Logik für Informatiker

Wintersemester 2005/06

Matthias Bernauer - Freiburg im Breisgau - 12. November 2005

Mitschrift zur Vorlesung "Logik für Informatiker" gelesen von Prof. Dr. Flum am Institut für Informatik der Universität Freiburg im Wintersemester 2005/2006.

Dieses Skript basiert auf den Folien von Herrn Prof. Flum:
http://home.mathematik.uni-freiburg.de/flum/

Hinweise bzgl. gefundener Fehler oder Vorschläge zur Erweiterung sind jederzeit willkommen an: `mail@matthias-bernauer.de`

Matthias Bernauer

Inhaltsverzeichnis

1 Einführung in die Logik

1.1 Was ist Logik?

Unter der mathematischen Logik wird heute im Allgemeinen eine, teils in der Philosophie, Mathematik und Informatik angesiedelte Theorie verstanden, die sich primär mit den Normen des korrekten Folgerns beschäftigt. Sie untersucht, unter welchen Bedingungen das Folgern einer Aussage aus einer Menge anderer Aussagen korrekt ist und entwickelt hierzu formale Sprachen zur exakten Beschreibung und Normierung der Schlussregeln.

Charakteristisch für die Regeln der deduktiven Logik (der Logik im engeren Sinne) ist im Gegensatz zur induktiven Logik, dass ein Übergang von einer Aussage zu einer anderen salva veritate, d.h. unter Erhaltung des Wahrheitswertes, möglich ist. Ein logisch gültiger Schluss ist ein solcher, der uns aufgrund seiner logischen Form nicht von wahren Prämissen (Voraussetzung, Annahme) zu einer falschen Konklusion (Schluss, Urteil) führen kann, also wahrheitserhaltend ist.

Daraus ergibt sich ein geläufiges Verfahren zur Überprüfung der Gültigkeit einer Folgerung, nämlich die Suche nach Gegenbeispielen. Gelingt es zu einem Argument, dessen logische Gültigkeit zweifelhaft ist, ein struktur- oder formgleiches Argument zu finden, dessen Prämissen wahr und dessen Konklusion falsch ist, so ist das Argument (und generell das Schlussschema) zu verwerfen.

1.2 Schule der Sophisten

- **Achilles und die Schildkröte**

  Achilles und die Schildkröte laufen ein Wettrennen. Achilles gewährt der Schildkröte einen Vorsprung. Dann kann Achilles die Schildkröte niemals einholen.

  Zenon von Elea (490 - 425 v. Chr.) gibt folgende Begründung: Zu dem Zeitpunkt, an dem Achilles den Startpunkt der Schildkröte erreicht, ist die Schildkröte schon ein Stück weiter. Etwas später erreicht Achilles diesen Punkt, aber die Schildkröte ist schon etwas weiter. Wenn Achilles diesen Punkte erreicht, ist die Schildkröte wieder etwas weiter. So kann Achilles zwar immer näher an die Schildkröte herankommen, sie aber nie erreichen.

- **Der Barbier**

  In einem Städtchen wohnt ein Barbier, der genau diejenigen männlichen Einwohner rasiert, die sich nicht selbst rasieren. Rasiert nun der Barbier sich selbst?

- **Antinomie des Lügners**

  Epimenides (Kreter, 600 v. Chr.)
  Brief des Paulus an Titus 1:12-13:
  Einer von ihren eigenen Landsleuten war ein Prophet, als er sagte: “Die Kreter lügen immer. Sie sind Raubtiere, liegen auf der faulen Haut und denken nur ans Fressen”. Er hat die Wahrheit gesagt.

- **Aus Don Quijote de la Mancha von Miguel de Cervantes (1517-1616)**

  Um eine gewisse Brücke zu überqueren, müssen alle Reisenden zunächst angeben, was ihr Ziel ist. Antworten sie wahrheitsgemäß, so dürfen sie die Brücke überqueren. Sonst werden sie gnadenlos an einem Galgen am Fußder Brücke erhängt.

  Eines Tages kommt ein Reisender, der angibt, sein Ziel sei es, am Galgen am Fußder Brücke erhängt zu werden.

1.3 Syllogismen - Überblick über die Regeln des Schließens

Prämisse	Alle Menschen sind sterblich
Prämisse	Sokrates ist ein Mensch
Konklusion	Sokrates ist sterblich

Prämisse	Alle a sind b
Prämisse	c ist ein a
Konklusion	c ist b

1.4 Welche Rolle spielt Logik in der Informatik?

- **Programmverifkation**: Man möchte wissen, dass ein Programm das *Richtige* tut oder dass es zumindest gewisse Eigenschaften hat.
- **Typechecking**: Der Compiler überprüft, dass eine Funktion immer einen Wert vom richtigen Typ zurückgibt.
- **Schaltkreisverikation**: Man möchte beweisen, dass ein Chip richtig funktioniert.
- **Protokollverikation**: Man möchte beweisen, dass die Komunikation zwischen zwei Agenten, die nach einem gewissen *Protokoll* abläuft, sicher ist.

Logic is the calculus of computer science

Logic permeates through computer science much more than it does through mathematics.

Anwendung der Logik in der Informatik:

- To model computer hardware
- As a database query language
- As a tool for representing and reasoning
- As a tool for specication and verication

2 Aussagenlogik

2.1 Begriffe

Definition (Alphabet): Ein *Alphabet* Σ ist eine nicht leere Menge von Zeichen (Buchstaben, Symbole).

Beispiel: $\Sigma_1 = \{0, 1, \ldots, 9\}$; $\Sigma_2 = \{0, 1\}$; $\Sigma_3 = \{a, b, \ldots, x, y, z\}$; $\Sigma_4 = \{a, d, f, x, \int,), (\}$

Es handelt sich häufig um endliche Alphabete konkreter Zeichen, doch es lassen sich auch unendliche Alphabete definieren: $\Sigma_5 = \{c_0, c_1, \ldots\}$; $\Sigma_6 = \{c_r | r \in \mathbb{R}\}$

Bemerkung:

- *Worte* sind endliche Aneinanderreihungen von Zeichen aus Σ.
- *Variablen* für Wörter: u, v, w.
- λ ist das *leere Wort*.
- Σ^* ist die Menge aller Wörter, die sich aus einem Alphabet Σ ergeben, bspw. $\int f(x)dx \in \Sigma_4^*$ aber auch $a \int \int dxd \in \Sigma_4^*$.
- $|w|$ ist die *Länge* eines Wortes. *Beispiel:* $|\int f(x)dx| = 7, |a \int \int dxd| = 6, |\lambda| = 0$.

Bemerkung: Das unendliche Alphabet $\Sigma_5 = \{c_0, c_1, \ldots\}$ lässt sich auch als endliches Alphabete $\Sigma_7 := \{c, 0, 1, \ldots, 9\}$ ersetzen.
Bei Identifikation von c_{25} mit $c25$ ergibt sich zum Beispiel $\Sigma_5 \subseteq \Sigma_7^*$ und $\Sigma_5^* \subseteq \Sigma_7^*$.

Definition (Syntax der Aussagenlogik):

$$\begin{aligned}\Sigma_a &= \{\neg, \wedge, \vee\} \cup \underbrace{\{X_1, X_2, \ldots\}}_{\text{Aussagenvariablen}} \cup \{), (\} \\ &= \{\neg, \wedge, \vee\} \cup \{X, 0, \ldots, 9\} \cup \{), (\}\end{aligned}$$

Bemerkung: Nicht alle Wörter sind aussagenlogischen Ausdrücke, z.B. $\vee X_5 \vee$.

Beispiel:
X: Die Sonne scheint.
Y: Die Logikvorlesung ist langweilig.
Z: Das Softwarepraktikum ist langweilig.

Y und Z: Die Logikvorlesung und das Softwarepraktikum sind langweilig.
Wenn nicht X, so Y : Wenn die Sonne nicht scheint, dann ist die Logikvorlesung langweilig.

Definition (Aussagenlogische Ausdrücke): *Aussagenlogische Ausdrücke* bzw. *aussagenlogische Formeln* sind die Wörter über Σ_a, die mit folgenden Regeln abgeleitet werden können:
(A1) Jede aussagenlogische Variable ist ein Ausdruck.
(A2) Ist α ein Ausdruck, so auch $\neg\alpha$.
(A3) Sind α und β aussagenlogische Ausdrücke, so auch $(\alpha \wedge \beta)$ und $(\alpha \vee \beta)$.

Kurzschreibweise:

$$\frac{}{X} \cdot \frac{\alpha}{\neg\alpha} \cdot \frac{\alpha, \beta}{(\alpha \wedge \beta)} \cdot \frac{\alpha, \beta}{(\alpha \vee \beta)}$$

oder in BNF:

$$\mathrm{AA} := \mathrm{AV} | \neg \mathrm{AA} | (\mathrm{AA} \wedge \mathrm{AA}) | (\mathrm{AA} \vee \mathrm{AA})$$

Beispiel: $(\neg X_5 \vee (X_7 \wedge X_5))$ ist ein aussagenlogischer Ausdruck.

Begründung: Angabe einer Ableitung

(A1) X_5

(A1) X_7

(A2) $\neg X_5$

(A3) $(X_7 \wedge X_5)$

(A3) $(\neg X_5 \vee (X_7 \wedge X_5))$

Die einzelnen Schritte lassen sich auch in anderer Reihenfolge anwenden, d.h. die Ableitungen eines Ausdrucks ist nicht eindeutig.

2.2 Aussagenkalkül

Satz (Rekursive Definition): Eine rekursive Definition einer Menge M wird durch einen Kalkül bestehend aus rekursiven Regeln gegeben, d.h. Regeln der Gestalt:
Wenn $m_1, \ldots, m_r \in M$, so auch $m \in M$ kurz:

$$\frac{m_1, \ldots, m_r}{m}$$

Falls $r = 0$, so spricht man auch von einer *Basisregel* oder *Aussagenregel*. Sie hat also die Gestalt: $m \in M$:

$$\frac{leer}{m}$$

M ist dann die Menge aller Objekte, deren Zugehörigkeit zu M durch endlichmalige Anwendung der Regeln des Kalküls gezeigt werden kann.

Beispiel: Sei $\Sigma = \{a, b, \ldots, y, z\}$ und M gegeben durch Kalkül:

(R0) $\frac{leer}{\lambda}$

(R1) $\frac{w}{w\xi\eta}, \xi \in \{\text{a, e, o, i, u}\}, \eta \in \Sigma$

$axei \in M$:

1	λ	(R0)
2	ax	(R1) auf 1 mit $\xi = a$ und $\eta = x$
3	$axei$	(R1) auf 2 mit $\xi = e$ und $\eta = i$

Satz (Induktionsprinzip für rekursiv definierte Mengen):
Will man zeigen, dass alle Elemente einer Menge M, die mit den Regeln des Kalküls K ableitbar sind, eine Eigenschaft E haben, so genügt hierzu der Nachweis, dass für jede Regel

$$\frac{m_1, \ldots, m_r}{m}$$

des Kalküls K gilt: Wenn $m_1, \ldots, m_n$ in K ableitbar sind und die Eigenschaft E haben (*Induktionsvoraussetzung*), so hat auch m die Eigenschaft E. Im Fall $r = 0$ müssen wir also zeigen, dass m die Eigenschaft E hat (*Induktionsanfang*).

Satz (Induktion über dem Kalkül K):
Beweis durch Induktion über dem Ausdruckskalkül (über den Aufbau der Ausdrücke):
Um nachzuweisen, dass alle aussagenlogischen Ausdrücke eine Eigenschaft E haben, reicht es zu zeigen:
(I1) Jede Aussagenvariable hat die Eigenschaft E.
(I2) Hat der Ausdruck α die Eigenschaft E, so auch $\neg\alpha$.
(I3) Haben die Ausdrücke α und β die Eigenschaft E, so auch $(\alpha \wedge \beta)$ und $(\alpha \vee \beta)$.

Bemerkung: Definition aussagenlogischer Ausdrücke mittels Aussagenkalkül ohne Klammern

$$\overline{X} \qquad \frac{\alpha}{\neg\alpha} \qquad \frac{\alpha, \beta}{\alpha \wedge \beta} \qquad \frac{\alpha, \beta}{\alpha \vee \beta}$$

Definition (Belegung):

$$b : \{X_1, X_2, \ldots\} \to \{\underset{\text{falsch}}{0}, \underset{wahr}{1}\}$$

Vorschriften zur Erweiterung auf alle aussagenlogische Ausdrücke:

$$\text{(F1)} \quad b(\neg\alpha) := \begin{cases} 1 & \text{wenn } b(\alpha) = 0 \\ 0 & \text{wenn } b(\alpha) = 1 \end{cases}$$

$$\text{(F2)} \quad b(\alpha \wedge \beta) := \begin{cases} 1 & \text{wenn } b(\alpha) = 1 \text{ und } b(\beta) = 1 \\ 0 & \text{sonst} \end{cases}$$

$$\text{(F3)} \quad b(\alpha \vee \beta) := \begin{cases} 0 & \text{wenn } b(\alpha) = 0 \text{ und } b(\beta) = 0 \\ 1 & \text{sonst.} \end{cases}$$

Beispiel: Sei $b(X) = b(Y) = 0$ und $b(Z) = 1$. Was ist $b(X \wedge Y \vee Z)$?
Je nachdem, ob die linke oder rechte Seite zuerst ausgewertet wird, ergibt sich $b(X \wedge Y \vee Z) = 1$ bzw. $b(X \wedge Y \vee Z) = 1$.

$$b(X \wedge Y) \overset{(F2)}{=} 0 \qquad b(Y \vee Z) \overset{(F3)}{=} 1$$
$$b(X \wedge Y \vee Z) \overset{(F3)}{=} 1 \qquad b(X \wedge Y \vee Z) \overset{(F2)}{=} 0$$

Satz (Eindeutige Zerlegbarkeit):
Jeder Ausdruck $\alpha \in AA$ ist entweder ein Ausdruck der Gestalt
1 X_i oder
2 $\neg\beta$ oder
3 $(\beta \wedge \gamma)$ oder
4 $(\beta \vee \gamma)$.
Daher ist β in (2) und β, γ in (3) und (4) eindeutig bestimmt, d.h. es ist immer klar, welche Regel zuletz angewendet wurde. (es ergibt sich ein eindeutiger Syntaxbaum

Bemerkung:
In Fall (2) nennt man α die Negation von β
In Fall (3) nennt man α die Konjunktion von β und δ
In Fall (4) nennt man α die Disjunktion von β und δ

Beweis:

- α hat Gestalt (1) gdw. α beginnt mit AV.
- α hat Gestalt (2) gdw. α beginnt mit $\neg$.
- α hat Gestalt (3) bzw. (4) gdw. α beginnt mit (.

Angenommen $(\beta \wedge \gamma) = (\beta' \circ \gamma')$ mit $\circ \in \{\wedge, \vee\} \Rightarrow \beta = \beta', \wedge = \circ, \gamma = \gamma'$ □

Bemerkung: Jeder Ausdruck enthält genauso viele) wie (.

Beweis: Induktion über Ausdrücke

(I1) X_i enthält kein) und kein (.

(I2) $\neg\alpha$: Nach Induktionsvoraussetzung ist die Anzahl von) in $\alpha = n_\alpha$ und die Anzahl von (in $\alpha = n_\alpha$. $\Rightarrow$ Anzahl $\neg = n_\alpha$

(I3) $(\alpha \vee \beta)$: Nach Induktionsvoraussetzung ist die Anzahl von) in $\alpha = n_\alpha$ und die Anzahl von (in $\alpha = n_\alpha$, ebenso ist die Anzahl von) in $\beta = n_\beta$ und die Anzahl von (in $\beta = n_\beta$. $\Rightarrow$ Anzahl von) $= n_\alpha + n_\beta + 1 =$ Anzahl von (. □

Bemerkung: Für alle $\alpha, \beta \in$ AA gilt: α ist kein echtes Anfangsstück von β, d.h. es gibt kein $w \in \Sigma_a^*$ mit $w \neq \lambda$ und $\beta = \alpha w$.

Beweis: mittels Induktion im Aussagenkalkül: E trifft auf alle $x \in \Sigma_a^*$ zu, d.h. $\forall \alpha \in AA : E\alpha$ wobei E die Eigenschaft bezeichne, dass x kein echtes Anfangsstück von β und β ist kein echtes Anfangsstück von x ist, $\forall \beta \in$ AA

(I1) $\alpha = X_i$: Sei $\beta \in$ AA. α ist kein Anfangsstück von β, da jeder von X_i verschiedene Ausdruck nicht mit X_i beginnt. β ist kein Anfangsstück von α, da $|\alpha| = 1$ und $|\beta| \geq 1$ für jedes $\beta \in$ AA.

(I2) $\alpha = \neg\alpha'$ und E trifft auf α' zu: Sei $\beta \in$ AA und etwa

$$\alpha = \beta w.$$

Zu zeigen $w = \lambda$. Es gibt x mit $\beta = \neg x$. Somit β mit (A2) gewonnen, also gibt es $\gamma \in$ AA mit $\beta = \neg\gamma$. Somit $\neg\alpha' = \neg\gamma w$; daher $\alpha' = \gamma w$. Wegen $E\alpha'$: $w = \lambda$.

(I3) $\alpha = (\alpha_1 \vee \alpha_2)$ und E trifft auf α_1 und α_2 zu: Sei $\beta \in$ AA und etwa

$$\alpha = \beta w.$$

Zu zeigen $w = \lambda$. β beginnt mit (; somit existieren $\beta_1, \beta_2 \in$ AA und $\circ \in \{\wedge, \vee\}$ mit $\beta = (\beta_1 \circ \beta_2)$. Dann

$$\alpha_1 \vee \alpha_2) = \beta_1 \circ \beta_2) w$$

Wegen $E\alpha_1$: $\alpha_1 = \beta_1$ und somit $\circ = \vee$ und

$$\alpha_2) = \beta_2) w$$

Wegen $E\alpha_2$: $\alpha_2 = \beta_2$ und daher $w = \lambda$. □

Folgerung: Sind $\alpha_1, \ldots, \alpha_m, \beta_1, \ldots, \beta_k \in$ AA $\Rightarrow$ $\alpha_1\alpha_2 \ldots \alpha_m = \beta_1 \ldots \beta_k$

Beweis: per Induktion ergibt sich
$\alpha_1 = \beta_1,\ \alpha_2 \ldots \alpha_m = \beta_2 \ldots \beta_k \ \Rightarrow\ \alpha_1 = \beta_1,\ \alpha_2 = \beta_2,\ \alpha_3 \ldots \alpha_m = \beta_3 \ldots \beta_k \ \Rightarrow\ m = k\ ,\ \alpha_i = \beta_i$ □

Beispiel: Syntaxbaum für die Formel $(\neg X_5 \vee (X_7 \wedge X_5))$

Bemerkung: Wegen Bemerkung 2 ist es möglich, eine induktive Definitionen über den Aufbau der Ausdrücke durchzuführen. Um in eindeutiger Weise eine Funktion für alle Ausdrücke zu definieren, genügt es,

(D1) jeder Aussagenvariablen einen Wert zuzuordnen,

(D2) jedem Ausdruck $\neg\alpha$ einen Wert zuzuordnen unter der Annahme, dass dem Ausdruck α bereits ein Wert zugeordnet ist,

(D3) jedem Ausdruck $(\alpha_1 \wedge \alpha_2)$ bzw. $(\alpha_1 \vee \alpha_2)$ einen Wert zuzuordnen unter der Annahme, dass den Ausdrücken α_1 und α_2 bereits je ein Wert zugeordnet ist.

Beispiel:

1. Wir definieren den Rang (maximale Tiefe des zugehörigen Syntaxbaumes) einer Formel $\mathrm{rg} : \mathrm{AA} \to \mathbb{N}$ durch

$$\begin{aligned} \mathrm{rg}(X) &= 0 \\ \mathrm{rg}(\neg\alpha) &= 1 + \mathrm{rg}(\alpha) \\ \mathrm{rg}((\alpha \wedge \beta)) &= 1 + \max\{\mathrm{rg}(\alpha), \mathrm{rg}(\beta)\} \\ \mathrm{rg}((\alpha \vee \beta)) &= 1 + \max\{\mathrm{rg}(\alpha), \mathrm{rg}(\beta)\} \end{aligned}$$

2. Die Menge der Teilausdrücke oder Subformeln: $\mathrm{TA} : \mathrm{AA} \to \mathrm{Pot}(\mathrm{AA})$:

$$\begin{aligned} \mathrm{TA}(X) &= \{X\} \\ \mathrm{TA}(\neg\alpha) &= \mathrm{TA}(\alpha) \cup \{\neg\alpha\} \\ \mathrm{TA}((\alpha \wedge \beta)) &= \mathrm{TA}(\alpha) \cup \mathrm{TA}(\beta) \cup \{(\alpha \wedge \beta)\} \\ \mathrm{TA}((\alpha \vee \beta)) &= \mathrm{TA}(\alpha) \cup \mathrm{TA}(\beta) \cup \{(\alpha \vee \beta)\} \end{aligned}$$

Beispiel:

3 Die Anzahl der Junktoren einer Formel $f : \mathrm{AA} \to \mathbb{N}$ durch

$$\begin{aligned} f(X) &= 0 \\ f(\neg\alpha) &= 1 + \mathrm{rg}(\alpha) \\ f((\alpha \wedge \beta)) &= 1 \oplus \{f(\alpha), f(\beta)\} \\ f((\alpha \vee \beta)) &= 1 \oplus \{f(\alpha), f(\beta)\} \end{aligned}$$

Bemerkung: Mit $\mathrm{var}(\alpha)$ bezeichnen wir die Menge der Aussagenvariablen in α.

Beispiel: $\mathrm{var}((X \wedge Y) \vee (X \vee Z)) = \{X, Y, Z\}$

2.3 Semantik der Aussagenlogik

Die Trennung zwischen Syntax und Semantik wurde das erste Mal konsequent durchgeführt in Alfred Tarski (1933): "Der Wahrheitsbegriff in den formalisierten Sprachen".

Definition (n-stellige Boolsche Funktion):
Sei $n \geq 1$. Eine *n-stellige Wahrheitswertfunktion WW* ist eine Abbildung $\{0,1\}^n \to \{0,1\}$.

Beispiel: Definition von Wahrheitswertfuntkionen durch Wahrheitstabellen

- einstellige Wahrheitswertfuntkion: $\dot{\neg} : \{0,1\} \to \{0,1\}$
- zweistellige Wahrheitswertfuntkionen: $\dot{\wedge}, \dot{\vee}, \dot{\to}, \dot{\leftrightarrow} : \{0,1\} \times \{0,1\} \to \{0,1\}$
- $\dot{\to}$ wird als "wenn so" gelesen.
- Syntax entscheidend: $(X_6 \wedge \neg X_4) \neq (\neg X_4 \wedge X_6)$
- Ist $\alpha \in AA$, so schreiben wir für α auch $\alpha(Y_1, \ldots, Y_n)$ um anzudeuten, dass $var(\alpha) \subseteq \{Y_1, \ldots, Y_n\}$ und Y_i paarweise verschieden sind.

x	y	$\dot{\neg}$	$\dot{\wedge}$	$\dot{\vee}$	$\dot{\to}$	$\dot{\leftrightarrow}$
1	1	0	1	1	1	1
1	0		0	1	0	0
0	1	1	0	1	1	0
0	0		0	0	1	1

Definition (Belegung):
Eine (aussagenlogische) *Belegung* ist eine Abbildung b mit $\mathrm{df}(b) \subseteq \mathrm{AV}$ und $b : \mathrm{df}(b) \to \{0,1\}$.
Sei $\alpha \in AA$, b heißt Belegung für α, wenn $var(\alpha) \subseteq \mathrm{df}(b)$.
b heißt *total*, falls b auf allen AV definiert ist: $\mathrm{df}(b) = \mathrm{AV}$.

Definition:
Sei b eine Belegung. Durch Induktion über den Aufbau der Ausdrücke definieren wir $\tilde{b}(\alpha)$ für alle $\alpha \in \mathrm{AA}$ mit $\mathrm{var}(\alpha) \subseteq \mathrm{df}(b), b(\alpha) \in \{0,1\}$ wie folgt:

- $\tilde{b}(X) := b(X)$ für $X \in \mathrm{df}(b)$
- $\tilde{b}(\neg\alpha) := \dot{\neg}(b(\alpha))$
- $\tilde{b}((\alpha \wedge \beta)) := \dot{\wedge}(\tilde{b}(\alpha), \tilde{b}(\beta))$
- $\tilde{b}((\alpha \vee \beta)) := \dot{\vee}(\tilde{b}(\alpha), \tilde{b}(\beta))$

Bemerkung: Für $\tilde{b}(\alpha) = 1$ sagt man "α gilt bei b", "α erfüllt b" oder "b ist Modell von α".

Bemerkung: Wegen $\tilde{b}(X) = b(X)$ für $X \in AV$ schreiben wir auch $b(\alpha)$ statt $\tilde{b}(\alpha)$.

Beispiel: Sei $b(X) = b(Y) = 1$, $b(Z) = 0$, und $\mathrm{df}(b) = \{X, Y, Z\}$.
Dann gilt $b(((X \wedge Y) \vee Z)) = \dot{\vee}(\dot{\wedge}(b(X), b(Y)),\ b(Z)) = 0$
Da $U \notin \{X, Y, Z\} = df(b)$, ist $b((X \wedge U) \vee Z)$ nicht definiert.

Lemma (Koinzidenzlemma):
Seien b und b' Belegungen für $\alpha \in \mathrm{AA}$, also $\mathrm{var}(\alpha) \subseteq \mathrm{df}(b) \cap \mathrm{df}(b')$, und gelte für alle $X \in \mathrm{var}(\alpha)$: $b(X) = b'(X)$. Dann gilt $b(\alpha) = b'(\alpha)$

Beweis: Zeige durch Induktion über den Aufbau der Ausdrücke β:
Wenn $\mathrm{var}(\beta) \subseteq \mathrm{df}(b) \cap \mathrm{df}(b')$, so $b(\beta) = b'(\beta)$.

Definition: Für $n \geq 1$ ist $\mathrm{AA}_n := \{\alpha | \mathrm{var}(\alpha) \subseteq \{X_1, \ldots, X_n\}\}$.

Definition: Ist $\alpha \in \mathrm{AA}_n$ und sind $b_1, \ldots, b_n \in \{0, 1\}$, so steht $\alpha[b_1, \ldots, b_n]$ für den Wert $b(\alpha)$, wobei b irgendeine Belegung ist mit $b(X_1) = b_1, \ldots, b(X_n) = b_n$

Beispiel: $((X_2 \wedge \neg X_4) \vee X_1)[0, 1, 0, 1] = 0$, aber $((X_2 \wedge \neg X_4) \vee X_1)[0, 1, 0]$ ist nicht definiert.

Beispiel: $\alpha(X_2, X_4, X_1) = (\neg X_2 \wedge \neg X_4) \vee (X_1 \wedge \neg X_4) = 1$ für $\alpha[X_2 = 1, X_4 = 0, X_1 = 1] = \alpha[1, 0, 1]$

Bemerkung: Für jede Belegungen von b gilt: $b((\neg\alpha \vee \beta)) = \dot{\rightarrow}(b(\alpha), b(\beta)))$. Wir fassen daher $(a \rightarrow b)$ als Abkürzung für $(\neg\alpha \vee \beta)$ auf. Entsprechend schreiben wir $(a \leftrightarrow b)$ für $((\neg\alpha \vee \beta) \wedge (\neg\beta \vee \alpha))$.

Definition (Tautologie, Erfüllbarkeit, logische Äquivalenz):
Bei Belegungen b gilt hier stets Belegung b mit $df(b) = AV$.

1. α ist *allgemeingültig*/eine *Tautologie* / $\vDash \alpha \Leftrightarrow \alpha$ gilt bei allen Belegungen
2. α ist *erfüllbar* (Erf.α) $\Leftrightarrow$ es gibt eine Belegung b, die α erfüllt.
3. α und β sind *logisch äquivalent*, $(\alpha \equiv \beta) \Leftrightarrow \vDash (\alpha \leftrightarrow \beta)$,
   $\Leftrightarrow$. für alle totalen Belegungen b gilt $b(\alpha) = b(\beta)$.
4. Sei $\Gamma \subseteq \mathrm{AA}$, b eine Belegung. $b(\Gamma) = 1$ bedeutet: $b(\alpha) = 1$ für alle $\alpha \in \Gamma$.
5. Γ ist *erfüllbar*, Erf.Γ gdw. es gibt b mit $b(\Gamma) = 1$.
6. α *folgt aus* Γ, $\Gamma \models \alpha$ gdw. für alle Belegungen b gilt: wenn $b(\Gamma) = 1$, so $b(\alpha) = 1$.

Bemerkung: Sei b eine totale Belegung.

1. $(\alpha \vee \neg\alpha)$ ist allgemeingültig, denn für jedes b gilt $\{b(\alpha), b(\neg\alpha)\} = \{1, 0\}$ und somit $b((\alpha \vee \neg\alpha)) = 1$.
2. $\neg(\alpha \vee \neg\alpha)$ ist nicht erfüllbar.

3. X_5 ist erfüllbar, aber nicht allgemeingültig.

4. α ist allgemeingültig gdw. $\neg\alpha$ nicht erfüllbar.
   Entsprechend ist α erfüllbar gdw. $\neg\alpha$ nicht allgemeingültig.

5. $\Gamma \vDash \alpha$ gdw. nicht Erf. $\Gamma \cup \{\neg\alpha\}$, denn

$$\begin{aligned} \Gamma \vDash \alpha \quad &\Leftrightarrow \quad \text{Jede Belegung } b, \text{ die } \Gamma \text{ erfüllt, erfüllt } \alpha \\ &\Leftrightarrow \quad \text{Es gibt keine Belegung } b, \text{ die } \Gamma \cup \{\neg\alpha\} \text{ erfüllt} \\ &\Leftrightarrow \quad \Gamma \cup \{\neg\alpha\} \text{ ist nicht erfüllbar} \end{aligned}$$

   D.h. die Folgerung $\Gamma \vDash \alpha$ lässt sich auf das Erfüllbarkeitsproblem $\Gamma \cup \{\neg\alpha\}$ reduzieren

6. α und $\neg\neg\alpha$ sind logisch äquivalent.
   $(\alpha \wedge \beta)$ und $(\beta \wedge \alpha)$ sind äquivalent.

7. Sind α_1 und β_1 logisch äquivalent und α_2 und β_2 logisch äquivalent,
   so auch $\neg\alpha_1$ und $\neg\beta_1$
   und $(\alpha_1 \wedge \alpha_2)$ und $(\beta_1 \wedge \beta_2)$
   und $(\alpha_1 \vee \alpha_2)$ und $(\beta_1 \vee \beta_2)$

8. $(\alpha \wedge \beta)$ und $\neg(\neg\alpha \vee \neg\beta)$ sind logisch äquivalent.

9. Als er ein Kaninchen verfolgte, merkte der Hund, dass der Weg sich in drei Richtungen gabelte. Er beschnüffelte den 1. Weg und fand keine Spur. Dann beschnüffelte er den 2. Weg und fand keine Spur. Dann rannte er den 3. Weg (ohne ihn zu beschnüffeln).
   X_i Das Kaninchen wählte den i-ten Weg.

   Frage: $\{X_1 \vee (X_2 \vee X_3))\neg X_1, \neg X_2, \} \quad \vDash \quad X_3$?

X_1	X_2	X_3	$(X_1 \vee (X_2 \vee X_3))$	$\neg X_1$	$\neg X_2$	X_3
1	1	1	1	0	0	1
1	1	0	**1**	0	0	**0**
1	0	1	1	0	1	1
0	1	1	1	1	0	1
1	0	0	**1**	0	1	**0**
0	1	0	**1**	1	0	**0**
...						

10. Gilt $((\neg\neg\neg X \vee Y) \wedge (Z \vee \neg\neg X)) \equiv ((\neg X \vee Y) \wedge (Z \vee \neg\neg X))$? Ja. (s.u.)

11. Ersetzungslemma

    Intuitiv: Ersetzt man in α einen Teilausdruck β durch einen zu β logisch äquivalenten Ausdruck, so erhält man einen zu α logisch äquivalenten Ausdruck.

    Gelte $\alpha_1 \equiv \beta_1$ und $\alpha_2 \equiv \beta_2$, dann: $\neg\alpha_1 \equiv \neg\beta_1$, $(\alpha_1 \wedge \alpha_2) \equiv (\beta_1 \wedge \beta_2)$, $(\alpha_1 \vee \alpha_2) \equiv (\beta_1 \vee \beta_2)$

$$\begin{aligned} \neg\neg X &\equiv X \\ \neg\neg\neg X &\equiv \neg X \\ (\neg\neg\neg X \vee Y) &\equiv (\neg X \vee Y) \\ ((\neg\neg\neg X \vee Y) \wedge (Z \vee \neg\neg X)) &\equiv ((\neg X \vee Y) \wedge (Z \vee \neg\neg X)) \end{aligned}$$

12. D'Morgan: $(\alpha \wedge \beta) \equiv \neg(\neg\alpha \vee \neg\beta)$
13. Kann man aus $(X \wedge Y) \equiv \neg(\neg X \vee \neg Y)$ schließen, dass $(\alpha \wedge \beta) \equiv \neg(\neg\alpha \vee \neg\beta)$, $\forall \alpha, \beta \in AA$? Ja.
14. Für $^* : \mathrm{AA} \to \mathrm{AA}(\neg, \vee)$ und jedes $\alpha \in \mathrm{AA}$ gilt: $\alpha \equiv \alpha^* \quad$ und $\quad \mathrm{var}(\alpha) = \mathrm{var}(\alpha^*)$
    * wird induktiv definiert durch:

$$\begin{aligned} X^* &:= X \\ \neg\alpha^* &= \neg\alpha^* \\ (\alpha \wedge \beta)^* &:= \neg(\neg\alpha^* \vee \neg\beta^*) \\ (\alpha \vee \beta)^* &:= (\alpha^* \vee \beta^*) \end{aligned}$$

Beweis (Induktion über α):

- $|\neg\alpha^+| = 1 + |\alpha^+| =^{I.V.} 1 + 2|\alpha| \leq 2|\neg\alpha|$
- $|(\alpha \vee \beta)^*| \leq \ldots$
- $|(\alpha \wedge \beta)^*| \leq |\alpha^*| + |\beta^*| + 6 \leq 2|\alpha| + 2|\beta| + 6 = 2(|\alpha| + |\beta| + 3) = |(\alpha \wedge \beta)|$

□

Lemma (Substitutionslemma): Intiutiv: Ersetzt man in einer Äquivalenz überall $Y_1, \ldots, Y_n$ durch Ausdrücke $\gamma_1, \ldots, \gamma_n$, so bleibt die Äquivalenz erhalten.

$Y_1, \ldots, Y_n$ seien paarweise verschieden und $\gamma, \ldots, \gamma_n \in \mathrm{AA}$. Für $\alpha \in \mathrm{AA}$ definieren wir $\alpha\frac{\gamma_1 \cdots \gamma_n}{Y_1 \ldots Y_n}$ durch Induktion über α.

- $X\frac{\gamma_1 \ldots \gamma_n}{Y_1 \ldots Y_n} := \begin{cases} \gamma_i & \text{falls } X = Y_i \text{ und } i \in \{1, \ldots, n\} \\ X & \text{sonst} \end{cases}$
- $[\neg\alpha]\,\frac{\gamma_1 \ldots \gamma_n}{Y_1 \ldots Y_n} := \neg\left[\alpha\frac{\gamma_1 \ldots \gamma_n}{Y_1 \ldots Y_n}\right]$
- $(\alpha \circ \beta)\frac{\gamma_1 \ldots \gamma_n}{Y_1 \ldots Y_n} := \left(\alpha\frac{\gamma_1 \ldots \gamma_n}{Y_1 \ldots Y_n} \circ \beta\frac{\gamma_1 \ldots \gamma_n}{Y_1 \ldots Y_n}\right)$ für $\circ \in \{\wedge, \vee\}$.

Ist $\alpha \equiv \beta$, dann gilt

$$\alpha\frac{\gamma_1 \ldots \gamma_n}{Y_1 \ldots Y_n} = \beta\frac{\gamma_1 \ldots \gamma_n}{Y_1 \ldots Y_n} \quad .$$

Beweis: Für eine totale Belegung b sei b_{sub} die Belegung

$$b_{\mathrm{sub}}(X) := \begin{cases} b(\gamma_i) & \text{falls } X = Y_i \text{ und } i \in \{1, \ldots, n\} \\ b(X) & \text{sonst} \end{cases}$$

Wir zeigen:

$$\text{für alle } \delta \in \mathrm{AA}: \quad b\left(\delta\frac{\gamma_1\dots\gamma_n}{Y_1\dots Y_n}\right) = b_{\mathrm{sub}}(\delta). \tag{1}$$

Aus (1) ergibt sich das Substitutionslemma: Für totales b gilt nämlich:

$$b\left(\alpha\frac{\gamma_1\dots\gamma_n}{Y_1\dots Y_n}\right) \overset{(1)}{=} b_{\mathrm{sub}}(\alpha) \overset{\alpha\equiv\beta}{=} b_{\mathrm{sub}}(\beta)) \overset{(1)}{=} b\left(\beta\frac{\gamma_1\dots\gamma_n}{Y_1\dots Y_n}\right).$$

Der Nachweis von (1) erfolgt durch Induktion über γ:

Basisregel: $\delta = Y_i$ mit $1 \le i \le n$: Dann $Y_i\frac{\gamma_1\dots\gamma_n}{Y_1\dots Y_n} = \gamma_i$ und $b_{\mathrm{sub}}(Y_i) = b(\gamma_i)$; daher

$$b\left(Y_i\frac{\gamma_1\dots\gamma_n}{Y_1\dots Y_n}\right) = b(\gamma_i) = b_{\mathrm{sub}}(Y_i).$$

Ist $\delta = X$ und $X \notin \{Y_1,\dots,Y_n\}$, dann

$$b\left(X\frac{\gamma_1\dots\gamma_n}{Y_1\dots Y_n}\right) = b(X) = b_{\mathrm{sub}}(X).$$

□

Definition (Wahrheitswertfunktion von Ausdrücken): Sei $n \ge 1, \alpha \in \mathrm{AA}_n$. Dann ist die *durch α definierte n-stellige Wahrheitswertfunktion* $h_\alpha : \{0,1\}^n \to \{0,1\}$ gegeben durch

$$h_\alpha(b_1,\dots,b_n) := \alpha[b_1,\dots,b_n] \text{ für alle } b_1,\dots,b_n \in \{0,1\} \quad .$$

Beispiel: $h_{(X_1\wedge X_2)} = \dot\wedge, \quad h_{(X_1\vee X_2)} = \dot\vee, \quad h_{(X_1\to X_2)} = \dot\to, \quad h_{(X_1\neg X_2)} = \dot\neg$

Bemerkung: Wir verwenden die folgenden Kurzschreibweisen:

- $(\alpha_1 \wedge \dots \wedge \alpha_k)$ Abkürzung für $(\dots((\alpha_1 \wedge \alpha_2) \wedge \alpha_3)\dots \wedge \alpha_k)$, die sog. *iterierte Konjunktion.*
- $(\beta_1 \vee \dots \vee \beta_k)$ Abkürzung für $(\dots((\beta_1 \vee \beta_2) \vee \beta_3)\dots \vee \beta_k)$die sog. *iterierte Disjunktion.*
- α *Literal*: $\underbrace{\alpha}_{\text{pos. Literal}}$ $\alpha \in \mathrm{AV}$ oder $X \in \mathrm{AV}$ mit $\underbrace{\alpha}_{\text{neg. Literal}} = \neg X$
- α in *NNF* (*Negations-Normalform*): Jedes Negationszeichen in α steht vor einer Variable.
- α in *DNF* (*Disjunktive Normalform*): α iterierte Disjunktion von iterierten Konjunktionen von Literalen.
- α in *KNF* (*Konjunktive Normalform*): α iterierte Konjunktion von iterierte Disjunktionen von Literalen.

Beispiel:

$$\begin{aligned} (X \vee Y \vee (\neg Z \wedge Y \wedge X)) &\quad \text{DNF} \\ ((X \wedge \neg Y) \vee (\neg Z \wedge X \wedge Z)) &\quad \text{KNF} \\ (X \vee \neg Y \vee Z) &\quad \text{DNF und KNF} \end{aligned}$$

Bemerkung: Für $\alpha, \beta \in \mathrm{AA}_n$ gilt: $\alpha \equiv \beta \quad \Leftrightarrow \quad h_\alpha = h_\beta$

Satz: Sei $n \geq 1$. Zu jeder n-stelligen Boolschen Funktion $h : \{0,1\}^n \to \{0,1\}$ gibt es ein $\alpha \in \mathrm{AA}_n$ mit $h = h_\alpha$. α kann in DNF bzw. KNF gewählt werden.

Beweis: Wir betrachten Beispiele für den Fall $n = 2$.

X_1	X_2	h_1	h_2	h_3	
1	1	0	1	0	$\mathrm{h}_1 = DNF = \mathrm{X}_1 \vee X_2 =$ 'bin ich in der 2ten oder 3ten Zeile?'
1	0	1	1	0	$\mathrm{h}_2 = KNF = \mathrm{X}_1 \wedge X_2 =$ 'bin ich nicht in der 1ten oder 4ten Zeile?'
0	1	1	1	0	
0	0	0	1	0	

DNF: $\alpha_1 = ((X_1 \wedge \neg X_2) \vee (\neg X_1 \wedge X_2)), \qquad \alpha_2 = (X_1 \wedge \neg X_1), \qquad \alpha_3 = (X_1 \vee \neg X_1)$

KNF: $\beta_1 = ((\neg X_1 \vee \neg X_2) \wedge (X_1 \vee X_2)), \qquad \beta_2 = (X_1 \wedge \neg X_1), \qquad \beta_3 = (X_1 \vee \neg X_1)$

Nach diesem Verfahren können auch alle anderen Wahrheitswertfunktionen behandelt werden. □

Folgerung (1): Jeder Ausdruck ist zu einem Ausdruck in DNF und zu einem Ausdruck in KNF logisch äquivalent.

Beweis: Sei $\alpha \in \mathrm{AA}$: Wähle n mit $\alpha \in \mathrm{AA}_n$, dann ist h_α n-stellige WW-Funktion. Nach Satz existiert dann ein β (etwa in DNF) mit $h_\alpha = h_\beta$. Mit Bemerkung 1 gilt dann: α und β sind logisch äquivalent. □

Definition (n-Atom): Ausdrücke der Gestalt $(\lambda_1 \wedge \ldots \wedge \lambda_n)$ mit $(\lambda_i = X_i$ oder $\lambda_i = \neg X_i)$ für $i = 1, \ldots, n$ sind *n-Atome*.

Folgerung (2): Jedes erfüllbare $\alpha \in \mathrm{AA}_n$ ist zu einer Disjunktion von n-Atomen logisch äquivalent.

Folgerung (3): Für $n \geq 1$ gibt es genau $2^{(2^n)}$ paarweise nicht logisch äquivalenten Ausdrücke in AA_n.

Beweis: Sind A und B endliche Mengen, dann gilt

$$|\{f|f : A \to B\}| = |B|^{|A|} \quad .$$

Daher gilt hier:

$$|\{h|h : \{0,1\}^n \to \{0,1\}\}| = 2^{(2^n)}$$

□

Bemerkung:

$$\begin{aligned} \mathrm{AA} &= \mathrm{AA}(\neg, \wedge, \vee) \qquad (\text{genauer } \mathrm{AA}(\dot{\neg}, \dot{\wedge}, \dot{\vee})) \\ &= \mathrm{AA}(\neg, \vee) \\ &= \mathrm{AA}(\neg, \wedge, \vee, \to, \leftrightarrow) \end{aligned}$$

Eine andere Definition ist mit nullstelligen Wahrheitswertfunktionen möglich.

Syntax für $\mathrm{AA}(\top, \bot, \neg, \wedge, \vee)$:

$$\overline{X}, \overline{\top}, \overline{\bot}, \frac{\alpha}{\neg\alpha}, \frac{\alpha, \beta}{(\alpha \wedge \beta)}, \frac{\alpha, \beta}{(\alpha \vee \beta)}$$

Die Belegung wird dann erweitert um $b(\top) = 1$ und $b(\bot) = 0$.

Definition (Funktionale Vollständigkeit): Eine Menge H von WW-Funktionen ist *funktional vollständig*, wenn es für $n \geq 1$ und $h : \{0,1\}^n \to \{0,1\}$ ein $\alpha \in \mathrm{AA}(H)$ mit $h = h_\alpha$ gibt.

Beispiel:

1. $, \{\dot{\neg}, \dot{\vee}\}$ und damit $\{\dot{\neg}, \dot{\vee}, \dot{\wedge}\}$ sind funktional vollständig.
2. $H = \left\{\dot{|}\right\}$ (Sheffer-Strich) ist funktional vollständig.

		$\dot{\|}$
1	1	0
1	0	1
0	1	1
0	0	1

Begründung: $h_{(X_1|X_1)} = \dot{\neg}$ und für $\alpha = \left(\left(X_1\dot{|}X_1\right)\dot{|}\left(X_2\dot{|}X_2\right)\right)$ ist $h_\alpha = \dot{\vee}$,
d.h. mit dem Sheffer-Strich lassen sich NOT und OR erzeugen, mit denen alle anderen aussagenlogischen Ausdrücke erzeugt werden können. Sheffer-Strich als Kalkül: $AA(|) : \ \frac{}{\overline{X}}, \frac{\alpha,\beta}{\alpha|\beta}$

Bei n Aussagenvariablen hat die Wahrheitstafel 2^n Zeilen. Verfahren mit Wahrheitstafeln sind daher sehr ineffizient.

Definition: Ein Ausdruck ist in *Negationsnormalform* (NNF), wenn in ihm Negationszeichen höchstens unmittelbar vor Variablen auftreten.

Beispiel: $\neg(X \wedge \neg Y) \notin \text{NNF}, \quad ((\neg X \vee Y) \wedge (Z \wedge \neg U)) \in \text{NNF}, \quad \neg\neg U \notin \text{NNF}$

Bemerkung: Jeder Ausdruck in DNF oder in KNF ist in NNF.

Bemerkung: Für die folgende induktiv definierte Abbildung $^* : \text{AA} \to \text{AA}$ gilt:

1. Für $\alpha \in \text{AA}$ ist $\alpha \equiv \alpha^*$ und α^* in NNF.
2. * ist in polynomieller Zeit berechenbar.

Wir definieren * durch:

- $X^* := X$
- $[\neg\alpha] := \begin{cases} \neg Y & \text{wenn } \alpha = Y \\ \beta^* & \text{wenn } \alpha = \neg\beta \\ ([\neg\beta]^* \vee [\neg\gamma]^*) & \text{wenn } \alpha = (\beta \wedge \gamma) \\ ([\neg\beta]^* \wedge [\neg\gamma]^*) & \text{wenn } \alpha = (\beta \vee \gamma) \end{cases}$
- $(\beta \wedge \gamma)^* := (\beta^* \wedge \gamma^*)$
- $(\beta \vee \gamma)^* := (\beta^* \vee \gamma^*)$

DNF bzw. KNF erstellen:

1. Zunächst von α nach α^* in NNF.
2. $\neg$ auflösen
3. Dann die Regel $(\alpha \wedge (\beta \vee \gamma)) = ((\alpha \wedge \beta) \vee (\alpha \wedge \gamma))$
   bzw. die entsprechend für $((\beta \vee \gamma) \wedge \alpha)$ anwenden.

Beispiel (Umwandlung in DNF):

$$(\neg X \wedge (Y \vee (Z \wedge U))) \rightsquigarrow ((\neg X \wedge Y) \vee (\neg X \wedge (Z \wedge U))) \rightsquigarrow ((\neg X \wedge Y) \vee (\neg X \wedge Z \wedge U))$$

3 Die Komplexität des Erfüllbarkeitsproblems

3.1 Der Endlichkeitssatz

Definition (Erfüllb.-Problem der Aussagenlogik): SAT $:= \{\alpha \in AA \mid \alpha \in \text{KNF}, \text{Erf. } \alpha\} \subseteq \Sigma_a^*$

Satz: SAT ist NP-vollständig, d.h. SAT $\in$ NPTIME.

Definition: $\mathrm{SAT}_0 := \{\alpha \in \mathrm{AA} \mid \text{Erf. } \alpha\}$

Folgerung: SAT_0 ist NP-vollständig, d.h. $\mathrm{SAT}_0 \in \mathrm{NPTIME}$.

Bemerkung: SAT und SAT_0 ist entscheidbar.

Bemerkung: $\{\alpha \in AA \mid \alpha \in \mathrm{DNF}, \text{Erf. } \alpha\} \in PTIME$,
d.h. die Erfüllbarkeit eines aussagenlogischen Ausdruckes in DNF ist in polynomieller Zeit entscheidbar und kann durch $p(|\alpha|)$ beschränkt werden: $p \in \mathbb{N}[X]$ Zeit: $\leq p(|\alpha|)$.

Beweis:

(1) Sei $\beta = (\lambda_1 \wedge \ldots \wedge \lambda_k)$ mit λ_j Literale. Dann Erf. β gdw. für alle $X \in \mathrm{var}(\beta)$ gilt: X und $\neg X$ kommen nicht beide unter den λ_i vor.

(2) Sei $\alpha = (\beta_1 \vee \ldots \vee \beta_s)$. Dann Erf. α gdw. ex. $i \in \{1, \ldots, s\} = [s]$ mit Erf. β_i.

Mit (1) und (2) folgt die Behauptung. □

Bemerkung: Seien $Y_i, \ldots, Y_n, Z_1, \ldots, Z_n$ paarweise verschiedene Variable. Jede zu

$$((Y_1 \leftrightarrow Z_1) \wedge (Y_2 \leftrightarrow Z_2) \wedge \ldots \wedge (Y_n \leftrightarrow Z_n))$$

logisch äquivalente Formel in DNF ist eine Disjunktion von mindestens 2^n iterierten Konjunktionen.

Satz (Endlichkeitssatz): Sei $\Gamma \subseteq \mathrm{AA}$. Ist jede endliche Teilmenge von Γ erfüllbar, so auch Γ.

Beweis: Sei $b_1, \ldots, b_n \in \{0, 1\}$ und jede endliche Teilmenge von Γ erfüllbar. Gesucht ist nun eine totale Belegung b mit $b(\Gamma) = 1$.

Wir nennen $b_1, \ldots, b_n$ gut, falls für jede endliche Teilmenge Γ_0 von Γ es eine totale Belegung b^0 mit $b(\Gamma_0) = 1$ und $b(X_1) = b_1, \ldots, b(X_n) = b_n$ gibt.

1. Die leere Folge ($n = 0$) ist gut.
2. $b_1, \ldots, b_n$ gut, $\alpha \in \Gamma \cap \mathrm{AA}_n \Rightarrow \alpha[b_1, \ldots, b_n] = 1$
   (setze $\Gamma_0 = \{\alpha\}$ in der Definition von "gut").
3. $b_1, \ldots, b_n$ gut $\Rightarrow$b$_1, \ldots, b_n, 0$ oder $b_1, \ldots, b_n, 1$ gut.

Wir definieren $b : \mathrm{AV} \to \{0, 1\}$ indem wir durch Induktion über $i \geq 0$ den Wert von $b(X_i)$ so festlegen, dass $b(X_1), \ldots, b(X_n)$ immer gut ist. Für $i = 0$ bedeutet das, dass die leere Folge gut ist. Außerdem setzen wir

$$b(X_{i+1}) = \begin{cases} 1 & \text{falls } b(X_1), \ldots b(X_i), 1 \text{ gut} \\ 0 & \text{sonst} \end{cases} .$$

Wegen (3) ist $b(X_1), \ldots b(X_{i+1})$ gut.

Behauptung: $b(\Gamma) = 1$.
Beweis: Sei $\alpha \in \Gamma$, etwa $\alpha \in \mathrm{AA}_n \Rightarrow$ nach (2)$\alpha[b(X_1), \ldots b(X_n)]$ =b$(\alpha) = 1$.

Beweis zu (3): Sei $b_1, \ldots, b_n$ gut. Sei (3) falsch, d.h. wir nehmen an, dass $b_1, \ldots, b_n, 0$ und $b_1, \ldots, b_n, 1$ beide nicht gut sind. $\Rightarrow$ endliches $\Gamma_0 \subseteq \Gamma$, so dass für alle totalen Belegungen b mit $b(\Gamma_0) = 1$ und $b(X_1) = b_1, \ldots b(X_n) = b_n$ gilt: $b(X_{n+1}) = 1 \Rightarrow$ endliches Teilmenge Γ_1 mit $b(X_{n+1}) = 0$.

Wir betrachten $\Gamma_0 \cup \Gamma_1$ endlich und $b_1, \ldots b_n$ gut. Es gibt eine totale Belegung b mit $b(\Gamma_0 \cup \Gamma_1) = 1$ mit $b(X_i) = b_i, i \in \{1, \ldots, n\}$. Ist $b(X_{n+1}) = 1$, dann ist das ein Widerspruch zur Existenz von Γ_1. Entsprechend ist $b(X_{n+1}) = 0$ ein Widerspruch zur Existenz von Γ_0.

Damit gibt es genau ein b_0 mit $b(\Gamma_0 \wedge \Gamma_1) = 1$ und $b(X_i) = b_i$ für $i \in \{1, \ldots, n\}$. □

Bemerkung: Lassen wir beim Aufbau der Ausdrücke der Aussagenlogik eine beliebige Menge $\{X_i \mid i \in I\}$ von Aussagenvariablen zu ($z.B. I = \mathbb{R}$), so gilt weiterhin der Endlichkeitssatz.

Folgerung: Wenn $\Gamma \vDash \alpha$, so existiert eine endliche Teilmenge $\Gamma_0 \subseteq \Gamma$ mit $\Gamma_0 \vDash \alpha$.

3.2 “Allgegenwart” der Aussagenlogik

3.2.1 Logische Folgerungen

Folgt α logisch aus Γ: $\Gamma \vDash^? \alpha$? (Beispiele Seite 5)

Sachverhalte $A_1, \ldots, A_n, B$. Ergibt sich B aus $A_1, \ldots, A_n$? Versuch: $\{\alpha_1, \ldots, \alpha_n\} \vDash \beta$.

Lemma (Deduktionslemma): $\{\alpha_1, \ldots, \alpha_n\} \vDash \beta$ gdw. $\vDash ((\alpha_1 \wedge \ldots \wedge \alpha_n) \rightarrow \beta)$, d.h. $((\alpha_1 \wedge \ldots \wedge \alpha_n) \rightarrow \beta)$ ist allgemeingültig.

Beweis:

“$\Rightarrow$”: Sei b eine Belegung und $b((\alpha_1 \wedge \ldots \wedge \alpha_n)) = 1$, also $b(\{\alpha_1, \ldots, \alpha_n\}) = 1$. Nach Voraussetzung ist dann $b(\beta) = 1$.

“$\Leftarrow$”: Sei b eine Belegung mit $b(\{\alpha_1, \ldots, \alpha_n\}) = 1$. Dann ist $b((\alpha_1 \wedge \ldots \wedge \alpha_n)) = 1$. Da $((\alpha_1 \wedge \ldots \wedge \alpha_n) \rightarrow \beta) = 1$ gilt, ist dann auch $b(\beta) = 1$. □

Einsatz bspw. in Diagnoseverfahren:
Wissensbasis $= \{(X_1 \rightarrow (Y_1 \vee Y_2 \vee \cdots \vee Y_n)), (\text{kein Benzin} \rightarrow (\text{Motor springt nicht an, Auto bleibt stehen}))\}$
Daten/Diagnose: $\beta_1, \ldots, \beta_r$, kein Benzin
Frage: $\Gamma \vDash Y_5$

3.2.2 Verifikation von Schaltungen

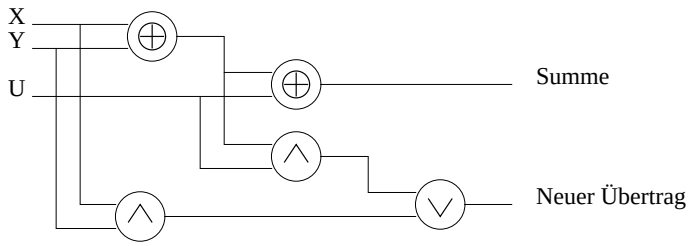
Beispiel: 1-Bit-Addierer mit Eingängen X, Y, Übertrag U und Ausgängen $S \in \{0, 1\}$ für die Summe und N für den Ausgangsübertrag.

X	Y	U	S	N
1	1	1	1	1
1	1	0	0	1
	⋮			

Das gewünschte Verhalten wird beschrieben durch:

$$\beta_S = ((X \wedge Y \wedge U) \vee (X \wedge \neg Y \wedge \neg U) \vee (\neg X \wedge Y \wedge \neg U) \vee (\neg X \wedge \neg Y \wedge U))$$
$$\beta_N = ((X \wedge Y \wedge U) \vee (X \wedge \neg Y \wedge U) \vee (\neg X \wedge Y \wedge U))$$

Realisierung durch folgenden Schaltkreis mit AND, OR, XOR und NOT.



Beschreibung *des* Verhaltens dieser Schaltung

$$\alpha_S = ((X \oplus Y) \oplus U)$$
$$\alpha_N = ((U \wedge (X \oplus Y)) \vee (X \wedge Y))$$

Verifikationsproblem: Gilt $\alpha_S \equiv \beta_S$ und $\alpha_N \equiv \beta_N$?

Bemerkung (XOR): $(a \oplus b)$ ist Abkürzung für $((\alpha \wedge \neg\beta)\ vee(\neg\alpha \wedge \beta))$.

Beispiel: Sei S ein Schaltkreis mit mit den Eingängen E_1, E_2 und Ausgängen A_1, A_2, der die folgende Funktion realisiert.

E_1	E_2	A_1 (NAND)	A_2 (AND)
1	1	0	1
1	0	1	0
0	1	1	0
0	0	1	0

Kombination von drei Schaltkreisen: (picture not embedded yet)
Eingangsvariablen X_1, X_2; Ausgangsvariablen Y_1, Y_2

Verhaltensbeschreibung: $(Y_1 \leftrightarrow \neg(X_1 \wedge X_2)), (Y_2 \leftrightarrow (X_1 \wedge X_2))$
Schaltelement K mit Eingängen X_1, X_2 und Ausgängen Y_1, Y_2, Y_3, Y_4
Ein- und Ausgänge der einzelnen Schaltelemente von K:
$X_1^u, X_2^u, Y_1^u, Y_2^u, X_1^m, X_2^m, Y_1^m, Y_2^m, X_1^o, X_2^o, Y_1^o, Y_2^o$

3.2.3 Fehlertolerante Datenübertragung

Erfahrung: Bei der Übertragung von 2^{n-1} Bits kommen weniger als ein Viertel falsch an, also weniger als 2^{n-3}.

Einfache Lösung: Jedes Bit wird 2^{n-1} Mal gesendet.