7 Logik

7.1 Aussagenlogik

7.1.1 Literatur

Literatur

Martin Kreuzer, Stefan Kühling.

Logik für Informatiker,

Pearson Studium.

Die folgenden Abschnitte wurden großteils aus dem Buch *Logik für Informatiker* übernommen.

7.1.2 Grundlagen

Syntax der Aussagenlogik

Definition 7.1. Formel

1. Eine **atomare Formel** ist von der Form A_i mit $i \in \mathbb{N}$, d.h. atomare Formeln sind nur die einfachen Aussagen.
2. Eine beliebige Formel entsteht induktiv aus atomaren Formeln, wobei die folgenden Schritte erlaubt sind:
   - Jede atomare Formel ist eine Formel.
   - Sind F, G zwei Formeln, so sind auch $F \wedge G$ sowie $F \vee G$ Formeln.
   - Für jede Formel F ist auch $\neg F$ eine Formel.

$F \wedge G$ nennt man **Konjunktion** von F und G. $F \vee G$ nennt man **Disjunktion** von F und G. $\neg F$ nennt man **Negation** von F.

Notationen

Für Aussagen verwenden wir statt $A_0, A_1, A_2, \ldots$ auch $A, B, C, \ldots$. Seien nun Formeln $F_1, F_2, F_3, \ldots$ gegeben:

1. Für $(\neg F_1 \vee F_2)$ schreiben wir auch $(F_1 \Rightarrow F_2)$. Wir nennen $F_1 \Rightarrow F_2$ auch eine **Folgerung** oder **Implikation**.
2. Für $(F_1 \wedge F_2) \vee (\neg F_1 \wedge \neg F_2)$ schreiben wir auch $(F_1 \Leftrightarrow F_2)$. Wir nennen $F_1 \Leftrightarrow F_2$ eine **Äquivalenz**.
3. Für $(\cdots((F_1 \vee F_2) \vee F_3) \vee \cdots \vee F_n)$ schreiben wir auch $\bigvee\limits_{i=1}^{n} F_i$.
4. Für $(\cdots((F_1 \wedge F_2) \wedge F_3) \wedge \cdots \wedge F_n)$ schreiben wir auch $\bigwedge\limits_{i=1}^{n} F_i$.

Semantik der Aussagenlogik (1/2)

Definition 7.2. Belegungen

1. Die Elemente der Menge $\{wahr, falsch\}$ heißen die **Wahrheitswerte**. Wir schreiben auch 1 statt *wahr* und 0 statt *falsch*.

2. Sei M eine Menge von atomaren Formeln. Eine **Belegung** von M ist eine Abbildung
$$\alpha : M \to \{0,1\}$$

3. Sei $\widehat{M}$ die Menge aller Formeln, die mit Hilfe der atomaren Formeln in M gebildet werden können, und sei $\alpha : M \to \{0,1\}$ eine Belegung. Dann erweitern wir α zu einer Abbildung $\widehat{\alpha} : \widehat{M} \to \{0,1\}$ gemäß den folgenden Vorschriften.

Semantik der Aussagenlogik (2/2)

Für $\widehat{\alpha} : \widehat{M} \to \{0,1\}$ gilt:

1. Für atomare Formeln $A \in M$ gilt $\widehat{\alpha}(A) = \alpha(A)$.
2. Für Formeln $F, G \in \widehat{M}$ gilt:
$$\widehat{\alpha}(F \wedge G) = \begin{cases} 1, & \text{falls } \widehat{\alpha}(F) = 1 \text{ und } \widehat{\alpha}(G) = 1, \\ 0, & \text{sonst.} \end{cases}$$
3. Für Formeln $F, G \in \widehat{M}$ gilt:
$$\widehat{\alpha}(F \vee G) = \begin{cases} 1, & \text{falls } \widehat{\alpha}(F) = 1 \text{ oder } \widehat{\alpha}(G) = 1, \\ 0, & \text{sonst.} \end{cases}$$
4. Für Formel $F \in \widehat{M}$ gilt:
$$\widehat{\alpha}(\neg F) = \begin{cases} 1, & \text{falls } \widehat{\alpha}(F) = 0 \\ 0, & \text{sonst.} \end{cases}$$

Im Folgenden schreiben wir der Einfachheit halber α statt $\widehat{\alpha}$. Ist eine Belegung der in einer Formel vorkommenden Aussagensymbole gegeben, so ist der Wahrheitswert der Formel gemäß dieser Definitionen leicht zu ermitteln.

Modelle und Erfüllbarkeit

Definition 7.3. Sei F eine aussagenlogische Formel und sei $\alpha : M \to \{0,1\}$ eine Belegung, dann:

1. Sind alle in F vorkommenden atomaren Formeln in M enthalten, so heißt α zu F **passend**.

2. Ist α zu F passend und gilt $\alpha(F) = 1$, so schreiben wir $\alpha \models F$. Wir sagen, dass F unter der Belegung α gilt und nennen α ein **Modell** für F.

3. Ist $\mathcal{F}$ eine Menge aussagenlogischer Formeln, so heißt α ein **Modell** für $\mathcal{F}$, wenn für alle $F \in \mathcal{F}$ gilt: $\alpha \models F$. In diesem Fall schreiben wir $\alpha \models \mathcal{F}$.

4. Eine Menge $\mathcal{F}$ von Formeln heißt **erfüllbar**, falls $\mathcal{F}$ mindestens ein Modell besitzt. Ansonsten heißt $\mathcal{F}$ **unerfüllbar**.

5. Eine Formel F heißt **allgemein gültig** oder auch **Tautologie**, wenn jede zu F passende Belegung ein Modell für F ist.

Äquivalenzen der Aussagenlogik

Theorem 7.4. *Die fundamentalen Äquivalenzen der Aussagenlogik Für aussagenlogische Formeln F, G, H gelten die folgenden Äquivalenzen:*

1. $(F \vee F) \equiv F$, *sowie* $(F \wedge F) \equiv F$ ***(Idempotenz)***

2. $(F \vee G) \equiv (G \vee F)$, *sowie* $(F \wedge G) \equiv (G \wedge F)$ ***(Kommutativität)***

3. $((F \vee G) \vee H) \equiv (F \vee (G \vee H))$, *sowie* $((F \wedge G) \wedge H) \equiv (F \wedge (G \wedge H))$ ***(Assoziativität)***

4. $(F \vee (F \wedge G)) \equiv F$, *sowie* $(F \wedge (F \vee G)) \equiv F$ ***(Absorption)***

5. $(F \vee (G \wedge H)) \equiv ((F \vee G) \wedge (F \vee H))$ ***(Distributivität)***

6. $(F \wedge (G \vee H)) \equiv ((F \wedge G) \vee (F \wedge H))$ ***(Distributivität)***

7. $\neg\neg F \equiv F$ ***(Doppelnegation)***

8. $\neg(F \wedge G) \equiv (\neg F \vee \neg G)$, *sowie* $\neg(F \vee G) \equiv (\neg F \wedge \neg G)$ ***(de Morgansche Regeln)***

9. *Ist F Tautologie, so gilt* $(F \vee G) \equiv F$, *sowie* $(F \wedge G) \equiv G$ ***(Tautologieregeln)***

10. *Ist F unerfüllbar, so gilt* $(F \vee G) \equiv G$, *sowie* $(F \wedge G) \equiv F$ ***(Unerfüllbarkeitsregeln)***

Normalformen

Definition 7.5. Ein **Literal** ist eine atomare Formel oder die Negation einer atomaren Formel. Im ersten Fall sprechen wir von einem **positiven Literal**. im zweiten Fall von einem **negativem Literal**.

1. Eine Formel F ist in **konjunktiver Normalform (KNF)**, falls sie eine Konjunktion von Disjunktionen von Literalen ist. Mit anderen Worten, es muss Literale L_{ij} geben, so dass F von folgender Form ist:

$$F = (L_{11} \vee \cdots \vee L_{1m_1}) \wedge \cdots \ldots (L_{n1} \vee \cdots \vee L_{nm_n})$$

2. Eine Formel F ist in **disjunktiver Normalform (DNF)**, falls sie eine Disjunktion von Konjunktionen von Literalen ist, d.h. falls F von folgender Form ist:

$$F = (L_{11} \wedge \cdots \wedge L_{1m_1}) \vee \cdots \vee (L_{n1} \wedge \cdots \wedge L_{nm_n})$$

Algorithmus Erzeugung KNF

Definition 7.6. Gegeben sei eine Formel F. Führe die folgenden Schritte durch:

1. Eliminiere $\Rightarrow$ und $\Leftrightarrow$ mittels ihrer Definitionen.
2. Ersetze in F jede Teilformel der Form $\neg\neg G$ durch G.
3. Ersetze in F jede Teilformel der Form $\neg(G \wedge H)$ durch $(\neg G \vee \neg H)$. Entsteht hierdurch eine Teilformel der Form $\neg\neg K$, so wende Schritt 2) an.
4. Ersetze in F jede Teilformel der Form $\neg(G \vee H)$ durch $(\neg G \wedge \neg H)$. Entsteht hierdurch eine Teilformel der Form $\neg\neg K$, so wende Schritt 2) an.
5. Wiederhole die Schritte 3) und 4) so oft wie möglich.
6. Ersetze in F jede Teilformel der Form $(G \vee (H \wedge I))$ durch $((G \vee H) \wedge (G \vee I))$.
7. Ersetze in F jede Teilformel der Form $((G \wedge H) \vee I)$ durch $((G \vee I) \wedge (H \vee I))$.
8. Wiederhole die Schritte 6) und 7) so oft wie möglich.

Die resultierende Formel ist dann in KNF.

KNF zu 3-KNF

Definition 7.7. Gegeben sei eine Formel F in KNF $F = \bigwedge_i (L_{i1} \vee L_{i2} \vee \ldots \vee L_{i,m_i})$. Führe die folgenden Schritte durch um eine erfüllbarkeitsäquivalente Formel zu bekommen, in der in jeder Disjunktion maximal drei Literale stehen.

1. Sei $F = K_1 \wedge K_2 \wedge \ldots \wedge K_n$ eine Formel mit n Disjunktionen.
2. Betrachte nacheinander alle Disjunktionen K_i mit $1 \leq i \leq n$.
3. Solange die Disjunktion noch mehr als drei Literale enthält:
   - Generiere neue Variable X.
   - Erstelle neue Disjunktion $K_i' = (L_{i1} \vee L_{i2} \vee X)$.
   - Ersetze $K_i = (\neg X \vee L_{i3} \vee \ldots \vee L_{im_i})$.

Die resultierende Formel ist dann in 3-KNF.

Definition 7.8. Gegeben sei eine aussagenlogische Formel F in KNF mit Literalen L_{ij}.

$$F = \\ (L_{11} \vee L_{12} \vee \ldots \vee L_{1\,m_1}) \wedge \ldots \wedge (L_{n1} \vee L_{n2} \vee \ldots \vee L_{n\,m_n})$$

Die **Klausel** $K_1, K_2, \ldots, K_n$ von F sind die Mengen der Literale der einzelnen Disjunktionen.

$$\begin{aligned} &K_1 = \{L_{11}, L_{12}, \ldots, L_{1\,m_1}\} \\ &K_2 = \{L_{21}, L_{22}, \ldots, L_{2\,m_2}\} \\ &\ldots \\ &K_n = \{L_{n1}, L_{n2}, \ldots, L_{n\,m_n}\} \end{aligned}$$

Die F zugeordnete **Klauselmenge** $\mathcal{K}(F)$ ist die Menge der Klauseln von F.

$$\mathcal{K}(F) = \{K_1, K_2, \ldots, K_n\}$$

Observation 7.9. *Verschiedene Formeln in KNF können dieselbe Klauselmenge haben.*

Aufgrund von Idempotenz und Kommutativität sind Formel mit derselben Klauselmenge äquivalent.

Example 7.10. Die folgenden Formeln haben die Klauselmenge $\{\{A\}, \{B, \neg C\}\}$ und sind nicht identisch, aber äquivalent.

$$F = (A \vee A) \wedge (B \vee \neg C)$$
$$G = A \wedge (\neg C \vee B)$$
$$H = A \wedge (B \vee \neg C \vee B)$$

7.1.3 Das Resolutionskalkül der Aussagenlogik

Lemma 7.11. *Sei F eine aussagenlogische Formel und $\mathcal{K}(F)$ ist die zugeordnete Klauselmenge.*

Dann ist F äquivalent zu jeder Formel G, für die gilt $\mathcal{K}(G) = \mathcal{K}(F) \cup \{R\}$, wobei R eine Resolvente zweier Klausel $K_1, K_2 \in \mathcal{K}(F)$ ist.

Beweis. Sei $\alpha : M \mapsto \{0, 1\}$ eine zu F passende Belegung (d.h. alle atomaren Formeln erhalten einen Wahrheitswert), dann ist α auch eine passende Belegung für G.

Fall 1

Es gilt $\alpha(F) = 0$, dann gibt es eine Klausel $K = \{L_1, \ldots, L_m\}$ in $\mathcal{K}(F)$, für die gilt $\alpha(L_i) = 0$ für alle $i \in \{1, \ldots m\}$.

Da $\mathcal{K}(F) \subset \mathcal{K}(G)$ ist die Klausel K, aus der $\alpha(F) = 0$ folgt, auch eine Klausel von G und daraus folgt $\alpha(G) = 0$.

Fall 2

Es gilt $\alpha(F) = 1$, dann gibt es in jeder Klausel $K_1, \ldots, K_n$ mindestens ein $L_i \in K_i$ mit $\alpha(L_i) = 1$.

Angenommen $L \in K_1$ und $\neg L \in K_2$. Es gilt $\alpha(L) \neq \alpha(\neg L)$, daraus folgt entweder in K_1 oder in K_2 gibt es ein Literal $L' \neq L$ mit $\alpha(L') = 1$.

$R = K_1 \backslash \{L\} \cup K_2 \backslash \{\neg L\}$, d.h. $L' \in R$ mit $\alpha(L') = 1$.

Aus $\mathcal{K}(G) = \mathcal{K}(F) \cup \{R\}$ folgt $\alpha(G) = 1$. □

Example 7.12. $F \equiv (P \vee S) \wedge (\neg P \vee \neg S) \wedge (\neg S \vee A) \wedge (\neg A \vee P) \wedge P \wedge S$

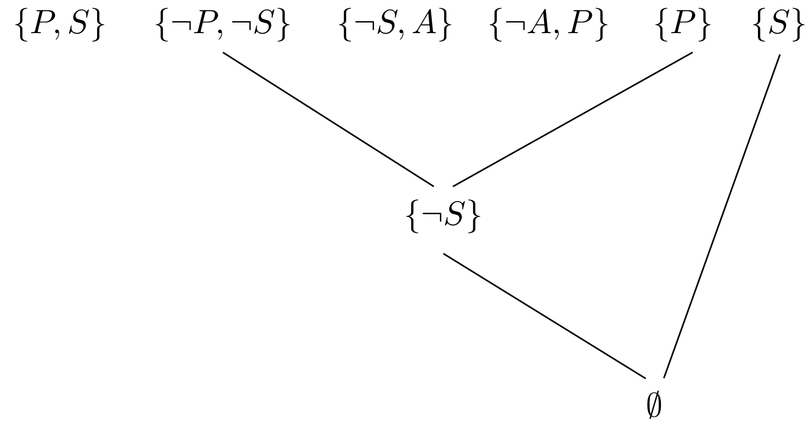
Klauselmenge

$\mathcal{K}(F) = \Big\{\{P, S\}, \{\neg P, \neg S\}, \{\neg S, A\}, \{\neg A, P\}, \{P\}, \{S\}\Big\}$

Resolventenmengen

$$
\begin{aligned}
\operatorname{Res}^0(\mathcal{K}(F)) =& \mathcal{K}(F)\\
\operatorname{Res}^1(\mathcal{K}(F)) =& \operatorname{Res}^0(\mathcal{K}(F)) \cup \\
& \Big\{\{P,\neg P\},\{P,A\},\{\neg A,\neg S\},\{S,\neg S\},\{\neg S\},\{\neg P\},\{P,\neg S\},\{A\}\Big\}\\
\operatorname{Res}^2(\mathcal{K}(F)) =& \operatorname{Res}^1(\mathcal{K}(F))\ \cup \Big\{\{\neg A\},\emptyset,\dots\Big\}
\end{aligned}
$$

Example 7.13. Das Beispiel grafisch.



7.1.4 KNF/DNF Nachtrag

Sei F eine aussagenlogische Formel in **KNF** mit Literalen L_{ij}.

$F = (L_{11} \vee L_{12} \vee \ldots \vee L_{1\,m_1}) \wedge (L_{21} \vee L_{22} \vee \ldots \vee L_{2\,m_2}) \wedge \ldots \wedge (L_{n1} \vee L_{n2} \vee \ldots \vee L_{n\,m_n})$

F ist eine **Tautologie**, wenn es in **jeder** Disjunktion (Klausel) Literale J, K gibt, für die gilt $J = \neg K$.

Sei G eine aussagenlogische Formel in **DNF** mit Literalen L_{ij}.

$G = (L_{11} \wedge L_{12} \wedge \ldots \wedge L_{1\,m_1}) \vee (L_{21} \wedge L_{22} \wedge \ldots \wedge L_{2\,m_2}) \vee \ldots \vee (L_{n1} \wedge L_{n2} \wedge \ldots \wedge L_{n\,m_n})$

G ist genau dann **unerfüllbar**, wenn es in **jeder** Konjuktion Literale J, K gibt, für die gilt $J = \neg K$.

Sowohl DNF als auch KNF werden im schlechtesten Fall (*worst case*) exponentiell groß, d.h. es gibt in der Anzahl der atomaren Formeln exponentiell viele Konjunktionen/Disjunktionen.

Beispiel

Seien $A_1, \ldots, A_n$ atomare Formeln, dann ist die n-äre **Paritätsfunktion** wie folgt definiert.

$$\bigoplus_{i=1}^{n} A_i = (\ldots(A_1 \oplus A_2) \oplus A_3) \ldots) \oplus A_n)$$

Seien α eine passende Belegung für $\{A_1, \ldots, A_n\}$, dann gilt Folgendes.

$$\alpha\left(\bigoplus_{i=1}^{n} A_i\right) = \begin{cases} 1 & \text{Anzahl } i \in \{1, \ldots, n\} \text{ mit } \alpha(A_i) = 1 \text{ ist ungerade} \\ 0 & \text{sonst} \end{cases}$$

Die Paritätsfunktion $\bigoplus_{i=1}^{n} A_i$ kann mit einer Formel, die linear in n ist (mit höchstens $2n$ Literalen), dargestellt werden, denn es gilt

$$A \oplus B = (A \vee B) \wedge (\neg A \vee \neg B) \ .$$

Die Paritätsfunktion hat dargestellt in DNF/KNF jeweils 2^{n-1} viele Konjunktionen/Disjunktionen (Klauseln).

A	B	C	$A \oplus B \oplus C$	A	B	C	$A \oplus B \oplus C$
0	0	0	0	1	0	0	1
0	0	1	1	1	0	1	0
0	1	0	1	1	1	0	0
0	1	1	0	1	1	1	1

DNF, Konstruktion aus den Zeilen für die $A \oplus B \oplus C = 1$ gilt.

$$A \oplus B \oplus C = (\neg A \wedge \neg B \wedge C) \vee (\neg A \wedge B \wedge \neg C) \vee (A \wedge \neg B \wedge \neg C) \vee (A \wedge B \wedge C)$$

KNF, Konstruktion aus den Zeilen für die $A \oplus B \oplus C = 0$ gilt.

$$\begin{aligned} A \oplus B \oplus C =& \neg\Big(\neg A \wedge \neg B \wedge \neg C) \vee (\neg A \wedge B \wedge C) \vee (A \wedge \neg B \wedge C) \vee (A \wedge B \wedge \neg C)\Big) \\ =&(A \vee B \vee C) \wedge (A \vee \neg B \vee \neg C) \wedge (\neg A \vee B \vee \neg C) \wedge (\neg A \vee \neg B \vee C) \end{aligned}$$

In DNF/KNF können keine Konjuktionen/Disjuktionen verschmolzen werden, weil für die Bestimmung der Parität immer alle Literale ausgewertet werden müssen.

Definition 7.14. Eine Formel F in KNF ist eine Hornformel, falls jede Disjunktion/Klausel maximal ein positives Literal enthält. Jede Klausel K kann dadurch als eine Implikation $T \to H$ gesehen werden. Dabei ist T eine Konjunktion von Atomen oder $T = \mathit{wahr}$ und H ist ein Atom oder $H = \bot$:

$$\begin{aligned} (\neg A \vee \neg B \vee C) &\equiv (A \wedge B \to C) \\ (\neg A \vee \neg B) &\equiv (A \wedge B) \to \bot \\ A &\equiv (\mathit{wahr} \to A) \end{aligned}$$

Definition 7.15. Gegeben eine Hornformel F. Wir prüfen die Erfüllbarkeit mit folgendem Algorithmus, welcher gleichzeitig ein Modell konstruiert:

1. Setze für alle Atome A_i in F die Belegung initial auf $\alpha(A_i) = 0$.
2. Solange in F Klauseln der Art $(T \to H)$ exisiteren mit $\alpha(T) = \mathit{wahr}$ und $\alpha(H) = 0$:
   - Falls $H = \bot$ gebe **unerfüllbar** aus.
   - Sonst setze $\alpha(H) = 1$

3. Gebe **erfüllbar** aus und $\alpha \models F$.

$F = (A_1) \wedge (\neg A_1 \vee A_2) \wedge (\neg A_1 \vee \neg A_3 \vee A_4) \wedge (\neg A_1 \vee \neg A_4 \vee A_5) \wedge (\neg A_2 \vee \neg A_4)$

Example 7.16.

$(1 \to A_1) \wedge (A_1 \to A_2) \wedge (A_1 \wedge A_3 \to A_4) \wedge (A_1 \wedge A_4 \to A_5) \wedge (A_2 \wedge A_4 \to \bot)$

setze $\alpha(A_1) = 1$.

$$(1 \to A_2) \wedge (1 \wedge A_3 \to A_4) \wedge (1 \wedge A_4 \to A_5) \wedge (A_2 \wedge A_4 \to \bot)$$

setze $\alpha(A_2) = 1$.

$$(1 \wedge A_3 \to A_4) \wedge (1 \wedge A_4 \to A_5) \wedge (1 \wedge A_4 \to \bot)$$

setze alle übrigen $\alpha(A_i) = 0$ und die Formel ist erfüllbar.

- Das Lösen von Formeln in 2-KNF (oder auch das 2-SAT Problem) ist effizient (polynomiell). **Idee:** In jeder Klausel muss mindestens eines der beiden Literale mit wahr belegt werden. Ist eines der Literale falsch, dann ist implizit, dass für eine erfüllbare Belegung das andere wahr sein muss. Diese Implikationen können in einem gerichteten Graphen modelliert werden. Wenn dieser Graph Kreise enthält, welche sowohl A und $\neg A$ enthalten, dann ist die Formel nicht erfüllbar, ansonsten ist die formel erfüllbar und die Pfade durch den Graphen implizieren ein Modell. Dies kann effizient durch Tiefensuche realisiert werden.
- Das Lösen von Formeln in 3-KNF (oder auch das 3-SAT Problem) ist NP-vollständig (Verweis auf Informatik 3 - Algorithmen und Datenstrukturen).

Mit dieser Art von Algorithmus werden SAT Probleme, also das Erfüllbarkeitsproblem für Formeln F in KNF gelöst.

1. Gegeben sei eine Formel F in KNF als Input.
2. Solange es Klauseln mit einem einzigen Literal $C = \{L\}$ als Klauselmenge gibt:
   - Setze L auf wahr.
   - Klauseln mit L werden entfernt, Literale $\neg L$ werden entfernt.

3. Solange es Literale L gibt, welche nur in einer Form (negativ/positiv) in F vorkommen:
   - Setze L auf wahr.
   - Klauseln mit L werden entfernt, Literale $\neg L$ werden entfernt.
4. Falls F keine Klauseln mehr enthält, dann gebe für F wahr zurück.
5. Falls F eine leere Klausel enthält, dann gebe für F falsch zurück.
6. Wähle ein Literal L.
7. Führe die Prozedur rekursiv mit $F \wedge \{L\}$ und $F \wedge \{\neg L\}$ und teste ob eines davon wahr ist.

7.2 Prädikatenlogik

7.2.1 Einleitung

Logik handelt vom Umgang mit **Aussagen**.

Aussagen sind sprachliche Gebilde, die entweder *wahr* oder *falsch* sind.

Beispiele

Aussagen

1. Es regnet.
2. Der Wal ist ein Fisch.
3. Die Variable x hat den Wert -1.

Keine Aussagen

1. Herzlichen Glückwunsch.
2. Wo ist der Bahnhof.

Die **Prädikatenlogik** ist eine Erweiterung der Aussagenlogik um folgende wesentliche Konzepte.

- Aussagen gelten für **Objekte** (Individuen) eines zu modellierenden **Individuenbereichs**.
- **Variablen** sind Platzhalter für Objekte des Individuenbereichs.
- **Formel** entstehen durch Variablen in Aussagen.
- Formeln werden kombiniert durch **Junktoren** und eingeschränkt durch **Quantoren**.
- Objekte können durch **Terme** bezeichnet werden.

Die heute gebräuchliche Syntax der Prädikatenlogik wurde 1889 eingeführt von Giuseppe Peano in *Arithmetices principia, nova methodo exposita* (Prinzipien der Arithmetik, nach einer neuen Methode vorgestellt).

Die Ursprünge gehen zurück auf eine Arbeit von Gottlob Frege *Begriffsschrift, eine der arithmetischen nachgebildete Formelsprache des reinen Denkens* von 1879.

Der Grundgedanke der Prädikatenlogik ist die Einführung **parametrisierter Aussagen**, in denen **Variablen** durch beliebige Objekte des zu modellierenden Individuenbereichs ersetzt werden können.

N ist eine Primzahl

> Variable N, Individuenbereich die natürlichen Zahlen $\mathbb{N}$

X ist kleiner als Y

> Variablen X und Y, Individuenbereich die reelen Zahlen $\mathbb{R}$

Erst nach dem Ersetzen der Variablen durch **konkrete** Objekte aus dem Individuenbereich kann ein Wahrheitswert festgestellt werden.

N ist eine Primzahl

> $N = 4$, daraus folgt die Aussage ist *falsch*
>
> $N = 5$, daraus folgt die Aussage ist *wahr*

7.2.2 Formeln

In der Prädikatenlogik heissen (parametrisierte) Aussagen **Formeln**.

- **Atomare Formeln** lassen sich nicht weiter in Teilformeln zerlegen.
- **Zusammengesetzte Formeln** sind aus Teilformeln aufgebaut.

Formale Notation für atomare Formeln, ist ein **Prädikat** gefolgt von einer Parameterliste.

Beispiel

$$\text{istPrimzahl}(N)$$

istPrimzahl ist das Prädikat, N ist die Variable.

Zweistellige Prädikate dürfen auch in **Infixnotation** verwendet werden.

$$\text{ist\_kleiner\_als}(X, Y) \quad \text{bzw.} \quad X \ \text{ist\_kleiner\_als} \ Y \quad \text{bzw.} \quad X < Y$$

ist_kleiner_als und $<$ sind Prädikate, X und Y sind Variablen.

Formeln lassen sich mit Hilfe der **Junktoren** ($\neg$, $\wedge$, $\vee$, $\oplus$, $\Rightarrow$, $\Leftrightarrow$), der Operatoren aus der Aussagenlogik, zu neuen Formeln kombinieren.

$$\text{istPrimzahl}(N) \ \wedge \ (N < 10)$$

$N = 4$, daraus folgt die Aussage ist *falsch* $N = 5$, daraus folgt die Aussage ist *wahr* $N = 11$, daraus folgt die Aussage ist *falsch*

Erinnerung

Die **Implikation** $x \Rightarrow y$ ist die Aussage *wenn x, dann y* oder x *impliziert* y.

Es gilt folgende Wahrheitstafel.

x	y	$x \Rightarrow y$
falsch	*falsch*	*wahr*
falsch	*wahr*	*wahr*
wahr	*falsch*	*falsch*
wahr	*wahr*	*wahr*

Die **Äquivalenz** $x \Leftrightarrow y$ ist die Aussage *wenn x, genau dann y* oder x *äquivalent* y.

Es gilt folgende Wahrheitstafel.

x	y	$x \Leftrightarrow y$
falsch	*falsch*	*wahr*
falsch	*wahr*	*falsch*
wahr	*falsch*	*falsch*
wahr	*wahr*	*wahr*

Quantoren können benutzt werden um Formeln zu bilden, die für manche bzw. für alle Individuen gelten sollen.

Existenzquantor

$$\exists N \in \mathbb{N} : (\text{istPrimzahl}(N) \ \wedge \ (N < 10)) \qquad \textit{wahr}$$

Es gibt eine natürliche Zahl N für die gilt, N ist eine Primzahlen und N ist kleiner als 10.

(Es gibt Primzahlen, die kleiner als 10 sind.)

Allquantor

$$\forall N \in \mathbb{N} : (\text{istPrimzahl}(N) \;\Rightarrow\; (N < 10)) \qquad \textit{falsch}$$

Für alle natürlichen Zahlen N gilt, N ist eine Primzahlen impliziert N ist kleiner als 10.

(Alle Primzahlen sind kleiner als 10.)

Schwierigkeiten

Schwierigkeiten beim Formulieren von Sachverhalten mit Quantoren und Variablen ist, dass Variablen erst **eingeführt** und **quantifiziert** werden müssen, bevor die eigentliche Aussage folgen kann.

<u>Beispiel</u>

Jede Primzahl ist kleiner als 10

$\downarrow$

Für alle N gilt, N ist eine Primzahlen impliziert N ist kleiner als 10

$\downarrow$

Für alle natürlichen Zahlen N gilt, N ist Primzahlen impliziert N ist kleiner als 10

$\downarrow$

$$\forall N \in \mathbb{N} : (\text{istPrimzahl}(N) \;\Rightarrow\; (N < 10))$$

Quantoren und Junktoren

Der Allquantor und die Konjunktion sind verträglich, folgende Formeln sind *wahr*.

$$\Big(\forall X : \big(P(X) \wedge Q(X)\big)\Big) \Leftrightarrow \Big(\big(\forall X : P(X)\big) \wedge \big(\forall X : Q(X)\big)\Big)$$
$$\Big(\forall X : \big(P(X) \vee Q(X)\big)\Big) \Leftarrow \Big(\big(\forall X : P(X)\big) \vee \big(\forall X : Q(X)\big)\Big) \quad (\not\Rightarrow)$$

Der Existenzquantor und die Disjunktion sind verträglich, folgende Formeln sind *wahr*.

$$\Big(\exists X : \big(P(X) \lor Q(X)\big)\Big) \Leftrightarrow \Big(\big(\exists X : P(X)\big) \lor \big(\exists X : Q(X)\big)\Big)$$

$$\Big(\exists X : \big(P(X) \land Q(X)\big)\Big) \Rightarrow \Big(\big(\exists X : P(X)\big) \land \big(\exists X : Q(X)\big)\Big) \quad (\not\Leftarrow)$$

Für die Negation gelten folgende Formeln, d.h. folgende Formeln sind *wahr*.

$$\Big(\neg\big(\forall X : P(X)\big)\Big) \Leftrightarrow \Big(\exists X : \big(\neg P(X)\big)\Big)$$

$$\Big(\neg\big(\exists X : P(X)\big)\Big) \Leftrightarrow \Big(\forall X : \big(\neg P(X)\big)\Big)$$

<u>Folgerung</u>

Ein Quantor wäre ausreichend, es gilt Folgendes (folgende Formeln sind *wahr*).

$$\big(\forall X : P(X)\big) \Leftrightarrow \Big(\neg\Big(\exists X : \big(\neg P(X)\big)\Big)\Big)$$

$$\big(\exists X : P(X)\big) \Leftrightarrow \Big(\neg\Big(\forall X : \big(\neg P(X)\big)\Big)\Big)$$

Die Verwendung beider Quantoren ist wegen der besseren Lesbar- und Benutzbarkeit sinnvoll.

Schachtelung

Bei Schachtelungen gleicher Quantoren können diese vertauscht werden, folgende Formeln sind *wahr*.

$$\Big(\forall X : \big(\forall Y : P(X,Y)\big)\Big) \Leftrightarrow \Big(\forall Y : \big(\forall X : P(X,Y)\big)\Big)$$

$$\Big(\exists X : \big(\exists Y : P(X,Y)\big)\Big) \Leftrightarrow \Big(\exists Y : \big(\exists X : P(X,Y)\big)\Big)$$

Bei Schachtelung verschiedener Quantoren darf in der Regel keine Vertauschung vorgenommen werden.

7.2.3 Gültigkeitsbereich

In folgenden Ausdrücken ist die Formel P der **Gültigkeitsbereich** (*scope*) des Quantors für die Variablen X.

$$\forall X : P(X) \quad \text{oder} \quad \exists X : P(X)$$

Eine Variable im Gültigkeitsbereich eines Quantors ist (durch den Quantor) **gebunden**, einen nicht gebundene Variable ist **frei**.

<u>Beispiel</u>

$\forall X : \Big(Q(X) \wedge (X > 2)\Big)$	X ist (durch $\forall$) gebunden
$Q(Y) \Rightarrow \Big(\exists X : R(X, Y, Z)\Big)$	X ist gebunden, Y und Z sind frei
$(X > Y) \wedge \Bigg(\forall X : \Big(\exists Y : (X < Y)\Big)\Bigg)$	X, Y sind gleichzeitig frei und gebunden

Geschlossene Formeln enthalten keine freien Variablen, Formeln mit mindestens einer freien Variable sind **offen**.

<u>Beispiel</u>

$\forall X : \Big(Q(X) \wedge (X > 2)\Big)$	geschlossen
$Q(Y) \Rightarrow \Big(\exists X : R(X, Y, Z)\Big)$	offen
$(X > Y) \wedge \Bigg(\forall X : \Big(\exists Y : (X < Y)\Big)\Bigg)$	offen

Das Variablen in einer Formel gleichzeitig gebunden und frei vorkommen kann durch Umbenennung der Variablen, ohne Änderung der Bedeutung, vermieden werden.

$$\Bigg((X > Y) \wedge \Big(\exists Y : P(Y)\Big)\Bigg) \Leftrightarrow \Bigg((X > Y) \wedge \Big(\exists Z : P(Z)\Big)\Bigg)$$

Ebenso ist die Bindung einer Variablen durch verschiedene Quantoren mit Umbenennung vermeidbar.

$$\Bigg(\Big(\forall X : P(X)\Big) \vee \Big(\exists X : Q(X)\Big)\Bigg) \Leftrightarrow \Bigg(\Big(\forall X : P(X)\Big) \vee \Big(\exists Y : Q(Y)\Big)\Bigg)$$

Üblich ist das Einhalten folgender **Konventionen**.

- Keine Variable in einer Formel ist gleichzeitig frei und gebunden.
- Jeder Quantor in einer Formel führt eine **neue** Variable ein, die ausserhalb seines Gültigkeitsbereichs nicht vorkommt.

Gültigkeitsbereich und Junktoren

Q macht keine Aussage über X.

$$\begin{aligned}
\Big(\big(\forall X : P(X)\big) \wedge Q\Big) &\Leftrightarrow \Big(\forall X : \big(P(X) \wedge Q\big)\Big)\\
\Big(\big(\exists X : P(X)\big) \wedge Q\Big) &\Leftrightarrow \Big(\exists X : \big(P(X) \wedge Q\big)\Big)\\
\Big(\big(\forall X : P(X)\big) \vee Q\Big) &\Leftrightarrow \Big(\forall X : \big(P(X) \vee Q\big)\Big)\\
\Big(\big(\exists X : P(X)\big) \vee Q\Big) &\Leftrightarrow \Big(\exists X : \big(P(X) \vee Q\big)\Big)\\
\Big(\big(\forall X : P(X)\big) \Rightarrow Q\Big) &\Leftrightarrow \Big(\exists X : \big(P(X) \Rightarrow Q\big)\Big)\\
\Big(\big(\exists X : P(X)\big) \Rightarrow Q\Big) &\Leftrightarrow \Big(\forall X : \big(P(X) \Rightarrow Q\big)\Big)\\
\Big(Q \Rightarrow \big(\forall X : P(X)\big)\Big) &\Leftrightarrow \Big(\forall X : \big(Q \Rightarrow P(X)\big)\Big)\\
\Big(Q \Rightarrow \big(\exists X : P(X)\big)\Big) &\Leftrightarrow \Big(\exists X : \big(Q \Rightarrow P(X)\big)\Big)
\end{aligned}$$

Die Namen der quantifizierten Variablen sind belanglos.

$$\big(\forall Y : P(Y)\big) \Leftrightarrow \big(\forall Z : P(Z)\big) \quad \text{und} \quad \big(\exists X : P(X)\big) \Leftrightarrow \big(\exists Y : P(Y)\big)$$

Gültigkeitsbereich und Semantik

Geschlossene Formeln sind Aussagen, d.h. der Wahrheitswert einer geschlossenen Formeln kann angegeben werden.

Offenen Formeln sind parametrisierte Aussagen, d.h. der Wahrheitswert einer offenen Formeln kann erst nach dem Ersetzen der freien Variablen durch konkrete Objekte aus dem Individuenbereich festgestellt werden.

Bemerkung

Eine offene Formel wird zu einer geschlossenen, wenn man alle freien Variablen durch konkrete Objekte ersetzt.

Beispiel

$\exists N$: istPrimzahl(N)	geschlossen, über dem Individuenbereich $\mathbb{N}$ ist der Wahrheitswert *wahr*
$\forall N$: istPrimzahl(N)	geschlossen, über dem Individuenbereich $\mathbb{N}$ ist der Wahrheitswert *falsch*
istPrimzahl(N)	offen
istPrimzahl(5)	geschlossen, Wahrheitswert *wahr*
istPrimzahl(99)	geschlossen, Wahrheitswert *falsch*

7.2.4 Terme

Terme

Terme sind Ausdrücke, die bestimmte Objekte des gerade betrachteten Individuenbereichs bezeichnen.

- **Atomare Terme** bezeichnen ein bestimmtes Objekt (Individuum) direkt.
- **Zusammengesetzte Terme** verwenden eine Funktion, die zunächst angewendet werden muss, um das bezeichnete Objekt (Individuum) zu identifizieren.

Beispiel

Die beiden nachfolgenden Termen beschreiben beide dasselbe Objekt aus dem Individuenbereichs aller bei Wikipedia aufgelisteter Personen.

„Horst Köhler, *22.02.1943“

praesident(„Bundesrepublik Deutschland“, 2010)

Die Funktion praesident hat als Argumente Terme eines möglicherweise anderen Individuenbereichs.

Atomare Terme

Die einzigen atomaren (unzerlegbaren) Terme sind Konstanten und Variablen.

- **Konstanten** sind Zeichen oder Zeichenreihen, die immer genau ein Objekt (Individuum) aus dem gerade betrachteten Individuenbereich bezeichnen.

  Beispiel

„Bundesrepublik Deutschland“	$\longrightarrow$	ein Staat, aus dem Individuenbereich der aktuell existierenden Staaten
2010	$\longrightarrow$	die Jahreszahl 2010 aus dem Individuenbereich der Jahreszahlen

- **Variablen** bezeichnen ebenfalls zu jedem Zeitpunkt genau ein Objekt, aber diese Bedeutung kann sich ändern je nachdem, mit welcher Bedeutung die Variable gerade **belegt** worden ist.

  Beispiel

  Die Variable X kann sowohl mit 2010 als auch mit „Bundesrepublik Deutschland“ belegt werden.

Zusammengesetzte Terme

Formale Notation für zusammengesetzte Terme, ist ein **Funktor** (Funktionssymbol oder Operator) gefolgt von einer Parameterliste.

Beispiel

$$\text{wurzel}(N)$$

wurzel ist ein Funktor, N ist ein Parameter.

Zweistellige Funktoren dürfen auch in **Infixnotation** verwendet werden.

$$\text{summe}(X, Y) \qquad \text{bzw.} \qquad X + Y$$

summe und + sind Funktoren, X und Y sind Parameter.

7.2.5 Sprache

Relationen

Prädikate können als **Relationsnamen** aufgefasst werden.

Aller Kombinationen von Individuen, die eine Formel wahr machen, wenn man sie für die Parameter der Formel einsetzt, bilden eine **Relation** (Tupelmenge).

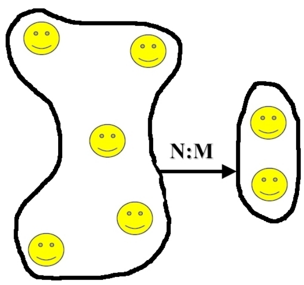
Beispiel

$$\text{ist\_echter\_teiler\_von}(X, Y)$$

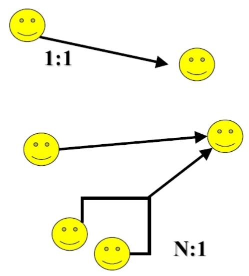
X	Y
2	4
2	6
3	6
2	8
4	8
3	9
2	10
5	10
⋮	⋮

Zuordnungen

Relation

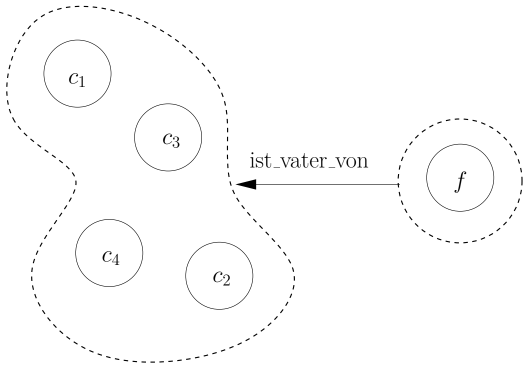


Funktion



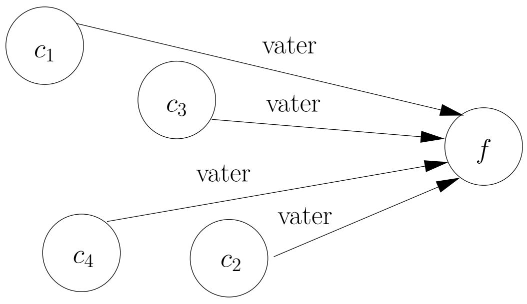
Festlegung konkreter Zuordnungen

Formel



Die **Formel** ist_vater_von(f, c_2) ist *wahr* und machte eine **Aussage über Objekte**.

Term



Der **Term** vater(c_2) bezeichnet das Objekt f, ist damit ein **Objektbezeichner**.

Formeln und Terme

Es ist nicht leicht, atomare Formeln und zusammengesetzte Terme rein syntaktisch zu unterscheiden.

- **Formel** (machen eine Aussage) `<Prädikat>(<Parameterliste>)`
- **Term** (bezeichnen Objekte) `<Funktor>(<Parameterliste>)`

Ob das syntaktische Gebilde

$$\text{magic}(1, 2^2, 3 \cdot 3)$$

ein Term oder eine Formel ist, hängt davon ab, ob das Symbol magic ein Funktor ist (eine Funktion) oder ein Prädikat (eine Relation) ist.

- ist_praesident(„Horst Köhler, *22.02.1943“) ist eine atomare Formel, die eine Aussage macht über das von dem atomaren Term „Horst Köhler, *22.02.1943“ bezeichnete Objekt.
- produkt(3, 42) ist ein zusammengesetzter Term, der ein Objekt bezeichnet und die atomaren Terme 3 und 42 als Parameter enthält.

Sprache

Prädikatenlogik bietet nur einen Sprachrahmen für das Formalisieren beliebiger Anwendungsprobleme. Die Sprache der Logik gibt es nicht.

Stattdessen benötigt jede Anwendung ihre eigene, ganz spezifische logische Sprache.

Syntaktische und semantische Grundlagen aller logischen Sprachen werden festgelegt durch Konventionen/Definitionen der traditionellen Logik.

Von Sprache zu Sprache verschieden ist das verwendetes **Vokabular**, d.h. die Menge derjenigen Zeichen, aus denen Formeln/Terme aufgebaut sind.

Das Vokabular einer konkreten logischen Sprache besteht aus zwei Teilmengen.

- Von Anwendung zu Anwendung verschieden sind Funktoren und Prädikate.

  Das anwendungsspezifisches Vokabular, die **Signatur einer logischen Sprache** muss vom jeweiligen Anwender selbst festgelegt werden.

- Für alle Anwendungen gleich sind Junktoren, Quantoren, Variablen etc.

  Der anwendungsunabhängiger Anteil ist bis auf Dialekte immer identisch (normiert).

Signatur

Die **Signatur** einer Anwendungssprache ist die Sammlung aller Funktoren und Prädikate. Frei wählbar, d.h vom Anwendungsdesigner festzulegen sind

- Prädikate (Relationsnamen),
- Funktoren (Funktionsnamen)

und deren Stelligkeiten (= Anzahl an Parametern).

<u>Formal</u>

Die Signatur einer Anwendungssprache über einem Individuenbereich Ω ist ein Paar $(\mathcal{P}, \mathcal{F})$ für das Folgendes gilt.

- $\mathcal{P}$ ist endliche Menge von Prädikaten P_j mit Stelligkeiten n_j $(1 \leq j \leq n)$, die Aussagen über Objekte aus Ω machen.
- $\mathcal{F}$ ist endliche Menge von Funktoren F_i mit Stelligkeiten m_i $(1 \leq i \leq m)$, die Objekte aus Ω bezeichnen.

7.2.6 Zusammenfassung

Syntax von Termen

1. Jede Variable ist ein Term.
2. Jede Konstante ist ein Term.
3. Wenn $T_1, \ldots, , T_n$ mit $n \geq 1$ Terme sind und wenn F ein n-stelliger Funktor ist, dann ist auch $F(T_1, \ldots, T_n)$ ein Term.

Wichtig

Stets bezogen auf die Signatur einer bestimmte Sprache, die einen entsprechenden Funktor definiert.

4. Sonst gibt es keine weiteren Terme.

Syntax von Formeln

1. *wahr* und *falsch* sind atomare Formeln.
2. Wenn P ein 0-stelliges Prädikat ist, dann ist P eine atomare Formel, die abkürzend für $P()$ steht.
3. Wenn $T_1, \ldots, T_n$ mit $n \geq 1$ Terme sind und wenn P ein n-stelliges Prädikat ist, dann ist $P(T_1, \ldots, T_n)$ eine atomare Formel.

   Wichtig

   Stets bezogen auf die Signatur einer bestimmte Sprache, die ein entsprechendes Prädikat definiert.
4. Wenn g eine Formel ist, dann ist auch $\neg g$ eine Formel.
5. Wenn g und h Formeln sind, dann sind auch folgende Ausdrücke Formeln.
$$g \wedge h, \quad g \vee h, \quad g \oplus h, \quad g \Rightarrow h \quad \text{und} \quad g \Leftrightarrow h$$
6. Wenn g eine Formel und X eine Variable ist, dann sind auch folgende Ausdrücke Formeln.
$$\forall X : g(X) \quad \text{und} \quad \exists X : g(X)$$
7. Sonst gibt es keine weiteren Formeln.

7.2.7 Anwendung

Anwendung der Prädikatenlogik

Prädikatenlogik ist ein Hilfsmittel vor allem für Informatik, Mathematik und Linguistik.

Bei der Konzeption und Programmierung von Expertensystemen und in der künstlichen Intelligenz kommt angewandte Prädikatenlogik zum Einsatz, z.B. die Programmiersprache Prolog (*programming in logic*).

Der Relationenkalkül, eine der theoretischen Grundlagen von Datenbankabfragesprachen wie etwa SQL, bedient sich der Prädikatenlogik als Ausdrucksmittel.

7.2.8 Beispiele

Betrachten Sie folgenden Signatur einer prädikatenlogischen Sprache mit einer Teilmenge der Lebewesen B als Individuenbereich.

Signatur

- Prädikate $\mathcal{P} = \{pilz, rosa, giftig, =\}$
  - Das Prädikat $=$ ist zweistellig, die anderen einstellig.
  - Die Bedeutung der einstelligen Prädikate ergibt sich aus der Namensgebung.
  - Für das Prädikat $=$ gilt, $X = Y$ ist genau dann *wahr*, wenn X und Y dasselbe Individuum bezeichnen.
- Funktoren $\mathcal{F} = \emptyset$

Aufgabe 1

Formulieren Sie für die folgenden Aussagen prädikatenlogischen Formeln, die genau dann *wahr* sind wenn die jeweilige Aussage gilt.

1. Alle rosa Pilze sind giftig.
2. Wenn ein Pilz giftig ist, dann ist er rosa.
3. Kein rosa Pilz ist giftig.

Lösung 1

Formeln

1. $\forall X : \Big((rosa(X) \wedge pilz(X)) \Rightarrow giftig(X)\Big)$

   oder

   $\forall X : \Big(rosa(X) \Rightarrow (pilz(X) \Rightarrow giftig(X))\Big)$

2. $\forall X : \Big((pilz(X) \wedge giftig(X)) \Rightarrow rosa(X)\Big)$

   oder

   $\forall X : \Big(pilz(X) \Rightarrow (giftig(X) \Rightarrow rosa(X))\Big)$

3. $\forall X : \neg\Big(rosa(X) \wedge pilz(X) \wedge giftig(X)\Big)$

   oder

   $\forall X : \Big((rosa(X) \wedge pilz(X)) \Rightarrow \neg giftig(X)\Big)$

Aufgabe 2

Ist die prädikatenlogischen Formel

$$\forall X : \Big(\forall Y : \Big((pilz(X) \wedge pilz(Y)) \Rightarrow (X = Y)\Big)\Big)$$

genau dann *wahr* wenn die Aussage

Es gibt genau einen Pilz.

gilt?

Begründen Sie Ihre Antwort oder geben Sie ein Gegenbeispiel an.

Hinweis

Achten Sie auf den Individuenbereich.

Lösung 2

Formel und Aussage sind nicht äquivalent.

Gegenbeispiel. Angenommen B enthält keine Pilze.

- Die Aussage

  Es gibt genau einen Pilz.

  ist *falsch*.

- Die Formel

  $$\forall X : \Big(\forall Y : \Big((pilz(X) \wedge pilz(Y)) \Rightarrow (X = Y)\Big)\Big)$$

  ist *wahr*

  Denn es gilt $(pilz(X) \wedge pilz(Y))$ ist immer *falsch*, d.h. der Gültigkeitsbereich von Y ist immer *wahr* und somit ist die Formel *wahr*.

Aufgabe 3

Korrekturvorschlag

Ist die prädikatenlogischen Formel

$$\Big[\forall X : \Big(\forall Y : \Big((pilz(X) \wedge pilz(Y)) \Leftrightarrow (X = Y)\Big)\Big)\Big]$$

genau dann *wahr* wenn die Aussage

Es gibt genau einen Pilz.

gilt?

Begründen Sie Ihre Antwort oder geben Sie ein Gegenbeispiel an.

Lösung 3

Formel und Aussage sind nicht äcquivalent.

Gegenbeispiel. Angenommen B enthält genau einen Pilz und ein Lebewesen Z, dass kein Pilz ist.

- Die Aussage

  Es gibt genau einen Pilz.

  ist *wahr*.

- Die Formel

  $$\forall X : \Big(\forall Y : \Big((pilz(X) \wedge pilz(Y)) \Leftrightarrow (X = Y)\Big)\Big)$$

  ist *falsch*.

  Denn $(pilz(Z) \wedge pilz(Z))$ ist *falsch* und $(Z = Z)$ ist *wahr*, daraus folgt die korrigierte Formel ist *falsch*.

Korrektur

Die prädikatenlogischen Formel

$$\Big[\exists Z : pilz(Z)\Big] \wedge \Big[\forall X : \Big(\forall Y : \Big((pilz(X) \wedge pilz(Y)) \Rightarrow (X = Y)\Big)\Big)\Big]$$

ist genau dann *wahr* wenn die Aussage

Es gibt genau einen Pilz.

gilt.

Beispiel

Satz 7.17 (Goldbach-Vermutung)**.** *Das Doppelte jeder natürlichen Zahl größer Zwei ist die Summe zweier Primzahlen.*

Aufgabe 1

Geben Sie über dem Individuenbereich der natürlichen Zahlen $\mathbb{N}$ die Signatur einer logischen Sprache an, die verwendet werden kann, um die **Goldbach-Vermutung** als prädikatenlogische Formel zu formulieren.

Hinweis

Für eigene Definitionen können Sie annehmen, dass Begriffe wie Summe, Differenz, Gleichheit, Ungleichheit, etc. bekannt sind und die in der Arithmetik übliche Bedeutung haben.

Lösung 1

Sei $\mathbb{N}$ der Individuenbereich der natürlichen Zahlen.

Signatur mit Prädikaten $\mathcal{P}$ und Funktoren $\mathcal{F}$.

$\mathcal{P} = \{prim, >, =\}$

$prim$	ist einstellig, $prim(n)$ ist genau dann wahr, wenn n eine Primzahl ist.
$=$	ist zweistellig, $n = m$ ist genau dann wahr, wenn n gleich m ist.
$>$	ist zweistellig, $n > m$ ist genau dann wahr, wenn n größer m ist (wird nicht gebraucht bei $\mathbb{N}_{>2}$).

$\mathcal{F} = \{+\}$

$+$	ist zweistellig, $n + m$ ist die Summe der beiden Zahlen.

Aufgabe 2

Benutzen Sie die logischen Sprache aus dem vorstehenden Aufgabenteil um eine prädikatenlogische Formel zu formulieren, die genau dann wahr ist wenn die Goldbach-Vermutung zutrifft.

Lösung 2

Formel über dem Individuenbereich $\mathbb{N}$.

$$\forall n : \Big((n > 2) \Rightarrow \\ \Big(\exists p \in \mathbb{N} : \Big(\exists q \in \mathbb{N} : \Big(\mathrm{prime}(p) \wedge \mathrm{prime}(q) \wedge ((n + n) = (p + q))\Big)\Big)\Big)\Big)$$