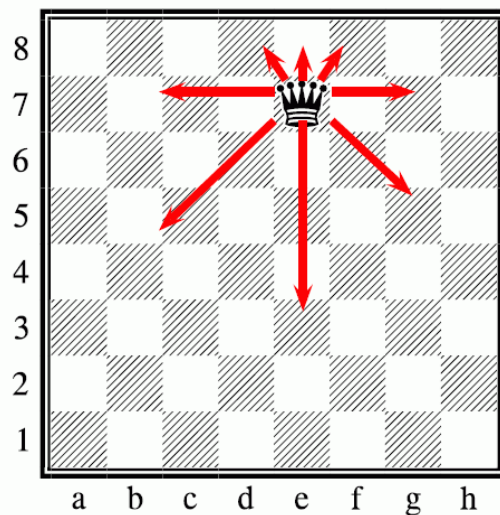
## Aussagenlogik

gleich ein Anwendungsbeispiel:

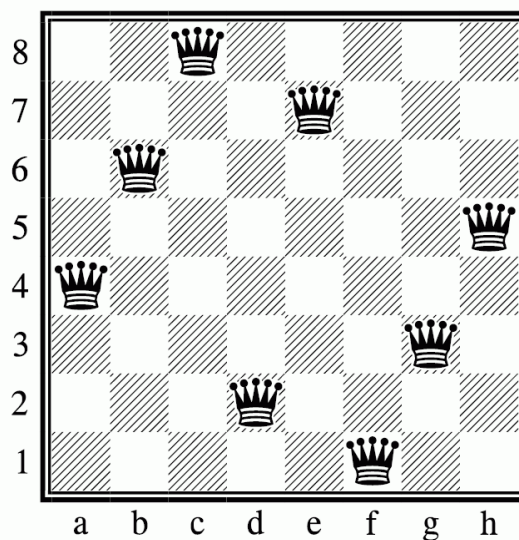
### *Das 8-Damen-Problem*

Man plaziere 8 Damen so auf einem Schachbrett, dass sie sich gegenseitig nicht bedrohen.

Bewegungs- (Bedrohungs-) möglichkeiten der Dame im Schach:



Eine Lösung des 8-Damen-Problems:



aussagenlogische Codierung des Problems:

boolesche Variable  $D_{i,j} = \text{wahr}$ , falls eine Dame auf Feld  $(i, j)$  steht, sonst *falsch* (benutze Zahlen statt Buchstaben fürs Schachbrett)

Bedingungen für Platzierung:

Dame auf Feld  $(1, 1)$  bedroht bestimmte Felder:

$$D_{1,1} \rightarrow \neg D_{1,2} \wedge \neg D_{1,3} \wedge \neg D_{1,4} \wedge \neg D_{1,5} \wedge \neg D_{1,6} \wedge \neg D_{1,7} \wedge \neg D_{1,8}$$

$$D_{1,1} \rightarrow \neg D_{2,1} \wedge \neg D_{3,1} \wedge \neg D_{4,1} \wedge \neg D_{5,1} \wedge \neg D_{6,1} \wedge \neg D_{7,1} \wedge \neg D_{8,1}$$

$$D_{1,1} \rightarrow \neg D_{2,2} \wedge \neg D_{3,3} \wedge \neg D_{4,4} \wedge \neg D_{5,5} \wedge \neg D_{6,6} \wedge \neg D_{7,7} \wedge \neg D_{8,8}$$

für Feld  $(5, 7)$ :

$$D_{5,7} \rightarrow \neg D_{5,8} \wedge \neg D_{5,6} \wedge \neg D_{5,5} \wedge \neg D_{5,4} \wedge \neg D_{5,3} \wedge \neg D_{5,2} \wedge \neg D_{5,1}$$

$$D_{5,7} \rightarrow \neg D_{4,7} \wedge \neg D_{3,7} \wedge \neg D_{2,7} \wedge \neg D_{1,7} \wedge \neg D_{6,7} \wedge \neg D_{7,7} \wedge \neg D_{8,7}$$

$$D_{5,7} \rightarrow \neg D_{6,8} \wedge \neg D_{4,6} \wedge \neg D_{3,5} \wedge \neg D_{2,4} \wedge \neg D_{1,3}$$

$$D_{5,7} \rightarrow \neg D_{4,8} \wedge \neg D_{6,6} \wedge \neg D_{7,5} \wedge \neg D_{8,4}$$

Für jedes Feld  $(i, j)$  sei  $FE_{i,j}$  die Konjunktion der Formeln, die die Einschränkungen für dieses Feld beschreiben.

Zusätzliche Bedingung: für genau 8 Felder  $(i, j)$  soll  $D_{i,j}$  wahr sein. Gleichwertig: für jedes  $k, k = 1, \dots, 8$ , soll  $R_k$  wahr sein:

$$D_{1,k} \vee D_{2,k} \vee D_{3,k} \vee D_{4,k} \vee D_{5,k} \vee D_{6,k} \vee D_{7,k} \vee D_{8,k}$$

Lösung des Problems:

finde Belegung aller 64 Variablen  $D_{i,j}$  so, dass alle Formeln  $FE_{i,j}$  und  $R_k$  wahr werden (*Erfüllbarkeitsproblem*).

# Vokabular der Aussagenlogik (Objektmathematik)

## Logische Zeichen

- 1** Symbol für den Wahrheitswert „wahr“
- 0** Symbol für den Wahrheitswert „falsch“
- $\neg$  Negationssymbol („nicht“)
- $\wedge$  Konjunktionssymbol („und“)
- $\vee$  Disjunktionssymbol („oder“)
- $\rightarrow$  Implikationssymbol („wenn ... dann“)
- $\leftrightarrow$  Symbol für beiderseitige Implikation („genau dann, wenn“)
- (,) die beiden Klammern

## Signatur

Eine (aussagenlogische) *Signatur* ist eine abzählbare Menge  $\Sigma$  von Symbolen, etwa

$$\Sigma = \{P_0, \dots, P_n\}$$

oder

$$\Sigma = \{P_0, P_1, \dots\}.$$

Die Elemente von  $\Sigma$  heißen auch *atomare Aussagen*, *Atome* oder *Aussagevariablen*.

## Aussagenlogische Formeln:

Zur Signatur  $\Sigma$  ist  $For0_\Sigma$ , die Menge der *Formeln über  $\Sigma$*  (oder der *Aussagen über  $\Sigma$* ) induktiv definiert durch

1.  $1, 0 \in For0_\Sigma, \Sigma \subseteq For0_\Sigma$

2. Mit  $A, B$  sind auch

$$\neg A, (A \wedge B), (A \vee B), (A \rightarrow B), (A \leftrightarrow B)$$

Elemente von  $For0_\Sigma$

Wir nennen die Sonderzeichen mit Ausnahme der Klammern auch die *logischen Operatoren*, unter ihnen  $\mathbf{1}$ ,  $\mathbf{0}$  die *logischen Konstanten*. Die Elemente von  $\Sigma$  heißen auch *atomare Aussagen*, *Atome* oder *Aussagevariablen*.

Wenn klar ist, um welches  $\Sigma$  es sich handelt, schreiben wir oft einfach  $For0$  statt  $For0_\Sigma$ .

## Beweisprinzip der *strukturellen Induktion*:

Gilt für eine Eigenschaft *Eig*

1.  $\mathbf{1}$ ,  $\mathbf{0}$  und jedes Atom  $p \in \Sigma$  haben die Eigenschaft *Eig*
2. Für beliebige  $A, B \in For0_\Sigma$ :
  - Hat  $A$  die Eigenschaft *Eig*, dann auch  $\neg A$ .
  - Haben  $A, B$  die Eigenschaft *Eig*, dann auch  $(A \wedge B)$ ,  $(A \vee B)$ ,  $(A \rightarrow B)$ ,  $(A \leftrightarrow B)$ .

dann gilt *Eig* für alle  $A \in For0_\Sigma$ .

Man nennt die *strukturelle Induktion* auch Induktion nach dem Aufbau der Formeln.

## Variante: Definition einer Funktion auf der Menge der Formeln

Ist eine Funktion  $f$

1. eindeutig definiert auf  $\mathbf{1}$ ,  $\mathbf{0}$  und den Atomen.
2. sind  $f(\neg A)$ ,  $f((A \wedge B))$ ,  $f((A \vee B))$ ,  $f((A \rightarrow B))$ ,  $f((A \leftrightarrow B))$  eindeutig definiert unter der Annahme, es seien  $f(A)$ ,  $f(B)$  schon definiert

dann ist  $f$  auf der gesamten Menge  $For0_\Sigma$  eindeutig definiert.

## Teilformeln:

Eine *Teilformel* einer Formel  $A$  ist ein Teilwort von  $A$ , welches Formel ist.

## Abkürzungen

1. Ganz außen stehende Klammern in einer Formel dürfen weggelassen werden.
2. Klammern dürfen weggelassen werden gemäß der Prioritätsregel:  $\wedge, \vee$  binden stärker als  $\rightarrow, \leftrightarrow$ . (Achtung: nach Def. 1.1 ist  $\neg A \wedge B$  immer als  $(\neg A \wedge B)$  und nicht als  $\neg(A \wedge B)$  zu lesen, da zu  $\neg$  keine Klammern gehören.)

Wegen der Assoziativgesetze für  $\wedge$  und  $\vee$  können wir für  $A \wedge (B \wedge C)$  oder  $(A \wedge B) \wedge C$  kurz  $A \wedge B \wedge C$  schreiben, entsprechend  $A_1 \wedge \dots \wedge A_n$ , entsprechend  $A_1 \vee \dots \vee A_n$ . Wir sprechen von Konjunktionen bzw. Disjunktionen.

# Semantik der Aussagenlogik

## Wahrheitswerte

Für alles Folgende seien zwei feste, ansonsten beliebige Objekte  $\mathbf{W}, \mathbf{F}$  ausgezeichnet, die wir die beiden *Wahrheitswerte* nennen. (Vorausgesetzt wird nur, daß beide voneinander verschieden sind.)

## Interpretation

Es sei  $\Sigma$  eine aussagenlogische Signatur. Eine **Interpretation** über  $\Sigma$  ist eine beliebige Abbildung

$$I : \Sigma \rightarrow \{\mathbf{W}, \mathbf{F}\}.$$

## Auswertung

Zu jedem  $I$  über  $\Sigma$  wird eine zugehörige **Auswertung** der Formeln über  $\Sigma$  definiert

$$val_I : For_0\Sigma \rightarrow \{W, F\}$$

mit:

$$val_I(\mathbf{1}) = W$$

$$val_I(\mathbf{0}) = F$$

$$val_I(P) = I(P) \quad \text{für jedes } P \in \Sigma$$

$$val_I(\neg A) = \begin{cases} F & \text{falls } val_I(A) = W \\ W & \text{falls } val_I(A) = F \end{cases}$$

$val_I$  auf  $(A \wedge B)$ ,  $(A \vee B)$ ,  $(A \rightarrow B)$ ,  $(A \leftrightarrow B)$  wird gemäß der folgenden Tabelle berechnet

$val_I(A), val_I(B)$	$val_I(C)$ für $C =$			
	$A \wedge B$	$A \vee B$	$A \rightarrow B$	$A \leftrightarrow B$
W,W	W	W	W	W
W,F	F	W	F	F
F,W	F	W	W	F
F,F	F	F	W	W

Welche der folgenden Aussagen sind stets wahr?

- 1  $(A \rightarrow B) \rightarrow (\neg B \rightarrow \neg A)$
- 2  $\neg(A \rightarrow B) \leftrightarrow (A \wedge \neg B)$
- 3  $\neg(A \vee B) \rightarrow (A \vee B)$
- 4  $(A \rightarrow B) \rightarrow (\neg A \rightarrow \neg B)$
- 5  $(\neg A \vee B) \vee (A \wedge \neg B)$



## Beispiel

Bei der Auswertung einer Formel werden der Übersichtlichkeit halber die Werte der Teilformeln mitnotiert.

$$\Sigma = \{P, Q, R\}$$

$$I : I(P) = \mathbf{W}, I(Q) = \mathbf{F}, I(R) = \mathbf{W}.$$

Wir berechnen  $val_I((P \wedge \neg R) \rightarrow \neg(R \vee Q))$

$P$	$Q$	$R$	$\neg R$	$(P \wedge \neg R)$	$(R \vee Q)$	$\neg(R \vee Q)$	$(P \wedge \neg R) \rightarrow \neg(R \vee Q)$
$\mathbf{W}$	$\mathbf{F}$	$\mathbf{W}$	$\mathbf{F}$	$\mathbf{F}$	$\mathbf{W}$	$\mathbf{F}$	$\mathbf{W}$

## Boolesche Funktionen

Eine *Boole'sche Funktion* ist eine Funktion von  $\{\mathbf{W}, \mathbf{F}\}^n$  nach  $\{\mathbf{W}, \mathbf{F}\}$ , für ein  $n \in \mathbb{N}$ . ( $\mathbb{N}$  ist die Menge der natürlichen Zahlen einschließlich 0.) Ist  $n$  die Anzahl der in einer Formel  $A$  auftretenden Atome, und legt man für diese eine *bestimmte Reihenfolge* fest – identifiziert sie also mit Argumentstellen –, so liefert die Wahrheitstafel von  $A$  eine Boole'sche Funktion  $\{\mathbf{W}, \mathbf{F}\}^n \rightarrow \{\mathbf{W}, \mathbf{F}\}$ . Es ist bekannt (leichte *Übung*), daß sich umgekehrt auch *jede* Boole'sche Funktion als Wahrheitstafel einer Formel in dieser Weise erhalten läßt.

## Modell, Allgemeingültigkeit, Erfüllbarkeit

- Ein **Modell** einer Formel  $A \in For_0_\Sigma$  ist eine Interpretation  $I$  über  $\Sigma$  mit  $val_I(A) = \mathbf{W}$ .
- Zu einer Formelmengemenge  $M \subseteq For_0_\Sigma$  ist ein Modell von  $M$  eine Interpretation  $I$ , welche Modell von jedem  $A \in M$  ist.
- $A \in For_0_\Sigma$  heißt **allgemeingültig**  
gdw  
 $val_I(A) = \mathbf{W}$  für jede Interpretation  $I$  über  $\Sigma$ .
- $A \in For_0_\Sigma$  heißt **erfüllbar**  
gdw  
es gibt eine Interpretation  $I$  über  $\Sigma$  mit  $val_I(A) = \mathbf{W}$ .

Es gilt

$A$  erfüllbar  $\Leftrightarrow \neg A$  nicht allgemeingültig,

$A$  allgemeingültig  $\Leftrightarrow \neg A$  nicht erfüllbar.

Man nennt die allgemeingültigen Formeln auch *Tautologien*. (Erst in der Prädikatenlogik werden beide Begriffe differieren.)

## Beispiele allgemeingültiger Formeln

$A \rightarrow A$	Selbstimplikation
$\neg A \vee A$	Tertium non datur
$A \rightarrow (B \rightarrow A)$	Abschwächung
$\mathbf{0} \rightarrow A$	Ex falso quodlibet
$A \wedge A \leftrightarrow A$	Idempotenz
$A \wedge (A \vee B) \leftrightarrow A$	Absorption
$A \wedge (B \vee C) \leftrightarrow (A \wedge B) \vee (A \wedge C)$	Distributivität
$A \vee (B \wedge C) \leftrightarrow (A \vee B) \wedge (A \vee C)$	Distributivität
$(A \rightarrow B) \leftrightarrow (\neg B \rightarrow \neg A)$	Kontraposition
$(A \rightarrow (B \rightarrow C)) \leftrightarrow$ $((A \rightarrow B) \rightarrow (A \rightarrow C))$	Verteilen
$\neg(A \vee B) \leftrightarrow \neg A \wedge \neg B$	De Morgan
$\neg(A \wedge B) \leftrightarrow \neg A \vee \neg B$	De Morgan

## Semantischer Folgerungsbegriff:

$\Sigma$  sei eine Signatur,  $M \subseteq For_{0\Sigma}$ ,  $A, B \in For_{0\Sigma}$ .

■  $M \models A$       lies: **aus  $M$  folgt  $A$**

gdw

Jedes Modell von  $M$  ist auch Modell von  $A$ .

■  $A, B \in For_{0\Sigma}$  heißen **logisch äquivalent**

gdw

$A \models_{\Sigma} B$  und  $B \models_{\Sigma} A$

## Einfache Sätze:

- $A$  erfüllbar gdw  $\neg A$  nicht allgemeingültig
- $\models A$  gdw  $A$  ist allgemeingültig
- $\models \neg A$  gdw  $A$  ist unerfüllbar
- $A \models B$  gdw  $\models A \rightarrow B$
- $M \cup \{A\} \models B$  gdw  $M \models A \rightarrow B$
- $A, B$  sind logisch äquivalent gdw  $A \leftrightarrow B$  ist allgemeingültig

## zum letztgenannten Satz:

Zwei Formeln  $A, B$  sind logisch äquivalent genau dann, wenn  $A \leftrightarrow B$  eine Tautologie ist.

**Beweis**  $A, B$  logisch äquivalent

$\Leftrightarrow \text{val}_I(A) = \text{val}_I(B)$  für alle Interpretationen  $I$  (über  $\Sigma$ )  
(d. h.  $A$  und  $B$  haben dieselben Modelle)

$\Leftrightarrow \models_{\Sigma} A \leftrightarrow B$

$\Leftrightarrow A \leftrightarrow B$  ist allgemeingültig.

## Satz:

Logische Äquivalenz ist bezüglich der aussagenlogischen Operatoren eine Kongruenzrelation auf  $For0_{\Sigma}$ . Insbesondere gilt für beliebige  $A \in For0_{\Sigma}$

$A$  allgemeingültig  $\Leftrightarrow A$  logisch äquivalent zu  $\mathbf{1}$

$A$  unerfüllbar  $\Leftrightarrow A$  logisch äquivalent zu  $\mathbf{0}$ .

## Def. "Interpolante":

Seien  $A, B$  aussagenlogische Formeln, so daß  $A \rightarrow B$  eine Tautologie ist. Eine Formel  $C$  heißt eine *Interpolante* von  $A \rightarrow B$ , falls

1.  $A \rightarrow C$  und  $C \rightarrow B$  Tautologien sind und
2. in  $C$  nur solche aussagenlogischen Atome  $P \in \Sigma$  vorkommen, die sowohl in  $A$  als auch in  $B$  vorkommen.  
An eventuelle Vorkommen von  $\mathbf{1}$  und  $\mathbf{0}$  in  $C$  werden keinerlei Einschränkungen gemacht.

Satz (*Craigsches Interpolationslemma*):

Es seien  $A$  und  $B$  zwei aussagenlogische Formeln und  $A \rightarrow B$  sei eine Tautologie. Dann existiert zu  $A$  und  $B$  eine Interpolante.

(Beweis siehe Schmitt 2008.)

## Disjunktive und konjunktive Normalform

Definitionen:

- Ein **Literal** ist ein Atom oder ein negiertes Atom
- Eine Formel ist in **disjunktiver Normalform** (DNF), wenn sie Disjunktion von Konjunktionen von Literalen ist.
- Eine Formel ist in **konjunktiver Normalform** (KNF), wenn sie Konjunktion von Disjunktionen von Literalen ist.

Sätze:

- 1 Zu jeder aussagenlogischen Formel  $A$  gibt es eine logisch äquivalente in disjunktiver Normalform und ebenso eine logisch äquivalente in konjunktiver Normalform.
- 2 Die Algorithmen zur Herstellung beider Normalformen ergeben sich unmittelbar aus elementaren Tautologien.
- 3 Ist die Wahrheitstafel einer Formel gegeben, so lassen sich disjunktive und konjunktive Normalform aus dieser „direkt“ ablesen.
- 4 Disjunktive und konjunktive Normalform einer Formel sind nicht eindeutig.

## Beispiel zur exponentiellen Länge einer KNF:

Um zu prüfen, ob

$$A_n = (\neg P_{1,1} \vee \neg P_{1,2}) \wedge \dots \wedge (\neg P_{n,1} \vee \neg P_{n,2})$$

eine Tautologie ist, wird die Unerfüllbarkeit von

$$\neg A_n = (P_{1,1} \wedge P_{1,2}) \vee \dots \vee (P_{n,1} \wedge P_{n,2})$$

geprüft. Die konjunktive Normalform von  $\neg A_n$  ist:

$$\bigwedge \{P_{1,f(1)} \vee \dots \vee P_{n,f(n)} \mid f : 1, \dots, n \rightarrow 1, 2\}.$$

Für  $n = 3$  ist das:

$$\begin{aligned} & (P_{1,1} \vee P_{2,1} \vee P_{3,1}) \wedge (P_{1,1} \vee P_{2,1} \vee P_{3,2}) \wedge \\ & (P_{1,1} \vee P_{2,2} \vee P_{3,1}) \wedge (P_{1,1} \vee P_{2,2} \vee P_{3,2}) \wedge \\ & (P_{1,2} \vee P_{2,1} \vee P_{3,1}) \wedge (P_{1,2} \vee P_{2,1} \vee P_{3,2}) \wedge \\ & (P_{1,2} \vee P_{2,2} \vee P_{3,1}) \wedge (P_{1,2} \vee P_{2,2} \vee P_{3,2}) \end{aligned}$$

In  $\neg A_n$  treten  $2 * n$  Literale auf, in der KNF  $n * 2^n$ .

Jedoch: mit Einführung von Hilfsatomen in die Formel lässt sich immer eine äquivalente "kurze KNF" (kknf) konstruieren.

*Zu jeder aussagenlogischen Formel A mit n Literalvorkommen gibt es eine konjunktive Normalform  $A_{kknf}$ , so dass*

- *A ist erfüllbar gdw  $A_{kknf}$  erfüllbar ist,*
- *$A_{kknf}$  enthält höchstens  $c * n$  Literalvorkommen für eine von n unabhängige Konstante c,*

$A_{kknf}$  kann effektiv aus A in linearer Zeit konstruiert werden. (siehe Schmitt 2008, S. 30ff.)

## Shannonsche Normalform

eine graphbasierte Normalform, die auf Shannon 1938, Church 1956 und Bryant 1986 zurückgeht

### Shannon-Formeln:

*Shannon Formeln* sind aussagenlogische Formeln, die aufgebaut sind aus

- dem dreistelligen Operator  $sh$
- den Konstanten 0 und 1
- Aussagevariablen  $P_1, \dots, P_n, \dots$

Der Wahrheitswerteverlauf von  $sh$  wird gegeben durch

$$sh(P_1, P_2, P_3) = \begin{cases} P_2 & \text{falls } P_1 = 0 \\ P_3 & \text{falls } P_1 = 1 \end{cases}$$

oder in Tabellenform:

$P_1$	1	1	1	1	0	0	0	0
$P_2$	1	1	0	0	1	1	0	0
$P_3$	1	0	1	0	1	0	1	0
$sh(P_1, P_2, P_3)$	1	0	1	0	1	1	0	0

### Eigenschaften des $sh$ -Operators:

- $sh(P_1, P_2, P_3) \leftrightarrow (\neg P_1 \wedge P_2) \vee (P_1 \wedge P_3)$
- $sh(0, P_2, P_3) \leftrightarrow P_2$
- $sh(1, P_2, P_3) \leftrightarrow P_3$
- $sh(P, 0, 1) \leftrightarrow P$
- $sh(P, 1, 0) \leftrightarrow \neg P$
- $sh(P_1, P_2, P_2) \leftrightarrow P_2$
- $sh(sh(P_1, P_2, P_3), P_4, P_5) \leftrightarrow sh(P_1, sh(P_2, P_4, P_5), sh(P_3, P_4, P_5))$
- $A \leftrightarrow sh(P, A_{P=0}, A_{P=1})$
- $\neg sh(A, B, C) \leftrightarrow sh(A, \neg B, \neg C)$

Wir fixieren eine Ordnung auf der Menge der Aussagevariablen, etwa die durch die Ordnung der Indizes gegebene.

## Definition

- ① Die Konstanten 0, 1 sind normierte *sh*-Formeln.
- ②  $sh(P_i, A, B)$  ist eine normierte *sh*-Formel wenn
  - $A$  und  $B$  normierte *sh*-Formeln sind und
  - für jede in  $A$  oder  $B$  vorkommende Aussagenvariable  $P_j$  gilt  $j > i$ .

## Theorem

*Zu jeder aussagenlogischen Formel  $A$  gibt es eine äquivalente normierte *sh*-Formel  $B$ .*

**Beweis:** Induktion nach der Anzahl  $n$  der in  $A$  vorkommenden Aussagevariablen.

Für  $n = 0$  kann  $A$  logisch äquivalent auf eine der Konstanten 0 oder 1 reduziert werden. Konstanten sind normierte *sh*-Formeln.

Im Induktionsschritt wählen wir die in  $A$  vorkommende Aussagevariable  $P_i$  mit dem kleinsten Index. Mit  $A_0$  bezeichnen wir die Formel, die aus  $A$  entsteht, indem jedes Vorkommen von  $P_i$  durch 0 ersetzt wird. Entsprechend wird  $A_1$  gebildet. Nach Induktionsvoraussetzung gibt es normierte *sh*-Formeln  $B_0, B_1$ , die logisch äquivalent (siehe Def. 2.16) sind zu  $A_0, A_1$ . Offensichtlich ist  $A$  äquivalent zu  $sh(P_i, B_0, B_1)$ , und  $sh(P_i, B_0, B_1)$  ist eine normierte *sh*-Formel.



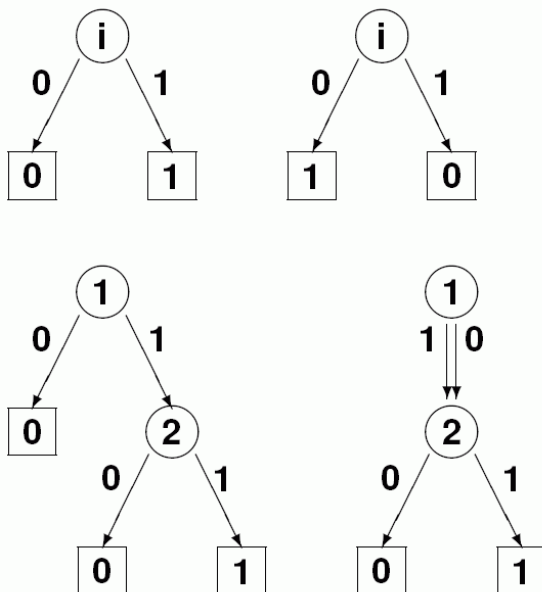
## Definition "Shannon-Graph":

Ein *sh-Graph* ist ein gerichteter, binärer, zusammenhängender Graph.

- Jedem nichtterminalen Knoten  $v$  ist eine natürliche Zahl  $index(v)$  zugeordnet.
- Von jedem nichtterminalen Knoten  $v$  gehen zwei Kanten aus. Eine davon ist mit 0, die andere mit 1 gekennzeichnet.
- Jedem terminalen Knoten  $v$  ist eine der Zahlen 0 oder 1 zugeordnet, bezeichnet mit  $wert(v)$ .
- Ist der nichtterminale Knoten  $w$  ein unmittelbarer Nachfolger von  $v$ , dann gilt  $index(v) < index(w)$ .
- Es gibt genau einen Wurzelknoten.

beachte: jeder Shannon-Graph ist azyklisch.

Beispiele für Shannon-Graphen:



Es gibt eine offensichtliche Entsprechung zwischen Shannon-Graphen und normierten Shannon-Formeln:

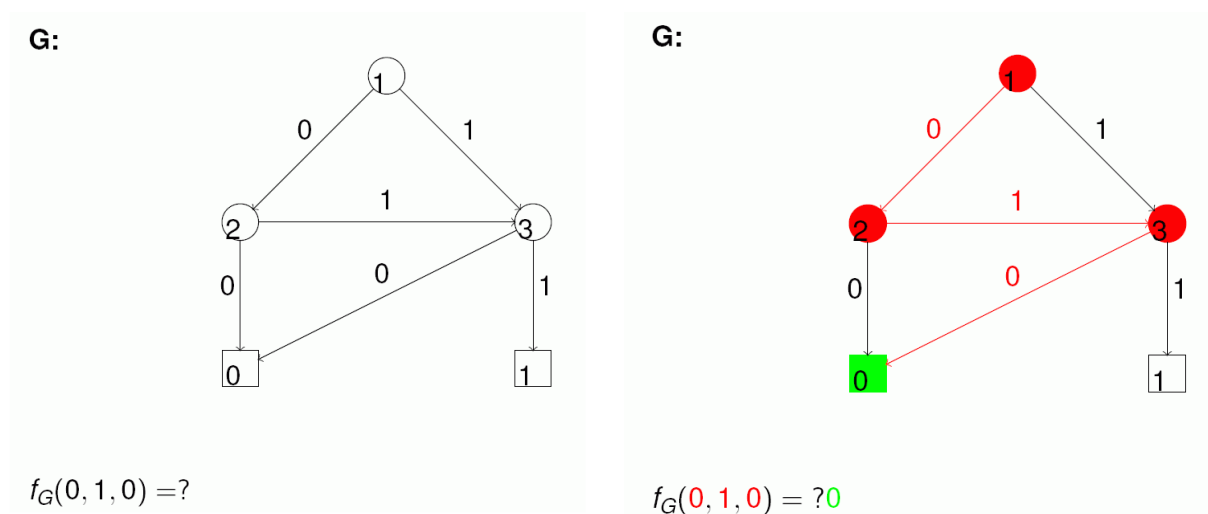
$n$ -te Variable entspricht den Knoten mit Index  $n$ .



## Shannon-Graphen und boolesche Funktionen:

- Jedem *sh*-Graphen  $G$  kann man eine  $m$ -stellige Boolesche Funktion  $f_G$  zuordnen, wobei  $m$  die Anzahl der in  $G$  vorkommenden verschiedenen Indizes  $i_1, \dots, i_m$  ist.
- Wir fassen  $f_G$  als eine Funktion mit den Eingabevariablen  $P_{i_1}, \dots, P_{i_m}$  auf und bestimmen den Funktionswert  $f_G(P_{i_1}, \dots, P_{i_m})$ , indem wir an der Wurzel von  $G$  beginnend einen Pfad durch  $G$  wählen. Am Knoten  $v$  folgen wir der Kante 0, wenn die Eingabevariable  $P_{index(v)}$  den Wert 0 hat, sonst der Kante 1.
- Der Wert des terminalen Knotens ist dann der gesuchte Funktionswert.

### Beispiel:



### Umgekehrt gilt auch:

Zu jeder Boole'schen Funktion  $f : \{0, 1\}^m \rightarrow \{0, 1\}$  und jeder aufsteigenden Folge  $i_1 < \dots < i_m$  von Indizes gibt es einen *sh*-Graphen  $G$  mit

$$f_G = f.$$

Beweis: s. Schmitt (2008), S. 37f.

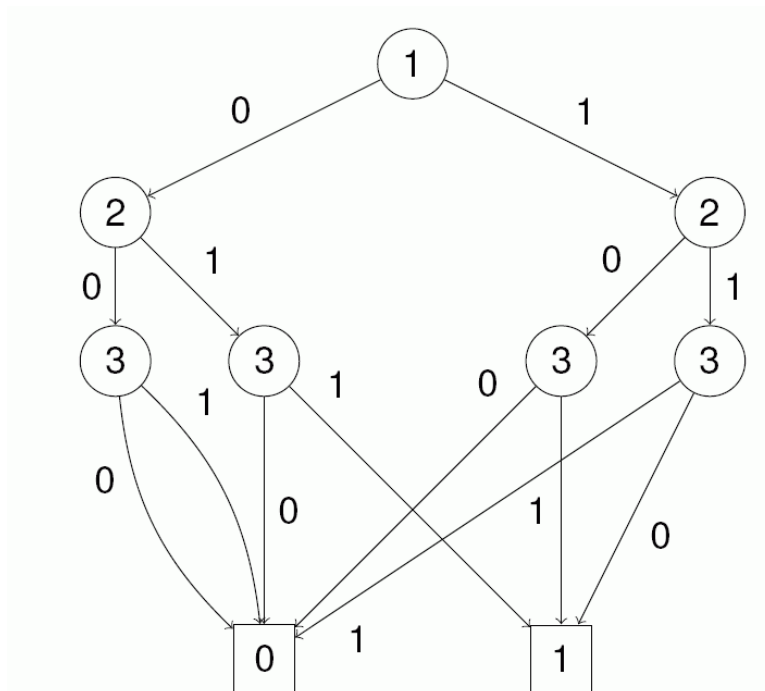
## Reduzierte Shannon-Graphen

Ein *sh*-Graph heißt *reduziert*, wenn

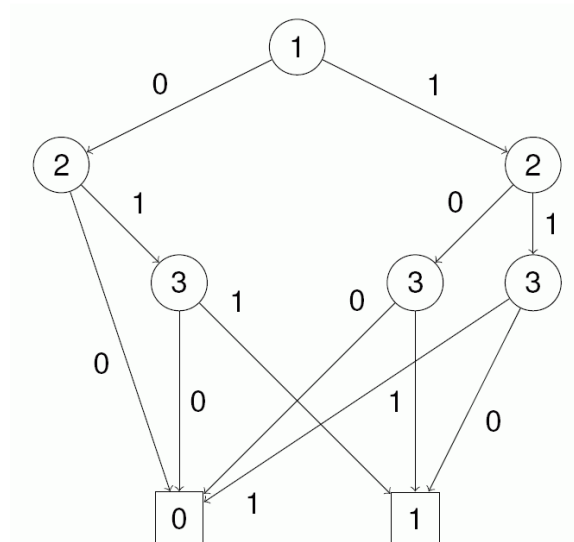
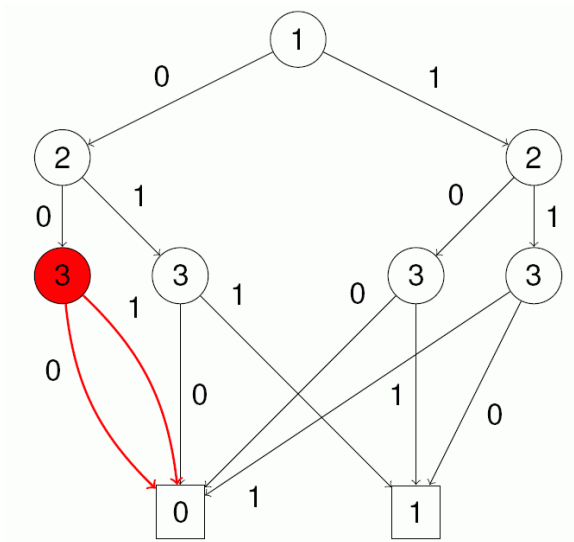
- 1 es keine zwei Knoten  $v$  und  $w$  ( $v \neq w$ ) gibt, so daß der in  $v$  verwurzelte Teilgraph  $G_v$  mit dem in  $w$  verwurzelten Teilgraph  $G_w$  isomorph ist.
- 2 es keinen Knoten  $v$  gibt, so dass die beiden von  $v$  ausgehenden Kanten zum selben Nachfolgerknoten führen.

Ein reduzierter Shannongraph heißt auch *ordered binary decision diagram* (OBDD).

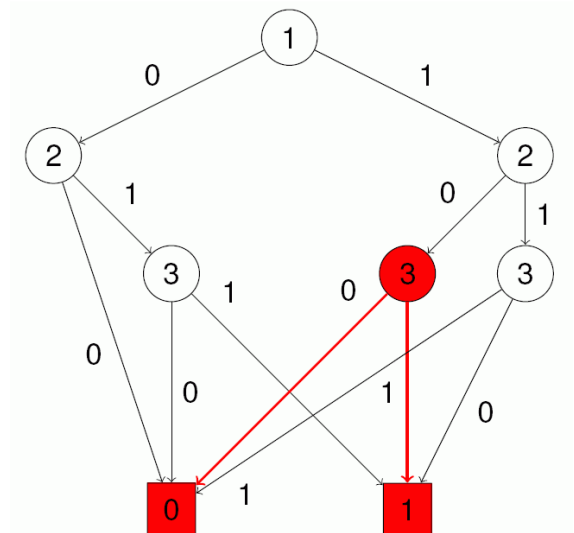
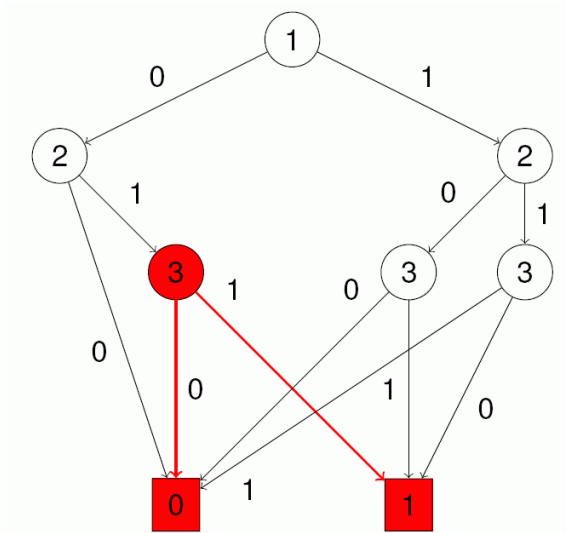
Beispiel für eine Reduktion eines Shannon-Graphen:



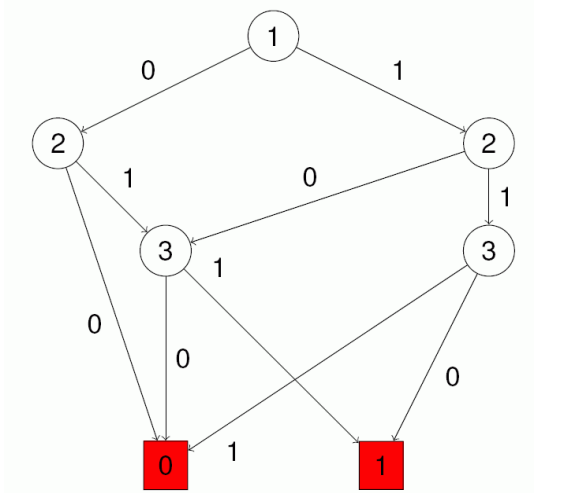
zuerst: Entfernen doppelter Kanten (mitsamt dem Knoten, von dem diese ausgehen)



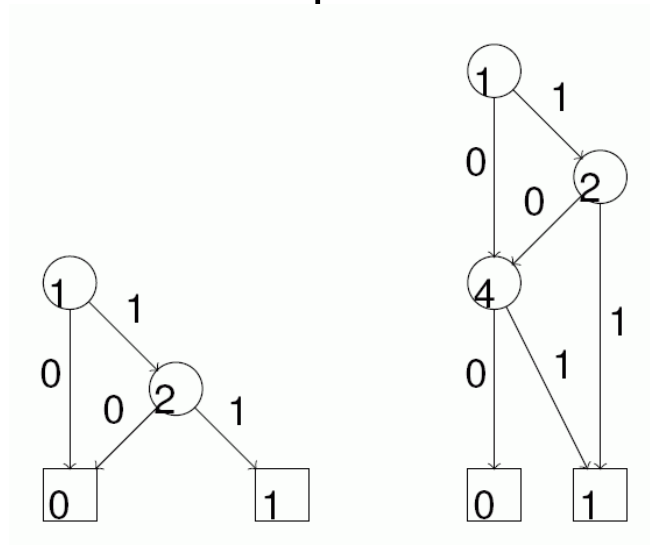
Auffinden isomorpher Teilgraphen:



einer davon darf entfernt werden (einlaufende Kante neu verknüpfen):



## Weitere Beispiele



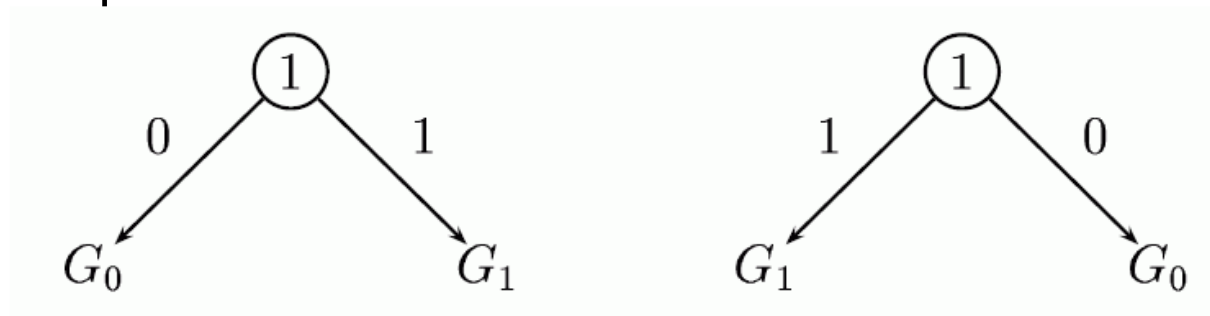
## Isomorphie von Shannon-Graphen

Seien zwei *sh*-Graphen  $H, G$  gegeben. Ihre Knotenmengen seien  $V_1, V_2$ .

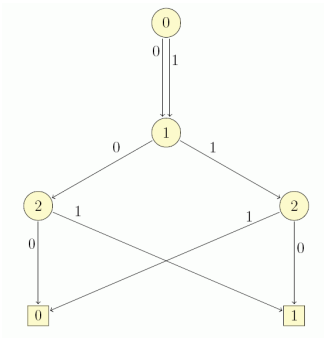
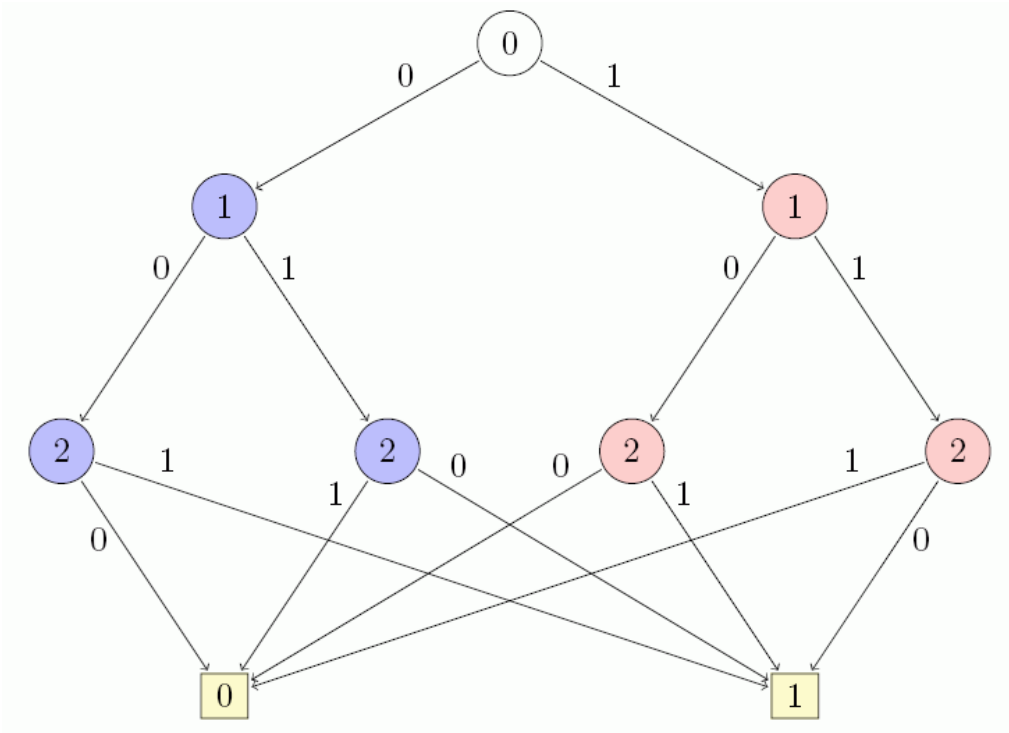
$H, G$  heißen zueinander *isomorph* ( $H \cong G$ ) genau dann, wenn es eine bijektive Abbildung  $\pi$  von  $V_1$  nach  $V_2$  gibt mit:

- ①  $index(k) = index(\pi(k))$  für jeden Nichtterminalknoten  $k \in V_1$
- ②  $wert(k) = wert(\pi(k))$  für jeden Terminalknoten  $k \in V_1$
- ③ Für jeden Nichtterminalknoten  $k \in V_1$ , dessen 0-Kante/1-Kante zu dem Knoten  $k_0/k_1$  führt, gilt: die 0-Kante von  $\pi(k)$  führt zu  $\pi(k_0)$ , die 1-Kante zu  $\pi(k_1)$ .

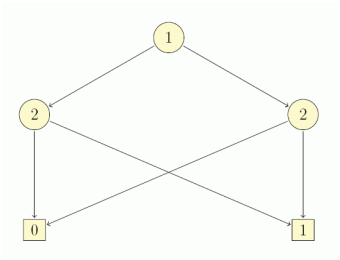
einfachstes Beispiel zweier isomorpher Shannon-Graphen:



komplexeres Beispiel: die beiden Teilgraphen unterhalb des Wurzelknotens sind isomorph:



(somit Reduktion möglich auf und dann auf



## Ein Kriterium für Reduziertheit:

*Sei  $G$  ein Shannongraph, so daß für jedes Paar von Knoten  $v, w$  gilt*

*wenn die 1-Nachfolger von  $v$  und  $w$  gleich sind und  
die 0-Nachfolger von  $v$  und  $w$  gleich sind  
dann  $v = w$*

*Dann erfüllt  $G$  die Bedingung (1) aus der Definition reduzierter Shannongraphen, d.h. für jedes Paar  $x, y$  von Knoten gilt*

*wenn  $G_x$  isomorph zu  $G_y$  ist  
dann  $x = y$*

Beweis: s. Schmitt (2008), S. 41.

## Eindeutigkeitssatz für reduzierte Shannon-Graphen:

*Sind  $G, H$  reduzierte sh-Graphen zu  $\Sigma = \{P_1, \dots, P_n\}$ , dann gilt*

$$f_G = f_H \Leftrightarrow G \cong H.$$

*(Zu jeder Booleschen Funktion  $f$  gibt es bis auf Isomorphie genau einen reduzierten sh-Graphen  $H$  mit  $f = f_H$ ).*

Beweis: s. Schmitt (2008), S. 41ff.

Wie die "einfache" KNN kann auch der Shannon-Graph in ungünstigen Fällen "groß" werden:

[BDD für Multiplikationen]

- $X$  enthalte  $2k$  Variablen  $\{x_0, \dots, x_{k-1}, y_0, \dots, y_{k-1}\}$
- $x = x_0 \dots x_{k-1}$  und  $y = y_0 \dots y_{k-1}$  bezeichnen  $k$ -stellige Binärzahlen.
- für  $0 \leq i < 2k$  bezeichne  $Mult_i$  die boolesche Funktion, die das  $i$ -te Bit des Produktes von  $x$  mit  $y$  beschreibt.

### Theorem

*Für jede Ordnung  $<$  der Variablen in  $X$  gibt es einen Index  $0 \leq i < 2k$ , so dass der BDD  $B_{Mult_i, <}$  mindestens  $2^{k/8}$  Knoten besitzt.*

# Das SAT-Problem

## SAT

Instanz: Eine aussagenlogische Formel  $F \in \text{For}_0$

Frage: Ist  $F$  erfüllbar?

Gibt es eine Interpretation  $I$  mit  $\text{val}_I(F) = \mathbf{1}$ ?

SAT ist ein *NP-vollständiges* Problem:

Gäbe es einen (deterministischen) polynomialen Entscheidungsalgorithmus für die Erfüllbarkeit, dann wäre  $NP = P$ , d. h. jedes nichtdeterministisch-polynomiale Entscheidungsproblem auch deterministisch-polynomial.

vereinfachte Varianten des Problems:

Das Erfüllbarkeitsproblem für Formeln  $A$

- in KNF ist NP-vollständig
- in 3-KNF ist NP-vollständig
- in 2-KNF ist polynomial entscheidbar
- in DNF ist polynomiell entscheidbar ( $O(n \log n)$  oder besser)
- $k$ -KNF Formeln sind Konjunktionen von Disjunktionen mit höchstens  $k$  Literalen.

## Horn-Formeln

Def.:

Eine *Horn-Formel* ist eine aussagenlogische Formel in konjunktiver Normalform, in der jede Disjunktion höchstens ein positives Literal enthält. Eine solche Disjunktion heißt eine *Horn-Klausel*.

Alternative Schreibweise:

$\neg B_1 \vee \dots \vee \neg B_m \vee A$	$B_1 \wedge \dots \wedge B_m \rightarrow A$
$\neg B_1 \vee \dots \vee \neg B_m$	$B_1 \wedge \dots \wedge B_m \rightarrow \mathbf{0}$
$A$	$A$

Dabei heißt  $B_1 \wedge \dots \wedge B_m$  der *Rumpf* und  $A$  der *Kopf* der Horn-Klausel  $B_1 \wedge \dots \wedge B_m \rightarrow A$ .

Beispiel einer Horn-Formel:

$$\begin{aligned} & \neg P \\ \wedge & (Q \vee \neg R \vee \neg S) \\ \wedge & (\neg Q \vee \neg S) \\ \wedge & R \wedge S \wedge (\neg Q \vee P) \end{aligned}$$

Alternative Schreibweise

$$\begin{aligned} & (P \rightarrow \mathbf{0}) \\ \wedge & (R \wedge S \rightarrow Q) \\ \wedge & (Q \wedge S \rightarrow \mathbf{0}) \\ \wedge & R \wedge S \wedge (Q \rightarrow P) \end{aligned}$$

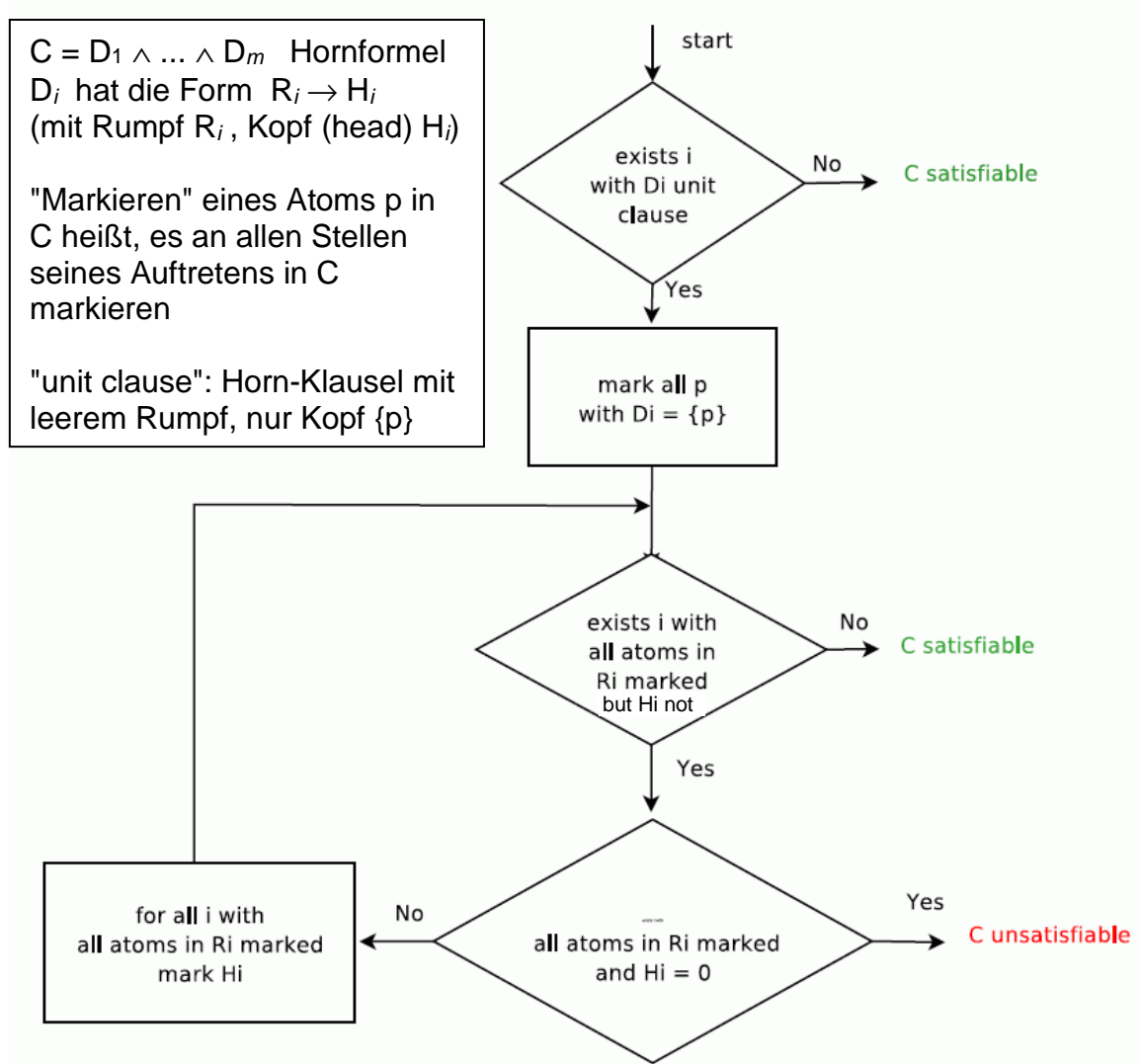


## Erfüllbarkeitsproblem für Horn-Formeln:

### Satz:

*Für Horn-Formeln ist die Erfüllbarkeit in quadratischer Zeit entscheidbar.*

### Beweis durch Angabe eines Entscheidungsalgorithmus:



Korrektheitsbeweis für diesen Algorithmus siehe Schmitt (2008), S. 49ff.

Ausschnitte entnommen aus  
Beckert (2010) und Schmitt (2008)