Autor: Xuân Baldauf <xuan--2003.07.24--logik--sem02@studium.baldauf.org>

Datum: 30.10.2002

Mengentheoretisch-algebraische Grundlagen der Informatik

gelesen von Gerhard Brewka

Organisatorisches

- Link (Übungen?): http://www.informatik.uni-leipzig.de/~rhartwig/
- Übungsaufgabenabgabe: vor der Vorlesung.
- 60% der Übungsaufgaben müssen richtig sein, um einen Übungsschein zu erhalten.
- Übungsschein ist Vorraussetzung zur Klausur.
- Klausur ist Bestandteil der Vordiplomprüfung.

2. Mengenbegriff

- Intuitiver Mengenbegriff:
  - Cantor (1845..1918): Eine Menge ist eine Zusammenfassung bestimmter, wohlunterschiedener Objekte unserer Anschauung oder unseres Denkens zu einem Ganzen; Diese Objekte heißen „Elemente der Menge“
- Notation:
  - $a \in M$ (a ist ein Element der Menge M)
  - $a \notin M$ (a ist kein Element der Menge M)
  - Endliche Mengen lassen sich einfach aufzählen.
    - Beispiel: $\{2,8,13\}$ (Die Menge, die 2, 8 und 13 enthält)
    - Beispiel: $\{\text{Hans}, \text{Peter}\}$ (Die Menge, die Hans und Peter enthält)
  - Unendliche Mengen können durch Angabe der Eigenschaften ihrer Elemente beschrieben werden.
    - Beispiel: $M = \{x \mid x \text{ ist eine Primzahl}\}$ (Die Menge aller x, wobei x eine Primzahl ist)
  - Mengen können auch induktiv beschrieben werden:
    - Beispiel:
      - M_1 ist die kleinste Menge, für die gilt:
        - Basisfälle:
          - „aa“ $\in M_1$ und
          - „bb“ $\in M_1$
        - abgeleitete Fälle: Wenn $w \in M_1$, dann gilt
          - $'a' + w + 'a' \in M_1$ und
          - $'b' + w + 'b' \in M_1$

Induktive Definitionen erlauben Induktionsbeweise:
Eine Eigenschaft E gilt für alle Elemente einer induktiv definierten Menge, wenn

1. E für alle Basisfälle gilt und
2. E für alle abgeleiteten Fälle gilt, vorausgesetzt E gilt für die Elemente, aus denen sie abgeleitet werden.

Beispiel

Zeige, dass jedes Element von M_1 eine gerade Anzahl von 'a's und 'b's besitzt.
1. „aa“ besitzt 2 'a's und 0 'b's. „bb“ besitzt 0 'a's und '2' b's.
2. Angenommen w besitzt m 'a's und n 'b's (m,n sind gerade), dann
   1. besitzt 'a'+w+'a' 2+m 'a's und n'b's;
   2. besitzt 'b'+w+'b m 'a's und 2+n 'b's.

Definition: gleich (Extensionalitätsprinzip)

Mengen heißen genau dann **gleich**, wenn sie die selben Elemente besitzen.
$\forall(M \in Mengen): \forall(N \in Mengen): \big((M = N) \Leftrightarrow \forall(x): ((x \in M) \Leftrightarrow (x \in N))\big)$

Beispiel

$\{a, b\} = \{b, a\}$

Bemerkung

Vorsicht: Es sind nicht beliebige Mengenkonstruktionen durch Aussagen möglich.

Beispiel: Russelsche Antonomie

„Es gibt sicher Mengen, die sich nicht als Element enthalten.“
Gibt es die Menge all dieser Mengen?
Die Menge aller Mengen, die sich nicht selbst enthält: $\{M \mid M \notin M\}$
Angenommen, es gäbe diese Menge. Enthält sie sich selbst?
- Angenommen, sie enthält sich nicht. Dann enthält sie sich.
- Angenommen, sie enthält sich doch. Dann enthält sie sich nicht.

Dies ist ein Widerspruch.

Grundbegriffe der Mengenlehre

Definition: istTeilmengeVon

Eine Menge M_0 **ist Teilmenge von** einer Menge M_1 (geschrieben als $M_0 \subseteq M_1$) genau dann, wenn für alle Elemente von M_0 gilt, dass sie auch Elemente von M_1 sind.
$\forall(M_0 \in Mengen): \forall(M_1 \in Mengen): \Big((M_0 \subseteq M_1) \Leftrightarrow \big(\forall(x): ((x \in M_0) \Rightarrow (x \in M_1))\big)\Big)$

Definition: istEchteTeilmengeVon

- **echte Teilmenge**: $(N \subset M) \Leftrightarrow ((N \subseteq M) \wedge \neg(M \subseteq N))$
- **leere Menge**: $\emptyset$
- **Potenzmenge**: $P(M) = \{X \mid X \subseteq M\}$
  - Beispiel: P({a,b,c})={∅,{a},{b},{c},{a,b},{a,c},{b,c},{a,b,c}}

Eigenschaften der Relation „Teilmenge“

Für alle Mengen M, N, P gilt:

- **Reflexivität**:
  - M ist immer Teilmenge von sich selbst.
  - $M \subseteq M$
- **Transitivität**:
  - Ist M Teilmenge von N und N Teilmenge von P, so ist M auch Teilmenge von P.
  - $\big((M \subseteq N) \wedge (N \subseteq P)\big) \Rightarrow (M \subseteq P)$
- **Antisymmetrie**:
  - Ist N eine Teilmenge von M und M eine Teilmenge von N, so ist $M = N$.
  - $\big((N \subseteq M) \wedge (M \subseteq N)\big) \Rightarrow (M = N)$

Mengenalgebra

- **Schnittmenge** (AND):
  - $M \cap N = \{x \mid (x \in M) \wedge (x \in N)\}$ „Schnitt von M und N“
- **Vereinigung** (OR):
  - $M \cup N = \{x \mid (x \in M) \vee (x \in N)\}$ „Vereinigung von M und N“
- **Differenz** (AND NOT):
  - $M \setminus N = \{x \mid (x \in M) \wedge (x \notin N)\}$ „Differenz von M und N“
- Falls $M \cap N = \emptyset$, dann nennt man M und N **disjunkt**.

Eigenschaften von Mengen-Operationen

- **Kommutativität**:
  - $M \cap N = N \cap M$
  - $M \cup N = N \cup M$
- **Assoziativität**:
  - $M \cap (N \cap P) = (M \cap N) \cap P$
  - $M \cup (N \cup P) = (M \cup N) \cup P$
- **Idempotenz**:
  - $M \cap M = M$
  - $M \cup M = M$
- **Distributivität**:
  - $M \cap (N \cup P) = (M \cap N) \cup (M \cap P)$
  - $M \cup (N \cap P) = (M \cup N) \cap (M \cup P)$

Definition: Komplement

Ist M Teilmenge einer Grundmenge G, so heißt $C_G(M) = G \setminus M$ **Komplement** von M bezüglich der Grundmenge G.

Häufig ist die Grundmenge, auf die sich das Komplement bezieht, klar, und wird deswegen weg gelassen. In diesem Fall kann man statt $C(M)$ auch $\overline{M}$ schreiben.
Es gilt also: $C_G(M) = G \setminus M = C(M) = \overline{M}$.

Satz: De-Morgansche Gesetze

- $\overline{M \cap N} = \overline{M} \cup \overline{N}$
- $\overline{M \cup N} = \overline{M} \cap \overline{N}$

Satz

$$\forall\left(M_0 \in Mengen\right): \forall\left(M_1 \in Mengen\right): \left(\left(M_0 \setminus M_1\right) = \left(M_0 \cap \overline{M_1}\right)\right)$$

Satz

$$\forall\left(M_0 \in Mengen\right): \forall\left(M_1 \in Mengen\right): \forall\left(M_2 \in Mengen\right): ($$
$$M_0 \setminus \left(M_1 \cup M_2\right) = \left(M_0 \setminus M_1\right) \cap \left(M_0 \setminus M_2\right)$$
$$)$$

Beweis

Seien $M_0 \in Mengen$, $M_1 \in Mengen$ $M_2 \in Mengen$. Dann gilt:

$$\begin{aligned} M_0 \setminus \left(M_1 \cup M_2\right) &= M_0 \cap \overline{\left(M_1 \cup M_2\right)} \\ &= M_0 \cap \left(\overline{M_1} \cap \overline{M_2}\right) \\ &= \left(M_0 \cap \overline{M_1}\right) \cap \left(M_0 \cap \overline{M_2}\right) \\ &= \left(M_0 \setminus M_1\right) \cap \left(M_0 \setminus M_2\right) \end{aligned}$$

Definition: Mengensystem

Ein **Mengensystem** ist eine Menge von Mengen.

- Schnitt eines Mengensystems M: $\bigcap M = \left\{x \middle| \forall (Y \in M): \left(x \in Y\right)\right\}$
- Vereinigung eines Mengensystems M: $\bigcup M = \left\{x \middle| \exists (Y \in M): \left(x \in Y\right)\right\}$

Mengensysteme sind oft durch **Indexmengen** charakterisiert:
Gegeben sei eine Menge I von Indizies, $M = \left\{M_i \middle| i \in I\right\}$. Dann schreiben wir:

- statt $\bigcup M$: $\bigcup_{i \in I} M_i$
- statt $\bigcap M$: $\bigcap_{i \in I} M_i$

Definition: kartesisches Produkt, Kreuzprodukt

Seien M und N Mengen. Das **kartesische Produkt** (auch **Kreuzprodukt** genannt) von M und N ist die Menge
$M \times N = \left\{(a, b) \middle| (a \in M) \wedge (b \in N)\right\}$

Falls $M_1 = M_2 = ... = M_n = M$, dann heißt $M_1 \times M_2 \times ... \times M_n = M^n$.

Satz

$$\forall\left(M_0 \in Mengen\right): \forall\left(M_1 \in Mengen\right): \left(\left|M \times N\right| = \left|M\right| \cdot \left|N\right|\right)$$

Definition: *n*-Tupel

Tupel sind Anordnungen [oder Listen]
- 2-Tupel: **Paar**
- 3-Tupel: **Tripel**
- 4-Tupel: **Quadrupel**
- 5-Tupel: **Quintupel**

3. Aussagenlogik

Einführung

Was ist Logik? Logik ist das Wissenschaftsgebiet, das Folgerungsbeziehungen untersucht.
Was ist mathematische oder formale Logik? Logik mit mathematischen Mitteln.

Beispiele

Beispiel 1

- Aussage: Wenn es regnet, ist es nass.
- Aussage: Es regnet.
- Schlussfolgerung: Es ist nass.
- Diese Aussage ist gültig.

Beispiel 2

- Aussage: Es ist Sonntag oder es ist Montag.
- Aussage: Es ist nicht Sonntag.
- Schlussfolgerung: Es ist Montag.
- Diese Aussage ist gültig.

Beispiel 3

- Aussage: Wenn es warm ist, ist es nicht kalt.
- Schlussfolgerung: Wenn es kalt ist, ist es nicht warm.
- Diese Aussage soll gültig sein.

Beispiel 4

- Aussage: Wenn Sommer ist, ist es warm.
- Aussage: Es ist warm.
- Schlussfolgerung: Es ist Sommer.
- Diese Aussage ist nicht gültig.

Folgerungsbeziehungen ergeben sich nicht daraus, über welche konkreten oder virtuellen Objekte (Sonne, Regen, Sommer, Montag) geredet wird, sondern nur aus Aussagen über diesen Objekte.

- Es gibt mehrere Logiken
  - nicht nur Aussagenlogik
  - sondern auch Prädikatenlogik
  - andere Logiken

Syntax und Semantik

- Syntax ist der Aufbau der formalen Sprache.
- Semantik ist die Bedeutung der in der Sprache vorkommenden Ausdrücke.

Verknüpfungen (Junktoren)

- „und“ $\wedge$
- „oder“ $\vee$
- „wenn“, ... „dann“ $\Rightarrow$
- „nicht“ $\neg$ oder $\sim$
- „genau dann, wenn“ $\Leftrightarrow$

Aussagen werden durch **Formeln** repräsentiert.

Definition: Formel

Sei V eine Menge von aussagenlogischen Variablen. Die Menge der Formeln über V ist die kleinste Menge, für die gilt:
1. Jedes Element von V ist eine Formel.
2. Wenn P und Q Formeln sind, dann sind auch
   $\neg P$
   $(P \wedge Q)$
   $(P \vee Q)$
   $(P \Rightarrow Q)$
   $(P \Leftrightarrow Q)$
   Formeln

Beispiel

- Sei: $V = \{V_1, V_1\}$
- Formeln sind dann z.B.
  - V_1
  - $\neg V_1$
  - $\neg(V_1 \wedge V_2)$
  - $\neg V_1 \vee (V_2 \wedge V_1)$

Bindung der Junktoren

Bindungsprioritäten von am stärksten bindend zu schwächer bindend:
- $\neg$
- $\wedge$
- $\vee$
- $\Rightarrow$
- $\Leftrightarrow$

Elemente von V nennt man auch atomare Formeln.

Wann sind Formeln wahr oder falsch? Diese Frage kann man nur beantworten, wenn man
1. die Bedeutung der Junktoren kennt und
2. die Wahrheitswerte der Variablen kennt.

„Wahr“ wird durch „1“ repräsentiert. (Manchmal auch „W“ (wahr) oder „W“ (true) oder „L“)
„Falsch“ wird durch „0“ repräsentiert. (Manchmal auch „F“ (falsch) oder „F“ (false) oder „0“)

Bedeutung der Junktoren

p	**q**	**¬p**	**p∨q**	**p∧q**	**p⇔q**	**p⇒q**
0	0	1	0	0	1	1
0	1	1	1	0	0	1
1	0	0	1	0	0	0
1	1	0	1	1	1	1

Satz

$(p \Rightarrow q) \Leftrightarrow (\neg p \vee q)$

Interpretation

Wie kennt man die Wahrheitswerte der Variablen? Man braucht eine Interpretation, die jeder Variablen einen Wahrheitswert zuordnet.

Definition: Interpretation

Eine **Interpretation** im engeren Sinne I ist eine Abbildung von Variablen auf Wahrheitswerte $I : V \to \{0,1\}$.

Definition: Interpretation

Eine **Interpretation** im weiteren Sinne I ist eine Abbildung von Formeln auf Wahrheitswerte $I : Formeln \to \{0,1\}$ anhand einer gegebenen Interpretation im engeren Sinne I. Dabei ist:

- $\forall (A \in Formeln) : (I(\neg A) = \neg I(A))$
- $\forall (A \in Formeln) : \forall (B \in Formeln) : (I(A \wedge B) = I(A) \wedge I(B))$
- $\forall (A \in Formeln) : \forall (B \in Formeln) : (I(A \vee B) = I(A) \vee I(B))$
- $\forall (A \in Formeln) : \forall (B \in Formeln) : (I(A \Rightarrow B) = I(A) \Rightarrow I(B))$
- $\forall (A \in Formeln) : \forall (B \in Formeln) : (I(A \Leftrightarrow B) = I(A) \Leftrightarrow I(B))$
- $I(0) = 0$
- $I(1) = 1$
- $I(\neg 0) = 1$
- $I(\neg 1) = 0$
- $I(0 \wedge 0) = 0$
- $I(0 \wedge 1) = 0$
- $I(1 \wedge 0) = 0$
- $I(1 \wedge 1) = 1$
- $I(0 \vee 0) = 0$
- $I(0 \vee 1) = 1$
- $I(1 \vee 0) = 1$
- $I(1 \vee 1) = 1$
- $I(0 \Rightarrow 0) = 1$
- $I(0 \Rightarrow 1) = 1$
- $I(1 \Rightarrow 0) = 0$

- $I(1 \Rightarrow 1) = 1$
- $I(0 \Leftrightarrow 0) = 1$
- $I(0 \Leftrightarrow 1) = 0$
- $I(1 \Leftrightarrow 0) = 0$
- $I(1 \Leftrightarrow 1) = 1$

Beispiel

Gegeben sei die Formel $F = ((A \vee B) \Rightarrow (C \vee D))$

- Die Interpretation I sei wie folgt definiert:
  - $I(A) = 1$
  - $I(B) = 0$
  - $I(C) = 1$
  - $I(D) = 0$
- Dann gilt für diese Interpretation:
  - $I(A \vee B) = I(I(A) \vee I(B)) = I(1 \vee 0) = 1$
  - $I(C \wedge D) = I(I(C) \wedge I(D)) = I(1 \wedge 0) = 0$
  - $I((A \vee B) \Rightarrow (C \wedge D)) = I(I(A \vee B) \Rightarrow I(C \wedge D)) = I(1 \Rightarrow 0) = 0$
  - Erkenntnis: $((A \vee B) \Rightarrow (C \wedge D)) = 0$

Folgerbarkeit

Definition: Modell

Jede Interpretation I, die eine Formel F zu 1 auswertet, heißt **Modell** der Formel F.
$\forall (F \in Formeln) : \forall (I \in Interpretationen) : (I.istModellVon(F) \Leftrightarrow (I(F) = 1))$

Definition: Modell

Jede Interpretation, die jede Formel der Formelmenge M zu 1 auswertet, heißt **Modell** der Formelmenge M.
$\forall (M \subseteq Formeln) : \forall (I \in Interpretationen) : (I.istModellVon(M) \Leftrightarrow \forall (F \in M) : (I(F) = 1))$

Definition: Modellmenge

Die **Modellmenge** einer Formel F ist die Menge aller ihrer Modelle.
$\forall (F \in Formeln) : (Mod(F) = \{ I \mid I.istModellVon(F) \})$

Definition: Modellmenge

Die **Modellmenge** einer Formelmenge M ist die Menge aller ihrer Modelle.
$\forall (F \in Formeln) : (Mod(M) = \{ I \mid I.istModellVon(M) \})$

Definition: folgerbar

Gegeben sei eine Formel F und eine Formel Q. Wenn jedes Modell von F auch ein Modell von Q ist, dann heißt Q aus F **folgerbar**.
$(F \models Q) \Leftrightarrow (Mod(F) \subseteq Mod(Q))$

Definition: folgerbar

Gegeben sei eine Formelmenge M und eine Formel Q. Wenn jedes Modell von M auch ein Modell von Q ist, dann heißt Q aus M **folgerbar**.
$(M \models Q) \Leftrightarrow (Mod(M) \subseteq Mod(Q))$

Wie überprüft man, ob eine Formel Q aus einer Formel F folgerbar ist? Man kann zum Beispiel alle Möglichkeiten durchprobieren in Form von Wahrheitstabellen. Dies funktioniert aber nur bei wenigen Modellen. Bei n Modellen müssten in eine Wahrheitstabelle 2^n Zeilen enthalten sein.

Beispiel 3

- Aussage: Wenn es warm ist, ist es nicht kalt.
- Schlussfolgerung: Wenn es kalt ist, ist es nicht warm.

warm	***kalt***	***warm*** $\Rightarrow \neg$ ***kalt***	***kalt*** $\Rightarrow \neg$ ***warm***	$kalt \vee \neg kalt$
0	0	1	1	1
0	1	1	1	1
1	0	1	1	1
1	1	0	0	1

Definition: äquivalent

2 Formeln heißen **äquivalent** (Symbol: '≡'), wenn jede Interpretation sie gleich auswertet.

Definition: Tautologie, allgemein

Formeln, die in alle Interpretation wahr sind, heißen **Tautologie**n. Solche Formeln werden als **allgemeingültig** bezeichnet. (Beispiel: (A∨¬A))

Definition: Kontradiktion

Formeln, die in allen Interpretation falsch sind, heißen **Kontradiktion**en. Solche Formeln werden als **widersprüchlich** bezeichnet. (Beispiel: (A∧¬A))

Definition: erfüllbar

Formeln, die mindestens ein Modell haben, heißen **erfüllbar**.

Satz

Die Menge der Tautologien ist eine Teilmenge der Menge der erfüllbaren Formeln.
Die Menge der Kontradiktionen und die Menge der erfüllbare Formeln sind disjunkt.

Notation

„Aus F folgt Q“: F ⊨ Q

Satz

Folgende Aussagen sind äquivalent:
1. P ⊨ Q
2. (P⇒Q) ist eine Tautologie
3. (P∧¬Q) ist eine Kontradiktion

Beweis:

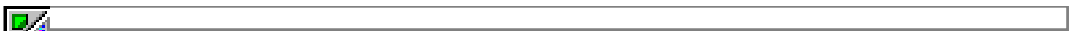
- Wir werden folgendes zeigen:
  - (1)⇒(2)
  - (2)⇒(3)
  - (3)⇒(1)

Beweis (1)⇒(2):

Es gelte P ⊨ Q. Berachte eine Interpretation I. Es gibt 2 Fälle:
1. I ist Modell von P. Da P ⊨ Q, ist jedes Modell von P auch Modell von Q. Damit wird P⇒Q zu 1 ausgewertet.
2. I ist kein Modell von P. Dann ist der Wahrheitswert von P⇒Q in I 1.

Da P⇒Q in allen Fällen zu 1 ausgewertet wird, ist P⇒Q eine Tautologie.

Die anderen Fälle werden ähnlich bewiesen.



Datum: 13.11.2002

Wir werden doch die anderen Fälle beweisen:

Satz

Folgende Aussagen sind äquivalent
1. $P \models Q$
2. $(P \Rightarrow Q)$ ist Tautologie
3. $P \wedge \neg Q$ ist Kontradiktion

Beweis (1) ⇒ (2) beim letzten Mal.

Exkurs: Formeln:
- $P \Leftrightarrow Q$ ist eine Formel (Aussage in Formel-Sprache)
- $P \equiv Q$ ist keine Formel (Aussage in Meta-Sprache)
- $P \Rightarrow Q$ ist eine Formel (Aussage in Formel-Sprache)
- $P \models Q$ ist keine Formel (Aussage in Meta-Sprache)

Beweis (2)⇒(3):

Wenn $P \Rightarrow Q$ eine Tautologie ist, so wird in jeder Interpretation von P zu 0 ausgewertet oder jede Interpretation von Q zu 1. Damit gibt es keine Interpretation, die P zu 1 und Q zu 0 auswertet. Also gibt es keine Interpretation,

die $P \wedge \neg Q$ zu 1 auswertet. Damit ist $P \wedge \neg Q$ eine Kontradiktion.

Beweis (1)⇒(1):

Sei $P \Rightarrow Q$ eine Kontradiktion. Dann gibt es keine Interpretation, die P zu 1 und Q zu 0 auswertet. Also wertet jede Interpretation, die P zu 1 auswertet, auch Q zu 1 aus. Damit gilt: $P \models Q$.

Satz: häufig verwendete Äquivalenzen

Folgende Äquivalenzen werden oft verwendet, um Formeln zu vereinfachen.

Seien P,Q,R beliebige Formeln. Dann gilt:

- $(\neg\neg P) \equiv P$
- $(P \Rightarrow Q) \equiv (\neg P) \vee Q$
- $(P \Leftrightarrow Q) \equiv (P \wedge Q) \vee ((\neg P) \wedge (\neg Q))$
- de-Morgansche Regeln:
  - $\neg(P \vee Q) \equiv (\neg P) \wedge (\neg Q)$
  - $\neg(P \wedge Q) \equiv (\neg P) \wedge (\neg Q)$
- Distributivität
  - $(P \wedge Q) \vee R \equiv (P \vee R) \wedge (Q \vee R)$
  - $(P \vee Q) \wedge R \equiv (P \wedge R) \vee (Q \wedge R)$

„Wann immer ich zusätzliche Informationen habe, dann wird nicht eine Interpretation plötzlich Modell. Das heißt, zusätzliche Information kann immer nur dazu führen, dass irgendeine Interpretation, die vorher Modell war, kein Modell ist."

3.3 Beispiel Geldautomat

Folgende Aussagen sind relevant:

eingegebene Karte ist gültig	K
eingegebene PIN ist okay	P
Kontostand ist okay	S
Geldbetrag auszahlen	A
Rücklgabe Karte	R

„Jetzt wollen wir bestimmtes Wissen darüber repräsentieren, wie dieser Geldautomat sich verhalten soll."

1. Die Karte wird einbehalten genau dann, wenn die eingegebene PIN falsch ist. $P \Leftrightarrow R$
2. Wir legen fest, dass der Betrag ausgezahlt wird, wenn die Karte gültig ist, die PIN okay und der Kontostand ausreichend ist. $(K \wedge P \wedge S) \Rightarrow A$
3. Nur unter obigen Bedingungen soll ausbezahlt werden. $A \Rightarrow (K \wedge P \wedge S)$

K	***P***	***S***	***A***	***R***	***ist ein Modell?***
0	0	0	0	0	
0	0	0	1	0	nein wegen 3

K	*P*	*S*	*A*	*R*	*ist ein Modell?*
0	0	1	0	0	
0	0	1	1	0	nein wegen 3
0	1	0	0	1	
0	1	0	1	1	nein wegen 3
0	1	1	0	1	
0	1	1	1	1	nein wegen 3
1	0	0	0	0	
1	0	0	1	0	nein wegen 3
1	0	1	0	0	
1	0	1	1	0	nein wegen 3
1	1	0	0	1	
1	1	0	1	1	nein wegen 3
1	1	1	0	1	nein wegen 2
1	1	1	1	1	

Jetzt sind nur noch 8 Intepretationen Modelle.

- Was können wir aus (1),(2),(3) und $\neg K$ folgern?
  - zum Beispiel: $\neg A$
- Was können wir aus (1),(2),(3) und A folgern?
  - zum Beispiel: K
  - zum Beispiel: P
  - zum Beispiel: S
- Was können wir aus (1),(2),(3) und $\neg A$ folgern?
  - $(\neg K)\vee(\neg P)\vee(\neg S)$

3.4 Inferenz

Eine **Inferenzregel** dient der Ableitung von Formeln aus bereits gegebenen Formeln.

Allgemeine Form: $\frac{\text{Prämissen}}{\text{Konklussion}}$ Beispiel: $\frac{F_{1,}F_{2,}\ldots,F_{n}}{F}$ wobei $\forall(j):\left(F_{i}\right)$ Prämissen sind und F die Schlussfolgerung.

Die Regel ist **korrekt**, falls $\left\{F_{1,}F_{2,}\ldots,F_{n}\right\}\models F$.

Beispiele korrekter Inferenzregeln. Seien P, Q, R beliebige Formeln sowie T beliebige Tautotlogien:

1. Modus ponens: $\frac{P,P\Rightarrow Q}{Q}$
2. Modus tollens: $\frac{P\Rightarrow Q,\neg Q}{\neg P}$
   - Beispiel: Wenn es regnet ist es nass. Wenn es nicht nass ist, regnet es nicht.

3. Negations-Elimination: $\frac{\neg\neg P}{P}$

4. Negations-Einführung: $\frac{P}{\neg\neg P}$

5. Kontraposition: $\frac{P \Rightarrow Q}{(\neg Q) \Rightarrow (\neg P)}$

6. Resolution: $\frac{P \wedge Q, (\neg P) \wedge R}{Q \vee R}$

7. T-Einführungs-Regel: $\frac{}{T}$ (Es gilt ohne Vorbedingung immer eine Tautologie)

Sei M eine Menge von Formeln, R eine Menge von Inferenzregeln.
$I \in R$ ist anwendbar in M, wenn M die Prämissen von I enthält. Ein formaler R-Beweis aus M für F_n ist eine Folge von Formeln $(F_1, F_2, \ldots, F_n)$, sodass für $i \in \{1, \ldots, n\}$ gilt:
1. $F_i \in M$ (Prämisse) oder
2. es gibt $I \in R$ mit Konklusion F_i, I ist anwendbar in $\{F_{1,} \ldots, F_{i_1}\}$

Beispiel

Informationen über Peter Müller:
P={
$(\text{istMathematiker}() \Rightarrow \neg \text{magComputer})$,
$\neg \text{istMathematiker}() \Rightarrow \text{istInformatiker}()$,
$\text{istInformatiker}() \Rightarrow \text{istSchlau}()$,
magComputer
}
$M \Rightarrow \neg C$
$\neg M \Rightarrow I$
$I \Rightarrow S$
C
R={(1),(4)}

Beweis: ($M \Rightarrow \neg C$, $\neg\neg C \Rightarrow \neg M$,C, $\neg\neg C$, $\neg M$, $\neg M \Rightarrow I$, I, $I \Rightarrow S$, S)

4. Relationen

4.1 Grundlegende Definitionen

Datum: 27.11.2002

Definition: Relation

Eine **Relation** R in einer Menge M ist eine Beziehung zwischen zwei Elementen von M.

Definition: Relation

Eine **Relation** R in einer Menge M ist eine Teilmenge M.

Beispiel

Relation '<' („kleiner als“) ist eine Relation $\mathbb{N}$:

$$(a<b)\Leftrightarrow(\exists(r\in\mathbb{N}^+):(a+r=b))$$

Falls „a<b“ gilt, dann sagt man auch: „'<' trifft auf (a,b) zu.“

Eine Relationen kann durch Paare charakterisiert werden, auf die sie zutrifft.

- Eine Relation R in M ist eine Teilmenge von $M\times M$
  Statt „ $(a,b)\in R$ “ schreibt man auch „ $a\,R\,b$ “.
- Entsprechend gilt: Eine Relation zwischen zwei Mengen M und N ist eine Teilmenge von $M\times N$.

Definition: Vorbereich, Nachbereich, Feld, inverse Relation

Sei R eine Relation in M. Dann ist:

- **Vorbereich** von R (VB von R): $\text{Vorbereich}(R)=\{x\in M\,|\,\exists(y\in M):(x\,R\,y)\}$
- **Nachbereich** von R (NB von R): $\text{Nachbereich}(R)=\{y\in N\,|\,\exists(x\in M):(x\,R\,y)\}$
- **Feld** von R (Fd von R): $\text{Feld}(R)=\text{Vorbereich}(R)\cup\text{Nachbereich}(R)$
- zu R **inverse Relation** R^{-1}: $R^{-1}=\{(y,x)\,|\,(x,y)\in R\}$

Eigenschaften von Relationen

R ist	***falls für alle x,y,z ∈ M gilt:***	***Beispiel***
reflexiv	$x\,R\,x$, $(x,x)\in R$	$\subseteq$
irreflexiv	$\neg(x\,R\,x)$	$\subset$
symmetrisch	$(x\,R\,y)\Rightarrow(y\,R\,x)$	$\equiv$
asymmetrisch	$(x\,R\,y)\Rightarrow\neg(y\,R\,x)$	$\subset$
antisymmetrisch	$((x\,R\,y)\wedge(y\,R\,x))\Rightarrow(x=y)$	$\subseteq$
transitiv	$((x\,R\,y)\wedge(y\,R\,z))\Rightarrow(x\,R\,z)$	$\vDash$
linear	$(x\,R\,y)\vee(y\,R\,x)$	$\geqslant$ auf $\mathbb{N}$
konnex	$(x\neq y)\Leftrightarrow((x\,R\,y)\vee(y\,R\,x))$	$<$
voreindeutig	$((y\,R\,x)\wedge(z\,R\,x))\Rightarrow(y=z)$	nur eine Kante zu Knoten in Nachbereich

R ist	***falls für alle x,y,z ∈ M gilt:***	***Beispiel***
eindeutig	$((x\,R\,y)\wedge(x\,R\,z))\Rightarrow(y=z)$	nur eine Kante zu Knoten in Vorbereich
eineindeutig	$(R.istEindeutig())\,und\,(R.istVoreindeutig())$	

Definition: Einschränkung

Sei R Relation in M, $N\subseteq M$.
Dann ist die **Einschränkung** von R **auf** N (geschrieben als $R\|N$) die Menge:
$R\|N=\{(a,b)\in R|(a\in N)\wedge(b\in N)\}$

Definition: Relations-Verknüpfung

Die Verknüpfung von Relationen R in M und S in M (geschrieben als $R\circ S$) ist die Menge:
$R\circ S=\{(x,z)|\exists(y):((x\,R\,y)\wedge(y\,S\,z))\}$

Beispiel

Sei M={Peter,Franz,Fritz,...} eine Menge von Personen und seien folgende Relationen definiert:
- $V=\{(x,y)|(x,y\in M)\wedge(x \text{ ist verwand mit } y)\}$
- $F=\{(x,y)|(x,y\in M)\wedge(x \text{ ist befreundet mit } y)\}$

Dann ist:
- $((x,y)\in(V\circ F))\Leftrightarrow(y \text{ ist Freund eines Verwandten von } x)$
- $((x,y)\in(F\circ V))\Leftrightarrow(y \text{ ist Verwandter eines Freundes von } x)$

Definition: Transitive Hülle

Sei R Relation in M. Die **transitive Hülle** von R (geschrieben als: R^t)[oder auch R^*) ist die kleinste Relation, für die gilt:
1. $R\subseteq R^t$
2. $(((x,y)\in R^t)\wedge((y,z)\in R))\Rightarrow((x,z)\in R^t)$

Beispiel

Sei $R=\{(x,x+1)|x\in\mathbb{N}\}$ die direkte Nachfolgerelation auf $\mathbb{N}$, dann ist R^t die '<'-Relation

4.2 Aquivalenzrelationen

Eine Relation R in M heißt **Äquivalenzrelation**, wenn R
- reflexiv,
- symmetrisch und
- transitiv

ist.

Beispiele

- $\equiv$ auf die Menge der aussagenlogischen Formeln
- $R=\{(x,y)|x \text{ und } y \text{ sind Studierende im gleichen Semester }\}$

Definition: Zerlegung

Eine **Zerlegung** („**Partition**“) einer Menge M ist die Menge K von nichtleeren Teilmengen von M, sodass
1. die Elemente von K sind paarweise disjunkt und
2. jedes Element von M ist Element eines Elementes von K

Das heißt, dass jedem Element aus M genau eine Teilmenge von M zugeordnet ist, die das Element ebenfalls enthält.

Definition: Äquivalenzklasse

Jede Äquivalenzrelation R in M induziert eine Zerlegung K von M in **Äquivalenzklasse**n, wobei a und b zu einer Klasse gehören, genau dann, wenn ($a R b$).

Die Klasse der zu a äquivalenten Elemente von M wird mit $[a]_R$ bezeichnet. Es gilt also: $[a]_R=\{b|((b\in M)\wedge(a R b))\}$.

4.3 Ordnungsrelationen

Sei R eine Relation in M. R heißt:
- **reflexive Halb-Ordnung** falls R
  - reflexiv,
  - transitiv und
  - antisymmetrisch ist.
  - Beispiel: $\subseteq$
- **reflexive Vollordnung** falls R
  - reflexive Ordnung und
  - linear ist.
  - Beispiel: $\leqslant$
- **irreflexive Halb-Ordnung** falls R
  - irreflexiv und
  - transitiv (und damit auch antisymmetrisch [nicht etwas asymmetrisch?]) ist.
  - Beispiel: $\subset$
- **irreflexive Vollordnung** falls R
  - irreflexive Ordnung und
  - konnex ist.
  - Beispiel: $<$

Anmerkung 1

- Vollordnungen werden auch **totale Ordnung**en genannt.
- Halbordnungen werden auch **partielle Ordnung**en genannt.
- Ordnungen, die nicht Vollordnungen sind, heißen **echte Halbordnung**en.

Anmerkung 2

Irreflexivität und Transitivität impliziert Asymmetrie.

Beweis

Wäre R nicht asymmetrisch, dann gäbe es zwei x und y mit x R y und y R x. Aufgrund der Transitivität würde daraus folgen: x R x. Dies ist aber im Widerspruch zur Irreflexivität.

Definition: Schranken, Extrema, Supremum, Infimum

Sei R eine Ordnung auf M sowie $N \subseteq M$. Dann ist

- $a \in M$ **obere Schranke** von N, falls $\forall (x \in N) : (x\,R\,a)$
- $a \in M$ **untere Schranke** von N, falls $\forall (x \in N) : (a\,R\,x)$
- $a \in M$ ist **maximales Element** von N, falls $\neg(\exists (x \in N) : ((a\,R\,x) \wedge (x \neq a)))$
- $a \in M$ ist **minimales Element** von N, falls $\neg(\exists (x \in N) : ((x\,R\,a) \wedge (x \neq a)))$
- $a \in M$ ist **Maximum** von N, falls $\forall ((x \in N) \wedge (x \neq a)) : (x\,R\,a)$
- $a \in M$ ist **Minimum** von N, falls $\forall ((x \in N) \wedge (x \neq a)) : (a\,R\,x)$
- $a \in M$ heißt **Supremum** (obere Grenze) von N (geschrieben als: „sup N"), falls a Minimum der Menge der oberen Schranken von N ist.
- $a \in M$ heißt **Infimum** (untere Grenze) von N (geschrieben als „inf N"), falls a Maximum der Menge der unteren Schranken von N ist.

Beispiel

1. geordnete Menge $(M, <)$, $M = \mathbb{N}$, $N = \{4,5,6,7\}$. (Die Relation '$<$' ist die „kleiner als“-Relation)
   - Jede Zahl größer 7 ist obere Schranken.
   - Jede Zahl kleiner als 4 ist untere Schranke.
   - 7 ist Maximum und maximales Element.
   - 4 ist Minimum und minimales Element.
   - 8 ist Supremum.
   - 3 ist Infimum.
2. geordnete Menge $(M, \leqslant)$, $M = \mathbb{N}$, $N = \{4,5,6,7\}$. (Die Relation '$\leqslant$' ist die „kleiner als oder gleich“-Relation)
   - Jede Zahl größer gleich 7 ist obere Schranken.
   - Jede Zahl kleiner gleich 4 ist untere Schranke.
   - 7 ist Maximum und maximales Element.
   - 4 ist Minimum und minimales Element.
   - 7 ist Supremum.
   - 4 ist Infimum.
3. geordnete Menge $(M, \subset)$, $M = \text{Potenzmenge}(\{a, b, c\})$, $N = \{\{a\}, \{b\}\}$
   - Obere Schranke ist: $\{a, b\}$
   - Obere Schranke ist: $\{a, b, c\}$
   - Untere Schranke ist: $\emptyset$
   - Maximales Element ist: $\{a\}$
   - Maximales Element ist: $\{b\}$
   - Minimales Element ist: $\{a\}$
   - Minimales Element ist: $\{b\}$
   - Maximum: existiert nicht
   - Minimum: existiert nicht
   - Supremum: $\{a, b\}$
   - Infimum: $\emptyset$

Definition: Wohlordnung

Eine Vollordnung R in M heißt **Wohlordnung**, falls jede nichtleere Teilmenge von M ein minimales Element besitzt.

Beispiel

- $\leqslant$ auf $\mathbb{N}$ ist eine Wohlordnung
- $\leqslant$ auf $\mathbb{Z}$ ist keine Wohlordnung (weil es z.B. $\mathbb{Z}$ selbst kein minimales Element hat).

5. Korrespondenzen und Abbildungen, Unendlichkeit

Definition: Relation, Korrespondenz

Eine Relation K (Korrespondenz) **zwischen** zwei Mengen M und N ist eine Relation in

$M \cup N$ mit $\text{Vorbereich}(K) \subseteq M$ und $\text{Nachbereich}(K) \subseteq N$.

Definition: Bild, Urbild

Bei einem Paar $(a,b) \in K$ nennt man
- a **Urbild** von b und
- b **Bild** von a.

Definition: Vorbereich, Definitionsbereich

$\text{Vorbereich}(K)$ heißt auch **Definitionsbereich**.

Definition: Nachbereich, Wertebereich

$\text{Nachbereich}(K)$ heißt auch **Wertebereich**.

Definition: Abbildung, Funktion

Ist eine Korrespondenz eindeutig (jedes Urbild hat maximal 1 Bild), dann heißt die Korrespondenz **Abbildung** oder **Funktion**.

Definition: partielle Abbildung

Gilt $\text{Vorbereich}(K) \subseteq M$, dann heißt die Abbildung „**Abbildung aus** M in N“. Eine Solche Abbildung heißt auch **partielle Abbildung**.
Gilt $\text{Vorbereich}(K) = M$, dann heißt die Abbildung „**Abbildung von** M in N“.
Eine „Abbildung f von M in N“ wird geschrieben als „ $f : M \to N$“

Definition: Verkettung

Seien f und g Abbildungen von M nach N und von N nach R. Dann heißt die Abbildung $f \circ g = \{(a,c) \mid \exists (b \in N) : ((a,b) \in A \wedge (b,c) \in g)\}$ **Verkettung** von f und g.

Definition: injektiv, surjektiv, bijektiv

$f : M \to N$ heißt injektiv, wenn kein Element von N mehr als ein Urbild hat.
$f : M \to N$ heißt surjektiv, wenn jedes Element von N mindestens ein Urbild hat.
$f : M \to N$ heißt bijektiv, wenn f injektiv und surjektiv ist.

Satz

Anmerkung: Wenn M und N endlich sind und es eine bijektive Funktion $f : M \to N$ gibt, dann sind M und N gleichmächtig, haben also die gleiche Länge.

Satz: Unendlichkeitsdefinition nach Dedekind

Eine Menge M ist **unendlich**, wenn es eine bijektive Abbildung von M in eine echte Teilmenge von M gibt.

Beispiel

Menge $\mathbb{N}$, Menge der geraden Zahlen $G = 2 \cdot \mathbb{N}$ (welche eine echte Teilmenge von $\mathbb{N}$ ist). Es gibt die Funktion $f : \mathbb{N} \to G$, $f(x) = 2 \cdot x$. Diese Funktion ist bijektiv.

Definition: abzählbar

Eine Menge M heißt **abzählbar**, wenn es eine surjektive Abbildung den natürlichen Zahlen $\mathbb{N}$ auf M gibt.

Definition: abzählbar unendlich

Eine Menge M heißt **abzählbar unendlich**, wenn M abzählbar und unendlich ist.

Beispiel

Die ganzen Zahlen $\mathbb{Z}$ sind abzählbar, weil es diese surjektive Funktion gibt: $f(n)=\begin{cases} k & \text{falls } n=2\cdot k \\ -k & \text{falls } n=2\cdot k+1 \end{cases}$ (Diese Funktion ist sogar bijektiv.)

Definition: überabzählbar

Nicht abzählbare Mengen heißen **überabzählbar**.

Satz

Die reellen Zahlen $\mathbb{R}$ sind überabzählbar.

Beweis

Cantorsches Diagonalverfahren:
Bereits die Menge $]0,1[$ aus den rellen Zahlen größer 0 und kleiner 1 ist überabzählbar:

- Sei $f:\mathbb{N}\to]0,1[$ und sei f surjektiv.
- Jedes $r\in]0,1[$ kann als nichtabzählbare Dezimalzahl dargestellt werden.
  - Sei $f(n)=[0,z_{n0,}z_{n1,}z_{n2,}\ldots]=10^{-1}\cdot z_{n0}+10^{-2}\cdot z_{n1}+10^{-3}\cdot z_{n2}+\ldots$
    - Man betrachte die relle Zahl $d=[0,d_0d_1d_2\ldots]_{10}=10^{-1}\cdot d_0+10^{-2}\cdot d_1+10^{-3}\cdot d_2+\ldots$ mit $d_j=\begin{cases} 2 & \text{falls } z_{jj}=1 \\ 1 & \text{sonst} \end{cases}$
    - d ist von allen Bildern von f verschieden, damit ist f nicht surjektiv.
    - d unterscheidet sich von $f(n)$ an der Stelle z_{nn} („der n-ten Stelle hinter'm Komma“)

6. Algebraische Strukturen

Definition: algebraische Struktur

Eine **algebraische Struktur** $A=(M_1,f_1,\ldots,f_s,R_1,\ldots,R_t)$ besteht aus folgenden Komponenten:

- nichtleere Menge M (heißt auch „Trägermenge“ oder „Universum“)
- Funktionen f_i (mit $1\leqslant i\leqslant s$) mit einer zugehörigen Stelligkeit m_i (das heißt, dass f_i m_i Argumente übergeben bekommt, also: $f_i:M^{m_i}\to M$)
- Relationen R_j (mit $1\leqslant j\leqslant t$) mit zugehöriger Stelligkeit n_j (das heißt: $R_j\subseteq M^{n_j}$)

Definition: Signatur, Typ

Die Folge $(m_1, m_2, \ldots, m_s, n_1, \ldots, n_t)$ heißt **Typ** oder **Signatur** der algebraischen Stuktur.

Definition: Algebra

Eine algebraische Struktur heißt **Algebra**, falls $(s>0)\wedge(t=0)$. (Wenn es also nur Funktionen gibt.)

Definition: Verknüpfung

Zweistellige Funktionen $f: M\times M\to M$ heißen auch **Verknüpfungen** in M.

- Beispiele:
  - Addition
  - Multiplikation

Definition: Gruppe

Eine Menge G mit einer Verknüpfung $\circ$ heißt **Gruppe**, falls folgendes gilt:

1. $\circ$ ist assoziativ: $(a\circ b)\circ c=a\circ(b\circ c)$
2. G enthält ein **Einselement** e, also ein Element, für das gilt:
   $\forall(a\in G): (a\circ e=e\circ a=a)$
3. $\forall(a\in G): \exists(a^{-1}): (a\circ a^{-1}=e)$ Für jedes a existiert ein inverses Element, genannt a^{-1}.

Datum: 11.12.2002

Definition: abelsche Gruppe

Eine Gruppe heißt **abelsch**, falls $\circ$ kommutativ ist.

Beispiele

- $\mathbb{Z}$ mit $+$
- $\mathbb{R}$ mit $+$
- $\mathbb{C}$ mit $+$

Definition: Untergruppe

$(U,\circ)$ heißt **Untergruppe** von $(G,\circ)$ falls
- $(U,\circ)$ Gruppe ist und
- $U \subseteq G$ und
- $U \neq \emptyset$

Satz

Falls $(U_1,\circ)$ und $(U_2,\circ)$ Untergruppen sind, dann ist auch $(U_1 \cap U_2,\circ)$ eine Untergruppe.

Für die Vereinigung $U_1 \cup U_2$ gilt dies [im Allgemeinen] nicht.

Beispiel

Für die Gruppe $(\mathbb{Z},+)$ gilt:
- Untergruppe sind die Geradenzahlen.
- Untergruppe sind die durch 3 teilbaren Zahlen.
- Der Schnitt dieser Untergrupen ist z.B. {...,6,12,18,...}
- Die Vereinigung ist keine Untergrupe, da z.B. $2+3=5$ und 5 ist nicht Element der Vereinigung

Definition: Gruppen-Homomorphismus

Seien $G_1=(M_1,op_1)$ und $G_2=(M_2,op_2)$ Gruppen. Eine Abbildung $\phi: M_1 \to M_2$ heißt Gruppe-**Homomorphismus** von G_1 in G_2, falls gilt:
$\forall (a,b \in M_1): \left(\phi(a\, op_1\, b)=\phi(a)\, op_2\, \phi(b)\right)$

Definition: Gruppen-Isomorphismus

Ein Homomorphismus heißt **Isomorphismus**, falls ϕ bijektiv ist.

Beispiel

Seien G_1 positive reelle Zahlen mit Multiplikation.
Seien G_2 positive reelle Zahlen mit Addition.
Sei $\phi=\log$, es gilt $\log(a\cdot b)=\log(a)+\log(b)$.

6.3 Verbände

Definition: Verband

Eine Menge V mit zwei Verknüpfungen $\cap$ und $\cup$ heißt Verband, wenn $\forall(a,b,c\in V)$ gilt:

1. Kommutativität
   1. $a\cap b=b\cap a$
   2. $a\cup b=b\cup a$
2. Assoziativität
   1. $(a\cap b)\cap c=a\cap(b\cap c)$
   2. $(a\cup b)\cup c=a\cup(b\cup c)$
3. Absorption
   1. $a\cap(a\cup b)=a$
   2. $a\cup(a\cap b)=a$

Beispiele

- Potenzmenge einer Menge M mit $\cap$, $\cup$
- positiv ganze Zahlen mit größtem gemeinsamen Teiler und kleinstem gemeinsame Vielfachem
- Die Menge der Äquivalenzklassen aussagenlogischer Formeln mit $\wedge$ und $\vee$, wobei
  $[F_1]_{\equiv}\wedge[F_2]_{\equiv}=[F_1\wedge F_2]_{\equiv}$
  $[F_1]_{\equiv}\vee[F_2]_{\equiv}=[F_1\vee F_2]_{\equiv}$

Definition: Verbands-Homomorphismus

Seien $V_1=(M_{1,}\cap_1,\cup_1)$ und $V_2=(M_{2,}\cap_2,\cup_2)$ Verbände. Eine Abbildung $\phi: M_1\to M_2$ heißt Verbands-**Homomorphismus** von V_1 in V_2, falls
$\forall(a,b\in M_1):\Big(\big(\phi(a\cap_1 b)=\phi(a)\cap_2\phi(b)\big)\wedge\big(\phi(a\cup_1 b)=\phi(a)\cup_2\phi(b)\big)\Big)$.

Definition: Verbands-Isomorphismus

Ein Homomorphismus heißt **Isomorphismus**, falls ϕ bijektiv ist.

Definition

1. Sei $V=(M,\cap,\cup)$ ein Verband. Wir definieren $H(V)=(M,\leqslant)$ mit $(a\leqslant b)\Leftrightarrow(a\cap b=a)$.
2. Sei $H=(M,\leqslant)$ Halbordnung, so dass je 2 Elemente aus M Infimum und Suprebum besitzen.
   Wir definieren $V(H)=(M,\cap,\cup)$ wobei
   $\big((a\cap b)=\text{infimum}\big(\{a,b\}\big)\big)\wedge\big((a\cup b)=\text{supremum}\big(\{a,b\}\big)\big)$

Satz

1. Wenn V Verband ist, so ist $H(V)$ partielle Ordnung in der je 2 Elemente Supremum und Infimum besitzen.
2. Wenn H Halbordnung ist, sodass je 2 Elemente Supremum und Infimum besitzen, dann ist $V(H)$ Verband.
3. $H=H(V(H))\quad V=V(H(V))$

Beweis

Beweis, dass $H(V)$ partielle Ordnung ist:
Sei $H(V)=(M,\leqslant)$ mit $(a\leqslant b)\Leftrightarrow(a\cap b=a)$. Zu zeigen: $\leqslant$ ist reflexiv, transitiv, antisymmetrisch.

1. $a\leqslant a$
   $a\cap a=a\cap(a\cup(a\cap b))$ (Absorption)
   $=a$ (Absorption)
   $\Rightarrow a\leqslant a$
2. $((a\leqslant b)\wedge(b\leqslant c))\Rightarrow(a\leqslant c)$ (also: $a\cap c=a$)
   $a\cap c=(a\cap b)\cap c$ $(a\leqslant b)\Rightarrow(a=a\cap b)$
   $=a\cap(b\cap c)$ Assoziativität
   $=a\cap b$ $(b\leqslant c)\Rightarrow(b\cap c=b)$
   $=a$ wie oben $(a\leqslant b)$
3. $((a\leqslant b)\wedge(b\leqslant a))\Rightarrow(a=b)$
   $a=a\cap b$ wegen $a\leqslant b$
   $=b\cap a$ Kommutativität
   $=b$ wegen $b\leqslant a$

Verbände lassen sich also durch Hasse-Diagramme darstellen:

[siehe Foto]

6.4 Boolesche Algebren

Definition: boolesch distributiver Verband

Ein Verband $V=(M,\cap,\cup)$ heißt **boolesch distributiv**, falls $\forall(a,b,c\in M)$:
1. $a\cap(b\cup c)=(a\cap b)\cup(a\cap b)$
2. $a\cup(b\cap c)=(a\cup b)\cap(a\cup c)$

Beispiel

$(Potenzmenge(M),\cap,\cup)$ (gemeint sind die mengentheoretischen $\cup$ und $\cap$)

Satz

Sei V ein distributiver Verband. $((a\cap b)=(a\cap c))\wedge((a\cup b)=(a\cup c))\Rightarrow(b=c)$
Beweis: Siehe Skript

Definition: Komplement

Sei $V=(M,\cap,\cup)$ ein Verband mit kleinstem Element 0 und größtem Element 1. $b\in M$ heißt **Komplement** von $a\in M$, wenn gilt:
- $a\cap b=0$
- $a\cup b=1$

Definition: komplementärer Verband

V heißt **komplementär**, wenn jedes Element mindestens 1 Komplement besitzt.

Beispiel

$(Potenzmenge(M), \cap, \cup)$

- Nullelement ist: $\emptyset$
- Einselement ist: M
- Komplement von X ist: $M \setminus X$

Satz

In einem distributiven Verband hat jedes Element höchstens 1 Element.

Definition: Boolesche Algebra

Ein komplementär distributiver Verband heißt **Boolesche Algebra**. Das eindeutige Komplement von a bezeichnet man mit a^c.

Beispiel

$(\{0,1\}, \wedge, \vee)$

- Nullelement: 0
- Einselement: 1

Satz

In einer booleschen Algebra gelten folgende Beziehungen:
1. $0^C=1$, $1^C=0$
2. $(a^C)^C=a$
3. $(x=y)\Leftrightarrow(x^C=y^C)$
4. $(a\cup b)^C=a^C\cap b^C$, $(a\cap b)^C=a^C\cup a^B$
5. $(a\leqslant b)\Leftrightarrow(b^C\leqslant a^C)$,
6. $(a\leqslant b)\Leftrightarrow(a\cap b^C)=\emptyset$

Beispiel: aus der Technischen Informatik:

Man betrachte die Menge aller 2-stelligen booleschen Funktionen $f:\{0,1\}\times\{0,1\}\rightarrow\{0,1\}$ mit den Verknüpfungen $\wedge$ und $\vee$, die folgendermaßen definiert sind:
$(f_1\wedge f_2)(x,y)=f_1(x,y)\wedge f_2(x,y)$
$(f_1\vee f_2)(x,y)=f_1(x,y)\vee f_2(x,y)$
Diese Funktionen bilden einen booleschen Verband mit:
- Nullelement: Die Funktion f^0 mit $\forall(x,y):\left(f^0(x,y)=0\right)$
- Einselement: Die Funktion f^1 mit $\forall(x,y):\left(f^1(x,y)=1\right)$
- $f^C(x,y)=\neg f(x,y)$

Beispiel

x	y	$f_1(x,y)$	$f_2(x,y)$	$(f_1\wedge f_2)(x,y)$	$(f_1\vee f_2)(x,y)$	$f_1^C(x,y)$
0	0	0	1	0	1	1
0	1	1	1	1	1	0
1	0	1	0	0	1	0
1	1	0	0	0	0	1

7. Graphentheorie

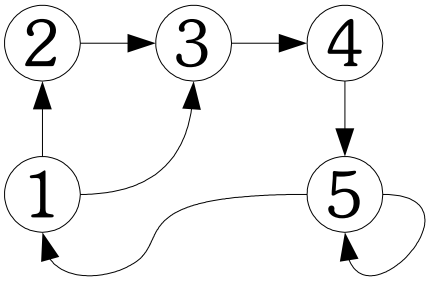
7.1 Grundlegende Definitionen

Definition: gerichteter Graph

Ein **gerichteter Graph** $G=(V,E)$ besteht aus
- einer Menge von Knoten („vertices“) V und
- einer Menge von Kanten („edges“) $E\subseteq V\times V$

Beispiel

Sei $V=\{1,2,3,4,5\}$ und $E=\{(1,2),(1,3),(2,3),(3,4),(4,5),(5,5),(5,1)\}$
Dann sieht der dadurch bezechnete Graph zum Beispiel so aus:



Definition: Pfad, Weg

Sei G ein gerichteter Graph. Eine endliche Folge $(a_1, a_2, \ldots, a_n)$ heißt dann **Pfad** in G (oder **Weg** in G), wenn $\forall(i=1,\ldots,n):\big((a_i, a_i+1)\in E\big)$.

Definition: zyklenfrei

Ein Weg heißt **zyklenfrei**, falls $\forall(i\neq j):\big(a_i\neq a_j\big)$.

Definition: zyklenfrei

Ein Graph heißt **zyklenfrei** genau dann, wenn jeder Weg in G zyklenfrei ist.

Definition: indegree, outdegree

- „Indegree“: $\text{indeg}(v)=\textit{Länge}\big(\{(x,y)\in E\,|\,y=v\}\big)$ („Eingangsgrad“)
- „Outdegree“: $\text{outdeg}(v)=\textit{Länge}\big(\{(x,y)\in E\,|\,x=v\}\big)$ („Ausgangsgrad“)

Definition: Baum

Sei $G=(V,E)$ ein gerichteter Graph. G heißt **Baum**, wenn folgendes gilt:
1. G ist zyklenfrei
2. $\exists(s\in V):\big(\big(\text{indeg}(s)=0\big)\wedge\big(\forall\big(v\neq s\big)\Rightarrow\big(\text{indeg}(v)=1\big)\big)\big)$

<joke>

Definition: X-Graph

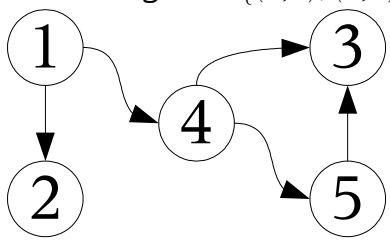
Sei $G=(V,E)$ ein gerichteter Graph. G heißt **X-Graph**, wenn folgendes gilt:
1. G ist zyklenfrei
2. $\exists\big(s\in V\big):\big(\big(\text{indeg}(s)=0\big)\wedge\big(\text{outdeg}(s)=3\big)\wedge\big(\forall\big(v\neq s\big):\big(\text{indeg}(v)=1\big)\big)\big)$
3. $\exists\big(f\in V\big):\big(\big(\text{outdeg}(f)=0\big)\wedge\big(\big((v,f)\in E\big)\Rightarrow\big(\text{outdef}(v)=1\big)\big)\big)$
4. $\big(\big(\text{outdeg}(v)=n\big)\wedge\big(n\neq 0\big)\wedge\big((v,f)\notin E\big)\big)\Rightarrow$ Es gibt genau einen Nachfolger v' von v mit outdeg(v')=n+2, und alle übrigens Nachfolger von v haben outdegree 0.

Satz

Jeder X-Graph ist ein Baum.
</joke>

Repräsentation endlicher Graphen

Gegeben sei ein Graph:
$G=(V,E)$
Knotenmenge $V=\{1,2,3,4,5\}$
Kantenmenge $E=\{(1,2),(1,4),(4,3),(4,5),(3,5)\}$



Adjazenmatrix

Speichere G durch eine $|V|\times|V|$ -Matrix A_G, wobei $\forall(i):\forall(j):\big((a_{ij}=1 \text{ (sonst 0)})\Leftrightarrow((v_i,v_j)\in E)\big)$.

Matrix für Beispiel:
$$\begin{bmatrix} 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

Adjazenzlisten

Liste $\forall\big(i\in\{1,\ldots,|V|\}\big):\big(L_i\big)$ enthält alle Nachfolger von v_i

Definition: ungerichteter Graph

Eine Struktur $G=(V,E)$ ist ein **ungerichteter Graph**, wenn folgendes gilt:
- V ist die Knotenmenge
- E ist die Menge der ungerichteten Kanten: $E\subseteq\big\{\{p,q\}\big|p,q\in V\big\}$

Definition: Weg

Sei G ein ungerichteter Graph.

$(a_1,\ldots,a_m)$ heißt **Weg** in G, falls $\{a_1,a_2\},\{a_2,a_3\},\ldots,\{a_{m-1},a_m\}\in E$.

Definition: zusammenhängend

G heißt **zusammenhängend**, wenn gilt:

$\big((p,q\in V)\wedge(p\neq q)\big)\Rightarrow$ Es gibt einen Weg von p nach q in G

Definition: zusammenhängend

Ein gerichteter Graph $G=(V,E)$ heißt **zusammenhängend**, wenn der zugehörige ungerichtete Graph $G'=(V',E')$ mit $V'=V$ und $E'=\{\{p,q\}|(p,q)\in E\}$ zusammenhängend ist.

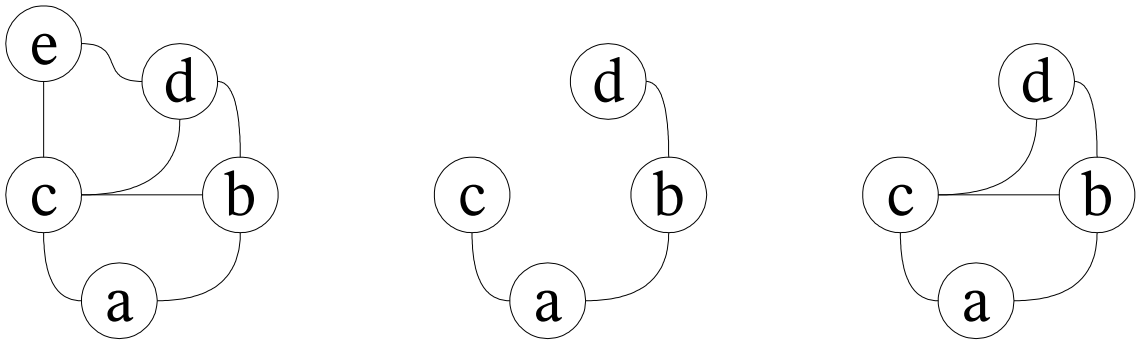
Definition: Teilgraph

Sei G ungerichteter Graph $G=(V,E)$. Der ungerichtete Graph $G'=(V',E')$ heißt:
- Teilgraph von G, genau dann wenn $(V'\subseteq V)\wedge(E'\subseteq E)$

Definition: Teilgraph

Sei G ungerichteter Graph $G=(V,E)$. Der ungerichtete Graph $G'=(V',E')$ heißt:
- Untergraph von G, genau dann wenn $(V'\subseteq V)\wedge\left(E'=\{\{p,q\}\in E|p,q\in V'\}\right)$

Beispiel

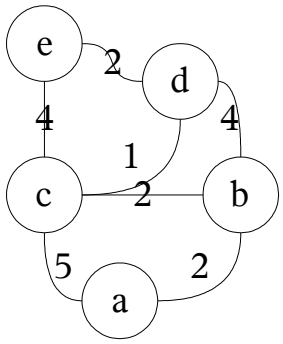


Ursprungsgraph Teilgraph Untergraph

Definition: kantenbewerteter Graph

Ein **kantenbewerteter Graph** ist ein ungerichteter Graph $G=(V,E)$ mit einer Wertungsfunktion $w:E\to\mathbb{R}^+$ die jeder Kante eine positive reelle Zahl als Kosten zuordnet.

Beispiel

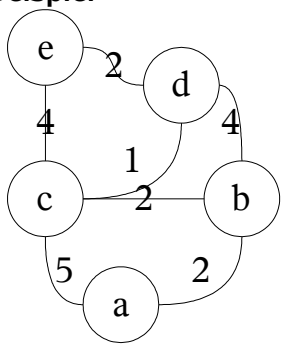


Solche Graphen sind nützlich, um z.B. die Länge eines Weges herauszufinden.

Sei G ein kantenbewerteter Graph mit n Knoten. Sei u einer der Knoten.
Gesucht ist der Teilgraph mit kürzesten Wegen von u zu anderen Knoten.
W: Liste der noch zu behandelnden Knoten

F: Liste von Kanten, die auf kürzestem Weg von u zu anderen Knoten liegen
l(v): kürzeste bisher gefundene Weglänge von u nach v
k(v): optimale von u aus zu v führende Kante

Beispiel



u=a
v:a,b,c,d,e
W:{a,b,c,d,e},{b,c,d,e},{c,d,e},{d,e},{e}, Ø
F: Ø ,{{a,b}},{{a,b},{b,c}},{{a,b},{b,c},{c,d}},... {{d,e}}

$l(a)$:	∞,0	$k(a)$:	
$l(b)$:	2	$k(b)$:	$\{a,b\}$
$l(c)$:	5,4	$k(c)$:	$\{a,c\},\{b,c\}$
$l(d)$:	∞,6,5	$k(d)$:	$\{b,d\},\{c,d\}$
$l(e)$:	∞,8,7	$k(e)$:	$\{c,e\},\{d,e\}$

[Algorithmus auf Folie zum bestimmen des kürzesten Weges]

Korrektheitsbeweis-Sketch

Wir zeigen: nach i Schleifendurchgängen sind die Entfernungen von u zu i Knoten, die am nächsten an u liegen, korrekt berechnet und diese Knoten sind aus W entfernt.
Induktionsanfang: Beim ersten Schleifendurchlauf wird u gewählt, l(u)=0
Induktionsschritt:

Nimm an, v wird im i+1 Durchlauf aus W genommen. Der kürzeste Pfad von v gehe über Vorgänger v' von v. Da v' näher an u liegt, ist nach Induktionsvorraussetzung v' bereits mit richtiger Länge aus W entfernt. Da der kürzeste Weg zu v die Länge l(v')+w (v'v) hat und dieser Wert v beim Entfernen von v' bereits zugewiesen wurde, wird von mit der korrekten Länge entfernt und die richtige Kante in F eingefügt.

Definition: indegree, outdegree

Sei $G=(V,E)$ gerichteter Graph. Es sei definiert

- $\forall(v\in V):\left(\mathbf{indeg}(v)=Länge\left(\{(x,y)\in E\,|\,y=v\}\right)\right)$ („Eingangspool“)
- $\forall(v\in V):\left(\mathbf{outdeg}(v)=Länge\left(\{(x,y)\in E\,|\,x=v\}\right)\right)$ („Ausgangspool“)

Definition: Baum

Sei $G=(V,E)$ ein gerichteter Graph. G heißt **Baum**, wenn folgendes gilt:

1. der G entsprechende ungerichtete Graph G' ist zyklenfrei
2. $\exists(s\in V):\left(\left(\text{indeg}(s)=0\right)\wedge\left(\forall\left(v\neq s\right)\Rightarrow\left(\text{indeg}(v)=1\right)\right)\right)$

Definition: Wurzel

Der Knoten s mit $\text{indeg}(s)=0$ heißt **Wurzel** des Baums.

Anmerkung

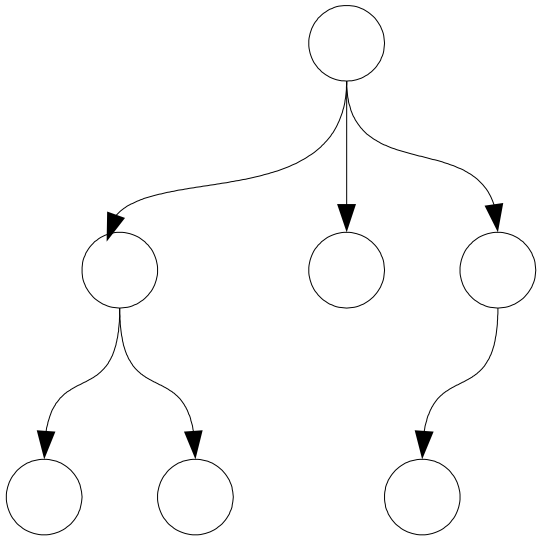
Es gibt genau einen 1 Weg von der Wurzel zu jedem anderen Knoten.

Definition: Blatt

Ein Knoten v mit $\text{outdef}(v)=0$ heißt **Blatt**.

Definition: innerer Knoten

Ein Knoten, der nicht Blatt ist, heißt **innerer Knoten**.



Baum mit Ordnung 3 und Tiefe 2

Definition: Tiefe eines Knotens

Die **Tiefe eines Knotens** v ist die Länge des gerichteten Weges (=Anzahl der Knoten) von der Wurzel zu v.

Definition: Tiefe eines Baumes

Die **Tiefe eines Baumes** ist die Länge des längsten Weges im Baum

Definition: Ordnung

Die **Ordnung eines Baums** ist die maximale Anzahl der Nachfolger eines Knotens dieses Baums.

Definition: Binärbaum

Ein Baum $G=(V,E)$ heißt **Binärbaum**, wenn $\text{Ordnung}(G)=2$.

Definition: vollständiger Baum

Sei G ein Baum der Ordnung k mit der Tiefe m. Dann heißt G **vollständig**, wenn es keinen Baum der Ordnung k mit Tiefe m gibt, der mehr Knoten als G besitzt.

Definition: geordneter Baum

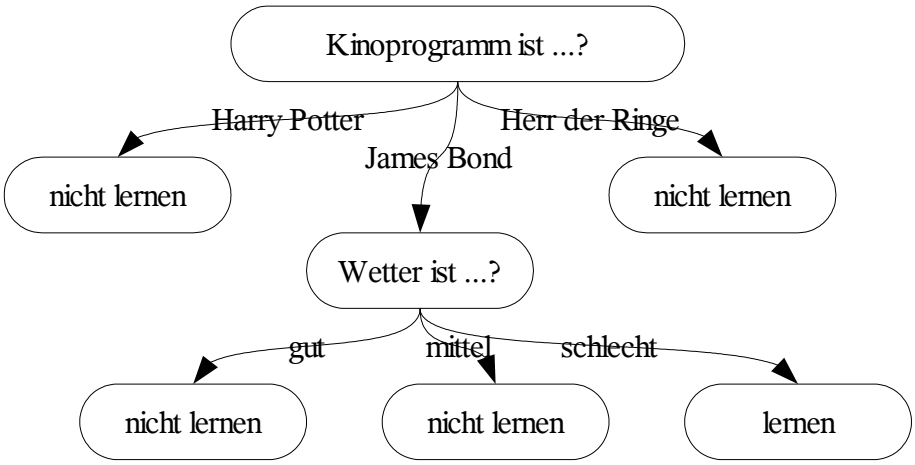
Ein Baum heißt **geordnet**, wenn die Nachfolger eines jeden Knotens geordnet sind (z.B. „1. Nachfolger“, „2. Nachfolger“, „3. Nachfolger“,...)

Definition: kantenmarkiert

Sei $G=(V,E)$ ein Baum sowie M eine Menge von Markierungen. Gibt es eine Abbildung $m: E \to M$, die jeder Kante eine Markierung zuordnet, so heißt G **kantenmarkiert**.

Beispiele für Bäume

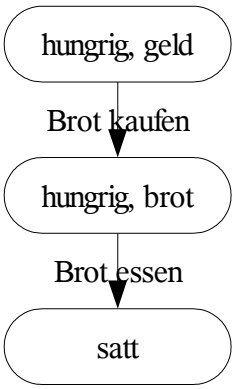
Bäume werden oft in der Informatik verwendet:
- Ableitungsbäume in der Logik
- Syntaxbäume
- Entscheidungsbäume
  - Innere Knoten entsprechen Attributen, Kanten von A nach B sind mit Werten des Attributs A markiert. Blätter entsprechend Aktionen



Entscheidungsbaums eines Studenten

- Suchbäume beim Problemlösen
  - Sei S eine Menge von Zuständen, wobei ein Zustand z.B. beschrieben sei durch eine Menge von Formeln.
    - Das Problem ist charakterisiert durch:
      - Anfangszustand s_0
      - Menge von Zielzuständen G
      - Operatoren, die Zustände in Nachfolgezustände überführen, falls sie anwendbar sind.
    - Gesucht ist: Eine Folge von Operatoren, die s_0 in einen Zielzustand überführt.
    - Lösung:
      - Erzeuge schrittweise einen Suchbaum:
        - Er ist ein markierter Baum mit der Wurzel s_0.
        - Eine Kante von s_i zu s_j hat die Markierung op, falls op in s_i anwendbar ist und s_j durch op aus s_i erzeugt werden kann.
  - Beispiel:
    - STRIPS-Planen (Stanford Research Institute Planning System)

- Zustände: Mengen von atomaren Formeln
- Operatoren:
  - Vorbedingungen „pre“: Atome, die gelten müssen, damit op anwendbar ist.
  - „delete“-Liste: Atome, die im Nachfolgezustand nicht gelten.
  - „add“-Liste: Atome, die im Nachfolgezustand gelten.
- Falls $\text{pre} \subseteq s$, dann erzeugt op aus s den Zustand $\text{op}(s) = (s \setminus \text{delete}) \cup \text{add}$
-
- $s_0 = \{\text{hungrig}, \text{geld}\}$, $G = \{s | \text{satt} \in S\}$
- Operatoren
  - „Brot kaufen“
    - $\text{pre} = \{\text{geld}\}$
    - $\text{delete} = \{\text{geld}\}$
    - $\text{add} = \{\text{brot}\}$
  - „Brot essen“
    - $\text{pre} = \{\text{brot}\}$
    - $\text{delete} = \{\text{brot}, \text{hungrig}\}$
    - $\text{add} = \{\text{satt}\}$



8. Grundlagen der Informationstheorie

Informatinstherie ist (nach Shannon) der Versuch, den Begriff der Information rein statistisch zu erfassen.

Definition: Alphabetmenge, Menge aller Zeichenketten.

Sei Σ eine Menge von Zeichen (**Alphabetmenge**).
Die **Menge aller Zeichenketten** (Wörter) über Σ (geschrieben als Σ^*) ist die kleinste Menge, für die gilt:
1. Die leere Zeichenkette ϵ ist Element von Σ^*
2. $((a \in \Sigma) \wedge (w \in \Sigma^*)) \Rightarrow (a \cdot w \in \Sigma^*)$

Definition: Nachricht

Eine **Nachricht** ist eine Folge von Zeichen, die von einem Sender (Quelle) an einen Empfänger (Senke) übermittelt wird. Damit ist eine Nachricht ein Element von Σ^* für geeignete Σ.

Definition: Information

(intuitiv): **Information** ist
- das, was durch eine Nachricht übermittelt wird oder
- die Bedeutung der Nachricht.

Definition: Informationsgehalt einer Nachricht

Der Informationsgehalt einer Nachricht ist (intuitiv) die Anzahl der Bits, die notwendig sind, um eine Nachricht in bezüglich der Wortlänge optimalem Code binär zu codieren.
- hängt ab von der Auftrittswahrscheinlichkeit der Zeichen

Definition: Wahrscheinlichkeit

Seien A und B Ereignisse. Für die die Wahrscheinlichkeiten w(A) bzw. w(B) gelten:
1. $0 \leqslant w(A) \leqslant 1$
2. $w(a)=1$ falls A sicher ist
3. $w(A \vee B)=w(A)+w(B)$ falls A und B sich ausschließen

Daraus ist vieles ableitbar, z.B.:
- $w(\neg A)=1-w(A)$

Forderungen an Informationsgehalt einer Nachricht:

1. Je seltener ein Zeichen auftritt, desto höher sollte der Informationsgehalt sein.
2. Der Informationsgehalt einer Zeichenkette soll sich aus der Summe des Informationsgehalts der einzelnen Zeichen ergeben: $I(x_1 \cdot \ldots \cdot x_n)=I(x_1)+\ldots+I(x_n)$
3. Der Informatinsgehalt eines absolut sicheren Zeichens ist 0.

Definition: Informationsgehalt, Bit

Der Logarithmus ist die einfachste Funktion, durch die diese Bedingungen erfüllt werden können. Wir definieren für ein Zeichen $x \in \Sigma$:

$$I(x)=\log_2\left(\frac{1}{w}(x)\right)=-\log_2\left(w(x)\right)$$

Die Einheit dieses Informationsgehalts ist **Bit**.

Datum: 22.01.2003

Informationstheorie nach Shannon

$$I(x)=\log_2\left(\frac{1}{w}(x)\right)=-\log_w\left(w(x)\right)$$

Die Einheit dieses Informationsgehalts ist **Bit**.

Beispiele

1. Sendet eine Quelle immer das selbe Zeichen x, so ist $w(x)=1$ und damit $I(x)=0$.
2. Haben wir ein Alphabet $\Sigma=\{0,1\}$ mit $w(0)=\frac{1}{2}$ sowie $w(1)=\frac{1}{2}$, dann ist $I(0)=1=I(1)$.
   Der Informationsgehalt von Zeichenketten ist hier die Länge der Zeichenkette.
3. Gibt es 2^k Symbole gleicher Wahrscheinlichkeit, so ist $\forall(s\in\Sigma):\left(I(s)=k\right)$
4. In deutschen Texten tritt 'b' mit einer Wahrscheinlichkeit von etwa 0.016 auf. Damit ist $I(\text{'b'})\approx\log_2\left(\frac{1}{0.016}\right)\approx 5.79$

Definition: Entropie

Die **Entropie** ist der mittlere Informationsgehalt eines Zeichens einer Quelle oder Nachricht (Abstrakt ist dies das selbe. Was man braucht ist ein Alphabet Σ und Wahrcheinlichkeiten der Symbole). Man erhält die Entropie durch Aufsummieren aller Informationsgehalte der Zeichen von Σ, gewichtet mit der Wahrscheinlichkeit der Zeichen.

Definiton: Entropie

Seien $\Sigma=\{x_1,\ldots,x_n\}$ ein Alphabet, w Wahrscheinlichkeitsverteilung über Σ. Die Entropie H von Σ und w ist:

$$H=\sum_{i=1}^{n}\left(w(x_i)\cdot I(x_i)\right)$$

Anmerkung

Die Entropie ist am größten, wenn alle Wahrscheihnlichkeiten gleich sind.

Beispiele

1. Zeichen a mit $w(a)=1$: H=0
2. Zeichen a,b mit $w(a)=w(b)=0.5$ H=1
3. Zeichen a,b,c,d mit jeweils gleicher Wahrscheinlichkeit 0.25 H=2
4. Zeichen a,b,c,d mit $w(a)=\frac{1}{2}$, $w(b)=\frac{1}{4}$, $w(c)=\frac{1}{8}$, $w(d)=\frac{1}{8}$

$$H=\frac{1}{2}\cdot 1+\frac{1}{4}\cdot 2+2\cdot\left(\frac{1}{8}\cdot 3\right)=1.75$$

Definition: Codierung

Seien A,B Alphabete. Eine **Codierung** C von A in B ist eine injektive Abbildung $C: A \to B^*$

Beispiel: Binärcodierung

$|B|=2$, meist ist $B=\{0,1\}$

Definition: Mittlere Wortlänge

Die **mittlere Wortlänge** eines Codes C für $A=\{a_1,\ldots,a_n\}$ und der Wahrscheinlichkeitsverteilung w für A ist $L=\sum_{i=1}^{n}\left(w(a_i)\cdot l_i\right)$ wobei l_i die Länge von $C(a_i)$ ist.

Theorem: Shannonsches Codierungstheorem

Seien $\Sigma=\{x_1,\ldots x_n\}$ ein Alphabet, w eine Wahrscheinlichkeitsverteilung über Σ, H die Entropie über Σ und w. Sei C eine Binärcodierung von Σ und L ihre mittlere Wortlänge. Dann gilt:

$H \leqslant L$

($H=L$ ist machbar, wenn man Codierungen $C: A^* \to B^*$ zulässt)

Intuitive Erklärung

Man kann für eine Nachricht mit Entropie H keine Binärcodierung fnden, die mit weniger als H Bits pro Zeichen (durchschnittlich) auskommt.

Code-Redundanz

Die Code-Redundanz ist: $R=L-H$
Gesucht ist oft: Code mit möglichst wenig Redundanz.

Beispiel

S	***w(s)***	***I(s)***
a	0,5	1
b	0,25	2
c	0,13	3
d	0,13	3

Die Entropie ist hier 1.75
Eine mögliche Codierung: (Codierung 1):

Zeichen	***Codierung***
a	00
b	01
c	10
d	11

Dieser Code ist mit fester Wortlänge $L=2$ und damit mit mittlerer Wortlänge $L=2$.

Eine andere mögliche Codierung: (Codierung 2)

Zeichen	***Codierung***
a	1
b	01
c	000
d	001

Dies ist ein Code mit variable Wortlänge. Wichtig ist: Fano-Eigenschaft: Kein Codewort ist ein Anfangsstück eines anderen.

$$L=\frac{1}{2}\cdot 1+\frac{1}{4}\cdot 2+\frac{1}{8}\cdot 3+\frac{1}{8}\cdot 8=1.75$$

Hier ist $H=L$, also die Redundanz=0 (bezogen auf die mittlere Wortlänge). Damit ist C optimal.

Datum: 29.01.2003

[...durch viele Abstürze verloren...]
[Hier kommt der Inhalt von TheorieI8.pdf]

[...Absturz...]

9. Einführung in die Kryptographie

Grundbegriffe

Definition: Kryptographie

Die **Kryptographie** ist die Wissenschaft von der Verschlüsselung der Daten

Definition: Kryptoanalyse

Die **Kryptoanalyse** ist die Analyse verschlüsselter Daten auf ihren Klartext.

Definition: Kryptoanalytiker

Ein **Kryptoanalytiker** ist jemand, der versucht, verschlüsselte Daten zu lesen bzw. zu entschlüsseln.

Definition: Angriff, Attacke

Der Versuch, verschlüsselte Botschaften zu entschlüsseln (zu „knacken“), wird als **Angriff** oder **Attacke** bezeichnet.

Definition: Chiffrierung

„Chiffrierung“ ist ein Synonym für „Verschlüsselung“.

Definiton: Chiffretext, Geheimtext

Das Resultat der Verschlüsselung wird **Chiffretext** oder **Geheimtext** genannt.

Definition: Klartext

Die Eingabe der Verschlüsselung („das, was verschlüsselt wird“), wird **Klartext** genannt.

9.1 Transpositionschiffren

Bei Transpositionschiffren werden Buchstaben an andere Stellen gesetzt.

Beispiel: Spaltentransposition

Klartext	„HALLO LEUTE WIE GEHT ES EUCH“
Verarbeitung	HALLO LEUTE WIEGE HTESE UCH
Chiffretext	„HLWHUAEITCLUEEHLTGS OEEE “

Verfahren:

- Man zerteile den Text in 5 Spalten und lese jede Spalte von oben nach unten.

9.2 Verschiebechiffren

(wurden von Cäsar benutzt)
Man benutze eine Zuordnung von Klartext-alphabet zu Geheimtext-Alphabet, wobei ein Buchstabe des Geheimtext-Alphabets seinem Buchstaben zum Klartext-Alphabet gegenüber um eine konstante Anzahl von k Stellen verschoben ist.

Beispiel

Klartext-Alphabet	a b c d e f g h i j k l m n o p q r s t u v w x y z
Geheimtext-Alphabet	d e f g h i j k l m n o p q r s t u v w x y z a b c

Verschiebechiffren heißen **additive Chiffren**.

9.3 Multiplikative Chiffren

Man verwende statt der Addition eine Multiplikation modulo (z.B. modulo 26 für ein Klartextalphabet der Länge 26).

Beispiel

Klartext-Alphabet	a b c d e f g h i j k l m n o p q r s t u v w x y z
Geheimtext-Alphabet bei $k=2$	b d f h j l n p q t v x z *b d f ...*
Geheimtext-Alphabet bei $k=3$	c f i l o r u x a d g j m p s v y b e h k n q t w z

Es funktionieren nur manche Offsets k, nämlich diese, die eine eineindeutige Zuordnung zwischen Klartext und Geheimtextalphbet ermöglichen. Das sind nur die, die keinen gemeinsamen Teiler mit der Länge des Alphabets haben. Im Fall einer Alphabet-Länge von 26 funktioniert $k \in \{5,7,9,11,15,17,19,21,23,25\}$.

9.4 Monoalphabetische Chiffren

Verfahren, bei denen jeder Buchstabe des Alphabets zu dem selben Geheimbuchstaben verschlüsselt wird. Damit gibt es $26 \cdot 25 \cdot 24 \cdot \ldots \cdot 1 = 26! \approx 4 \cdot 10^{26}$ Möglichkeiten.
Das Entziffern wird jedoch arg erleichtert durch eine Häufigkeitsanalyse. Für eine bestimmte Sprache ist meist bekannt, wie häufig welche Zeichen darin vorkommen. Für den Geheimtext kann man ebenfalls herausfinden, wie häufig welches Chiffre-Zeichen darin vorkommt. Idealerweise stimmen die Häufigkeiten der Klartext-Zeichen mit den Häufigkeiten der Chiffre-Zeichen überein. Daraus lässt sich eine Zuordnung zwischen Klartext-Zeichen und Chiffre-Zeichen bilden, sodass eine Dechiffrierung möglich ist.

9.5 Polyalphabetische Chiffren

Ordne einem Klartextbuchstaben mehrere Geheimzeichen zu. Die Anzahl der Geheimzeichen für einen Klartextbuchstaben sollte der Häufigkeit dieses Buchstabens entsprechen. Die jeweilige Auswahl des Geheimzeichens sei zufällig.

Vigenère-Chiffre

Dieses Verfahren stammt aus 1586.

Chiffrierung

Man benötigt zum Chiffrieren ein Schlüsselwort und das Vigenère-Quadrat.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Man schreibe das Schlüsselwort unter Klartext, wiederhole, wenn nötig. Der Buchstabe unter dem Klartextbuchstaben gibt das Alphabet des Quadrats an, das verwendet wird.

Beispiel

Klartext: P O L Y A L P H A B E T I S C H
Schlüssel: K R Y P T O K R Y P T O K R Y P T O K R Y P T O
Chiffre-Text: Z F J N T ...

9.6 Moderne Verfahren

Diese Verfahren verarbeiten Bits. Fast immer wird die XOR-Operation verwendet.

Sei s ein binärer Schlüssel. Dann gilt:
- $(y = x \text{ XOR } s) \Leftrightarrow (x = y \text{ XOR } s)$

Data Encryption Standard (DES)

Arbeitsweise

DES arbeitet auf Blöcken zu 64 Bits.
Diese Blöcke werden zu Teil-Blöcken à 32 Bits (linke Hälfte, rechte Hälfte) zerteilt.
- Diese Block-Paare durchlaufen 16 Runden. In jeder Runde wird die rechte Hälfte auf 48 Bit

erweitert, mit dem aktuellen Schlüssel geXORt, das Ergebnis in 8 Blöcke à 6 Bits aufgeteilt, diese anhand von Tabellen in 4 Bits umgewandelt. Die so entstandenen 32 Bit werden permutiert und es wird XOR mit der linken Seite durchgeführt. Das Ergebnis wird der rechten Hälfte zugewiesen, der alte Wert der rechten Hälfte wird der neuen linken Hälfte zugewiesen.
- Nach 16 Runden werden L und R zusammengeführt und nochmal permutiert.
- Der in einer Runde verwendete 48-Bit-Schlüssel wird jeweils aus dem ursprünglichen 64-Bit-Schlüssel generiert.

Eigenschaften

- Die DES-Kodierung kann zum Codieren und zum Decodieren verwendet werden (sie ist symmetrisch).
- DES galt lange Zeit als recht sicher. Jedes Eingangsbit hat Einfluss auf jedes Ausgangsbit. Eine Änderung eines einzigen Eingangsbits hat eine Änderung von etwa 50% der Ausgangsbits zur Folge.

Autor: Xuân Baldauf <xuan--2003.06.18--logik--sem02@studium.baldauf.org>

Datum: 09.04.2003

Logik

gelesen von Gerhard Brewka

1. Aussagenlogik

Definition: Logik

Was ist Logik? Die **Logik** ist die Lehre von den formalen Beziehungen (insbesondere Folgerungsbeziehungen) zwischen Sätzen.

Geschichte der Logik

- Als eigenständige Wissenschaft wurde sie begründet von Aristoteles. Dazu gehört:
  - Lehre vom Begriff
    - Klassifikation
    - Definitionslehre („Wie kann man aus vorgegebenen Begriffen neue Begriffe definieren?“)
  - Lehre vom Urteil
    - Struktur und Klassifikation von Aussagen
      - („Wie kann ich komplexe Aussagen aus einfachen erzeugen?“)
  - Lehre vom Schluss
    - Folgerung auf Grund der Strukur der Sätze
- Beiträge der Stoiker
  - Theorie der Implikation
  - logische Antinomien (wie „Dieser Satz ist falsch.“)
- mittelalterliche Scholastik
  - Arbeiten zur Sprachphilosophie und logischen Semantik
- Leibnitz
  - Vorläufer der modernen Logik
  - logische Symbolsprache
- Entwicklung der mathematischen Logik ab etwa 1850: Frege, Boole, Russel
  - Logik wird ausgedrückt mittels einer
    - Kunstsprache mit exakt definierter Syntax,
    - modelltheoretischer Semantik und
    - korrekten und vollständigen Kalkülen.

Relevanz

Warum ist die Logik relevant für die Informatik?
- Die Logik ist Beschreibungsmittel für Sachverhalte.
- Semantik für die Programmiersprachen
- Programmverifikation (Sicherstellung, dass ein Programm korrekt ist, erfordert Aussagen darüber, was korrekt ist.)

- Schaltalgebra
- Wissensrepräsentation, automatisches Beweisen
- Logikprogrammierung
  - Spezifikation von abstrakten Zielen und deren abstrakten Erreichungsmöglichkeiten (Programm)
  - Spezifikation eines konkreten Zieles, was auf das Muster eines abstrakten Zieles passt (Aufruf)
  - Generierung von konkreten Schritten zum Erreichen eines konkreten Ziels.
  - Beispiel: Prolog, make

Logik untersucht Folgerung auf Grund des Struktur der Sätze
- Beispiel:
  - Aus „Wenn es regnet, dann ist es nass." und „Es regnet." lässt sich schließen: „Es ist nass."
  - Aus „Wenn es grubelt, dann brabelt es." und „Es grubelt." lässt sich schließen: „Es brabelt."
  - Die Ableitung „Aus „Peter ist mein Bruder." zu „Peter ist mit mir verwandt."." ist kein gültiger logischer Schluss, weil die Prämissen fehlen, die Verwandtschaft mit „Bruderschaft" verknüpfen. Es wäre ein gültiger logischer Schluss, wenn die Ableitung noch die Prämisse „Wenn jemand mein Bruder ist, ist er mit mir verwandt." enthalten würde.

Logische vs. natürliche Sprache

Der Operator „und"

Das „und" der natürlichen Sprache ist nicht genau dem „und" der Logik gleich. Beispiel:
- Formal logisch interpretiert würden folgende Sätze das selbe aussagen:
  - „Peter bekam Fieber und der Arzt verschrieb ein Medikament."
  - „Der Arzt verschrieb ein Medikament und Peter bekam Fieber."

Das natürlichsprachliche „und" hat offensichtlich eine kausale Implikation impliziert.

Der Operator „dann"

„Wenn ich kein Bier zum Essen trinke, dann esse ich immer Fisch."
„Wenn ich Eis esse oder kein Bier trinke, dann esse ich nie Fisch."
Daraus lässt sich ableiten:
„Ich trinke immer Bier zum Essen und keinen Fisch oder kein Eis."

Wahrheitstabelle

Bier	***Fisch***	***Eis***	***passt?***
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1

Bier	*Fisch*	*Eis*	*passt?*
1	0	1	1
1	1	0	1
1	1	1	0

$\text{Bier} \wedge (\neg \text{Fisch} \vee \neg \text{Eis})$

1.1 Syntax der Aussagenlogik

Sei A eine Menge von Symbolen (aussagenlogische Variablen, atomare Formeln). (Beispiel nach Schöning: $A = \{A_1, A_2, \ldots\}$). Die Menge der aussagenlogischen Formeln (über A) ist induktiv wie folgt definiert:

1. Jede atomare Formel ist eine aussagenlogische Formel
2. Wenn F und G aussagenlogische Formeln sind, dann sind auch $(F \wedge G)$ und $(F \vee G)$ aussagenlogische Formeln.
3. Wenn F aussagenlogische Formel ist, dann auch $\neg F$.

Abkürzungen

$(F_1 \Rightarrow F_2)$ statt $(\neg F_1 \vee F_2)$

$(F_1 \Leftrightarrow F_2)$ statt $((F_1 \wedge F_2) \vee (\neg F_1 \wedge \neg F_2))$

$\left(\bigwedge_a^n (F_i)\right)$ statt $(\ldots((F_1 \vee F_2) \vee F_3) \vee \ldots F_n)$

$\left(\bigvee_a^n (F_i)\right)$ statt $(\ldots((F_1 \wedge F_2) \wedge F_3) \wedge \ldots F_n)$

1.2 Semantik der Aussagenlogik

Definition: Belegung, Interpretation

Eine **Interpretation** („**Belegung**“) ist eine Funktion $I: A \mapsto \{0,1\}$, wobei A die Menge der atomaren Formeln ist.

Definition: Wahrheitswert von Formeln

Sei I eine Interpretation. Wir erweitern I zu einer Funktion $\hat{I}: E \mapsto \{0,1\}$, wobei E die Menge aller Formeln über A ist.

1. $\forall (A_i \in A): \left(\hat{I}(A_i) = I(A_i)\right)$
2. $\hat{I}((F \wedge G)) = \begin{matrix} 1 & \text{falls } (\hat{I}(F) = 1) \wedge (\hat{I}(G) = 1) \\ 0 & \text{sonst} \end{matrix}$
3. $\hat{I}((F \vee G)) = \begin{matrix} 1 & \text{falls } (\hat{I}(F) = 1) \vee (\hat{I}(G) = 1) \\ 0 & \text{sonst} \end{matrix}$
4. $\hat{I}(\neg F) = \begin{matrix} 1 & \text{falls } \hat{I}(F) = 0 \\ 0 & \text{sonst} \end{matrix}$

Datum: 16.04.2003

[...30 Minuten später wegen Software-Technik...]
<import from=„Gerhard Brewka“>

Beispiel

Sei I eine Belegung, die
- A zu 1,
- B zu 0 und
- C zu 1

auswertet.
Wir bestimmen den Wahrheitswert der Formel $((A \vee B) \wedge (\neg C \vee \neg B))$

$$\begin{aligned}(I(((A \vee B) \wedge (\neg C \vee \neg B))) = 1) &\Leftrightarrow (I((A \vee B)) = 1) \wedge (I((\neg C \vee \neg B)) = 1) \\ &\Leftrightarrow ((I(A) = 1) \vee (I(B) = 1)) \wedge ((I(\neg C) = 1) \vee (I(\neg B) = 1)) \\ &\Leftrightarrow ((I(A) = 1) \vee (I(B) = 1)) \wedge ((I(C) = 0) \vee (I(B) = 0))\end{aligned}$$

Da $I(A) = 1$ und $I(B) = 0$, ist diese Bedingung erfüllt, das heißt: die Formel wird zu wahr ausgewertet.

Weniger umständlich ist: Wir ersetzen die atomaren Formeln durch ihre Wahrheitswerte und propagieren:

$$\begin{aligned}&((1 \vee 0) \wedge (\neg 1 \vee \neg 0)) \\ \Leftrightarrow &((1 \vee 0) \wedge (0 \vee 1)) \\ \Leftrightarrow &(1 \vee 1) \\ \Leftrightarrow &1\end{aligned}$$

Dies enspricht folgender Baumdarstellung.
[Grafik: Baumdarstellung]

Wahrheitstafeln

Die Wirkung der Junktoren lässt sich durch Wahrheitstafeln darstellen

F	G	$\neg F$	$F \wedge G$	$F \vee G$	$F \Rightarrow G$	$F \Leftrightarrow G$
0	0	1	0	0	1	1
0	1	1	0	1	1	0
1	0	0	0	1	0	0
1	1	0	1	1	1	1

</import>

Definition: Modell, Gültigkeit, Erfüllbarkeit

Sei F eine Formel.
- Eine Belegung, die F zu 1 auswertet, heißt **Modell** von F. Falls I Modell von $F(I(F) = 1)$ so schreiben wir $I \models F$, falls nicht $I \not\models F$.
- F heißt **erfüllbar**, wenn F mindestens 1 Modell besitzt.
- Unerfüllbare Formeln heißen **Kontradiktion**en.

- F heißt (allgemein-)**gültig** (**Tautologie**) (Notation: $\models F$), falls jede Belegung Modell ist.
- Eine **Belegung** ist Modell einer Menge von Formeln M, wenn sie jedes Element von M zu 1 auswertet.
- $I.istModellVon(\{F_1,\ldots,F_n\}) \Leftrightarrow I.istModellVon(F_1 \wedge \ldots \wedge F_n)$

Satz

$F.istTautologie() \Leftrightarrow (\neg F).istUnerfüllbar()$.

Beweis

offensichtlich.

gültig	nicht gültig	
erfüllbar		unerfüllbar
G	H $\neg H$	$\neg G$

Definition: logische Folgerung

Seien $F_1,\ldots,F_k,G$ Formeln.
G folgt aus $\{F_1,\ldots,F_k\}$ genau dann, wenn jedes Modell von $\{F_1,\ldots,F_k\}$ auch Modell von G ist.

Satz

Folgende Aussagen sind äquivalent:
1. $\{F_1,\ldots,F_k\} \Rightarrow G$
2. $\left(F_1 \wedge \left(F_2 \wedge \left(\ldots F_k\right)\right) \Rightarrow G\right).istTautologie()$
3. $\left(F_1 \wedge \left(F_2 \wedge \left(\ldots F_k\right)\right) \wedge \neg G\right).istKontradiktion()$

Definition: äquivalent

Zwei Formeln F und G heißen **äquivalent** (Notation: $F \equiv G$), falls für alle Belegungen I gilt: $I(F) = I(G)$.

Satz

$(F \equiv G) \Leftrightarrow \left((F \Leftrightarrow G)\right).istTautologie()$

Satz: Ersetzbarkeit

Seien F und G äquivalente Formeln. Sei H Formel, die F als Teilformel besitzt. H' entsteht aus H durch Ersetzen eines Vorkommens von F in H durch G. Dann gilt: $H \equiv H'$.

$\left((A \wedge \neg\neg B) \wedge (\neg\neg B \equiv B)\right) \Leftrightarrow \left((A \wedge B) \equiv (A \wedge \neg\neg B)\right)$

- Beweis: durch Induktion

Satz: Äquivalenzen

Es gelten folgende Äquivalenzen:

- **Idempotenz**
  - $\forall(F):\big((F\wedge F)\equiv F\big)$
  - $\forall(F):\big((F\vee F)\equiv F\big)$
- **Kommutativität**
  - $\forall(F,G):\big((F\wedge G)\equiv(C\wedge G)\big)$
  - $\forall(F,G):\big((F\vee G)\equiv(G\vee F)\big)$
- **Assoziativität**
  - $\forall(F,G,H):\big(((F\wedge G)\wedge H)\equiv(F\wedge(G\wedge H))\big)$
  - $\forall(F,G,H):\big(((F\vee G)\vee H)\equiv(F\vee(G\vee H))\big)$
- **Absorption**
  - $\forall(F,G):\big((F\wedge(F\vee G))\equiv F\big)$
  - $\forall(F,G):\big((F\vee(F\wedge G))\equiv F\big)$
- **Distributivität**
  - $\forall(F,G,H):\big((F\wedge(G\vee H))\equiv((F\wedge G)\vee(F\wedge H))\big)$
  - $\forall(F,G,H):\big((F\vee(G\wedge H))\equiv((F\vee G)\wedge(F\vee H))\big)$
- **doppelte Negation**
  - $\forall(F):\big((\neg\neg F)\equiv(F)\big)$
- **de Morgan**
  - $\forall(F,G):\big(\neg(F\wedge G)\equiv((\neg F)\vee(\neg G))\big)$
  - $\forall(F,G):\big(\neg(F\vee G)\equiv((\neg F)\wedge(\neg G))\big)$

Bindungsregeln

Die Operatoren binden in dieser Reihenfolge (vom stärksten zum schwächsten)

- $\neg\cdot$
- $\cdot\wedge\cdot$
- $\cdot\vee\cdot$
- $\cdot\Rightarrow\cdot$
- $\cdot\Leftrightarrow\cdot$

Beispiel

- Folgende Aussagen sind gegeben:
  - $\neg\text{Bier}\Rightarrow\text{Fisch}$
  - $\text{Eis}\wedge\neg\text{bier}\Rightarrow\neg\text{Fisch}$
  - $\neg\neg\text{Bier}\vee\text{Fisch}$
  - $\neg(\text{Eis}\vee\neg\text{Bier})\wedge\neg\text{Fisch}$
- Schlussfolgerung:

- $(\text{Bier} \vee \text{Fisch}) \wedge (\neg(\text{Eis} \vee \neg\text{Bier}) \vee \neg\text{Fisch})$
  $\equiv (\text{Bier} \vee \text{Fisch}) \wedge ((\neg\text{Eis} \wedge \text{Bier}) \vee \neg\text{Fisch})$
  $\equiv (\text{Bier} \vee \text{Fisch}) \wedge (\neg\text{Eis} \vee \neg\text{Fisch}) \wedge (\text{Bier} \vee \neg\text{Fisch})$
  $\equiv (\text{Bier} \vee \text{Fisch}) \wedge (\text{bier} \vee \neg\text{Fisch}) \wedge (\neg\text{Eis} \vee \neg\text{Fisch})$
  $\equiv \text{Bier} \wedge (\text{Fisch} \vee \neg\text{Fisch}) \wedge (\neg\text{Eis} \vee \neg\text{Fisch})$
  $\equiv \text{Bier} \wedge (\neg\text{Eis} \vee \neg\text{Fisch})$

Definition: Literal

Ein **Literal** ist
- eine atomare Formel (**positives Literal**) (zum Beispiel: A) oder
- eine Negation einer atomaren Formel (**negatives Literal**) (zum Beispiel: $\neg A$).

Definition: konjunktive Normalform

F ist in **konjunktiver Normalform** („KNF“), falls F Konjunktion von Disjunktionen von Literalen ist.

Beispiel

$(A \vee B \vee \neg C) \wedge (\neg A \vee C) \wedge \ldots$

Definition: disjunktive Normalform

F ist in **disjunktiver Normalform** („DNF“), falls F Disjunktion von Konjunktionen von Literalen ist.

Beispiel

$(A \wedge B \wedge \neg C) \vee (\neg A \wedge C) \vee \ldots$

Datum: 23.04.2003

[Folie]

$(\text{Bier} \vee \text{Fisch}) \wedge (\neg(\text{Eis} \vee \neg \text{Bier}) \vee \neg \text{Fisch})$ de Morgan

[...] [...]

Wiederholung

Definition: konjunktive Normalform

Eine Formel ist in **konjunktiver Normalform**, wenn sie eine Konjunktion von Disjunktionen von Literalen ist.

Definition: disjunktive Normalform

Eine Formel ist in **disjunktiver Normalform**, wenn sie eine Disjunktion von Konjunktionen von Literalen ist.

Satz

Für jede Formel F gibt es eine äquivalente Formel in konjunktiver Normalform.

Beweis

durch Induktion über Formelaufbau

Satz

Für jede Formel F gibt es eine äquivalente Formel in disjunktiver Normalform.

Beweis

durch Induktion über Formelaufbau

Bemerkung

Es gibt unterschiedliche Formeln in disjunktiver Normalform, die äquivalent sind.

Bemerkung

Es gibt unterschiedliche Formeln in konjunktiver Normalform, die äquivalent sind.

Beispiel

$(A \wedge B \wedge C) \vee (A \wedge B \wedge \neg C) \vee F \equiv (A \wedge B) \vee F$

Beispiel

Die Formel $(A \wedge B \wedge C)$ ist sowohl in konjunktiver als auch in disjunktiver Normalform.

Umformung in konjunktive Normalform

Gegeben sei eine Formel F, in der die Operatoren $\Rightarrow$, $\Leftrightarrow$ bereits ersetzt sind.

1. Ersetze in F jedes Vorkommen einer Teilformel der Form
   - $\neg\neg G$ durch G

- $\neg(G \vee H)$ durch $(\neg G \wedge \neg H)$
- $\neg(G \wedge H)$ durch $(\neg G \wedge \neg H)$

bis keine solche Teilformel mehr vorkommt.

2. Ersetze in F jedes Vorkommen einer Teilformel der Form
   - $(E \vee (G \wedge H))$ durch $((E \vee G) \wedge (E \vee H))$
   - $((E \wedge G) \vee H)$ durch $((E \vee H) \wedge (G \vee H))$

   bis keine solche Teilformel mehr vorkommt. (Dieser Vorgang heißt „Ausdistributieren“)

Umformung in distributive Normalform

Gegeben sei eine Formel F, in der die Operatoren $\Rightarrow$, $\Leftrightarrow$ bereits ersetzt sind.

2. Ersetze in F jedes Vorkommen einer Teilformel der Form
   - $\neg\neg G$ durch G
   - $\neg(G \vee H)$ durch $(\neg G \wedge \neg H)$
   - $\neg(G \wedge H)$ durch $(\neg G \wedge \neg H)$

   bis keine solche Teilformel mehr vorkommt.
3. Ersetze in F jedes Vorkommen einer Teilformel der Form
   - $(E \wedge (G \vee H))$ durch $((E \wedge G) \vee (E \wedge H))$
   - $((E \vee G) \wedge H)$ durch $((E \wedge H) \vee (G \wedge H))$

   bis keine solche Teilformel mehr vorkommt.

Beispiel

$(A \Rightarrow B) \Rightarrow (B \Rightarrow C)$	Elimination von $\Rightarrow$
$\neg(A \Rightarrow B) \vee (\neg B \vee C)$	Elimination von $\Rightarrow$
$\neg(\neg A \vee B) \vee (\neg B \vee C)$	de Morgan, doppelte Negation
$(A \wedge \neg B) \vee \neg B \vee C$	Distribuieren
$((A \vee \neg B) \wedge (\neg B \vee \neg B))$	Distribuieren
$(A \vee \neg B \vee C) \wedge (\neg B \vee \neg B \vee C)$	fertig: Konjunktive Normalform
$(A \vee \neg B \vee C) \wedge (\neg B \vee C)$	wir können aber weiter vereinfachen
$(\neg B \vee c)$	

Bemerkung

Die konjunktive Normalform und die disjunktive Normalform einer Formel kann auch direkt aus der Wahrheitstafel von F abgelesen werden.

- Disjunktive Normalform:
  - Die Konjunktionen von Literalen entstehen aus Zeilen, für die der Wert von F gleich 1 ist:
    - Atome, deren Wert in der Zeile 1 ist, werden positive Literale.
    - Atome, deren Wert in der Zeile 0 ist, werden negative Literale.
- Konjunktive Normalform:
  - Die Disjunktionen von Literalen entstehen aus Zeilen, für die der Wert von F gleich 0 ist.
    - Atome, deren Wert in der Zeile 0 ist, werden positive Literale.
    - Atome, deren Wert in der Zeile 1 ist, werden negative Literale.

Beispiel

A	B	C	$(A \Rightarrow B) \Rightarrow (B \Rightarrow C)$
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

Ablesen der disjunktiven Normalform

$(\neg A \wedge \neg B \wedge \neg C) \vee (\neg A \wedge \neg B \wedge C) \vee \ldots$ (insgesamt 6 Konjunktionen)

Ablesen der konjunktiven Normalform

$(A \vee \neg B \vee C) \wedge (\neg A \vee \neg B \vee C)$ (insgesamt 6 Disjunktionen)

Hinweis

zur Korrektheit der Methode für konjunktive Normalform:
- Bilde disjunktive Normalform für $\neg F$,
- wende de Morgan an

das liefert eine Formel der Form $\neg G$, sodass G in konjunktiver Normalform ist, also ist G eine konjunktive Normalform der Formel F.
In G werden alle positiven Literale der distributiven Normalform zu negativen Literalen und umgekehrt.

Hornformeln

Hornformeln sind ein wichtiger, effizient zu behandelnder Spezialfall nach Alfred Horn.

Definition: Hornformel

Eine Formel F ist eine Hornformel, falls
- F in konjunktiver Normalform ist und
- jedes Konjunktionsglied (also jede Disjunktion) höchstens ein positives Literal enthält.

Beispiel

$(A \vee \neg B) \wedge (\neg C \vee \neg A \vee D) \wedge (\neg A \vee \neg B) \wedge D \wedge \neg E$. Dies ist äquivalent zu
$(B \Rightarrow A) \wedge (C \wedge A \Rightarrow D) \wedge (A \wedge B \Rightarrow 0) \wedge (1 \Rightarrow D) \wedge (E \Rightarrow 0)$

Der Vorteil von Hornformeln ist, dass es für sie einen sehr effizienten Erfüllbarkeitstest gibt.

Hornformel-Erfüllbarkeitstest

Gegeben sei eine Hornformel F (in implikativer Form).
1. Markiere jedes Vorkommen von A in F, falls F eine Teilformel $1 \Rightarrow A$ besitzt.
2. So lange
   - ((es gibt in F eine Teilformel $G=(A_1 \wedge \ldots \wedge A_n \Rightarrow B)$) oder ($G=(A_1 \wedge \ldots \wedge A_n \Rightarrow 0)$ mit $A_1, \ldots, A_n$ markiert und B unmarkiert))

   tue
   - Wenn (G hat erste Form) dann
     - markiere jedes Vorkommen von B

     sonst
     - gib aus „unerfüllbar“ und stoppe
3. gib aus „erfüllbar“ und stoppe

Beispiel

- Wissensbasis:
  - $4\text{-Beiner} \wedge \text{miaut} \Rightarrow \text{Katze}$
  - $\text{Katze} \Rightarrow \text{Haustier}$
  - $\text{Katze} \Rightarrow \text{jagt Mäuse}$
  - $1 \Rightarrow 4\text{-Beiner}$
  - $4\text{-Beiner} \wedge \text{bellt} \Rightarrow \text{Hund}$
  - $\text{Hund} \Rightarrow \text{Haustier}$
  - $\text{Hund} \Rightarrow \text{jagt Katzen}$
  - $1 \Rightarrow \text{bellt}$
- Frage:
  - Ist $G = \text{Haustier} \wedge \text{jagt Katzen}$ ableitbar?
  - Dies ist genau dann der Fall, wenn $WB \wedge \neg(\text{Haustier} \wedge \text{jagt Katzen})$, also $\text{Haustier} \wedge \text{jagt Katzen} \Rightarrow 0$ unerfüllbar ist.
  - Die Frage können wir beantworten, indem wir wahre Aussagen durch die Wissensbasis schicken:
- Antwort-Weg:
  - $\underline{1} \Rightarrow \underline{4\text{-Beiner}}$
  - $\underline{1} \Rightarrow \underline{\text{bellt}}$
  - $\underline{4\text{-Beiner}} \wedge \underline{\text{bellt}} \Rightarrow \underline{\text{Hund}}$
  - $\underline{\text{Hund}} \Rightarrow \underline{\text{Haustier}}$
  - $\underline{\text{Hund}} \Rightarrow \underline{\text{jagt Katzen}}$
  - $\underline{\text{Haustier}} \wedge \underline{\text{jagt Katzen}} \Rightarrow 0$ ist unerfüllbar
  - also $\text{Wissensbasis} \models G$

Satz

Der Markierungsalgorithmus für implikative Hornformeln ist korrekt und stoppt nach maximal n Markierungsschritten, wobei n die Anzahl der Atome in F ist.

Beweisskizze

Die Komplexitätsableitung ist offensichtlich.

Korrektheit

Schritte 1 und 2 markieren nur solche Atome, die in allen Modellen von F den Wert 1 haben müssen. Falls 0 markiert werden müsste, kann es kein Modell geben und die Ausgabe in Schritt 2 ist korrekt. Ansonsten ist nach Verlassen von Schritt 2 die Belegung I mit $\left(I(A_j)=1\right)\Rightarrow\left(A_j\right)$ markiert ein Modell von F, denn für jede Implikation G in F gilt: entweder die Vorbedingung und die Nachbedingung von G sind wahr in I, oder die Vorbedingung von G ist falsch. In beiden Fällen ist $I(G)=1$.

Datum: 30.04.2003

Bemerkung: Markierungsalgorithmus

1. Der Algorithmus berechnet das „kleinste Modell“ von F. Hierbei gilt: $\forall(\text{Atome } A): (M \subseteq M') \Leftrightarrow (M(A) \subseteq M'(A))$. Hierbei werden alle Atome auf „wahr“ gesetzt, die wahr werden müssen, damit die Formel überhaupt mindestens 1 Modell haben kann. Das kleinste Modell ist ein Modell, was die kleinste Anzahl von Atomen hat, die notwendig ist, um eben diese Formel erfüllbar zu machen.

Exkurs: Größenrelation zwischen Modellen

$(A \Rightarrow B) \wedge (A) \wedge (C \Rightarrow D)$:
Hierfür gibt es z.B.

- das Modell $M_1: (A=1, B=1, C=0, D=0)$ aber auch
- das Modell $M_2: (A=1, B=1, C=1, D=1)$

Im Sinne des „kleineren Modells“ ist hier $M_1 < M_2$, weil in M_1 weniger Atome auf wahr gesetzt sind.

2. Falls F in implikativer Hornform ist und in F keine Implikation der Form $(A_1 \wedge \ldots \wedge A_n) \Rightarrow 0$ vorkommt, so ist F erfüllbar. Ebenso, wenn keine Implikation $1 \Rightarrow A$ vorkommt.

Satz: Endlichkeitssatz, Kompaktheitssatz

Eine Menge M von Formeln ist erfüllbar genau dann, wenn jede endliche Teilmenge von M erfüllbar ist.

- äquivalent ist: „M unerfüllbar genau dann, wenn es eine endliche Teilmenge von M gibt, die unerfüllbar ist.“

Beweis

...würde fast eine ganze Vorlesung einnehmen und ist nich sehr attraktiv. Deswegen verschieben wir ihn auf später im Hauptstudium...

Resolution

(von A. Robinson)
Resolution ist eine syntaktische Umformungsregel, die aus 2 Formeln eine neue erzeugt (nämlich die **Resolvente**). Sie wird verwendet, um eine Formelmenge auf Unerfüllbarkeit zu testen.

Anmerkung: Kalkül

Mengen von syntaktischen, mechanisch ausführbaren Umformungsregeln heißen **Kalkül**e.

Definition: korrekt

Kalküle heißen **korrekt**, wenn sie nur die gewünschten Formeln herleiten.

Definition: vollständig

Kalküle heißen vollständig, wenn sie alle gewünschten Formeln herleiten.

Anmerkung

Für Logikkalküle gilt: gewünschte Formeln und folgerbare Formeln sind die selben.

Resolution setzt vorraus, dass die Formelmenge in konjunktiver Normalform vorliegt.

Mengennotation für konjunktive Normalform

Seien $\forall(i):\forall(j):\left(L_{i,j}\right)$ Literale.
Statt $\left(L_{1,1}\vee L_{1,2}\vee\ldots\vee L_{1,n_1}\right)\wedge\left(L_{2,1}\vee L_{2,2}\vee\ldots\vee L_{2,n_1}\right)\wedge\ldots\wedge\left(L_{k,1}\vee L_{k,2}\vee\ldots\vee L_{k,n_k}\right)$
schreiben wir $\left\{\left\{L_{1,1},L_{1,2},\ldots,L_{1,n_1}\right\},\left\{L_{2,1},L_{2,2},\ldots,L_{2,n_1}\right\},\ldots,\left\{L_{k,1},L_{k,2},\ldots,L_{k,n_k}\right\}\right\}$.

Definition: Klausel

Jedes Element einer Menge, die eine Formel in Mengennotation darstellt, heißt **Klausel**. Eine Klausel entspricht damit einer Disjunktion einer Formel in konjunktiver Normalform.

Definition: Resolvente

Seien K_1, K_2 Klauseln. R heißt **Resolvente** von K_1 und K_2, falls
- es ein Literal L gibt mit $\left(L\in K_1\right)\wedge\left((-L)\in K_2\right)$ und
- $R=\left(K_1\setminus\{L\}\right)\cup\left(K_2\setminus\{-L\}\right)$

Hierbei ist:
- $-L=\neg L$, falls L ein Atom ist und
- $-L=A$ falls $L=\neg A$

Beispiel

Sei $K_1=\{A,B\}$, $K_2=\{\neg B,\neg C\}$, $R=\{A,\neg C\}$.
$A\vee B,\neg B\vee\neg C\vDash A\vee\neg C$

Spezialfall: leere Klausel

Eine leere Klausel, geschrieben als $\{\}$ oder $\square$ oder $\bot$, repräsentiert falsch (also Widerspruch).

Satz: Resolutionslemma

Sei
- F eine Formel in Klauselform
- R Resolvente zweier Klauseln in F.

Dann sind F und $F\cup\{R\}$ äquivalent.

Beweis

Sei I eine Belegung. Zu zeigen ist: $\left(I\vDash F\cup\{R\}\right)\Leftrightarrow\left(I\vDash F\right)$
- Teilbeweis 0: Fall: $\left(I\vDash F\cup\{R\}\right)\Rightarrow\left(I\vDash F\right)$
  - Falls $I\vDash F\cup\{R\}$, dann offensichtlich $I\vDash F$

- Teilbeweis 1: Fall: $\left(I \vDash F \cup \{R\}\right) \Leftarrow \left(I \vDash F\right)$
  - Es gelte $I \vDash F$.
  - Dann gilt:
    - $I \vDash K$ für jede Klausel K in F.
  - Ferner sei $R = \left(K_1 \setminus \{L\}\right) \cup \left(K_2 \setminus \{-L\}\right)$
    - Fall 1: $I \vDash L$
      - Dann gilt: $I \vDash K_2 \setminus \{-L\}$
      - und deshalb $I \vDash R$
    - Falls 2: $I \vDash -L$
      - Dann gilt: $I \vDash K_1 \setminus \{L\}$
      - und deshalb $I \vDash R$

Beispiel

$K_1 = \{A, \neg C, D\}$, $K_2 = \{\neg A, B\}$
$R = \{\neg C, D, B\}$

Definition: Resolvierungsiteration

Sei F eine Klauselmenge. Wir definieren:

- $\mathrm{Res}(F) := F \cup \{R \mid R$ ist Resolvente von 2 Klauseln in $F\}$
- $\mathrm{Res}^0(F) := F$
- $\mathrm{Res}^{n+1}(F) = \mathrm{Res}\left(\mathrm{Res}^n(F)\right)$
- $\mathrm{Res}^*(F) = \bigcup_{n \geq 0} \left(\mathrm{Res}^n(F)\right)$

Beispiel

Bestimme $Res^1(F)$ für $F = \{\{A, \neg B, C\}, \{B, C\}, \{\neg A, C\}, \{B, \neg C\}, \{\neg C\}\}$ und $F \cup \{\{A, C\}, \{\neg B, C\}, \{A, C, \neg C\}, \{A, B, \neg B\}, \{A, \neg B\}\}$.

Frage: Wieviele verschiedene Klauseln aus n Atomen kann es geben?
Antwort: 4^n und damit endlich viele

Satz: Resolutionssatz der Aussagenlogik

Eine Klauselmenge F ist unerfüllbar genau dann, wenn $\square \in \mathrm{Res}^*(F)$

Beweis

...das nächste Mal...

Algorithmus zum Test der Erfüllbarkeit einer Formel in Klauselform

(auf der Basis des Resolutionssatzes)

- Eingabe: Formel F in Klauselform
- Algorithmus:
  REPEAT
  G:=F;

F:=Res(F)
UNTIL ($\square \in F$) OR ($F = G$);
RETURN $\square \in F$

- Ausgabe:
  - true: die Formelmenge ist nicht erfüllbar
  - false: die Formelmenge ist erfüllbar

Definition: Deduktion, Herleitung, Beweis

Eine **Deduktion** („**Herleitung**“, „**Beweis**“) der leeren Klausel aus einer Klauselmenge F ist eine Folge $(K_1, \ldots, K_n)$ von Klauseln, sodass

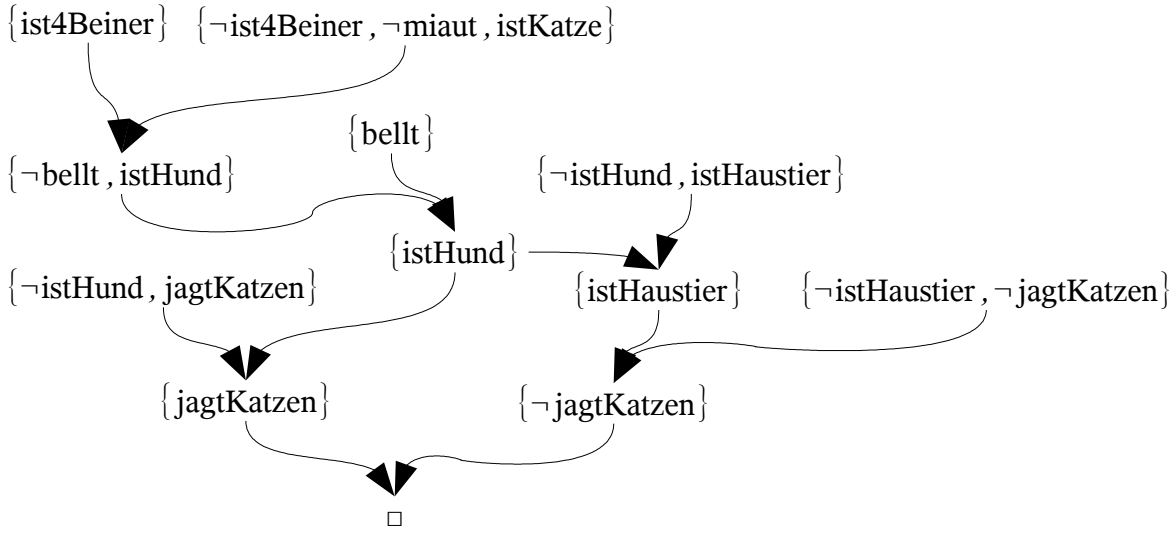
1. $K_m = \square$
2. $\forall\left(i \in (\mathbb{N} \cap [1, m])\right)$:
   - Entweder: $K_i \in F$
   - oder: K_i ist Resolvente von K_j, K_h mit $(j < i) \wedge (h < i)$

Beispiele

1. $(\text{4-Beiner} \wedge \text{miaut}) \Rightarrow \text{Katze}$	entspricht	$\{\neg \text{ist4Beiner}, \neg \text{miaut}, \text{istKatze}\}$
2. $\text{4-Beiner} \wedge \text{bellt} \Rightarrow \text{Hund}$	entspricht	$\{\neg \text{4-Beiner}, \neg \text{bellt}, \text{Hund}\}$
3. $\text{Katze} \Rightarrow \text{Haustier}$	entspricht	$\{\neg \text{istKatze}, \text{istHaustier}\}$
4. $\text{istHund} \Rightarrow \text{istHaustier}$	entspricht	$\{\neg \text{istHund}, \text{istHaustier}\}$
5. $\text{istKatze} \Rightarrow \text{jagtMäuse}$	entspricht	$\{\neg \text{istKatze}, \text{jagtMäuse}\}$
6. $\text{istHund} \Rightarrow \text{jagtKatzen}$	entspricht	$\{\neg \text{istHund}, \text{jagtKatzen}\}$
7. ist4Beiner	entspricht	$\{\text{ist4Beiner}\}$
8. bellt	entspricht	$\{\text{bellt}\}$
Fragestellung		
9. $\neg \text{istHaustier} \vee \neg \text{jagtKatzen}$	entspricht	$\{\neg \text{istHaustier}, \neg \text{jagtKatzen}\}$
Konklusionen		
10.aus 2,7 folgt:		$\{\neg \text{bellt}, \text{istHund}\}$
11.aus 10,8 folgt:		$\{\text{istHund}\}$
12.aus 11,4 folgt:		$\{\text{istHaustier}\}$
13.aus 12,9 folgt:		$\{\neg \text{jagtKatzen}\}$
14.aus 11,6 folgt:		$\{\text{jagtKatzen}\}$
15.aus 13,14 folgt:		$\square$

Die Frage kann also mit „nein“ beantwortet werden.

Resolutionsgraph



Satz: Resolutionssatz der Aussagenlogik

Eine Klauselmenge F ist unerfüllbar genau dann, wenn $\square \in \mathrm{Res}^*(F)$

Beweis

Teilbeweis

Die Rückrichtung $\Leftarrow$ folgt aus dem Resolutionslemma: $\forall(n):\left(F \equiv Res^n(F)\right)$. $\forall(k):\left(\left(\square \in Res^k(F)\right) \Rightarrow \left(Res^k(F).istUnerfüllbar()\right)\right) \Rightarrow F.istUnerfüllbar()$

Teilbeweis

für die Hinrichtung $\Rightarrow$: Sei F unerfüllbar. Ist F unendlich, so muss es wegen der Kompaktheit eine endliche Teilmenge von F geben, die unerfüllbar ist. Wir können uns deshalb auf endliche Formelmengen beschränken.
Wir zeigen $\square \in Res^*(F)$ durch Induktion über Anzahl n der in F vorkommenden Atome:

Induktionsanfang

$n=0$. Dann ist $F=\{\square\}$ und damit $\square \in Res^*(F)$

Induktionsschritt

Sei F eine Klauselmenge mit den Atomen $A_1,\ldots,A_{n+1}$. Wir definieren aus F zwei Klauselmengen F_1 und F_0 wie folgt:

- F_0 entsteht aus F durch Streichen jedes Vorkommen von A_{n+1} in einer Klausel sowie duch Streichen der Klauseln, in denen $\neg A_{n+1}$ vorkommt. (Belegung von A_{n+1} mit 0 fixieren)
- F_1 entsteht aus F durch Streichen jedes Vorkommen von $\neg A_{n+1}$ in einer Klausel sowie

durch Streichen der Klauseln, in denen A_{n+1} vorkommt. (Belegung von A_{n+1} mit 1 fixieren)

F_0 und F_1 sind unerfüllbar. Wäre etwa F_0 erfüllbar, so könnte man aus Modell M für F_0 ein Modell für F konstruieren. (A_{n+1} zu 0 auswerten, alle anderen Atome wie in M).

F_0 und F_1 haben höchstens n Atome. Damit ist nach Induktionsvoraussetzung $\square$ aus F_0 und aus F_1 herleitbar. Benutzt man in den jeweiligen Herleitungen für $\square$ die ursprünglichen Klauseln aus F, so entstehen Beweise für $\square$ oder A_{n+1} oder $\neg A_{n+1}$. In höchstens einem weiteren Schritt wird $\square$ hergeleitet.

Beispiel

$F=\{\{A, B, \neg C\}, \{\neg A, D\}\}$

- $A_{n+1}=A$
  $F_0=\{\{B, \neg C\}\}$
  $F_1=\{\{D\}\}$

Davis-Putnam-Verfahren

(nicht „Putman“)
Das Verfahren ist ein Erfüllbarkeitstest für die Formel F in Klauselform.

Definition: reduzierte Klauselmenge

Sei F eine Menge von Klauseln, L ein Literal. Die um L reduzierte Klauselmenge F_L entsteht aus F durch:
1. Streichen aller Klauseln, die L enthalten.
2. Streichen des Komplements von L aus den verbleibenden Klauseln.

Beispiel

$F=\{\{A, B\}, \{C, \neg A\}, \{\neg B\}\}$

- Reduzieren um A :
  - $F_A=\{\{C\}, \{\neg B\}\}$
- Reduzieren um $\neg B$:
  - $F_{\neg B}=\{\{A\}, \{C, \neg A\}\}$

Anmerkung

Jedes Modell M von F, das L zu wahr auswertet, ist Modell von F_L.

Beispiel

Für $F=\{\{A, B\}, \{C, \neg A\}, \{\neg B\}\}$ gilt: ($M(A)=1$, $M(B)=0$, $M(C)=1$).
Für die reduzierte Formel $F_A=\{\{C\}, \{\neg B\}\}$ gilt: ($M(B)=0$, $M(C)=1$).

Anmerkung

Zu jedem Modell M' von F_L gibt es ein Modell M von F, das die Atome in F_L gleich auswertet wie M' ($M(L)=1$, $M(A)=M'(A)$ für Atome, die in F_L vorkommen)

Daraus folgt:
F_L ist erfüllbar genau dann, wenn F ein Modell hat, was L zu wahr auswertet.

Beobachtungen

1. Wenn $\square \in F$, dann ist F unerfüllbar.
2. Wenn eine Einerklausel (Klausel der Form $\{L\}$) in F vorkommt, dann ist F erfüllbar genau dann, wenn F_L erfüllbar ist.
3. Sei L ein beliebiges Literal in F.
   $F.istErfüllbar() \Leftrightarrow \left(\left(F_L\right).istErfüllbar() \vee \left(F_{\neg L}\right).istErfülbbar() \right)$

Algorithmus: Erfüllbarkeitstest

```
boolean satisfiable(Klauselmenge F) {
    if ( S={} ) {
        return true;
    }
    if ( □∈S ) {
        return false;
    }
    if ( ∃(L):({L}∈S) ) {
        return satisfiable( S_L );
    } else {
        Literal L = pickLiteral( S );

        return (satisfiable( S_L )||satisfiable( S_¬L ));
    }
}
```

Beispiel

$$\begin{aligned} \text{erfüllbar}\left(\{\{A,B\},\{C,\neg A\},\{\neg B\}\}\right) &= \text{erfüllbar}\left(\{\{A\},\{C,\neg A\}\}\right) \\ &= \text{erfüllbar}\left(\{\{C\}\}\right) \\ &= \text{erfüllbar}\left(\{\}\right) \\ &= 1 \end{aligned}$$

Beispiel

$$\begin{aligned} \text{erfüllbar}\left(\{\{A,B\},\{\neg A,B\},\{A,\neg B\}\}\right) &= \text{erfüllbar}\left(\{\{B\}\}\right) \vee \text{erfüllbar}\left(\{\{B\},\{\neg B\}\}\right) \\ &= \text{erfüllbar}\left(\{\}\right) \vee \text{erfüllbar}\left(\{\square\}\right) \\ &= 1 \vee 0 \\ &= 1 \end{aligned}$$

Noch zur Resolution

Einschränkungen des Resolutionsverfahrens (diese Resolutionen sind effizienter für ihre Spezialfälle als die allgemeine Resolution)

- Unit-Resolution: Spezialfall der Resolution, wo in jedem Resolutionsschritt mindestens eine resolvierte Klausel 1-elementig ist.
- Input-Resolution: Spezialfall der Resolution, wo in jedem Resolutionsschritt mindestens eine resolvierte Klausel Eingabeklausel aus F ist.

Satz

Unit-Resolution und Input-Resolution sind vollständig für Hornklauseln. (Das heißt: Falls F eine Hornklauselmenge ist, so ist $\square$ ableitbar genau dann, wenn $F.istUnerfüllbar()$.)

Gegenbeispiel

für den allgemeinen Fall (Unit):
$\{\{\neg A, \neg B\}, \{\neg A, B\}, \{A, \neg B\}, \{A, B\}\}$ ist unerfüllbar, aber keine Unit-Resolvierung möglich.

Datum: 14.05.2003

Tableauverfahren

Ein Tableau für eine Formel F ist ein Baum, dessen Knoten mit Formeln markiert sind. Die Wurzel ist mit F markiert.

- Der Pfad von der Wurzel zum Blatt repräsentiert eine Konjunktion der Formeln an Knoten [?]
- der gesamte Baum repräsentiert Disjunktionen dieser Konjunktionen.

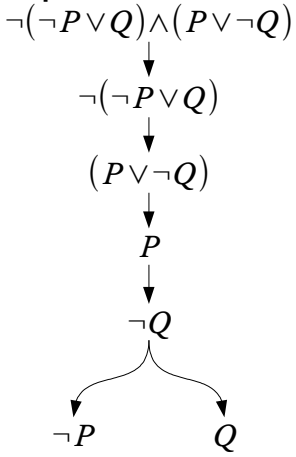
Die Grundidee ist: F ist unerfüllbar, wenn für jeden Pfad von der Wurzel zu einem Blatt gilt: die entsprechende Konjunkion ist unerfüllbar.

Alternative Grundidee: Wenn die Formel erfüllbar sein soll, dann müssen bestimmte Teilaussagen auch erfüllbar sein.

Erzeugungsregeln für Tableaus

1. Wenn im Pfad $\neg\neg H$ vorkommt, dann erweitere den Pfad um H.
2. Wenn im Pfad $G_1 \wedge G_2$ vorkommt, dann erweitere den Pfad um G_1 und anschließend um G_2
3. Wenn im Pfad $\neg(G_1 \vee G_2)$ vorkommt, dann erweitere den Pfad um $\neg G_1$ und anschließend um $\neg G_2$
4. Wenn im Pfad $\neg(G_1 \wedge G_2)$ vorkommt, dann verzweige, sodass der linke Nachfolger $\neg G_1$ und der rechte Nachfolger $\neg G_2$ ist.
5. Wenn im Pfad $G_1 \vee G_2$ vorkommt, dann verzweige, sodass der linke Nachfolger G_1 und der rechte Nachfolger G_2 ist.

Beispiel



Wenn $\neg(\neg P \vee Q) \wedge (P \vee \neg Q)$ gelten soll, dann muss $\neg(\neg P \vee Q)$ und $(P \vee \neg Q)$ gelten. Wenn $\neg(\neg P \vee Q)$ gelten soll, dann muss P und $\neg Q$ gelten. Wenn $(P \vee \neg Q)$ gelten soll, dann muss $\neg P$ oder Q gelten. Es kann aber nicht $\neg P$ gelten, wenn gleichzeitig P gelten soll. Es kann auch nicht Q gelten, wenn gleichzeitig $\neg Q$ gelten soll. Damit kann $\neg(\neg P \vee Q) \wedge (P \vee \neg Q)$ nicht gelten.

Definition: Tableau

Die Menge der **Tableau**s für eine Formel F ist induktiv wie folgt definiert:

1. Der Baum, der aus einem mit F markierten Knoten besteht, ist ein Tableu für F.
2. Wenn T ein Tableau für F ist und T' aus T durch Anwendung einer Erzeugungsregel ensteht, so ist T' ein Tableau für F.

Definition: Ast

Ein **Ast** eines Tableaus ist ein Pfad von der Wurzel zu einem Blatt

Definition: abgeschlossen

Ein Ast heißt **abgeschlossen**, wenn in ihm eine Formel und ihre Negation vorkommt.

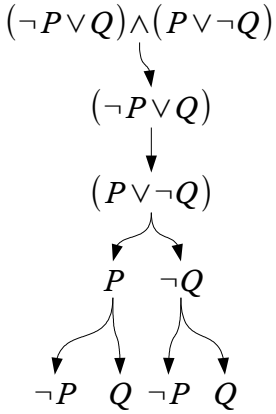
Definition: abgeschlossen

Ein Tableau heißt **abgeschlossen**, wenn jeder Ast abgeschlossen ist.

Satz

F ist unerfüllbar genau dann, wenn es ein abgeschlossenes Tableau für F gibt.

Beispiel



2. Prädikatenlogik

In der Aussagenlogik interessiert die Struktur der Sätze nur, insofern sie durch „und“, „oder“, „nicht“ entsteht.

Beispiel

- Verbal:
  - Aussagen:
    - Alle Informatiker sind schlau.
    - Der Vater von Peter ist Informatiker.
  - Schlussfolgerung:
    - Der Vater von Peter ist schlau.
- Formal:
  - $\forall(x):\big(Informatiker(x)\Rightarrow schlau(x)\big)$
  - $Informatiker(Vater(Peter))$
  - $schlau(Vater(Peter))$

Für die Definition von Formeln muss festgelegt werden, welche Variablen welche Prädikaten und welche Funktionssymbole verwendet werden können (bei letzterem zusätzlich die Anzahl der Argumente).
Konstanten können als nullstellige Funktionssymbole aufgefasst werden.

Definition: Signatur

Eine **Signatur** $\Sigma=(var, pre, fun, s)$ besteht aus:
1. einer Menge von Variablen var,
2. einer Menge von Prädikatensymbolen pre,
3. einer Menge von Funktionssymbolen fun sowie
4. einer Stelligkeitsfunktion $s:(pre\cup fun)\rightarrow\mathbb{N}$

Hierbei sind var, pre, fun paarweise disjunkt.

Definition: Syntax der Prädikatenlogik

Sei $\Sigma=(var, pre, fun, s)$ eine Signatur.
Terme (über Σ) werden induktiv wie folgt definiert:
1. Jede Variable aus var ist ein Term.
2. Falls $f\in fun$ mit $s(f)=k$ und $t_1,\ldots,t_k$ Terme sind, so ist $f(t_1,\ldots,t_k)$ ein Term.

Beispiel

- Gegeben sei:
  - $Vater\in fun$, $s(Vater)=1$ (1 Argument)
  - $Peter\in fun$, $s(Peter)=0$ (0 Argumente, also Konstante)
  - $x\in var$
- dann sind zum Beispiel Terme:
  - x
  - $Peter$
  - $Vater(Peter)$

- $Vater(x)$
- $Vater(Vater(x))$

Definition: Formel

Formeln (über Σ) sind wie folgt definiert:

1. Falls $P \in pre$ mit $s(P)=k$ und $t_1,\ldots,t_k$ Terme sind, so ist $P(t_1,\ldots,t_k)$ eine (atomare) Formel
2. Falls F eine Formel ist, so ist auch $\neg F$ eine Formel.
3. Falls F und G Formeln sind, so sind
   1. $(F \wedge G)$ und
   2. $(F \vee G)$ Formeln.
4. Falls x eine Variable ist und F eine Formel, so sind
   1. $\forall x\, F$ und
   2. $\exists x\, F$ Formeln.

Beispiel

- Sei (zusätzlich) gegeben:
  - $schlau \in pre^1$
  - $Informatiker \in pre^1$
- dann sind zum Beispiel Formeln
  - $schlau(Peter)$
  - $Informatiker(Vater(Vater(Peter)))$
  - $\neg(schlau(Peter) \vee Informatiker(Vater(Vater(Peter))))$
  - $\forall x(Informatiker(x) \Rightarrow schlau(x))$

Definition: Struktur

Eine **Struktur** $A=(U_A, I_A)$ ist ein Paar bestehend

- aus einer beliebigen nichtleeren Menge U_A („Universum“, „Domain“, „Individuenbereich“) und
- einer Abbildung I_A, für die gilt:
  1. Jedem k-stelligen Funktionssymbol f wird eine k-stellige Funktion über U_A zugeordnet.
  2. Jedem k-stelligen Prädikatensymbol P wird ein k-stelliges Prädikat über U_A zugeordnet.
  3. Jeder Variablen wird von I_A ein Element von U_A zugeordnet.

Semantik der Prädikatenlogik

Datum: 21.05.2003

Definition: Struktur

Eine **Struktur** ist ein Universum [Individuenmenge], Zuordnung von Symbolen der Signatur (Funktionssymbole, Prädikatensymbole, Variablen) zu Funktionen über das Universum, Prädikaten des Universums und Elemente des Universums.

Definition: Prädikat

Ein n-stelliges **Prädikat** über M ist eine Teilmenge von M^n.

Definition: Funktion

Eine n-stellige **Funktion** über M ist eine Abbildung $M^n \to M$

Schreibweise

Sei A eine Struktur, $A=(U_A, I_A)$.
Dann schreiben wir
- P^A statt $I_A(P)$,
- f^A statt $I_A(f)$ und
- x^A statt $I_A(x)$

Definition: Semantik

Sei F eine Formel (über Σ), A eine Struktur (über Σ). Der Wert eines Termes t in A (geschrieben als $A(t)$) ist induktiv wie folgt definiert:
1. falls $t=x$ für die Variable x, so ist $A(t)=x^A$
2. falls $t=f(t_1,\ldots,t_n)$, so ist $A(t)=f^A(A(t_1),\ldots,A(t_n))$

Der Wahrheitswert einer Formel F unter A (geschrieben als $A(F)$) ist induktiv wie folgt definiert:
1. Falls $F=P(t_1,\ldots,t_k)$, so ist
   1. $A(F)=1$, wenn $(A(t_1),\ldots,A(t_k))\in P^A$,
   2. $A(F)=0$, sonst.
2. Falls $F=\neg G$, so ist
   1. $A(F)=1$, wenn $A(G)=0$,
   2. $A(F)=0$ sonst.
3. Falls $F=(G\wedge H)$, so ist
   1. $A(F)=1$, wenn $(A(G)=1)\wedge(A(H)=1)$
   2. $A(F)=0$ sonst.
4. Falls $F=(G\vee H)$, so ist
   1. $A(F)=1$, wenn $(A(G)=1)\vee(A(H)=1)$
   2. $A(F)=0$ sonst.
5. Falls $F=\forall(x):(G)$, so ist

1. $A(F)=1$, wenn $\forall(d\in U_A):\left(A_{[x/d]}(G)=1\right)$
2. $A(F)=0$ sonst.

6. Falls $F=\exists(x):(G)$, so ist
   1. $A(F)=1$, wenn $\exists(d\in U_A):\left(A_{[x/d]}(G)=1\right)$
   2. $A(F)=0$ sonst.

Dabei bedeutet $A_{[x/d]}$ die Struktur, die mit A übereinstimmt, bis auf den Wert von $x^{A_{[x/d]}}=d$.

Definition: Modell

Falls $A(F)=1$, so heißt A **Modell** von F . (Notation: $A\vDash F$)

[Ein Modell ist **unendlich**, wenn das Universum des Modells unendlich ist.]

Definition: erfüllbar

F heißt **erfüllbar**, wenn F ein Modell besitzt.

Definition: allgemeingültig

F heißt **allgemeingültig**, falls jede Struktur F zu 1 auswertet.

Die Begriffe sind entsprechend auch für Mengen von Formeln definiert.

Eigenschaften

Wieder gilt:
- $(F).istAllgemeingültig()\Rightarrow(\neg F).istUnerfüllbar()$

Beispiel

Struktur A_1 :
- Das Universum sei $\{\text{Peter},\text{Claudia},\text{Klaus}\}$
- $P(x,y)$ soll bedeuten: x mag y
- Es gilt:
  - Peter mag Claudia
  - Claudia mag Klaus
  - Klaus mag Peter
- also gilt:
  - $P^{A_1}=\{(\text{Peter},\text{Claudia}),(\text{Claudia},\text{Klaus}),(\text{Klaus},\text{Peter})\}$

$\left(A_1(F)=1\right)\Leftrightarrow\left(\left(A_{1[x/\text{Peter}]}(\exists(y):P(x,y))=1\right)\wedge\left(A_{1[x/\text{Claudia}]}(\exists(y):P(x,y))=1\right)\wedge\left(A_{1[x/\text{Klaus}]}(\exists(y\right.\right.$

- Gibt es ein d , sodass $A_{1[x/\text{Peter}]}(P(x,y))=1$?
  - Ja, nämlich $d=\text{Claudia}$, denn $(\text{Peter},\text{Claudia})\in P^{A_1}$
- Analog gilt es für $d=\text{Peter}$ und $d=\text{Klaus}$

Damit gilt $A(F)=1$.

Beispiel

$F=\forall(x):\exists(y):(P(x,y))$

Die Struktur A_2 ist:
- Das Universum ist $\mathbb{N}$.
- $P(x,y)$ soll bedeuten: x ist größer als y
- Also: $P^{A_2}=\{(n,m)|n>m\}$
- $(A_2(F)=0)\Leftrightarrow$
  - es gibt ein d, sodass $A_{2[x/d]}(\exists(y):(P(x,y)))$ für $d=0$ gilt:
    - Es gibt kein d' mit $(d,d')\in p^{A_2}=0$

  Also $A_{2[x/0]}(\exists(y):(P(x,y)))=0$ und damit $A_2(F)=0$
  Also ist F keine Tautologie.

Bemerkung

Die Aussagenlogik ist ein Spezialfall der Prädikatenlogik. Alle Prädikatensymbole sind in diesem Fall 0-stellig, Terme erübrigen sich dann.

Bemerkung: Prädikatenlogik mit Identität

Eine häufig verwendete Erweiterung ist die Prädikatenlogik mit Identität. In diesem Fall ist
- $=$ ein 2-stelliges Prädikatensymbol
- die Standardinterpretation in allen Strukturen ist:
  - $A:=(t_1,t_2)$ ist wahr genau dann, wenn ${t_1}^A={t_2}^A$.

Die Infixnotation ist: $t_1=t_2$

Gesucht ist die Formel, die besagt:
1. dass P eine antisymmetrische Relation ist:
   $(\forall(x):\forall(y):(P(x,y)\Rightarrow\neg P(y,x)))\vee(\forall(x):\forall(y):(\neg(P(x,y)\wedge P(y,x))))$

2. f injektiv ist:
   $\forall(x):\forall(y):((f(x)=f(y))\Rightarrow(x=y))$
3. f surjektiv ist
   $\forall(x):\exists(y):(f(y)=x)$

Für jedes Modell soll gelten: Kardinalität des Universums ist nicht größer als 2:
$\forall(x):\forall(y):\forall(z):((x=y)\vee(x=z)\vee(y=z))$

Datum: 28.05.2003

heute gehalten von Rolf Hartwig.

Wiederholung

Beispiel

„Es ist nicht alls Gold, was glänzt“ bedeutet in der Prädikatenlogik $\neg\forall(x):\big(glänzt(x)\Rightarrow istGold(x)\big)$.

Allgemein ist die Struktur der logischen Formel $\neg\forall(x):\big(b(x)\Rightarrow A(x)\big)$.
Es gibt auch andere Interpretationen:

- Vögel
  - $B(x)=istVogel(x)$
  - $A(x)=kannFliegen(x)$
- Teilbare Zahlen
  - $B(x)=istDurch2Teilbar(x)$
  - $A(x)=istDurch3Teilbar(x)$

Beispiel

Gegeben sei die Formel $\exists(x):P\big(f(x,x)\big)$

- mögliche Interpretationen
  - f ist die Addition und P entscheidet, ob das Argument eine gerade Zahl ist.
  - f ist die Multiplikation und P entscheidet, ob das Argument gleich 0 ist.

Semantik

Semantik bedeutet Interpretation der Formeln.
Gegeben sei eine Signatur Σ. Wir legen den Bereich U fest („Individuenbereich“, „Universum“).
Die Funktion „Interpretation“ $I(x)$ ist überladen:

- Sie bildet bei Variablen als Parameter in U ab:
  $I(x)=x^A\in U$
- Sie bildet bei Funktionen als Parameter in Funktionen über U ab:
  $I(f)=f^A\subseteq U\times U\times\ldots\times U\rightarrow U$
- Sie bildet bei Prädikaten als Parameter in Relationen über U ab:
  $I(P)=P^A\subseteq U\times U\times\ldots\times U$

Ein Paar $A=\big(U_A,I_A\big)$ heißt algebraische Struktur in der Mathematik.

Analog wie in der Aussagenlogik wird die Interpretation A der Symbole erweitert zu einer Interpretation der

- Terme: $A(t)$ und der
- Formeln: $A(F)$

Wie wird interpretiert?

- Ist F der Form $P(t_1,\ldots,t_k)$, dann ist
  - $A(F)=\begin{cases}1 & \text{falls } P^A\big(A(t_1),\ldots,A(t_k)\big)\\ 0 & \text{sonst}\end{cases}$
- Ist F der Form $\forall x\, G$
  - $A(F)=\begin{cases}1 & \text{falls für alle } x-\text{Variationen } A' \text{ von } A \text{ gilt: } A'(G)=1\\ 0 & \text{sonst}\end{cases}$
  - Eine x-Variation A' von A ist so definiert, dass sie mit A bis auf die Interpretation von x überein stimmt.
  - $x^{A'}=I_A(x)=d\in U_A$
  - [... abgewischt ...]

Definition: Modell

Eine Struktur A ist **Modell** von der Formel F genau dann, wenn $A(F)=1$.

Definition: Folgerung

$X\models F$ (F **folgt** aus X, wobei X eine Formelmenge ist) genau dann, wenn jedes Modell von X auch ein Modell von F ist.

Definition: Äquivalenz

$F\equiv G$ (F ist **äquivalent** zu G) genau dann, wenn für alle Strukturen A gilt: $A(F)=A(G)$

Satz

$(F\equiv G)\Leftrightarrow\big(\big(A\models F\big)\Leftrightarrow\big(A\models G\big)\big)$

Lemma

Sei F eine Formel, die genau die Variablen $x_1, x_2, \ldots, x_k$ frei enthält. Dann gilt:

1. $\big(\models F\big)\Leftrightarrow\big(\models \forall(x_1):\forall(x_2):\ldots:\forall(x_k):(F)\big)$
2. $F.istUnerfüllbar()\Leftrightarrow\big(\exists(x_1):\exists(x_2):\ldots:\exists(x_k):(F)\big).istErfüllbar()$

Beweis

Klar nach Definition.

2.3 Äquivalente Umformungen und Normalformen

Satz

Es ist offensichtlich: Wenn F und G aussagenlogische Formeln sind, und man F' aus F und G' aus G erhält dadurch, dass man für die aussagenlogischen Variablen in F beziehungsweise G beliebige prädikatenlogische Formeln einsetzt, so gilt:

- Wenn $F\equiv G$ (in der Aussagenlogik), dann gilt $F'\equiv G'$ (in der Prädikatenlogik)

Beispiel

$\neg(A \wedge B) \equiv \neg A \vee \neg B$

Beispiel

$\neg\big(\forall(x):(P(x)) \wedge \exists(y):(Q(y,x))\big) \equiv \neg A(x):(P(x)) \vee \neg\exists(y):(Q(y,x))$

Satz

Für beliebige Formeln F und G gilt:
$\neg\forall(x):(F) \equiv \exists(x):(\neg F)$

Satz

Für beliebige Formeln F und G gilt:
$\neg\exists(x):(F) \equiv \forall(x):(\neg F)$

Satz

Für beliebige Formeln F und G gilt, falls x in G nicht frei vorkommt:
$(\forall(x):(F) \wedge G) \equiv \forall(x):(F \wedge G)$

Satz

Für beliebige Formeln F und G gilt, falls x in G nicht frei vorkommt:
$(\forall(x):(F) \vee G) \equiv \forall(x):(F \vee G)$

Satz

Für beliebige Formeln F und G gilt, falls x in G nicht frei vorkommt:
$(\exists(x):(F) \wedge G) \equiv \exists(x):(F \wedge G)$

Satz

Für beliebige Formeln F und G gilt, falls x in G nicht frei vorkommt:
$(\exists(x):(F) \vee G) \equiv \exists(x):(F \vee G)$

Satz

Für beliebige Formeln F und G gilt:
$(\forall(x):(F) \wedge \forall(x):(G)) \equiv \forall(x):(F \wedge G)$

Satz

Für beliebige Formeln F und G gilt:
$(\forall(x):(F) \vee \forall(x):(G)) \equiv \forall(x):(F \vee G)$

Satz

Für beliebige Formeln F gilt:
$\forall(x):\forall(y):(F) \equiv \forall(y):\forall(x):(F)$

Satz

Für beliebige Formeln F gilt:
$\exists(x):\exists(y):(F)\equiv\exists(y):\exists(x):(F)$

Beweis

Obige Sätze werden mit Hilfe der Definition der Interpretation bewiesen.

Bemerkung

Nicht äquivalent sind:
- $\forall(x):(F)\vee\forall(x):(G)$ und $\forall(x):(F\vee G)$
  - Beispiel:
    - $F=P(x)$ mit $(P^A(x)=1)\Leftrightarrow(x.istGeradeZahl())$ und
      $G=Q(x)$ mit $(Q^A(x)=1)\Leftrightarrow(x.istUngeradeZahl())$
- $\exists(x):(F)\wedge\exists(x):(G)$ und $\exists(x):(F\wedge G)$
- $\exists(x):\forall(y):(F)$ und $\forall(y):\exists(x):(F)$

Lemma

Für beliebige Formeln F und G gilt:
$(F\equiv G)\Leftrightarrow(\vDash(F\Leftrightarrow G))$

Beweis

Der Beweis ist offensichtlich.

Bemerkung

$(\vDash(F\Leftrightarrow G))\Leftrightarrow((\vDash(F\Rightarrow G))\wedge(\vDash(G\Rightarrow F)))$

Bemerkung

Wenn $\neg(F\equiv G)$, ist die Frage immer noch sinnvoll, ob vielleicht
- $\vDash(F\Rightarrow G)$ oder
- $\vDash(G\Rightarrow F)$

Datum: 04.06.2003

Heute gelesen wieder von Gerhard Brewka.

Definition: Substitution

Sei
- F eine Formel,
- x eine Variable und
- t eine Term

$F[x/t]$ bezeichnet die Formel, die man aus F erhält, indem jedes freie Vorkommen von x in F durch t ersetzt wird. Eine solche Ersetzung heißt **Substitution**.

Definition: freies Vorkommen

Eine Variable kommt **frei vor**, wenn sie nicht durch einen Quantor gebunden ist.

Beispiel

Gegeben sei die Formel $\forall(x):\exists(y):\big(P(x,y)\wedge\forall(z):\big(Q(z)\big)\big)$. Dann sind die Variablen x, y und z gebunden.

Beispiel

Gegeben sei die Formel $F=„P(x)\wedge\forall(x):\big(Q(x)\big)“$. Dann ist
$F[x/a]=„P(a)\wedge\forall(x):\big(Q(x)\big)“$

Lemma

Sei
- $F=Q„(x):(G)“$ eine Formel mit dem Quantor $Q\in\{\forall,\exists\}$,
- y eine Variable, die nicht in G vorkommt.

Dann gilt: $Q„(x):(G)“=Q„(y):(G)“[x/y]$

Beispiel

$\forall(x):\big(P(x)\vee Q(y)\big)\equiv\forall(y):\big(P(y)\vee Q(y)\big)$

Definition: bereinigt

Eine Formel heißt **bereinigt**,
- wenn es keine Variable gibt, die sowohl gebunden wie frei auftritt, und
- wenn alle Quantoren verschiedene Variablen haben.

Beispiel

$F_{unbereinigt}=„\forall(x):\exists(y):\big(P(x,f(y))\wedge\forall(y):\big(Q(x,y)\vee R(x)\big)\big)“$
$F_{bereinigt}=„\forall(u):\exists(v):\big(P(u,f(v))\wedge\forall(y):\big(Q(x,y)\vee R(x)\big)\big)“$

Definition: Pränex-Normalform

Eine Formel ist in **Pränex**(normal)**form** (PNF), falls sie die Gestalt
$F = Q_1 „ (y_1) : “ Q_2 „ (y_2) : “ \ldots „ : “ Q_n „ (y_n) : “ G$ besitzt.
Hierbei sind $\forall (i) : (Q_i \in \{\forall, \exists\})$ Quantoren. G hat keine Quantoren.

Definition: Matrix

Die **Matrix** einer Formel in Pränex-Normalform ist der Teil der Formel, der den Quantoren folgt.

Satz

Für jede Formel F gibt es eine äquivalente Formel F' in Pränex-Normalform.

Beweis

Induktion über den Formelaufbau.

Induktionsanfang

Voraussetzung: F ist atomar.
Behauptung: F ist in Pränex-Normalform.
Beweis: F ist in Pränex-Normalform.

Induktionsschritt

- 1. Fall: $F = „ \neg “ F_1$ wobei $(F_1).istInPränexNormalform()$
  $G_1 = Q_1 „ (y_1) : “ \ldots Q_n „ (y_n) : “ G'$
  Die Pränex-Normalform von F_1 existiert nach Induktionsvoraussetzung.
  Dann gilt: $F = -Q_1 „ (y_1) : “ \ldots -Q_n „ (y_n) : \neg “ G'$, wobei
  - $-\exists = \forall$
  - $-\forall = \exists$

  - Beispiel
    - [...abgewischt...]
- 2. Fall: $F = „ (F_1 “ op „ F_2) “$ mit $op \in \{\wedge, \vee\}$.
  Seien $G_1 = Q_{11} „ (y_{11}) : “ \ldots Q_{1n} „ (y_{12}) : “ G_1'$ mit $F_1 \equiv G_1$ und
  $G_2 = Q_{21} „ (y_{21}) : “ \ldots Q_{2m} „ (y_{2m}) : “ G_2'$ mit $F_2 \equiv G_2$ Formeln in Pränex-Normalform.
  Weiter gelte, dass die gebundenen Variablen in G_1 und G_2 disjunkt sind (das kann durch Umbennenen erreicht werden). Dann ist F äquivalent zu
  $Q_{11} „ (y_{11}) : “ \ldots Q_{1j} „ (y_{1n}) : “ Q_{21} „ (y_{21}) : “ \ldots Q_{2m} „ (y_{2m}) : (“ G_1' op G_2' „) “$
  - Beispiel
    - $F = F_1 „ \wedge “ F_2$
    - $(F_1) : getPränexNormalform() = „ \forall (x) : \exists (y) : (P(x, y)) “$
    - $(F_2) : getPränexNormalform() = „ \forall (z) : R(z, z) “$
    - $(F) : getPränexNormalform() = „ \forall (x) : \exists (y) : \forall (z) : (P(x, y) \wedge R(z, z)) “$
- 3. Fall: $F = Q „ (x) : “ F_1$ mit $Q \in \{\exists, \forall\}$

$Q_1 „ (y_1) : “ \ldots Q_n „ (y_n) : “ F_1'$ sei die Pränex-Normalform von F_1. Durch Umbenennen kann x verschieden gemacht werden von allen y_i. Damit ist F äquivalent zu $Q „ (x) : “ Q_1 „ (x_1) : “ \ldots Q_n „ (y_n) : “ F_1'$.

Bemerkung

Im Beweis steht implizit der Algorithmus
1. Bereinigen.
2. Negationen wandern hinter die Quantoren.
3. Quantoren wandern nach vorne.

Beispiel

$F = „ \neg \exists (x) : (R(x)) \vee (P(a) \wedge \forall (x) : (Q(a, y))) “$
$F.bereinige() = „ \neg \exists (x) : (R(x)) \vee (P(a) \wedge \forall (y) : Q(a, y)) “$
$F.bereinige().pränexNormalformSchritt() = „ \forall (x) : (\neg R(x)) \vee \forall (y) : (P(a) \wedge Q(a, y)) “$
$F.konvertiereInPränexNormalform() = „ \forall (x) : \forall (y) : (\neg R(x) \vee (P(a) \wedge Q(a, y))) “$

Definition: Skolem-Form

Sei $F = Q_1 „ (y_1) : “ \ldots Q_n „ (y_n) : “ F_1$ in Pränex-Normalform. Die **Skolemform** von F entsteht aus F durch:
1. Ersetzen jeder existenzquantifizierten Variable x in F_1 durch einen Funktionsterm $f(y_1, \ldots, y_k)$: Dabei ist f ein neues, nicht in F vorkommendes Funktionssymbol. Die Stelligkeit von f ist die Anzahl der in F links vom Existenzquantor von x vorkommenden Allquantoren. Die Argumente sind die Variablen dieser Allquantoren von links nach rechts.
2. Streichen aller Existenzquantoren:
   $\forall (x) : \exists (y) : \exists (z) : (S(y, x) \wedge P(z, x))$ ist nicht äquivalent zu
   $\forall (x) : (S(sk_1(x), x) \wedge P(sk_2(x), x))$

Satz

Sei F eine Formel in Pränex-Normalform. Dann gilt:
$F.istErfüllbar() \Leftrightarrow F.skolemform().istErfüllbar()$

Gegeben sei eine Formel F
1. Bereinigen von F durch Umbennenen von Variablen
2. Binden freier Variablen durch Vorstellen von Existenzquantoren (erfüllbarkeits[?])
3. Herstellen der Pränex-Normalform
4. Herstellen der Skolemform
5. Umformung der Matrix in Konjunktive Normalform.
6. Notieren in Klauselform

Beispiel

$\neg \exists (x) : (P(x, z) \vee \forall (y) : Q(x, y)) \vee \forall (y) : (P(y, z))$
$\neg \exists (x) : (P(x, z) \vee \forall (y) : Q(x, y)) \vee \forall (w) : (P(w, z))$ (bereinigt)
$\exists (z) : (\neg \exists (x) : (P(x, z) \vee \forall (y) : (Q(x, y))))$

$$\exists(z):\Big(\forall(x):\big(\neg P(x,z)\wedge\neg\forall(y):\big(Q(x,y)\big)\big)\Big)$$

Datum: 18.06.2003

[Folie mit dem Algorithmus zur Transformation von prädikatenlogischen Formeln in prädikatenlogische Klauselmengen]

Unentscheidbarkeit

In der Aussagenlogik gibt es für jede Formel eine endliche Menge von Belegungen.
In der Prädikatenlogik ist eine Beschränkung auf eine endliche Menge von Strukturen nicht möglich. Sogar Strukturen mit unendlichem Universum zu betrachten. [?]

Beispiel

$$\forall(x):\big(P\big(x,f(x)\big)\big)\wedge\forall(y):\big(\neg P\big(y,y\big)\big)\wedge\forall(u):\forall(v):\forall(w):\big(\big(P\big(u,v\big)\wedge P\big(v,w\big)\big)\Rightarrow P\big(u,w\big)\big)$$

Hierfür gibt es nur unendliche Modelle, etwa

- $A=\big(U_A,I_A\big)$ mit
  - $U_A=\{0,1,2,\ldots\}$
  - $P^A=\{(m,n)\,|\,m<n\}$
  - $f^A(n)=n+1$

Kann man entscheiden, ob eine Formel F allgemeingültig oder erfüllbar ist?

Definition: entscheidbar

Ein Problem heißt **entscheidbar**, wenn es einen Algorithmus gibt, der bei Eingabe einer Instanz des Problems (hier eine Formel F)
- mit Ausgabe „ja“ terminiert, falls die Eingabe zu einer gesuchten Teilklasse gehört und
- ansonsten mit Ausgabe „nein“ terminiert.

Definition: semi-entscheidbar

Ein Problem heißt **semi-entscheidbar**, wenn es einen Algorithmus gibt, der bei Eingabe einer Instanz des Problems (hier eine Formel F)
- mit Ausgabe „ja“ terminiert, falls die Eingabe zu einer gesuchten Teilklasse gehört und
- ansonsten nicht terminiert.

Satz: Church-sche These

Das Gültigkeitsproblem der Prädikatenlogik ist unentscheidbar.

Korollar

Das Erfüllbarkeitsproblem der Prädikatenlogik ist unentscheidbar.

Beweis

Wäre Erfüllbarkeit entscheidbar, so auch Gültigkeit, denn F ist gültig genau dann, wenn $\neg F$ nicht erfüllbar ist.

Korollar

Folgerbarkeit der Prädikatenlogik ist unentscheidbar.

All diese Probleme sind semi-entscheidbar.

Herbrand-Theorie

Die Definition von Strukturen lässt beliebige Mengen als Universum zu.
Algorithmische Suche nach Modellen kann auf kanonische Weise erfolgen.

Definition: Herbrand-Universum

Das **Herbrand-Universum** $D(F)$ einer geschlossenen Formel in Skolemform ist die Menge aller variablenfreien Terme, die aus in F vorkommenden Symbolen gebildet werden können. Falls in F keine Konstante vorkommt, kann als Symbol zusätzlich eine neue Kontante a verwendet werden.

Induktive Definition:
$D(F)$ ist die kleinste Menge, sodass:
1. Alle Konstanten in F sind in $D(F)$. Falls keine Konstante in F vorkommt, so ist die Konstante $a \in D(F)$.
2. Falls f mit Stelligkeit n in F vorkommt und $t_1, \ldots, t_n \in D(F)$, so auch $f(t_1, \ldots, t_n) \in D(F)$.

Beispiel

Für die Formel $\forall(x): \left(\neg P\left(a, f(x)\right) \vee Q\left(g(b)\right)\right)$ ist
$D(F) = \left\{a, b, f(a), f(b), g(a), g(b), f(f(a)), \ldots, f(g(b)), \ldots, g(f(a)), \ldots, g(g(b)), \ldots\right\}$
Das Herbrand-Universum ist unendlich, falls ein Funktionssymbol vorkommt.

Definition: Herbrand-Struktur

Sei F eine geschlossene Formel in Skolemform. Eine Strukur $A = \left(U_A, I_A\right)$ heißt Herbrand-Struktur von F, falls gilt:
1. $U_A = D(F)$
2. für jedes n-stellige, in F vorkommende Funktionssymbol f und $t_1, \ldots, t_n \in D(F)$ gilt: $f^A(t_1, \ldots, t_n) = f(t_1, \ldots, t_n)$

Erklärung

Die Idee ist, dass alle Grundfunktionen in jeder möglichen Kombination auf sich selbst abgebildet werden.

Eigenschaften

Herbrand-Strukturen lassen nur die Wahl von P^A für jedes Prädikatensymbol P offen.

Satz

Sei F eine geschlossene Formel in Skolemform. Dann ist F erfüllbar genau dann, wenn F ein Herbrand-Modell besitzt.
Ein Herbrand-Modell von F ist eine Herbrand-Struktur, die F zu 1 auwertet.

Definition: Herbrand-Expansion

Sei $F = „\forall(y_1):\ldots:\forall(y_n):“F'$ eine geschlossene Formel in Skolemnform. Die **Herbrand-Expansion** von F, $E(F)$, ist wie folgt definiert:
$E(F)=\{F'[y_1/t_1]\ldots[y_n/t_n] | t_1,\ldots,t_n \in D(F)\}$
(wobei $F[y/t]$ die Formel ist, die durch Ersetzen von y in F durch t entsteht)

Satz

(von Gödel, Herbrand, Skolem)
Für jede geschlossene Formel F in Skolemform gilt: $F.istErfüllbar() \Leftrightarrow E(F).istErfüllbar()$

Hat $E(F)$ keine Quantoren und keine Variablen, dann können wir zurück zur Aussagenlogik gehen.
Daraus, und aus dem Endlichkeitssatz der Aussagenlogik folgt:

Satz

(von Herbrand)
Für jede geschlossene Formel F in Skolemform gilt:
F ist unerfüllbar genau dann, wenn es eine endliche Teilmenge von $E(F)$ gibt, die unerfüllbar ist.

Algorithmus von Gilmore

(Semi-Entscheidungsverfahren für Unerfüllbarkeit:)

Sei $E(F)=\{F_1, F_2, F_3, \ldots\}$.
Eingabe sei: die geschlossene Formel F in Skolemform.

```
n = 0;
do {
   n++;
} while ( {F_1,...,F_n} .istErfüllbar())
return „unerfüllbar“;
```

Satz

Das Unerfüllbarkeitsproblem der Prädikatenlogik ist semi-entscheidbar.

Satz

Das Gültigkeitsproblem der Prädikatenlogik ist semi-entscheidbar.

Satz

Das Folgerbarkeitsproblem der Prädikatenlogik ist semi-entscheidbar.

Datum: 25.06.2003

Zur Klausur

- Die Klausur wird von mir (Gerhard Brewka) gestellt.
- In der letzten Woche wird eine Musterklausur vorgestellt.
- Musterlösungen zu den Logik-Aufgaben werden in den Übungen vermittelt.
- Es gibt einen „abgespeckten“ Aufgabenzettel nächste Woche,
  - welcher nicht abgegeben und korrigiert wird
  - außer für Leute, die knapp unter der Punktzahl-Grenze sind.

Resolution in der Prädikatenlogik

Der Unerfüllbarkeitstest im Gilmore-Algorithmus kann durch aussagenlogische Resolution durchgeführt werden.

Grundresolutionsalgorithmus

Sei $E(F)=\{F_1, F_2, F_3, \ldots\}$
Die Eingabe ist eine geschlossene Formel F in Skolemform mit der Matrix F^* in konjunktiver Normalform.

```
i = 0;
M = {};
do {
   i++;
   M = M ∪ {F_i};
   M := Res*(M)
} while( □ ∉ M );
stoppe mit Ausgabe „unerfüllbar“;
```

Satz

Bei Eingabe einer geschlossenen Formel F mit Matrix F^* in konjunktiver Normalform stoppt obiger Algorithmus mit der Ausgabe „unerfüllbar“ genau dann, wenn $F.istUnerfüllbar()$.

Beispiel

$F=\{\{\neg P(x), \neg P(f(a)), Q(y)\}, \{P(y)\}, \{\neg P(g(b,x)), \neg Q(b)\}\}$
Diese prädikatenlogische Klauselmenge bedeutet:
$\forall(a):\forall(b):\forall(x):\forall(y):\big((\neg P(x)\vee\neg P(f(a))\vee Q(y))\wedge(P(y))\wedge(\neg P(g(b,x))\vee\neg P(b))\big)$
Daraus lässt sich schließen:

- Mittels $[x/f(a), y/b]$ zu $\{\{\neg P(f(a)), Q(b)\}\}$ (Gilt etwas für alle x, dann gilt es inbesondere für alle $f(a)$. Gilt etwas für alle y, dann gilt es auch für alle b.)
- Mittels $[y/f(a)]$ zu $\{\{P(f(a))\}\}$ (Gilt etwas für alle y, dann gilt es insbesondere für alle $f(a)$.)
- Mittels $[y/g(b,a)]$ zu $\{\{P(g(b,a))\}\}$ (Gilt etwas für alle y, dann gilt es insbesondere für

alle $g(b,a)$.)

- Mittels $[x/a]$ zu $\{\{\neg P(g(b,a)), \neg Q(b)\}\}$ (Gilt etwas für alle x, so gilt es auch für alle a.)

- Aus $\{\{\neg P(f(a)), Q(b)\}\}$ und $\{\{P(f(a))\}\}$ lässt sich $\{\{Q(b)\}\}$ schließen.
- Aus $\{\{P(g(b,a))\}\}$ und $\{\{\neg P(g(b,a)), \neg Q(b)\}\}$ lässt sich $\{\{\neg Q(b)\}\}$ schließen.
- Aus $\{\{Q(b)\}\}$ und $\{\{\neg Q(b)\}\}$ lässt sich $\{\square\}$ schließen.

Damit ist die Formel unerfüllbar.

Satz: Grundresolutionssatz

Eine Formel F in Skolemform mit Matrix F^* in konjunktiver Normalform ist unerfüllbar genau dann, wenn es eine Folge von $K_1, \ldots, K_n$ gibt, sodas

1. $K_n = \square$
2. $\forall (i \in \{1, \ldots, n\})$ gilt
   K_i ist Grundinstanz einer Klausel aus F^* oder
   K_1 ist aussagenlogische Resolvente zweier Klauseln K_a, K_b mit $(a<i) \wedge (b<i)$

Die Suche nach Grundinstanzen ist oft ineffizient. Daraus erwächst die Idee, dass die Intanziierung so zurückhaltend wie möglich gemacht werden sollte.

Beispiel

$\{\{P(x), \neg Q(g(x))\}, \{\neg P(f(y))\}\}$

- Mittels $[x/f(y)]$ lässt sich $\{\{P(f(y)), \neg Q(g(f(y)))\}\}$ schließen.
- Aus $\{\{P(x), \neg Q(g(x))\}, \{\neg P(f(y))\}\}$ und $\{\{P(f(y)), \neg Q(g(f(y)))\}\}$ lässt sich $\{\{\neg Q(g(f(y)))\}\}$ schließen.

Definition: Substitution

Eine **Substitution** $\sigma = [x_1/t_1, \ldots, x_k/t_k]$ ist eine Abbildung von Variablen x_i auf Terme t_i. Sei A ein Ausdruck (Term oder Formel). $A\sigma$ (Die Substitution σ angewendet auf A, besser geschrieben als $\sigma(A)$) bezeichnet den Ausdruck, der entsteht, indem jedes Vorkommen von x_i in A durch t_i ersetzt wird.
σ heißt Unifikator einer Menge von Ausdrücken $L = \{L_1, \ldots, L_n\}$ falls $\sigma(L_1) = \ldots = \sigma(L_n)$.

Beispiel

Gegeben sei die Formelmenge $F = \{\{P(x)\}\}$.
Dann lässt sich daraus schließen:

- Aus $\{\{P(x)\}\}$ mittels $[x/f(y)]$ (Unifikator) zu $\{\{P(f(y))\}\}$
- Aus $\{\{P(x)\}, \{P(f(y))\}\}$ mittels $\{x/f(a), y/a\} = [x/f(y)]$ zu $\{\{P(f(a))\}\}$

σ heißt allgemeinster Unifikator („most general unifier“, „MGU“) von L, falls für jeden

Unifikator σ' von L gilt:
$\sigma' = \sigma\sigma''$ für eine geeignete Substitution σ''

Satz: Unifikationssatz (Robinson)

Jede unifizierbare Menge von Literalen besitzt einen allgemeinsten Unifikator („MostGeneralUnifier“).

Unifikationsalgorithmus

- Eingabe:
  - nicht leere Literalmenge $\sigma(L) = \{\sigma(L_i) \mid L_i \in L\}$
- Code:
  - $\sigma = [\,]$
  - while ($|\sigma(L)| > 1$) {
    - Durchsuche Literale in $\sigma(L)$ von links nach rechts bis zur ersten Position, an der sich 2 Literale unterscheiden.
    - if (keines der sich unterscheidenden Symbole ist Variable) {
      - return „fail“;
    - } else {
      - Sei
        - x eine Variable der gefundenen, sich unterscheidenden Literale und
        - t der Term, der an gefundener Position im anderen Literal beginnt.
      - if (x kommt in t vor) {
        - return „fail“; /* occur check */
      - } else {
        - $\sigma = \sigma \cdot [x/t]$ (Dem Unifikator wird eine weitere Substitution hinzugefügt)
      - }
    - }
  - }
  - return σ ; /* return the most general unificator */

Beispiel

[...30 Sekunden nach Anschreiben wieder abgewischt...]
<import from=„Anne Güpner“>

- $\{\{P(\underline{a}, f(x,b))\}, \{P(\underline{y}, f(g(y),z))\}\}$. Darauf wird $\sigma = [y/a]$ angewendet.
- $\{\{P(a, f(\underline{x},b))\}, \{P(a, f(\underline{g(a)}))\}\}$. Darauf wird $\sigma = [y/a]\cdot[x/g(a)]$ angewendet.
- $\{\{P(a, f(g(a),\underline{b}))\}, \{P(a, f(g(g),\underline{z}))\}\}$. Darauf wird $\sigma = [y/a]\cdot[x/g(a)]\cdot[z/b]$ angewendet.
- $\{\{P(a, f(g(a),b))\}, \{P(a, f(g(a),b))\}\}$. Dies ist eine Einer-Menge
- $\{\{P(a, f(g(a),b))\}\}$

</import>

Bemerkung: Occur-Check

Gegeben sei die Klauselmenge $\{\{P(\underline{x}, x)\}, \{P(y, f(y))\}\}$. Wie würde man hier vorgehen? Man würde $[x/y]$ anwenden. Es würde die Klauselmenge $\{\{P(y, \underline{y})\}, \{P(y, \underline{f(y)})\}\}$ entstehen. Aber wie würde ohne Occur-Check diese Klauselmenge weiter verarbeitet werden?

Bemerkung

$[y/a]\cdot[x/g(a)]\cdot[z/b]=[y/a, x/g(a), z/b]$ aber
$[y/f(x,b)]\cdot[x/g(a)]\neq[y/f(x,b), x/g(a)]$

Datum: 02.07.2003

Definition: Resolvente, Prädikatenlogische Resolution

Seien K_1, K_2 und R Klauseln, s_1 und s_2 Substitutionen, sodass $s_1(K_1)$ und $s_2(K_2)$ keine gemeinsamen Variablen haben. R heißt **Resolvente** von K_1 und K_2, wenn gilt:

1. Es gibt Literale $L_1,\ldots,L_m$ in $s_1(K_1)$ und $L'_1,\ldots,L'_n$ in $s_2(K_2)$, sodass $\{-L_1,\ldots,-L_m,L'_1,\ldots,L'_n\}$ durch einen MostGeneralUnifier sub unifizierbar ist.
2. $R=\text{sub}\left(\left(s_1(K_1)\setminus\{L_1,\ldots,L_m\}\right)\cup\left(s_2(K_2)\setminus\{L'_1,\ldots,L'_n\}\right)\right)$

Hierbei ist $-L$ das Komplement von L:

- $(L=\neg A)\Rightarrow(-L=A)$
- $(L=A)\Rightarrow(-L=\neg A)$

Beispiel

Gegeben sei die Klauselmenge $F=\{\{P(g(y)),S(h(a,y))\},\{Q(y),\neg S(h(y,b))\}\}$

- nach Anwendung von $[y/z]$ auf die zweite Klauselmenge: $\{\{P(g(y)),S(h(a,y))\},\{Q(z),\neg S(h(z,b))\}\}$
- nach Anwendung von $\text{sub}=[z/a,y/b]$: $R=\{P(g(b)),Q(a)\}$

Definition: Resolutionsschritt

Wie in der Aussagenlogik definieren wir:
Sei F eine Klauselmenge.

- $Res(F)=F\cup\{R\,|\,R.istResolventeZweierKlauselnVon(F)\}$
- $Res^0(F)=F$
- $Res^{n+1}(F)=Res(Res^n(F))$
- $Res^*=\bigcup_{n\geq 0}(Res^n(F))$

Satz: Resolutionssatz der Prädikatenlogik

Sei F eine geschlossene Formel in Skolemform mit Matrix F^* in Klauselform. $F.istUnerfüllbar()\Leftrightarrow(\square\in Res^*(F^*))$

Beispiel

- natürlichsprachlich
  - Gegeben sei
    - Normale Vögel fliegen.
    - Pinguine fliegen nicht.
    - Strauße fliegen nicht.
    - Pinguine und Strauße fliegen.
    - Tweety ist Pinguin oder Strauß.
  - Die Frage ist:
    - Ist daraus ableitbar? „Tweety ist nicht normal.“

- formal
  - Gegeben ist:
    - $\forall(x):\big((istVogel(x)\wedge istNormal(x))\Rightarrow fliegt(x)\big)$
    - $\forall(x):\big(istStrauß(x)\Rightarrow\neg\, fliegt(x)\big)$
    - $\forall(x):\big(istPinguin(x)\Rightarrow\neg\, fliegt(x)\big)$
    - $\forall(x):\big(istPinguin(x)\Rightarrow istVogel(x)\big)$
    - $\forall(x):\big(istStrauß(x)\Rightarrow istVogel(x)\big)$
    - $istStrauß(tweety)\vee istPinguin(tweety)$
  - Das negierte Ziel ist:
    - $istNormal(tweety)$

Die Formeln in Klauselform sind:
1. $\{\neg istVogel(x)\vee\neg istNormal(x)\vee fliegt(x)\}$
2. $\{\neg istStrauß(x)\vee\neg fliegt(x)\}$
3. $\{\neg istPinguin(x)\vee\neg fliegt(x)\}$
4. $\{\neg istPinguin(x)\vee istVogel(x)\}$
5. $\{\neg istStrauß(x)\vee istVogel(x)\}$
6. $\{istStrauß(tweety)\vee istPinguin(tweety)\}$
7. $\{istNormal(tweety)\}$

Diese können nun resolviert werden

8.	$\{istStrauß(tweety)\vee istVogel(tweety)\}$	aus 6,4
9.	$\{istVogel(tweety)\}$	aus 8,5
10.	$\{istPinguin(tweety)\vee\neg fliegt(tweety)\}$	aus 6,2
11.	$\{\neg fliegt(tweety)\}$	aus 10,3
12.	$\{\neg istNormal(tweety)\vee fliegt(tweety)\}$	aus 9,1
13.	$\{\neg istNormal(tweety)\}$	aus 11,12

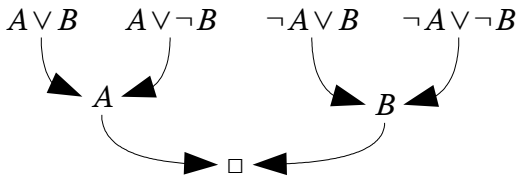
14. □

Es ist also tatsächlich so, Tweety ist nicht normal.

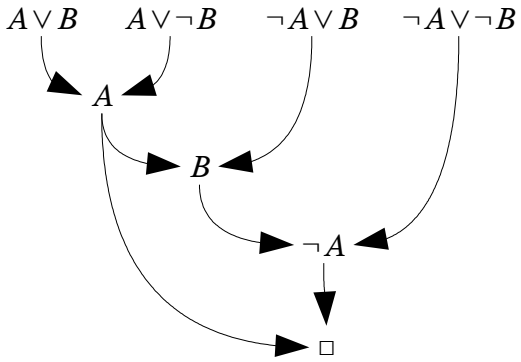
Effizientes Finden von Resolutionsbeweisen ist erschwert durch die kombinatorische Explosion. Restriktionen der Resolution schränken die Wahlfreiheit bei Resolventenbildung ein und führen somit möglicherweise schneller zum Widerspruch.

1. P-Restriktion: eine Elternklausel ist positiv (kein negatives Literal).
2. N-Restriktion: eine Elternklausel ist negativ (kein positives Literal).
3. lineare Resolution: eine Elternklausel ist die zuletzt erzeugte Resolvente.
4. Input-Restriktion: eine Elternklausel ist aus der Ausgangsklauselmenge.
5. Einheits-Restriktion: eine Elternklausel ist einelementig.
6. SLD-Resolution:
   Die SLD-Resolution ist nur für Hornklauseln definiert. Es wird Input-Resolution angewendet. Es wird mit einer nicht negativen Klausel begonnen und diese mit nicht negativen Klauseln resolviet. Es entstehen dabei nur negative Resolventen.

Beispiel: Resolution

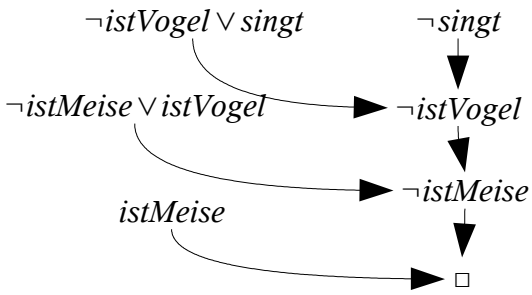


Beispiel: lineare Resolution



Beispiel: SLD-Resolution

Gegeben sei die Klauselmenge $\{\{istMeise\},\{\neg istMeise \vee istVogel\},\{\neg istVogel \vee singt\}\}$. Dies sind Horn-Klauseln mit höchstens einem Literal. Die Frage ist, ob die Meise singt. Das negierte Ziel für den Widerspruchsbeweis ist also: $\{\neg singt\}$

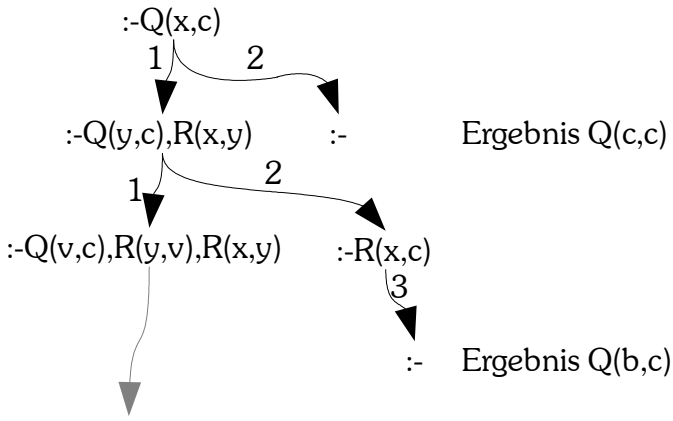


Satz

1. Resolution unter P-Restriktion ist vollständig.
2. Resolution unter N-Restriktion ist vollständig.
3. lineare Resolution ist vollständig.
4. Input-Resolution ist vollständig für Hornklauseln.
5. Einheitsresolution ist vollständig für Hornklauseln.
6. SLD-Resolution ist vollständig für Hornklauseln.

Beispiel

- Gegeben seien die Regeln
  - Q(x,z):-Q(y,z),R(x,y)
  - Q(x,x)
  - R(b,c)
- Die Zielklausel sei
  - :-Q(x,c)



Satz: Satz von Clark

Sei F ein Logik-Programm und $G=:-A_1, A_2, \ldots, A_k$ eine Zielklausel.

Dann gilt:

1. Korrektheit:
   Falls es eine erfolgreiche Berechnung von F bei Eingabe G gibt, so ist jede Grundinstanz des Rechenergebnisses $\text{sub}\left(A_1 \wedge A_2 \wedge \ldots \wedge A_k\right)$ Folgerung von F.
2. Vollständigkeit:
   Falls jede Grundinstanz von $\text{sub}'\left(A_1 \wedge A_2 \wedge \ldots \wedge A_k\right)$ Folgerung von F ist, so gibt es eine erfolgreiche Rechnung von F bei Eingabe G mit Ergebnis $\text{sub}\left(A_1 \wedge A_2 \wedge \ldots \wedge A_k\right)$, so dass für geeignete Substittionen [...in einem Anfall von Verwirrung abgewischt...]

Zur Klausur

Mengentheorie-Teil

- 1 Stunde, in der Vorlesung 25 Minuten zum Vortragen gebraucht.

Aufgabe 1

Welche Formeln sind Tautologien, Kontradiktionen, keines davon?

- Aufgabe a)
  Formel: $p \wedge (q \vee \neg p)$
  Diese Formel ist keine Tautotologie und erfüllbar, da es Modelle $(p=1) \wedge (q=1)$ gibt, aber die Interpretation $(p=0) \wedge (q=1)$ kein Modelle ist
- Aufgabe c)
  Formel: $p \wedge (q \wedge \neg p)$.
  Die Formel ist eine Kontradiktion, weil ein Modell p und $\neg p$ wahr machen müsste.

Aufgabe 2

Aufgabe 2.a

Ist die Aussage „Wenn $A \subseteq B$, dann $A \cap C \subseteq B \cap C$.“ wahr?

Ja, da $\forall(x):\big((x\in A\cap C)\Leftrightarrow(x\in A\wedge x\in C)\Rightarrow(x\in B\wedge x\in C)\Leftrightarrow(x\in B\cap C)\big)$

Aufgabe 2.b

Ist die Aussage „Wenn $A\cap C\subseteq B\cap C$, dann $A\subseteq B$.“ wahr?

Nein (nicht für alle A, B, C): Gegenbeispiel: $C=\emptyset$

Aufgabe 3

Aufgabe 3.a

Zeigen Sie, dass die Relation $\equiv$ auf der Menge der aussagenlogischen Formeln Äquivalenzrelation.

„trivial“

Aufgabe 3.b

Wieviele Äquivalenzklassen gibt es, wenn man 3 aussagenlogische Variablen zulässt?

Es gibt $2^3=8$ Interpretation [bis auf Isomorphie]. Für jede Teilmenge der Interpretation gibt es eine Formel, die genau diese Interpretationen als Modelle haben. Damit gibt es $2^{(2^3)}=2^8=256$ Äquivalenzklassen.

Aufgabe 4

Man betrachte $(\mathbb{R},\leqslant)$ (wobei $\mathbb{R}$ die reellen Zahlen sind und $\leqslant$ das übliche „kleiner gleich“ ist). Es sei $N=\{x\in\mathbb{R}\,|\,1\leqslant x<2\}$. Bestimmen Sie, ob Maximum, Minimum, Supremum und Infimum von N existieren. Falls ja, geben Sie diese Werte jeweils an.

- Das Maximum (größter Wert der Menge) existiert nicht. Warum? $\forall(x<2):\exists(x'<2):(x<x')$
- Das Supremum (kleinste obere Schranke) existiert und ist 2.
- Das Minimum und Infimum ist 1.

Aufgabe 5: Huffman-Algorithmus

$\Sigma=\{a,b,c,d,e\}$

- $w(a)=0.15$
- $w(b)=0.1$
- $w(c)=0.1$
- $w(d)=0.15$
- $w(e)=0.5$

(e,a,d,b,c)	0.5, 0.15, 0.15, 0.1, 0.1
(e,(c,d),a,b)	0.5, 0.2, 0.15, 0.15
(e,(a,b),(c,d))	0.5, 0.3, 0.2
(e,((a,b),(c,d)))	0.5, 0.5

e=0
a=100
b=101
c=110
d=111

Logik-Teil

- 1 Stunde

Aufgabe 1: Unifizierbarkeit

Seien x,y,z Variablen, a,b,c Konstanten, f,g,h Funktionssymbole.

Aufgabe 1.a

Sind folgende Formeln unifizierbar?
- „f(a,x)“
- „f(y,y)“

Ja, und zwar durch den MostGeneralUnifier „[y/a,x/a]“.

Aufgabe 1.b

Sind folgende Formeln unifizierbar?
- „f(g(x),z)“
- „f(g(y),g(z))“

Nein, weil [z/g(z)] angewendet werden müsste.

Aufgabe 1.c

Sind folgende Formeln unifizierbar?
- „f(x,g(x))“
- „f(g(y),x)“

Ja, durch [x/g(y)].

Aufgabe 2: Resolution

Zeigen Sie mit Resolution, dass folgende Formeln Tautologien sind.

Aufgabe 2.a

$F = „ \exists(x):\forall(y):\left(P(y,x)\right)\Rightarrow\forall(x):\exists(y):\left(P(x,y)\right)“$

Die Resolution terminiert nur bei einem Widerspruch. Aus diesem Grund nehmen wir das gegenteil an und führen es zum Widerspruch.

$\exists(x):\forall(y):\left(P(y,x)\right)\wedge\neg\forall(x):\exists(y):\left(P(x,y)\right)$ Implikation auflösen

$\exists(x):\forall(y):\left(P(y,x)\right)\wedge\exists(x):\forall(y):\left(\neg P(x,y)\right)$ Bereinigen

$\exists(x):\forall(y):\left(P(y,x)\right)\wedge\exists(z):\forall(v):\left(\neg P(z,v)\right)$

Trick

Beim Überführen einer Konjunktion in Klauselform dürfen Konjunktionsglieder (der äußersten Konjunktion) einzeln in Klauselform überführt werden.

Dieser Satz vereinfach die Skolemisierung.

Resultierende Klauseln wären dann:
- $\{P(y, sk_1)\}$ und
- $\{\neg P(sk_1, y)\}$

statt
- $\{P(y, sk_1)\}$ und
- $\{\neg P(sk_1(y), y)\}$

Man sieht sofort, dass aus dieser Klauselmenge sich die leere Klausel $\square$ resolvieren lässt.

Aufgabe 3

Gegeben seien folgende Aussagen:
1. „Jeder Barbier rasiert alle Personen, die sich nicht selbst rasieren."
   $\forall(x):\forall(y):((barbier(x)\wedge\neg rasiert(y,y))\Rightarrow rasiert(x,y))$
2. „Kein Barbier rasiert jemanden, der sich selbst rasiert."
   $\forall(x):\forall(y):((barbier(x)\wedge rasiert(y,y))\Rightarrow\neg rasiert(x,y))$

Wir wollen ableiten:
- „Es gibt keine Barbiere."
  $\neg\exists(x):(barbier(x))$

Die Widerspruchsannahme ist $\exists(x):(barbier(x))$. Die Skolemisierung dieser Annahme ist $barbier(sk_1)$.

Wir müssen nun die anderen Aussagen in Skolemform bringen:
1. $\{\neg barbier(x)\vee rasiert(y,y)\vee rasiert(x,y)\}$
2. $\{barbier(sk_1)\}$
3. $\{\neg barbier(v)\vee\neg rasiert(w,w)\vee\neg rasiert(v,w)\}$
4. (aus 1. und 2.): $rasiert(y,y)\vee rasiert(sk,g)$
5. (aus 2. und 3.): $\neg rasiert(w,w)\vee\neg rasiert(sk,w)$
6. Hier können wir 4 Literale auf einmal aufheben mit dem MostGeneralUnifier $[y/sk, w/sk]$:
   $\square$

Weitere Informationen:
- Bearbeitungszeit: 2 Stunden
- beliebige Skripte dürfen mitgebracht werden

Inhaltsverzeichnis

Organisatorisches..... 1
2. Mengenbegriff..... 1
Beispiel 2
Satz: Extensionalitätsprinzip..... 2
Beispiel 2
Bemerkung 2
Russelsche Antonomie..... 2
Grundbegriffe der Mengenlehre..... 2
Eigenschaften von Mengen..... 2
Mengenalgebra..... 3
Eigenschaften von Mengen-Operationen..... 3
Definition: Komplement..... 3
Satz: De-Morgansche Gesetze..... 3
Definition: Mengensystem..... 3
Definition: kartesisches Produkt, Kreuzprodukt..... 4
Definition: n-Tupel..... 4
3. Aussagenlogik..... 5
Einführung..... 5
Beispiele 5
Beispiel 1 5
Beispiel 2 5
Beispiel 3 5
Beispiel 4 5
Syntax und Semantik..... 5
Verknüpfungen (Junktoren)..... 6
Definition: Formel..... 6
Beispiel 6
Bindung der Junktoren..... 6
Bedeutung der Junktoren..... 7
Interpretation 7
Folgerbarkeit..... 7
Definition: Modell..... 7
Beispiel 3 7
Definition: äquivalent..... 8
Definition: Tautologie, allgemein..... 8
Definition: Kontradiktion..... 8
Definition: erfüllbar..... 8
Satz 8
Notation 8
Satz 8
Beweis: 8
Satz 9
Satz: häufig verwendete Äquivalenzen..... 9

3.3 Beispiel Geldautomat ... 10
3.4 Inferenz ... 11
Beispiel ... 12
4. Relationen ... 13
4.1 Grundlegende Definitionen ... 13
Definition: Relation ... 13
Definition: Relation ... 13
Beispiel ... 13
Definition: Vorbereich, Nachbereich, Feld, inverse Relation ... 13
Eigenschaften von Relationen ... 13
Definition: Einschränkung ... 14
Definition: Relations-Verknüpfung ... 14
Beispiel ... 14
Definition: Transitive Hülle ... 14
Beispiel ... 14
4.2 Aquivalenzrelationen ... 14
Beispiele ... 14
Definition: Zerlegung ... 15
Definition: Äquivalenzklasse ... 15
4.3 Ordnungsrelationen ... 15
Anmerkung 1 ... 15
Anmerkung 2 ... 15
Definition: Schranken, Extrema, Supremum, Infimum ... 16
Beispiel ... 17
Definition: Wohlordnung ... 17
Beispiel ... 17
5. Korrespondenzen und Abbildungen, Unendlichkeit ... 17
Definition: Relation, Korrespondenz ... 17
Definition: Bild, Urbild ... 18
Definition: Vorbereich, Definitionsbereich ... 18
Definition: Nachbereich, Wertebereich ... 18
Definition: Abbildung, Funktion ... 18
Definition: partielle Abbildung ... 18
Definition: Verkettung ... 18
Definition: injektiv, surjektiv, bijektiv ... 18
Satz ... 18
Satz: Unendlichkeitsdefinition nach Dedekind ... 18
Beispiel ... 18
Definition: abzählbar ... 19
Definition: abzählbar unendlich ... 19
Beispiel ... 19
Definition: überabzählbar ... 19
Satz ... 19
Beweis ... 19
6. Algebraische Strukturen ... 19
Definition: algebraische Struktur ... 19

Definition: Signatur, Typ..... 20
Definition: Algebra..... 20
Definition: Verknüpfung..... 20
Definition: Gruppe..... 20
Definition: abelsche Gruppe..... 21
Definition: Untergruppe..... 21
Satz 21
Beispiel 21
Definition: Gruppen-Homomorphismus..... 21
Definition: Gruppen-Isomorphismus..... 21
Beispiel 21
6.3 Verbände..... 22
Definition: Verband..... 22
Beispiele 22
Definition: Verbands-Homomorphismus..... 22
Definition: Verbands-Isomorphismus..... 22
Definition 22
Satz 22
Beweis 23
6.4 Boolesche Algebren..... 23
Definition: boolesch distributiver Verband..... 23
Beispiel 23
Satz 23
Definition: Komplement..... 23
Definition: komplementärer Verband..... 23
Beispiel 24
Satz 24
Definition: Boolesche Algebra..... 24
Beispiel 24
Satz 25
Beispiel: aus der Technischen Informatik:..... 25
7. Graphentheorie..... 25
7.1 Grundlegende Definitionen..... 25
Definition: gerichteter Graph..... 25
Beispiel 25
Definition: Pfad, Weg..... 26
Definition: zyklenfrei..... 26
Definition: zyklenfrei..... 26
Definition: indegree, outdegree..... 26
Definition: Baum..... 26
Definition: X-Graph..... 26
Satz 26
Repräsentation endlicher Graphen..... 27
Adjazenmatrix 27
Adjazenzlisten 27
Definition: ungerichteter Graph..... 27

Definition: Weg ... 27
Definition: zusammenhängend ... 27
Definition: zusammenhängend ... 28
Definition: Teilgraph ... 28
Definition: Teilgraph ... 28
Beispiel ... 28
Definition: kantenbewerteter Graph ... 28
Beispiel ... 28
Beispiel ... 29
Definition: indegree, outdegree ... 30
Definition: Baum ... 30
Definition: Wurzel ... 30
Anmerkung ... 30
Definition: Blatt ... 30
Definition: innerer Knoten ... 30
Definition: Tiefe eines Knotens ... 30
Definition: Tiefe eines Baumes ... 30
Definition: Ordnung ... 30
Definition: Binärbaum ... 30
Definition: vollständiger Baum ... 30
Definition: geordneter Baum ... 31
Definition: kantenmarkiert ... 31
Beispiele für Bäume ... 31
8. Grundlagen der Informationstheorie ... 32
Definition: Alphabetmenge, Menge aller Zeichenketten ... 32
Definition: Nachricht ... 32
Definition: Information ... 33
Definition: Informationsgehalt einer Nachricht ... 33
Definition: Wahrscheinlichkeit ... 33
Forderungen an Informationsgehalt einer Nachricht: ... 33
Definition: Informationsgehalt, Bit ... 33
Informationstheorie nach Shannon ... 34
Beispiele ... 34
Definition: Entropie ... 34
Definiton: Entropie ... 34
Anmerkung ... 34
Definition: Codierung ... 35
Beispiel: Binärcodierung ... 35
Definition: Mittlere Wortlänge ... 35
Theorem: Shannonsches Codierungstheorem ... 35
Intuitive Erklärung ... 35
Code-Redundanz ... 35
Beispiel ... 35
9. Einführung in die Kryptographie ... 38
Grundbegriffe ... 38
Definition: Kryptographie ... 38

Definition: Kryptoanalyse..... 38
Definition: Kryptoanalytiker..... 38
Definition: Angriff, Attacke..... 38
Definition: Chiffrierung..... 38
Definiton: Chiffretext, Geheimtext..... 38
Definition: Klartext..... 38
9.1 Transpositionschiffren..... 38
Beispiel: Spaltentransposition..... 38
9.2 Verschiebechiffren..... 39
Beispiel 39
9.3 Multiplikative Chiffren..... 39
Beispiel 39
9.4 Monoalphabetische Chiffren..... 39
9.5 Polyalphabetische Chiffren..... 39
Vigenère-Chiffre..... 40
Chiffrierung 40
Beispiel 40
9.6 Moderne Verfahren..... 40
Data Encryption Standard (DES)..... 40
Eigenschaften..... 41
1. Aussagenlogik..... 42
Definition: Logik..... 42
Geschichte der Logik..... 42
Relevanz 42
Logische vs. natürliche Sprache..... 43
Der Operator „und“..... 43
Der Operator „dann“..... 43
Syntax der Aussagenlogik..... 44
Abkürzungen 44
Semantik der Aussagenlogik..... 44
Definition: Belegung, Interpretation..... 44
Definition: Wahrheitswert von Formeln..... 44
Definition: Modell, Gültigkeit, Erfüllbarkeit..... 45
Satz 45
Beweis 45
Definition: logische Folgerung..... 45
Satz 45
Definition: äquivalent..... 45
Satz 45
Satz: Ersetzbarkeit..... 46
Satz: Äquivalenzen..... 46
Bindungsregeln..... 46
Beispiel 46
Definition: Literal..... 47
Definition: konjunktive Normalform..... 47
Beispiel 47

Definition: disjunktive Normalform 47
Beispiel 47
Wiederholung 48
Definition: konjunktive Normalform 48
Definition: disjunktive Normalform 48
Satz 48
Satz 48
Bemerkung 48
Bemerkung 48
Beispiel 48
Beispiel 48
Umformung in konjunktive Normalform 48
Umformung in distributive Normalform 49
Beispiel 49
Bemerkung 49
Beispiel 50
Hinweis 50
Hornformeln 50
Definition: Hornformel 50
Beispiel 50
Hornformel-Erfüllbarkeitstest 51
Beispiel 51
Satz 51
Beweisskizze 51
Korrektheit 52
Bemerkung: Markierungsalgorithmus 53
Exkurs: Größenrelation zwischen Modellen 53
Satz: Endlichkeitssatz, Kompaktheitssatz 53
Beweis 53
Resolution 53
Anmerkung: Kalkül 53
Definition: korrekt 53
Definition: vollständig 53
Anmerkung 54
Mengennotation für konjunktive Normalform 54
Definition: Klausel 54
Definition: Resolvente 54
Beispiel 54
Spezialfall: leere Klausel 54
Satz: Resolutionslemma 54
Beweis 54
Beispiel 55
Definition: Resolvierungsiteration 55
Beispiel 55
Satz: Resolutionssatz der Aussagenlogik 55
Beweis 55

Algorithmus zum Test der Erfüllbarkeit einer Formel in Klauselform 55
Definition: Deduktion, Herleitung, Beweis 56
Beispiele 56
Resolutionsgraph 57
Satz: Resolutionssatz der Aussagenlogik 57
Beweis 57
Davis-Putnam-Verfahren 58
Definition: reduzierte Klauselmenge 58
Beispiel 58
Anmerkung 58
Beispiel 58
Anmerkung 58
Beobachtungen 59
Algorithmus: Erfüllbarkeitstest 59
Beispiel 59
Beispiel 59
Noch zur Resolution 59
Satz 60
Gegenbeispiel 60
Tableauverfahren 61
Erzeugungsregeln für Tableaus 61
Beispiel 61
Definition: Tableau 61
Definition: Ast 62
Definition: abgeschlossen 62
Definition: abgeschlossen 62
Satz 62
Beispiel 62
2. Prädikatenlogik 63
Beispiel 63
Definition: Signatur 63
Definition: Syntax der Prädikatenlogik 63
Beispiel 63
Definition: Formel 64
Beispiel 64
Definition: Struktur 64
Semantik der Prädikatenlogik 65
Definition: Struktur 65
Definition: Prädikat 65
Definition: Funktion 65
Schreibweise 65
Definition: Semantik 65
Definition: Modell 66
Definition: erfüllbar 66
Definition: allgemeingültig 66
Eigenschaften 66

Beispiel 66
Beispiel 66
Bemerkung 67
Bemerkung: ..... Prädikatenlogik mit Identität 67
Wiederholung..... 68
Beispiel 68
Beispiel 68
Semantik 68
Wie wird interpretiert?..... 69
Definition: Modell..... 69
Definition: Folgerung..... 69
Definition: Äquivalenz..... 69
Satz 69
Lemma 69
Beweis 69
2.3 Äquivalente Umformungen und Normalformen..... 69
Satz 69
Beispiel 70
Beispiel 70
Satz 70
Satz 70
Satz 70
Satz 70
Satz 70
Satz 70
Satz 70
Satz 70
Satz 70
Satz 70
Satz 71
Beweis 71
Bemerkung 71
Lemma 71
Beweis 71
Bemerkung 71
Bemerkung 71
Definition: Substitution..... 72
Definition: freies Vorkommen..... 72
Beispiel 72
Beispiel 72
Lemma 72
Beispiel 72
Definition: bereinigt..... 72
Beispiel 72
Definition: Pränex-Normalform..... 73
Definition: Matrix..... 73

Satz 73
Beweis 73
Bemerkung 74
Beispiel 74
Definition: Skolem-Form 74
Satz 74
Beispiel 74
Unentscheidbarkeit 75
Beispiel 75
Definition: entscheidbar 75
Definition: semi-entscheidbar 75
Satz: Church-sche These 75
Korollar 75
Beweis 75
Korollar 75
Herbrand-Theorie 76
Definition: Herbrand-Universum 76
Beispiel 76
Definition: Herbrand-Struktur 76
Erklärung 76
Eigenschaften 76
Satz 76
Definition: Herbrand-Expansion 77
Satz 77
Satz 77
Algorithmus von Gilmore 77
Satz 77
Satz 77
Satz 77
Zur Klausur 78
Resolution in der Prädikatenlogik 78
Grundresolutionsalgorithmus 78
Satz 78
Beispiel 78
Satz: Grundresolutionssatz 79
Beispiel 79
Definition: Substitution 79
Beispiel 79
Unifikationsalgorithmus 80
Beispiel 80
Bemerkung: ..... Occur-Check 80
Bemerkung 81
Definition: Resolvente, Prädikatenlogische Resolution 82
Beispiel 82
Definition: Resolutionsschritt 82

Satz: Resolutionssatz der Prädikatenlogik..... 82
Beispiel 82
Beispiel: Resolution..... 84
Beispiel: lineare Resolution..... 84
Beispiel: SLD-Resolution..... 84
Satz 84
Beispiel 85
Satz: Satz von Clark..... 85
Zur Klausur..... 85
Mengentheorie-Teil..... 85
Aufgabe 1 85
Aufgabe 2 85
Aufgabe 2.a 85
Aufgabe 2.b 86
Aufgabe 3 86
Aufgabe 3.a 86
Aufgabe 3.b 86
Aufgabe 4 86
Aufgabe 5: Huffman-Algorithmus..... 86
Logik-Teil..... 87
Aufgabe 1: Unifizierbarkeit..... 87
Aufgabe 1.a 87
Aufgabe 1.b 87
Aufgabe 1.c 87
Aufgabe 2: Resolution..... 87
Aufgabe 2.a 87
Trick 88
Aufgabe 3 88
Inhaltsverzeichnis..... 89