Inhaltsverzeichnis

1 Aussagenlogik

1.1 Grundbegriffe

Defintion: (Syntax) Eine aussagenlogische Formel ist eine Zeichenkette, die sich aus den Variablen A_0, A_1, ... und den Symbolen $(,), \wedge, \vee, \neg$ zusammensetzt und folgende Regeln einhält:

- Variablen sind Formeln
- Wenn F_1 und F_2 Formeln sind, dann ist $(F_1 \wedge F_2)$ eine Formel (*Konjunktion*)
- Wenn F_1 und F_2 Formeln sind, dann ist auch $(F_1 \vee F_2)$ eine Formel (*Disjunktion*)
- Wenn F eine Formel ist, ist auch $\neg F$ eine Formel (*Negation*)

Alle Formeln enstehen auf diese Weise.
Beispiel: $(\neg A_0 \wedge A_1)$
Lemma: (Eindeutige Lesbarkeit) Jede Formel ist

- eine Variable
- eine Konjunktion $(F_1 \wedge F_2)$
- eine Disjunktion $(F_1 \vee F_2)$
- eine Negation $\neg F$

Es tritt immer genau einer dieser Fälle ein, und ggf sind F_1 und F_2, bzw. F eindeutig bestimmt.
Beispiel: Wir bezeichnen das erste Zeichen der gegebenen Formel. Es treten die Fälle auf:

- es ist eine Variable A_i. Dann ist A_i nach Definition die ganze Formel.
- es ist das Zeichen "$\neg$", dann ist nach Definition unserer Formel $\neg F$. F ist also der Rest unserer Zeichenkette.
- es ist "(". Nach Definition ist unsere Formel vom Typ $(F_1 \wedge F_2)$ / $(F_1 \vee F_2)$.

Wir nehmen an, dass wir die Formel außerdem auch als $(F_1' \wedge F_2')$ / $(F_1' \vee F_2')$ lesen können. Dann ist entweder F_1' Anfangsstück von F_1 oder umgekehrt. Nach dem folgenden Hilfssatz gilt $F_1 = F_1'$ und das unmittelbar folgende Zeichen muss "$\wedge$" oder "$\vee$" sein und legt den Typ der Formel fest. Wenn man eine Formel nach obigem Lemma zerlegt hat und die Formel keine Variable war, dann wendet man das Lemma anschie´ssend weiter auf F_1 und F_2 bzw. F an bis die Formel komplett zerlegt wurde.
Hilfssatz: Kein echtes Anfangsstück einer Formel ist selbst eine Formel. Mit "*echtem Anfangsstück*" meinen wir ein Anfangsstück, das nicht die ganze Formel ist.
Beispiel: $(F_1$ ist Anfangsstück von $(F_1 \wedge F_2)$.
Beweis: Durch Induktion über die Länge der Formel. Sei F Formel der Länge 1, dann hat F das echte Anfangsstück " " (*leere Formel*), aber das ist keine Formel.

Wir nehmen jetzt an, dass der Hilfssatz für alle kürzeren Formeln bewiesen werden kann. Wir wenden eine Fallunterscheidung nach dem ersten Buchstaben an.

- Variable A_1: Dann hat F die Länge 1.
- "$\neg$": Nach Definition gilt: $F = \neg F_1$. Sei F' echtes Anfangsstück von F_1, dann folgt $F' = \neg F_1$, und F_1' ist echtes Anfangsstück von F_1. Nach Induktionsvoraussetzung ist aber kein echtes Anfangsstück von F_1 eine Formel. Also kann F' keine Formel sein.
- "(": Wie im obigen Beweis gilt dann $F = (F_1 \wedge F_2)$ / $(F_1 \vee F_2)$. Sei F' echtes Anfangsstück von F, dann folgt $F' = (F_1 \wedge F_2)$ / $(F_1 \vee F_2)$. Da F_1 und F_2 kürzer als F sind und F_1 Anfangsstück von F_1' oder F_1' Anfangsstück von F_1 ist, muss $F_1 = F_1'$ gelten. Dann muss F_2') ein Anfangsstück von F_2) sein, also F_2' ein Anfangsstück von F_2. Nach Induktionsvoraussetzung kann F_2' kein echtes Anfangsstück gewesen sein.

Damit wäre der Hauptsatz bewiesen. □

Definition: Es sei $\mathcal{D}$ eine Menge von Variablen, dann ist eine Belegung eine Abbildung $\mathcal{A}$:

$$\mathcal{A} : \mathcal{D} \to \{0, 1\}$$

Dabei stehen 0, 1 für die Wahrheitswerte "*falsch*" oder "*wahr*". Sei also $A_3 \in \mathcal{D}$ und $\mathcal{A}(A_3) = 1$, dann hat A_3 unter der Belegung $\mathcal{A}$ den Wahrheitswert 1 ("*wahr*"). Wir definieren Verknüpfungen von Wahrheitswerten:

$$w_1 \wedge w_2 = \begin{cases} 1 & w_1 = 1 \text{ und } w_2 = 1 \\ 0 & w_1 = 0 \text{ oder } w_2 = 0 \end{cases}$$

$$w_1 \vee w_2 = \begin{cases} 1 & w_1 = 1 \text{ oder } w_2 = 1 \\ 0 & w_1 = 0 \text{ und } w_2 = 0 \end{cases}$$

$$\neg w_1 = \begin{cases} 1 & w = 0 \\ 0 & w = 1 \end{cases}$$

Definition: (Semantik) Sei $\mathcal{A}$ eine Belegung der Variablen einer Formel F, dann hat F den Wahrheitswert $\mathcal{A}(F)$ von F:

- $\mathcal{A}(A_1)$, falls $F = A_1$ Variable ist
- $\mathcal{A}(F_1) \wedge \mathcal{A}(F_2)$, falls $F = (F_1 \wedge F_2)$
- $\mathcal{A}(F_1) \vee \mathcal{A}(F_2)$, falls $F = (F_1 \vee F_2)$
- $\neg\mathcal{A}(F')$, falls $F = \neg F'$

Wegen des obigen Lemma ist $\mathcal{A}(F)$ dadurch wohldefiniert. Wir verwenden folgende Abkürzungen:

- $(F_1 \to F_2)$ für $(\neg F_1 \vee F_2)$
- $(F_1 \Leftrightarrow F_2)$ für $((F_1 \to F_2) \wedge (F_2 \to F_1))$
- $\bigwedge_{i<n} F_i$ für $(F_1 \wedge ...(Fn-2 \wedge Fn-1)...)$
- $\bigvee_{i<n} F_i$ für $(F_1 \vee ...(Fn-2 \vee Fn-1)...)$
- $\top$ ("*wahr*") für $(\neg A_0 \vee A_0)$
- $\bot$ ("*falsch*") für $(\neg A_0 \wedge A_0)$

Konventionen: Wir können $\top$, $\bot$ auch als "*nicht zusammengesetzte*" Formeln betrachten

- $\bigwedge\limits_{i<0} F_i = \top$
- $\bigvee\limits_{i<0} F_i = \bot$

Außerdem können wir $(F_1 \vee F_2)$ als Abkürzungen für $\neg(\neg F_1 \wedge \neg F_2)$ auffassen. Gelegentlich lassen wir Klammern weg, wenn keine Missverständnisse auftreten. Unsere Abkürzungen haben genau wie alle Formeln Wahrheitswerte unter vorgegebenen Bedingungen.

$$\mathcal{A}(\top) = 1, \bot = 0, \text{ usw.}$$

Sei (A) eine Belegung der Variablen von F. Wenn $\mathcal{A}(F) = 1$, sagen wir, "*$\mathcal{A}$ erfüllt F*", "*$\mathcal{A}$ ist Modell von F*", "$\mathcal{A} \models F$".
Definition: Eine Formel heißt "*allgemeingültig*" oder "*Tautologie*", wenn $\mathcal{A}(F) = 1$ für alle möglichen Belegungen gilt. Eine Formel F heißt "*erfüllbar*", wenn es eine Belegung $\mathcal{A}$ mit $\mathcal{A}(F) = 1$ gibt.
Beispiel:

- $\top = (\neg A_0 \vee A_0)$ ist allgemeingültig, A_0 ist erfüllbar.
- $\bot = (\neg A_0 \wedge A_0)$ ist nicht erfüllbar. A_0 ist nicht allgemeingültig.

Lemma:

- F ist genau dann eine Tautologie, wenn $\neg F$ nicht erfüllbar ist.
- F ist erfüllbar genau dann, wenn $\neg F$ keine Tautologie ist.

1.2 Äquivalenz von Formeln und Normalformen

Definition: Zwei Formeln F,G in den gleichen Variablen heißen äquivalent, kurz $F \equiv G$, wenn $\mathcal{A}(F) = \mathcal{A}(G)$ für alle Belegungen der Variablen.

Beispiel: F = A, G = A∧(A∨B)

Bemerkung:

- F$\equiv G$ genau dann, wenn $F \leftrightarrow G$ allgemeingültig ist.
- F allgemeingültig (kurz $\models$F) genau dann, wenn $F \equiv \top$
- F erfüllt (kurz $\not\models \neg$ F) genau dann, wenn F$\not\equiv \bot$
- " $\equiv$ " ist eine Äquivalenzrelation. [d.h. F≡F für alle Formeln F
  F$\equiv G \Rightarrow G \equiv F$ für alle Formel n F, G
  (F$\equiv G$ und $G \equiv H) \Rightarrow F \equiv H$ für alle Formeln F, G, H
  Das bedeutet: Die Menge aller Formeln zerfällt in Äquivalnzklassen [F] mit G∈[F]$\Leftrightarrow$ $F \equiv G$]

Bemerkung: Methoden um Äquivalenz von Formeln zu zeigen:

- Fallunterscheidungen für die Belegungen (siehe Bsp, mündlich)
- Alle Belegungen einsetzen und vergleichen: (A∧$B) \equiv (B \wedge A)$

$A \backslash B$	0	1
0	0	0
1	0	1

$A \backslash B$	0	1
0	0	0
1	0	1

- Venn-Diagramme: Für jede Variable A male eine "Menge" M_A. Punkte in M_A entsprechen Belegungen mit $\mathcal{A}(A) = 1$, Punkte außerhalb:$\mathcal{A}(A) = 0$

  Bsp: $((A \wedge B) \vee C) \equiv ((A \vee C) \wedge (B \vee C))$

  Hier noch Grafik einfügen !!!

Definition Sei F eine Formel, aufgebaut aus Variablen, den Junktoren $\neg, \wedge, \vee$, sowie $\top, \bot$. Die zu F duale Formel F* entsteht aus F durch Vertauschen von $\wedge, \vee$ sowie von $\top, \bot$.

Lemma:F≡G genau dann, wenn F* ≡G

Beweis: Es sei $\mathcal{A}$ eine Belegung . Vertausche in $\mathcal{A}$ die Werte 0,1 und erhalte eine neue Belegung $\mathcal{A}^*$

Behauptung: es gilt $\mathcal{A}(F) = 1$ genau dann, wenn $\mathcal{A}^*(F^*) = 0$.

Begründung: Durch Induktion über Länge der Formel mit dem Lemma über eindeutige Lesbarkeit.

$F = A \Rightarrow F^* = A$
$\mathcal{A}(F) = \mathcal{A}(A) = 1 \Leftrightarrow \mathcal{A}^*(A) = 0 = \mathcal{A}^*(F^*))$
$F = \top \Rightarrow F^* = \bot$ Induktionsschritt:
$\mathcal{A}(\top) = 1$ gilt immer genau $A^*(\bot) = 0$
Analog für $F = \bot$

- F = ¬ G :

$$\begin{aligned} \mathcal{A}(F) &= 1 \Leftrightarrow \mathcal{A}(G) &&= 0 \\ \overset{IV}{\Leftrightarrow} \mathcal{A}^*(G^*) &= 1 \Leftrightarrow \mathcal{A}^*(F^*) &&= 0 \end{aligned}$$

- F = $(F_0 \wedge F_1), F^* = (F_0^* \vee F_1^*)$:

$$\begin{aligned} \mathcal{A}(F) &= 1 \Leftrightarrow \mathcal{A}(F_0) = 1 \textit{ und } \mathcal{A}(F_1) = 1 \\ \Leftrightarrow \mathcal{A}^*(F_0^*) &= 0 \textit{ und } \mathcal{A}^*(F_1^*) = 0 \\ \Leftrightarrow \mathcal{A}^*(F^*) &= 0 \end{aligned}$$

- F = $(F_0 \vee F_1)$ analog. □

Satz: Es gelten folgenden Äquivalnzen sowie ihre Duale:

1. $(A \wedge A) \equiv A$ (Idempotenz)
2. $(A \wedge B) \equiv (B \wedge A)$ (Kommutativität)
3. $((A \wedge B) \vee C) \equiv ((A \vee C) \wedge (B \vee C))$ (Distributivität)
4. $(A \wedge (A \wedge B)) \equiv A$ (Absorption)
5. $(A \wedge (B \wedge C)) \equiv ((A \wedge B) \wedge C)$ Assoziatovität)
6. $(\bot \vee A) \equiv A$
7. $(\top \wedge A) \equiv \top$
8. $(A \wedge \neg A) \equiv \bot$ "(dual zu tertum non datur")

Beweis: Kommutativität, Distributivität, Absorption siehe oben. Rest analog.

Definition: Es seien F,H Formeln und A eine Variable. Dann bezeichnet die Formel F(H/A) die Formel die aus F entsteht, indem man jedes Vorkommen der Variable A durch die Formel H ersetzen. (Dabei gehen wir nicht rekursiv vor d.h. wenn H selbst die Variable A enthält, lassen wir A danach stehen).

Beispiel: F=$(A \wedge B)$, G=$(B \wedge A)$, H=$(B \vee C)$
F(H/B)=$(A \wedge (B \vee C))$, G(H/B)=$((B \vee C) \wedge A)$

Lemma(Ersetzungslemma): Es seien F,G,H Formeln und A eine Variable. Wenn G≡ H gilt, dann gelten auch:

$$F(G/A) \equiv F(H/A) \quad (1)$$

und

$$G(F/A) \equiv H(F/A) \quad (2).$$

Beweis: Zu (1) : Bei allen Belegungen $\mathcal{A}$ gilt $\mathcal{A}(G) = \mathcal{A}(H)$. Bei der rekursiven Definition von $\mathcal{A}(F)$ kann ich anstelle $\mathcal{A}(A)$ kann ich $\mathcal{A}(G)$ *oder* $\mathcal{A}(H)$ einsetzen und erhalte beidemal das gleiche Ergebniss für alle Belegungen $\Rightarrow$ (1).
Zu (2): In der rekursiven Definition von $\mathcal{A}(G), \mathcal{A}(H)$ ersetze wieder $\mathcal{A}(A)$ *durch* $\mathcal{A}(F)$, es folgt wieder die Äquivalenz (2). □

<u>**WARNUNG**</u>: Im Allgemeinen folgt aus (1) oder (2) nicht, dass G≡H.

Satz: Es gelten die de Morgenschen Regeln:
$\neg(A \wedge B) \equiv (\neg A \vee \neg B)$
dual dazu: $\neg(A \vee B) \equiv (\neg A \wedge \neg B)$
$\neg\neg\neg A \equiv A$.

Beweis: Wie oben, siehe auch Beweis des Dualitätslemma.□

Satz: Ein <u>Literal</u> ist eine Variable A, oder ¬A. Ein Ausdruck der Form:

$$\bigvee_{i<m} \bigwedge_{j<n_i} L_{ij}$$

wobei L_{ij} Literale sind, heißt "disjunktive Normalform". Dual dazu heißt:

$$\bigwedge_{i<m'} \bigvee_{j<n'_i} L'_{ij}$$

"konjunktive Normalform".

Satz: Jede Formel F ist zu einer Formel in konjunktiver, bzw. disjunktiver Normalform äquivalent.

WARNUNG: Diese Normalformen sind nicht eindeutig, z.B. sind:

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

beide in disjunktiver Normalform.

Beweis: Induktiv für beide Normalformenzusammen:
Sei etwa F = $(F_0 \wedge F_1)$, dann bringe zunächst F_0 und F_1 in die gewünschte Normalform.
Für die konjunktive Normalform hänge die beiden großen Konjunktionen zusammen:

$$\bigwedge_{i<m_0} \bigvee_{...} L'_{ij} \wedge \bigwedge_{i<m_1} \bigvee_{...} L''ij$$

$$\rightarrow \bigwedge_{i<m_o+m_1} \bigvee_{...} L_{ij}$$

.
Für die disjunktive Normalform benutzt man das Distributivgesetz:

$$\bigvee_{i<m_0} \bigwedge_{...} L'_{ij} \wedge \bigvee_{i<_m 1} \bigwedge_{...} L''_{ij}$$

$$\rightarrow \bigvee_{i<m_0 \cdot m_1} (\bigwedge_{...} L'_{ij} \wedge \bigwedge_{...} L''_{ij})$$

Analog verfahre mit F=$(F_0 \vee F_1)$ (Beides ist analog zum Ausmultiplizieren von Polynomen in mehreren Veränderlichen ...)
Um F=$\neg$ G in konjunktive Normalform zu bringen, bringe G in disjunktive Normalform und wende dann die de Morgenschen Regeln:

$$\neg(\bigvee_{i<m} \bigwedge_{j<n_i} L_{ij}) = \bigwedge_{i<m} \bigvee_{j<n_i} \neg L_{ij}$$

anschließend beseitige doppelte Veneinungen:$\neg\neg A \equiv A$. Analog für disjunktive Normalform. $\square$

1.3 Boole'sche Algebren

Definition: Eine boole'sche Algebra $(B, 0, 1, \sqcap, \sqcup, ^{\complement})$ besteht aus einer Menge B, Elenenten $0, 1 \in B$, Verknüpfungen $\sqcap, \sqcup : B \times B \to B$ und einem Komplement $^{\complement} \in B : a \mapsto a^{\complement}$, so dass folgende Axiome für $\forall a, b, c, \in B$ gelten:

(1.1)	$a \sqcup a = a$	$a \sqcap a = a$	(Idempotenz)
(1.2)	$a \sqcup b = b \sqcup a$	$a \sqcap b = b \sqcap a$	(Kommutativität)
(1.3)	$(a \sqcap b) \sqcap c = a \sqcap (b \sqcap c)$	$(a \sqcup b) \sqcup c = a \sqcup (b \sqcup c)$	(Assoziaivität)
(1.4)	$a \sqcup (a \sqcap b) = a$	$a \sqcap (a \sqcup b) = a$	(Absorption)
(1.5)	$a \sqcap (b \sqcup c) = (a \sqcap b) \sqcup (a \sqcap c)$	$a \sqcup (b \sqcap c) = (a \sqcup b) \sqcap (a \sqcap c)$	(Distributivität)
(1.6)	$0 \sqcap a = 0$	$1 \sqcup a = 1$	
(1.7)	$a \sqcap a^{\complement} = 0$	$a \sqcup a^{\complement} = 1$	

Bemerkungen:

- Eines der Distributivitätsgesetze (1.5) ist überflüssig
- Es gelten die de Morgan'schen Regeln

Beispiel einer Folgerung:
$a \sqcup 0 \overset{1.6}{=} a \sqcup (a \sqcap 0) \overset{1.4}{=} a$
Analog für $a \sqcap 1 = a$

Beispiel 1: Es sei X eine Menge. Dann ist die Potenzmenge $\mathcal{P}(X)$ eine boole'sche Algebra $(\mathcal{P}(X), \varnothing, X, \cap, \cup, \setminus\cdot)$.
Für $A \subset X$ ist $A^{\complement} = X \setminus A$

Beispiel 2: Ein n-Byte sei eine Folge von n Bits aus $\{0, 1\}$. Dann bildet die Menge aller n-Bytes eine boole'sche Albegra mit AND, OR, NOT.

Diese Beispiele sind isomorph. Dabei entspricht dem n-Byte $(b_0, ..., b_{n-1})$ die Teilmenge $\{i \in \{0, ..., n-1\} \mid b_i = 1\} \in \mathcal{P}(\{0, ..., n-1\})$

Es gibt für boole'sche Algebren das Prinzip der Dualität:
Wenn eine Gleichung gilt, dann gilt sie auch nach Vertauschen von $\sqcap$ und $\sqcup$, sowie von 0 und 1.

Definition: Eine Struktur $(M, \sqcap, \sqcup)$ heißt <u>Verband</u>, wenn (1.1) bis (1.4) gelten.

Bemerkung: Zu jedem Verband gehört eine <u>partielle Ordnung</u> $\leq$ mit:

$$a \leq b \Leftrightarrow a \sqcap b = a$$

Es gelten:

Reflexivität: $a \sqcap a = a \Rightarrow a \leq a$

Antisymmetrie: Es gelte $a \leq b, b \leq a; a \sqcap b = a, b \sqcap a = b$
Aus (1.2) folgt $a = a \sqcap b = b$

Transitivität: Es gelte $a \leq b, b \leq c$
$a \sqcap b = a, b = b \sqcap c$
$a = a \sqcap b = a \sqcap (b \sqcap c) = (a \sqcap b) \sqcap c = a \sqcap c$

Beispiel: Die Potenzmenge ist ein Verband mit $A \leq B \Leftrightarrow A \cap B = A \Leftrightarrow A \subseteq B$

Umgekehrt sei $(M, \leq)$ eine Menge mit partieller Ordnung

Ein Infimum von $a, b \in M$ ist ein Element

$$c = \inf(a, b) \text{ mit } c \leq a, c \leq b$$

so dass

$$\forall d \in M : c \leq d \leq a, c \leq d \leq b \Rightarrow d = c$$

Ein Supremum von $a, b \in M$ ist ein Element

$$c = \sup(a, b) \text{ mit } a \leq c, b \leq c$$

so dass

$$\forall d \in M : a \leq d \leq c, b \leq d \leq c \Rightarrow d = c$$

Wenn in einer partiellen Ordnung $(M, \leq)$ je zwei $a, b, \in M$ ein Infimum oder Supremum haben, erhalten wir einen Verband mit $a \sqcap b = \inf(a, b)$ und $a \sqcup b = \sup(a, b)$.

Wenn es Elemente $0, 1$ mit $0 \leq a \leq 1$ $\forall a \in M$ gibt, dann gilt (1.6).

Zu einer boole'schen Algebra fehlen noch die Komplemente, sowie die Distributivität.

Beispiel: (Verband, aber keine boole'sche Algebra)
Sei V ein Vektorraum. Dann bilden die Untervektorräume von V einen Verband $U(V)$ mit $U \leq W \Leftrightarrow U \subseteq W$.
Dann
$U \sqcap W = U \cap W$
$U \sqcup W = U + W = \{u + w \mid u \in U, w \in W\}$
Aber (1.5) ist verletzt! Sei $V = \mathbb{R}^2$:
$U = \mathbb{R} \times \{0\}, W = \{0\} \times \mathbb{R}$ (x- und y-Achse)
Sei $X = \{(x, y) \mid x \in \mathbb{R}\}$, dann:
$U \sqcup W = \mathbb{R}^2$
$X \sqcap (U \sqcup W) = X$
$X \sqcap U = \{0\}$
$X \sqcap W = \{0\}$
$(X \sqcap U) \sqcup (X \sqcap W) = \{0\}$

Satz: (Stone'scher Darstellungssatz)
Jede boole'sche Algebra ist Unteralgebra einer Potenzmengenalgebra und jede endliche boole'sche Algebra ist isomorph zu einer Potenzmengenalgebra.

Beweis der zweiten Aussage:
Sei $(B, 0, 1, \sqcap, \sqcup, ^{\complement})$ eine boole'sche Algebra, so dass B eine endliche Menge ist.
Dann ist B ein Verband mit Minimum 0.

Ein <u>Atom</u> in B ist ein Element $a \in B \setminus \{0\}$, so dass aus $0 \leq b \leq a$ stets $b = 0$ oder $b = a$ folgt.

Sei x die Menge der Atome in B.
Definiere $\Phi : B \to \mathcal{P}(X)$ durch

$$\Phi(b) = \{a \in X \mid a \leq b\} \in \mathcal{P}(X)$$

Zeige zunächst: Φ ist bijektiv:
Surjektivität: Sei $U \subset X$, dann betrachte

$$b = \bigsqcup_{a \in U} a = a_0 \sqcup ... \sqcup a_{k-1}, \text{ falls } U = \{a_0, .., a_{k-1}\}$$

Sei $a \in X$, so gilt

$$a \leq b \Leftrightarrow a \sqcap (a_0 \sqcup ... \sqcup a_{k-1}) = a = (a \sqcap a_0) \sqcup ... \sqcup (a \sqcap a_{k-1})$$

Da $a \sqcap a_i \leq a, a \sqcap a_i \leq a_i$ und a, a_i Atome gilt entweder

$$a \sqcap a_i = 0 \Leftrightarrow a \neq a_i$$

oder

$$a = a \sqcap a_i = a_i$$

ist der obige Ausdruck entweder 0 (falls $a \notin U$) oder $a = a_i \in U$
$\Rightarrow \Phi(b) = U$

Injektivität:
Es gelte $\Phi(b) = \Phi(a)$,
dann gilt $\Phi(b \sqcap c) = \Phi(c)$, denn

Sei $a \leq b, a \leq c, a$ Atom:
$a = a \sqcap b = a \sqcap c = (a \sqcap b) \sqcap c = a \sqcap (b \sqcap c)$,
also o.B.d.A. $b \leq c$.
Betrachte $c \sqcap b^{\complement}$:
Falls $b \leq c$ und $c \sqcap b^{\complement} = 0$ folgt $b = c$,
Falls $c \sqcap b^{\complement} \neq 0$, finden wir $a \in X$
mit $a \leq c \sqcap b^{\complement} \Rightarrow a \in \Phi(c) \setminus \Phi(b)$ $\lightning$

Dazu nutzen wir aus, dass B endlich ist:
jede Kette $0 < a < ... < c \sqcap b^{\complement}$ hat höchstens Länge $\#B$ und für längstmögliche Kette ist a ein Atom.

$\Rightarrow \Phi$ ist bijektiv

Noch zu zeigen: $\Phi(0) = \varnothing, \Phi(1) = X$, usw. $\square$

Beispiel: Lindenbaum-Algebra
Die Lindenbaum-Algebra in n Variablen $A_0, ..., A_{n-1}$ ist die Menge der Äquivalnenzklassen aussagenlogischer Formeln in den Variablen $A_0, ..., A_{n-1}$ bezüglich der Äquivalenzen(?) aus 1.2.

Schreibe Elemente als $F/\equiv$ (z.B. $(\neg A_0 \vee A_1)/\equiv$) also

$$LA_n = \{F/\equiv \mid F \text{ ist aussagenlogische Formel in } A_0, ..., A_{n-1}\}$$

Wir setzen

- $0 = \bot/\equiv = (A_0 \wedge \neg A_0)/\equiv$
- $1 = \top/\equiv = (A_u \vee \neg A_0)/\equiv$
- $(F/\equiv) \sqcup (G/\equiv) = (F \vee G)/\equiv$
- $(F/\equiv) \sqcap (G/\equiv) = (F \wedge G)/\equiv$
- $(F/\equiv)^{\complement} = \neg F/\equiv$

Aus den elementaren Äquivalenzen (Abschnitt 1.2) und dem Ersetzungslemma folgt, dass allee Axiome einer boole'schen Algebra gelten.

Warnung: LA_n ist nicht isomorph zu $\mathcal{P}(\{0, ..., n-1\})$

1.4 Resolutionsmethode

Bemerkung: Um F durch eine Formel in KNF zu ersetzen, braucht man nur 'elementare Äquivalenzen' und das Ersetzungslemma. Die 'elementaren Äquivalenzen' (Satz 1.2) entsprechen den Axiomen der Boolschen Algebra (für die Lindenbaum Algebra).
Bemerkung: Um die konjunktive Normalform eindeutig zu machen, vereinbaren wir, dass in

$$\bigwedge_{i<m} \bigvee_{j<n_i} L_{ij}$$

die Ausdrücke $\bigvee\limits_{j<n_i} L_{ij}$ Atome sind.

Das heißt $(\neg)\ A_0\ \vee (\neg)\ A_1\ \vee \cdots \vee (\neg)\ A_{n-1}$ falls $A_0, \cdots A_{n-1}$ alle vorkommenden Variablen sind.

Beispiel: $(a \vee b) \wedge (\neg a \vee b) \wedge (a \vee \neg b) \wedge (\neg a \vee \neg b)$ nicht erfüllbar

Beispiel: Falls {A,B} die Variablen sind:

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

(Fallunterscheidung nach $\mathcal{A}$(B))

Beweis: Diese Zerlegung in Atome macht die konjunktive Normalform eindeutig (bis auf Reihenfolge), aber die neue Formel hat Länge

$$O(2^{\#Variablen})$$

- Eine solche Formel F ist allgemeingültig genau dann, wenn sie alle mögliche keine Atome enthält
- Eine solche Formel F ist nicht erfüllbar genau dann, wenn sie alle mögliche Atome enthält

Wir erhalten einen Algorithmus, der 'Allgemeingültigkeit' entscheidet, aber mit einer Laufzeit von $O(2^{\#Variablen})$.
Äquivalenz zu diesem Verfahren ist die 'Brute Force'-Methode:
Es gibt $2^{\#Variablen}$ viele Belegungen. Berechne $\mathcal{A}$(F) für jede Belegung $\mathcal{A}$, um Allgemeingültigkeit (oder Erfüllbarkeit) zu entscheiden.

Notation: Schreibe eine Formel

$$\bigwedge_{i<m} \bigvee_{j<n_i} L_{ij}$$

$$\bigwedge_{i<m} \bigvee_{j<n_i} L_{ij}$$

in KNF als Menge von Klauseln

$$\bigvee_{j<n_i} L_{ij} = \{L_{i,0}, \cdots, L_{i,n-1}\}$$

Eine Klausel ist also eine Menge von Literalen (Variablen oder negierte Variablen).

Sei also $\mathcal{C} = \{C_0, \cdots, C_{m-1}\}$ eine Menge von Klauseln und $\mathcal{A}$ eine Belegung, dann gilt:

$\mathcal{A} \models \mathcal{C} \Leftrightarrow \mathcal{A} \models C_i$ für alle $i < m$.

$\mathcal{A} \models \mathcal{C}_i = \{L_{i,0}, \cdots, L_{i,n-1}\} \Leftrightarrow$ Es gibt ein $j < n$ mit $A \models L_{ij}$

Definition: Es seien

$$C_1 = \{A\} \cup P$$

$$C_2 = \{\neg A\} \cup Q$$

wobei die Klauseln P, Q weder A noch $\neg A$ enthalten.

Dann heißt $C = P \cup Q$ eine Resultante von C_1, C_2.
Eine Menge $\mathcal{C}$ von Klauseln heißt unter Resultanten abgeschloßen, wenn sie mit je zwei Klauseln C_1, C_2 auch alle ihre Resultanten enthält.

Lemma: Es sei $\mathcal{A}$ eine Belegung der Variablen von C_1, C_2.
Dann gilt für die Resultante C, dass $\mathcal{A} \models \{C_1, C_2\} \Leftrightarrow \mathcal{A} \models \{C_1, C_2, C\}$.

Beweis: Fallunterscheidung nach $\mathcal{A}(A)$

1. $\mathcal{A}(A) = 0$: Dann gilt $\mathcal{A} \models C_1 \Leftrightarrow \mathcal{A} \models P$, mit $C_1 = \{A\} \cup P$

   Insbesondere folgt $A \models P \cup Q = C$

2. $\mathcal{A}(A) = 1$: Dann gilt $\mathcal{A} \models C_2 = \{\neg A\} \cup Q \Leftrightarrow A \models Q$, also folgt $A \models P \cup Q = C$

Daraus folgt:
$\mathcal{A} \models \{C_1, C_2\} \Rightarrow \mathcal{A} \models \{C_1, C_2, C\}$
Umgekehrt ist klar:
$\mathcal{A} \models \{C_1, C_2, C\} \Rightarrow \mathcal{A} \models \{C_1, C_2\}$

1.4.1 Satz

Es sei $\mathcal{C}$ eine Menge von Klauseln die unter Resultanten abgeschloßen ist. Dann ist $\mathcal{C}$ erfüllbar genau dann, wenn $\emptyset \notin \mathcal{C}$.

Beweis: Klausel $\emptyset \notin \mathcal{C} \Rightarrow A \models \mathcal{C}$ für alle $\mathcal{A}$ dann $\mathcal{A} \neq \emptyset$.
Für '$\Leftarrow$' sei $\mathcal{C}$ unter Resultanten abgeschlossen.

Induktion über Anzahl der Variablen:
Es sei $A = A_{n-1}$ die 'größte' Variable (größter Index), die in $\mathcal{C}$ vorkommt.
Wäre $\{A\}$, $\{\neg A\} \in \mathcal{C}$, dann wäre $\emptyset$ als Resultante in $\mathcal{C}$ enthalten.
Also dürfen wir annehmen, dass $\{\neg A\} \notin \mathcal{C}$ (ohne Einschränkung).
Wir betrachten jetzt nurnoch Belegungen $\mathcal{A}$ mit $\mathcal{A} = 1$.
Dann bearbeite $\mathcal{C}$ wie folgt:

- Falls $C \in \mathcal{C}$ weder A noch $\neg A$ enthält, behalte $C \in \mathcal{C}'$
- Falls $C \in \mathcal{C}$ die Variable A enthält, gilt $A \models C$. Dann streiche C in $\mathcal{C}$
- Falls $C \in \mathcal{C}$ die Variable $\neg A$ enthält, sei $C \models \{\neg A\} \cup Q$, dann streiche $\neg A$, somit sei $Q \in \mathcal{C}'$

- (Falls $C \in \mathcal{C}$ weder A noch $\neg A$ enthält, wenn c bereits allgemeingültig, wir hätten also $\mathcal{C} \backslash \{C\}$ betrachten können)

So erhalten wir eine neue Klauselmenge $\mathcal{C}'$, und es gilt $\mathcal{A}' \models \mathcal{C}' \Rightarrow \mathcal{A} = \mathcal{A}' \cup \{A \to 1\} \models \mathcal{C}$
Wenn also $\mathcal{C}'$ erfüllbar ist, ist $\mathcal{C}$ erst recht erfüllbar

- $\mathcal{C}' \notin \emptyset$ dann nach Annahme $\emptyset \notin \mathcal{C}$ und $\{\neg A\} \notin \mathcal{C}$
- $\mathcal{C}'$ ist unter Resultanten abgeschloßen.

Dann sei B eine Variable ($B \neq A$) und $C'_1 = \{B\} \cup P' \in \mathcal{C}'$, $C'_2 = \{\neg B\} \cup Q' \in \mathcal{C}'$

Dann bekommen $\mathcal{C}'_\in$, $\mathcal{C}'_\in$ von Klauseln der Form

$C_1 = \{\neg A, B\} \cup P'$ oder $C_1 = C'_1$
$C_2 = \{\neg A, \neg B\} \cup Q'$ oder $C_2 = C'_2$

Somit enthielt $\mathcal{C}$ die Resultante $\{\neg A\} \cup P' \cup Q'$ oder $P' \cup Q'$.
Damit enthält $\mathcal{C}'$ dann die Resultante $P' \cup Q'$ von C'_1 und C'_2.
Also ist $\mathcal{C}'$ unter Resultanten abgeschlossen mit $\emptyset \notin \mathcal{C}'$ und enthält eine Variable weniger, so dass wir die Induktionsvorraussetzung anwenden dürfen. Wenn wir keine Variablen übrig haben, gibt es nur die leere Klausel $\emptyset$, aber $\emptyset \notin \mathcal{C}$ nach Annahme, also $\mathcal{C} = \emptyset$, und $\mathcal{C}$ steht dann

$$\bigwedge_{i<0} ? = \top$$

also ist $\mathcal{C}$ erfüllbar.

Bemerkung: Der Beweis liefert auch eine Belegung $\mathcal{A}$, die die ursprüngliche Klauselmenge $\mathcal{C}$ erfüllt.
Aber: Abschließen unter Resultanten kannn eine Laufzeit bis zu $4^{\#Variablen}$ ($3^{\#Variablen}$, falls A_i, $\neg A_i$ nicht simultan vorkommen) haben = nicht sehr effizient.

1.5 Der Satz von Cook

Ziel: SAT = 'Saturierbarkeit', 'Erfüllbarkeit von Formeln' der Aussagenlogik ist ein NP-vollständiges Problem.

Definition: Es sei $\mathbb{A}$ ein Alphabet, also eine endliche Menge von Zeichen.
Dann bezeichnet $\mathbb{A}^*$ die Menge aller endlichen Wörter mit Symbolen aus $\mathbb{A}$ und $W \subset \mathbb{A}^*$ heißt ein Problem.

Beispiel: Es sei $\mathbb{A} = \{\wedge, \vee, \neg, (,), A_0, \cdots, A_{n-1}\}$
$W \subset \mathbb{A}^*$ sei die Menge aller Aussagenlogischen Formeln. Sei $w \in \mathbb{A}$ ein Wort der Länge n. Dann können sie in $O(n)$ Schritten entscheiden, ob $w \in W$.

Betrachte dazu ein Unterprogramm 'Formel', das sich anhand des nächsten Zeichens wie folgt verhält:

- $\vee, \wedge,)$ - Fehler
- $\neg$ - lasse das Programm 'Formel' ab dem nächsten Zeichen laufen. Wenn nicht $\vee$ oder $\wedge$ kommt - Fehler. Lasse 'Formel' ab dem Zeichen danach weiterlaufen.
-) - Überspringe und ok. Wenn) kommt, überspringe und ok, ansonsten Fehler
- Sonst - Zeichen ist Variable, überspringe sie und ok.

Wiederholung: Erfüllbarkeit

- brute force - $O(2^k)$
- Resolutionsmethode - $O(3^k)$

k - Anzahl der Variablen
Auf der anderen Seite - gegeben $\mathcal{A}$ lässt sich $A \models F$ in polynomialer Zeit (in $|F|$) prüfen.

Ziel: Erfüllbarkeit ist ein NP-vollständiges Problem.

Bemerkung: F ist Tautologie $\Leftrightarrow \neg F$ nicht erfüllbar.
$F \models G \Leftrightarrow F \rightarrow G$ allgemeingültig.
$\Leftrightarrow F \wedge \neg G$ nicht erfüllbar

Erinnerung: Turing Maschinen.

- Γ - Alphabet $\Gamma \supset \mathbb{A} \cup \{\_\}$ (Leerzeichen)
- Q - Zustände, $q \in Q$, $A \subset F \subset Q$
- F - Haltezustände
- A - Akzeptable Zustände, $\delta \subset ((Q \backslash F) \times \Gamma) \times (Q \times \Gamma \times \{-1, 0, 1\})$
- "Programm"

Dabei muss jedem $g \in Q \backslash F, a \in \Gamma$ Werte, $g \in Q, a \in \Gamma, d = +1/-1$ existieren, sodass $(q, a, q', a', d) \in \delta$. Eine Turing-Maschine heißt deterministisch, wenn für alle q, a wie oben die Werte (q', a', d) eindeutig sind. Wenn wir von nicht-deterministischen Turing-Maschinen sprechen, meinen wir alle.
Eine Eingabe ist ein Wert der Länge n aus $\mathbb{A}$, den wir auf das Band schreiben und mit "äuffüllen. Ein "Laufist eine Folge von Zuständen des Bandes und der Maschine

und Position des Schreiblesekopfes, sodass wenn auf der Position des Kopfes "a steht und die Maschine im Zustand "q_n" ist und im nächsten Schritt an der Stelle "a_{n+1}" steht und die Maschine im Zustand "q_{n+1}" ist und der Kopf sich um d bewegt, das Tupel $(q_n, a_n, q_{n+1}, a_{n+1}, d) \in \delta$. Außerdem verändert sich der Zustand nicht, wenn $q_n \in F$.

Definition: Ein Problem $w \in \mathbb{A}$ ist von der Klasse P, wenn es eine deterministische Turing-Maschine gibt, die für jedes Wort $w \in \mathbb{A}^*$ der Länge n in $P(n)$ Schritten entscheidet, ob $w \in W$ oder nicht. NP, wenn es eine nicht deterministische Turing-Maschine gibt, sodass für alle $w \in \mathbb{A}^*$, $n = |w|$ einen Lauf gibt, sodass für alle $w \in \mathbb{A}^*, n = |w|$ einen lauf gibt, der in $P(n)$ Schritten entscheidet ob $w \in W$. NP-vollständig, wenn es in NP liegt und jedes andere NP-Problem $w' \subset \mathbb{A}^*$ sich in polynomialer Zeit in ein äquivalentes Problem aus $w \subset \mathbb{A}'^*$ überführen lässt, d.h. zu jedem $w' \in \mathbb{A}'^*$ lässt sich in $P(|w'|)$ Schritten ein $w \in \mathbb{A}^* \Leftrightarrow w \in W$ bilden.

Bemerkung: Ein Problem ist in NP, wenn man einen Lösungsweg (z.B. eine Belegung $\mathcal{A}$ im Falle der Erfüllbarkeit) raten kann und dann innerhalb polynomialer Zeit prüfen kann ob es korrekt ist.

Beispiel: Primfaktorzerlegung ist in NP. Primzahltest in P.

Bemerkung: Vermutung: P $\neq$ NP. P $\subset$ NP gilt.
Falls P $\neq$ NP: P $\subset$ NP und NP-vollständig $\subset$ NP.
Falls P $=$ NP: NP-vollständig $\subset$ P $=$ NP.

Satz (Cook): Erfüllbarkeit ist NP-vollständig.

Bemerkung: Erfüllbarkeit ist in NP, denn man kann zu jeder möglichen Belegung in polynomialer Zeit prüfen, ob $\mathcal{A} \models F$.
Zu NP-Vollständigkeit sei $W \subset \mathbb{A}^*$ ein NP-Problem und $\delta \subset (Q \backslash F \times P) \times (Q \times \Gamma \times \{-1, 0, 1\})$ eine Turing-Maschine, die das Problem im Sinne der Definition löst. Dabei sei $P(n)$ die Laufzeit des gesuchten Laufes, dabei ist n die Länge des Wortes. Wir brauchen die Variablen:

- T_{ijk} für $-P(n) \leq i \leq P(n), j \in \Gamma, 0 \leq k \leq P(n)$, ßur Zeit k steht an der Stelle i des Bandes der Buchstabe j." - $O(P(n)^2)$
- H_{ik} wie oben "der Schreiblesekopf steht zur Zeit k an der Stelle i" - $O(P(n)^2)$
- Q_{qk} wie oben, $q \in Q$ "Maschine ist zur zeit k im Zustand q" - $O(P(n))$

Wir geben die Formel als Menge von Klammern an, dabei gibt es verschiedene Typen:

- T_{ij0} für alle i wie oben, wobei j jetzt der i-te Buchstabe des Eingabewortes $w \in \mathbb{A}^* sei.\ O(P(n)$
- H_{00} Startpunkt (Initialisierung)
- $Q_{q_0 0}$ Startzustand (Initialisierung)
- $\neg T_{ijk} \vee \neg T_{ij'k}$ für alle i, k wie oben und alle $j \neq j' \in \Gamma$ $O(P(n)^2$ (Eindeutigkeit der Zustände)
- $\neg Q_{qk} \vee \neg Q_{q'k}$ für alle k wie oben und alle $q, q' \in Q$ (Eindeutigkeit der Zustände)
- $\neg H_{ik} \vee \neg H_{i'k}$ für k wie oben und alle $i \neq i$ (Eindeutigkeit der Zustände)
- $\neg T_{ijk} \vee \neg T_{ij',k+1} \vee H_{ik}$ für alle $j \neq j'$ wie oben "Band ändert sich nur am Schreiblesekopf"
- $\neg H_{ik} \vee Q_{qk} \vee \neg T_{ijk} \vee (H_{i+0,k+1} \wedge Q_{q',k+1} \wedge T_{ij'k})(q, j, q', j', d) \in \delta$ (Lauf)
- "Übergangsfunktion"lässt sich umwandeln in insgesamt $O(P(n)^2)$ Klauseln (Lauf)
- Analoge Klauseln, die besagen, dass sich der Zustand nicht ändert, wenn Q_{qk} mit $q \in F$ gilt. $O(P(n)^2)$ Stück (Lauf)
- Schluss $\bigvee\limits_{q \in A} Q_g P(n)$ beschränkte Anzahl

Eine Belegung $\mathcal{A}$ der obigen Variablen erfüllt diese Aussage genau dann, wenn sie einen Lauf der Turing-Maschine mit Eingabe w beschreibt, der nach maximal $P(n)$ Schritten in einem akzeptablen Zustand terminiert. Also ist die Klauselmenge äquivalent. Zu "$w \in W$Zeitaufwand zur Erstellung der Klauselmenge ist polynomial in $|n|$. Also:: Erfüllbarkeit ist NP-vollständig. $\square$

Bemerkung: P = NP gilt genau dann, wenn es einen polynomialen Algorithmus gibt, das die Erfüllbarkeit testet. Ein Problem $w \subset \mathbb{A}^*$ ist genau dann NP-vollständig wenn sich jede Formal F in polynomialer Zeit $P(|F|)$ in ein Wort $w \in \mathbb{A}^*$ übersetzen lässt, sodass F erfüllbar $\Leftrightarrow w \in W$.

1.6 Der Kompaktheitssatz

Satz: Eine beliebige Menge X von Formeln ist genau dann erfüllbar, wenn jede endliche Teilmenge erfüllbar ist.

Bemerkung: Wenn X unendlich ist, muss man ünendlich lange arbeitenüm Erfüllbarkeit "von Handßu beweisen.

Aber: Um zu zeigen, dass X **nicht** erfüllbar ist, muss man nur die richtige endliche Teilmenge "ratenünd dann mit der Resolutionsmethode einen Widerspruch herleiten.
<u>Wdh:</u> Turing Maschinen, P,NP, SAT (Erfüllbarkeit in Aussagenlogik) ist NP-Vollständig.

1.6 Der Kompaktheitsatz

Satz: Eine Menge X von Formeln ist genau dann erfüllbar, wenn jede endliche Teilmenge erfüllbar ist.

Beweis:Falls X nicht abzählbar ist, benötigen wir Zorn's Lemma (äquivalent zum Auswahlaxiom).
Sei X abzählbar, dann ist auch die Menge V der Variablen, die in X vorkommen, abzählbar:

$$V = \{A_n | n \in \mathbb{N}\}$$

Annahme: Jede endliche Teimenge von X ist erfüllbar. Wir suchen induktiv $w_i \in \{0, 1\}$, für alle i$\in \mathbb{N}$, so dass für jedes n$\in \mathbb{N}$ folgendes gilt:
Für jede endliche Teilmenge Y$\subset$X existiert eine Belegung $\mathcal{A}$ mit $\mathcal{A} \models$Y und $\mathcal{A}(A_i) = w_i$ für alle i$< n$.

Induktionsanfang: n=0 ✓
Seien $w_0, ..., w_{n-1}$ wie oben konstruiert.
Annahme:das gesuchte w_i existiert nicht. Das heißt, für alle $w_n = 0$ existiert eine endliche Teilmenge Y_0, so dass $\mathcal{A} \not\models Y_0$ für alle $\mathcal{A}$ mit $\mathcal{A}(A_i) = w_i$ für i$< n$, und $\mathcal{A}(A_n) = 0$.
Analog: $Y_1 \subset X$ endlich, so das für alle $\mathcal{A}$ mit $\mathcal{A}(A_i) = w_i$ für alle $i < n$ und $\mathcal{A}(A_n) = 1$ gilt:

$$\mathcal{A} \not\models Y_1$$

Dann ist $Y_0 \cup Y_1$ endlich, und es existiert keine Belegung $\mathcal{A}$ mit $\mathcal{A}(A_i) = w_i$ für $i < n$ und $\mathcal{A} \models Y_0 \cup Y_1$, denn A_n lässt sich nach Annahme nicht so belegen, dass die Formel aus Y_0 und Y_1 erfüllt wenden. Das ist ein Widerspruch zur Wahl von $w_0, ..., w_{n-1}$. Also lassen sich die w_n indukt festlegen, und wir erhalten eine Belegung $\mathcal{A}$. Sei jetzt F∈X, dann sei n der maximale Index einer Variablen, die in F vorkommt (F ist endlich!). Aus der Wahl der $w_0, ..., W_{n-1}$ folgt $\mathcal{A} \models \{F\}$. □

Beispiel: "$A_{X,r}$=1⇔ Var. X hat den Wert r∈ ℝ"
X heißt abzählbar genau dann, wenn es eine Folge (x_n) $n \in \mathbb{N}$ gibt so dass:

$$X = \{x_n | n \in \mathbb{N}\}$$

ℚ abzählbar, ℝ überabzählbar (=nicht abzählbar)

Folgerung: Es sei X eine Menge von Formeln, und es sei G eine Formel, so dass X⊨ G, dann existiert eine endliche Teilmenge $X_0 \subset X$ und X⊨G.

Beweis: ÜBUNG!!! □

Beispiel: Ein Graph ist eine Menge von Punkten Ëckenünd eine Menge von Kanten (=Linien zwischen diesen Punkten).Er heißt "n-färbbar", wenn ich jeder Ecke eine "Farbe"0,...,n-1 zuordnen kann, so dass eine Kante zwei Ecken gleicher Farbe verbindet (z.B Eine pro Land auf der Landkarte, eine Kante pro gemeinsamer Grenze). Ein unendlicher Graph ist genau dann n-färbbar, wenn jeder endliche Teilgraph n-färbbar ist.
Färbungen: Fehlen im Ziegler-Skript!!

2 Prädikatenlogik

"Logik der ersten Stufe"
"$\forall_{x \in \mathbb{R}} \forall_{\delta > 0} \exists_{\varepsilon > 0} \forall_{y \in \mathbb{R}} (|y - x| < \varepsilon \rightarrow |f(y) - f(x)| < \delta)$" "f ist stetig".

2.1 Strukturen und Formeln

Definition: Eine Sprache L besteht aus:

- einer Menge von Konstanten
- einer Menge von Funktionszeichen
- einer Menge vo Relationszeichen

Funktionen und Relationen haben eine feste "Stelligkeit" (Zahl der erlaubten Argumente)$\geq$ 1

Beispiel: Um Körper zu beschreiben, wählen wir Konstanten $\{0, 1\}$, Funktionen $\{+, \cdot, -\}$ dabei haben $+, \cdot$ Stelligkeit 2 und - hat Stelligkeit 1. Für angeordnete Körper nehmen wir die Relation $\{<\}$ hinzu. Funktionen und Relationen werden in Präfix-Notation benutzt:

$$+ab \textit{ statt } a+b$$

Definition: Sei L eine Sprache. Eine L-Struktur $\mathfrak{A} = \{A, (Z^{\mathfrak{A}})_{Z\in L}\}$ wobei A eine Meinge ist und $Z^{\mathfrak{A}}$ je nach Typ von Z:

- $Z^{\mathfrak{A}} \in A$ falls Z Konstante ist
- $Z^{\mathfrak{A}}$:$A^n \to A$ Abbildung, falls Z eim n-stelliges Funktionssymbol ist
- $Z^{\mathfrak{A}} \subset A^n$ falls ein n-stelliges Relationssymbol ist

Beispiel: L wie oben, dann sind

$$(\mathbb{R}, 0_{\mathbb{R}}, 1_{\mathbb{R}}, +_{\mathbb{R}}, \cdot_{\mathbb{R}}, -_{\mathbb{R}}, <_{\mathbb{R}})$$

$$(\mathbb{Q}, 0_{\mathbb{Q}}, 1_{\mathbb{Q}}, +\mathbb{Q}, \cdot_{\mathbb{Q}}, -_{\mathbb{Q}}, < \mathbb{Q})$$

Beispiele von L-Strukturen.

Definition: Ein Isomorphismus zwischen L-Strukturen $\mathfrak{A} = (A, (Z^{\mathfrak{A}})_{Z\in L})$ und $\mathfrak{B} = (B, (Z^{\mathfrak{A}})_{Z\in L})$ ist eine bijektive Abbildung F:A →B, die mit den Interpretationen der Zeichen aus L verträglich ist:

- F($Z^{\mathfrak{A}}$=Z$^{\mathfrak{B}}$ für Konstanten
- F$\circ Z^{\mathfrak{A}} = Z^{\mathfrak{B}} \circ F^n : A^n \to B$ für Funktionssymbole
- F^n($Z^{\mathfrak{A}} = Z^{\mathfrak{B}}$ für Relationssymbole

Eine L-Antommorphismus von $\mathfrak{A}$ ist ein L-Isomorphismus von $\mathfrak{A}$ *nach* $\mathfrak{A}$. Die Menge der L-Antomophismen bildet eine Gruppe Ant $\mathfrak{A}$.
Fixiere eine Menge von Variablen $v_0, v_1, v_2, ...$

Definition: Ein L-Term ist eine Zeichenfolge, die nach folgenden Regeln gebildet ist:

1. Jede Konstante ist ein L-Term
2. Jede Variable ist ein L-Term

3. Sei f ein n-stelliges Funktionssymbol und seien $t_1, ..., t_n$ Terme, dann ist auch f$t_1, ..., t_n$ ein Term.

Beispiel:" $+ \cdot 2\pi v_0$" bedeutet $2 \cdot \pi + v_0$
" $\cdot + v_0 v_1 \pi$" bedeutet $(v_0 + v_1) \cdot \pi$

Lemma: Für jeden term gilt genau einer der drei folgenden Fälle:

- t ist eine (eindeutige) Konstante
- t ist eine (eindeutige) Variable
- t ist von der Form t=f $t_1 ... t_n$, dabei ist f ein n-stelliges Funktionssymbol und $t_1, .., t_n$ sind eindeutig bestimmte Terme.

Beweis: Analog zu eindeutigen Lebarkeit aussagenlogischer Formeln. □

Hilfsatz: Kein Term ist ein echtes Anfangsstück eines anderen Termes.

Beweis: Analog zur Aussagenlogik... □

Zur besseren Lesbarkeit schreiben wir manchmal f$(t_1, ..., t_n)$ für f$t_1 ... t_n$

Definition Sei L eine Sprache. Eine L-Formel ist

- $t_0 \doteq t_1$, wobei t_0, t_1 Terme. "t_0 ist gleich t_1"
- $Rt_0 \dots t_{n-1}$, $t_0, \dots, t_{n-1}$ Terme, R n-stelliges Relationssymbol
- $\neg F_0$, F_0 Formel
- $(F_0 \wedge F_1)$, wobei F_0, F_1 Formeln
- $\exists v\ F$, wobei v Variable, F Formel

Alle Formeln entstehen auf diese Weise.

Beispiel: $L = (0, f, +, <)$
$\forall v_0 \forall v_1 (< 0 v_1 \rightarrow \exists v_2 (< 0 v_2 \wedge ((< v_0 + v_2 v_3 \wedge < v_3 + v_0 v_2) \rightarrow (< f v_0 + v_1 f v_3 \wedge < f v_3 + f v_0 v_1))))$
"f ist stetig" mit $v_0 = x, v_1 = \epsilon, v_2 = \delta, v_3 = y$

Abkürzungen:

Abkürzung	Bedeutung
$\forall v\ F$	$\neg\exists v\ \neg F$
$(F_0 \vee F_1)$	$\neg(\neg F_0 \wedge \neg F_1)$
$(F_0 \to F_1)$	$(\neg F_0 \vee F_1)$
$(F_0 \leftrightarrow F_1)$	$((F_0 \to F_1) \wedge (F_1 \to F_0))$
$(F_0 \wedge \cdots \wedge F_{n-1})$	$(F_0 \wedge \cdots \wedge (F_{n-2} \wedge F_{n-1}) \dots)$
$(F_0 \vee \cdots \vee F_{n-1})$	$(F_0 \vee \cdots \vee (F_{n-2} \vee F_{n-1}) \dots)$
$\exists v_0 \dots v_{n-1}$	$\exists v_0 \exists v_1 \dots \exists v_{n-1}$
$\forall v_0 \dots v_{n-1}$	$\forall v_0 \forall v_1 \dots \forall v_{n-1}$

Wenn Klammern weggelassen werden, gelten folgende Präzedenzregeln in absteigender Priorität:

1. $\exists, \forall, \neg$
2. $\wedge$
3. $\vee$
4. $\to, \leftrightarrow$

Lemma: L-Formeln sind eindeutig lesbar.
Das heißt: Auf jede Formel trifft genau einer der Fälle aus obiger Definition zu und die dort eventuell vorkommenden Terme und Teilformeln sind ebenfalls eindeutig.

Hilfssatz: Keine Formel ist echtes Anfangsstück einer anderen Formel.

Beweis: Lemma und Hilfssatz durch Induktion über Länge der Formeln.
Fallunterscheidung nach erstem Symbol der Formel F:

- Falls erstes Symbol eine Variable, Konstante oder Funktion ist, muss F vom Typ $F = t_0 \mathrel{\mathring{=}} t_1$ sein. Da Terme eindeutig lesbar sind, ist klar, wie t_0 und t_1 aufgebaut sind und wo F aufhört.
- Falls erstes Symbol ein Relationssymbol ist, gilt $F = Rt_0 \dots t_{n-1}$. Dabei legt R die Stelligkeit n fest und $t_0 \dots t_{n-1}$ sind eindeutig.
- Falls erstes Symbol $\neg$ ist, gilt $F = \neg F_0$. Da F_0 kürzer als F ist, ist F_0 nach IV eindeutig lesbar.
- Falls erstes Symbol (ist, gilt $F \in \{(F_0 \wedge F_1), (F_0 \vee F_1), \dots\}$
- Falls erstes Symbol ein Quantor ist, gilt $F = \exists v_1\ F_0$ bzw. $F = \forall v_1\ F_0$. Nach IV ist F eindeutig lesbar.

□

2.2 Semantik

Definition: Sei $\mathfrak{A} = (A, (Z^{\mathfrak{A}})_{Z\in L})$ eine L-Struktur. Eine Belegung β ist eine Abbildung

$$\beta\colon (v_0, v_1 \dots) \to A$$

wobei $(v_0, v_1 \dots)$ die Menge der Variablen ist.

Definition: Sei $\mathfrak{A}$ eine L-Struktur, β eine Belegung.
Dann definiere für L-Terme t rekursiv

- $t^{\mathfrak{A}}[\beta] = \beta(v_i)$ falls $t = v_i$
- $t^{\mathfrak{A}}[\beta] = c^{\mathfrak{A}}$ falls $t = c$ Konstantensymbol
- $t^{\mathfrak{A}}[\beta] = f^{\mathfrak{A}}(t_1^{\mathfrak{A}}[\beta], \dots, t_{n-1}^{\mathfrak{A}}[\beta])$

Beispiel: $L = (e, 0, \mathrm{Inv})$ Sprache für Gruppen.
Sei $\mathbb{K}$ ein Körper. Dann kann man 2 L-Strukturen betrachten:
$\mathfrak{A} = (\mathbb{K}, 0, +, -)$
$\mathfrak{L} = (\mathbb{K} \setminus \{u\}, \wedge, \cdot, ^{-1})$
Sowie jede Menge möglicher anderer Strukturen.

Definition: Sei $\mathfrak{A}$ eine L-Struktur, β eine Belegung.
Der Wahrheitswert von Formeln wird rekursiv wie folgt definiert:

- $\varphi = F_0 \doteq t_1$, dann ist $\varphi^{\mathfrak{A}}[\beta]$ wahr, gdw. $t_0^{\mathfrak{A}}[\beta] = t_1^{\mathfrak{A}}[\beta] \in A$
- $\varphi = Rt_0 \dots t_{n-1}$, dann ist $\varphi^{\mathfrak{A}}[\beta]$ wahr, gdw. $(t_0^{\mathfrak{A}}[\beta], \dots, t_{n-1}^{\mathfrak{A}}[\beta]) \in R^{\mathfrak{A}} \subset A^n$
- $\varphi = \neg\varphi_0$, dann ist $\varphi^{\mathfrak{A}}[\beta]$ wahr, gdw. $\varphi_0^{\mathfrak{A}}[\beta]$ falsch ist
- $\varphi = (\varphi_0 \wedge \varphi_1)$ dann ist $\varphi^{\mathfrak{A}}[\beta]$ wahr, gdw. sowohl $\varphi_0^{\mathfrak{A}}[\beta]$ als auch $\varphi_1^{\mathfrak{A}}[\beta]$ wahr sind
- $\varphi = \exists v_i\ \varphi_0$, dann ist $\varphi^{\mathfrak{A}}[\beta]$ wahr, gdw. $\exists a \in A\colon \varphi_0^{\mathfrak{A}}[\beta \frac{a}{v_i}]$, $(\beta \frac{a}{v})(v_j) = \begin{cases} a & i = j \\ \beta[v_j] & i \neq j \end{cases}$

Also hat $\exists$ in $\exists v_i\ \varphi_0$ den Gültigkeitsbereich φ_0. Dort wird die "globale" Bedeutung $\beta(v_i) \in A$ durch eine "lokale" Bedeutung $a \in A$ ersetzt.

Globale Variablen: frei
Lokale Variablen: gebunden

Definition:
Sei v_i eine Variable, φ eine L-Formel. Dann kommt v_i in φ frei vor, wenn

- $\varphi = t_0 \doteq t_1$ und v_i kommt in t_0 oder t_1 vor.
- $\varphi = Rt_0 \dots t_{n-1}$ und v_i kommt in einem der Terme vor.

- $\varphi = \neg\varphi_0$ und v_i kommt in φ_0 frei vor.
- $\varphi = \varphi_0 \wedge \varphi_1$ und v_i kommt in φ_0 oder φ_1 frei vor.
- $\varphi = \exists v_j\ \varphi_0$ und $v_i \neq v_j$ und v_i kommt frei in φ_0 vor.

Insbesondere ist v_j in $\exists v_j\ \varphi_0$ nicht frei.

Lemma
Seien β, γ Belegungen in L-Struktur $\mathfrak{A}$.

- Für einen L-Term t gilt
  $t^{\mathfrak{A}}[\beta] = t^{\mathfrak{A}}[\gamma]$, wenn β und γ auf allen Variablen, die in t vorkommen übereinstimmen.
- Für eine Formel φ gilt
  $\varphi^{\mathfrak{A}}[\beta]$ gdw. $\varphi^{\mathfrak{A}}[\gamma]$, falls β und γ auf allen Variablen, die in φ frei vorkommen übereinstimmen.

Beweis:
Erste Aussage folgt induktiv über Länge von t.

Zweite Aussage folgt ebenfalls induktiv:
Klar, falls $\varphi = t_0 \doteq t_1, \varphi = Rt_0 \ldots t_{n-1}, \varphi = (\varphi_0 \wedge \varphi_1), \varphi = \neg\varphi_0$.

Sei $\varphi = \exists v_i\ \varphi_0$ und $a \in A$ so, dass $\varphi_0^{\mathfrak{A}}[\beta\frac{a}{v_i}]$ wahr ist. Dann stimmt $\gamma\frac{a}{v_i}$ mit $\beta\frac{a}{v_i}$ auf allen Variablen überein, die in φ_0 frei vorkommen.
Somit gilt auch $\varphi_0^{\mathfrak{A}}[\gamma\frac{a}{v_i}]$, somit $\varphi^{\mathfrak{A}}[\beta] \Rightarrow \varphi^{\mathfrak{A}}[\gamma]$.

Genau so: $\varphi^{\mathfrak{A}}[\gamma] \Rightarrow \varphi^{\mathfrak{A}}[\beta]$ $\square$

Definition: Eine Aussage ist eine L-Formel ohne freie Variablen.

Eine Aussage φ gilt in $\mathfrak{A}$ also unabhängig von Belegungen.
Dafür schreibt man $\mathfrak{A} \models \varphi$, was man "$\mathfrak{A}$ ist Modell von φ", "φ gilt in $\mathfrak{A}$" oder "$\mathfrak{A}$ erfüllt φ" liest.