Logik für Informatiker

3. Prädikatenlogik

Teil 1

01.06.2021

Viorica Sofronie-Stokkermans

Universität Koblenz-Landau

e-mail: sofronie@uni-koblenz.de

Rückblick: Vor- und Nachteile von Aussagenlogik

+ Aussagenlogik ist deklarativ: Syntaxelemente entsprechen Fakten

Rückblick: Vor- und Nachteile von Aussagenlogik

+ Aussagenlogik ist deklarativ: Syntaxelemente entsprechen Fakten
+ Aussagenlogik erlaubt konjunktive / disjunktive / negative Aussagen (im Gegensatz zu vielen Datenstrukturen und Datenbanken)

Rückblick: Vor- und Nachteile von Aussagenlogik

+ Aussagenlogik ist deklarativ: Syntaxelemente entsprechen Fakten
+ Aussagenlogik erlaubt konjunktive / disjunktive / negative Aussagen (im Gegensatz zu vielen Datenstrukturen und Datenbanken)
+ Aussagenlogik ist kompositional: Bedeutung von $F_1 \wedge F_2$ leitet sich ab aus der von F_1 und F_2

Rückblick: Vor- und Nachteile von Aussagenlogik

+ Aussagenlogik ist deklarativ: Syntaxelemente entsprechen Fakten
+ Aussagenlogik erlaubt konjunktive / disjunktive / negative Aussagen (im Gegensatz zu vielen Datenstrukturen und Datenbanken)
+ Aussagenlogik ist kompositional: Bedeutung von $F_1 \wedge F_2$ leitet sich ab aus der von F_1 und F_2
+ Bedeutung in Aussagenlogik ist kontextunabhängig (im Gegensatz zu natürlicher Sprache)

Rückblick: Vor- und Nachteile von Aussagenlogik

+ Aussagenlogik ist deklarativ: Syntaxelemente entsprechen Fakten

+ Aussagenlogik erlaubt konjunktive / disjunktive / negative Aussagen (im Gegensatz zu vielen Datenstrukturen und Datenbanken)

+ Aussagenlogik ist kompositional: Bedeutung von $F_1 \wedge F_2$ leitet sich ab aus der von F_1 und F_2

+ Bedeutung in Aussagenlogik ist kontextunabhängig (im Gegensatz zu natürlicher Sprache)

− Aussagenlogik hat nur beschränkte Ausdruckskraft (im Vergleich zu natürlicher Sprache)

Beispiele:

- Die Aussage "Jede natürliche Zahl ist entweder gerade oder ungerade" erfordert eine Formel für jede Zahl.

Weitere Logiken

Logik		
Aussagenlogik	Fakten	wahr/falsch
Prädikatenlogik	Objekte, Funktionen, Relationen	wahr/falsch
Temporallogik	Fakten, Zeitpunkte	wahr/falsch
Mehrwertige Logik	Fakten	wahr/falsch/unbekannt
Fuzzy-Logik	Fakten	[0, 1]
...	...	...

Weitere Logiken

Logik		
Aussagenlogik	Fakten	wahr/falsch
Prädikatenlogik	Objekte, Funktionen, Relationen	wahr/falsch
Temporallogik	Fakten, Zeitpunkte	wahr/falsch
Mehrwertige Logik	Fakten	wahr/falsch/unbekannt
Fuzzy-Logik	Fakten	[0, 1]
...	...	...

Prädikatenlogik

- **Syntax**
- **Semantik**
- **Kalküle**

    Resolution

    Semantische Tableaux

Prädikatenlogik

- **Syntax**
- **Semantik**
- **Kalküle**

  Resolution

  Semantische Tableaux

Prädikatenlogik

Reichere Struktur

- Objekte (Elemente)
  Leute, Häuser, Zahlen, Theorien, Farben, Jahre, ...

Prädikatenlogik

Reichere Struktur

- Objekte (Elemente)
  Leute, Häuser, Zahlen, Theorien, Farben, Jahre, ...

- Relationen (Eigenschaften)
  rot, rund, prim, mehrstöckig, ...
  ist Bruder von, ist größer als, ist Teil von, hat Farbe, besitzt, ..
  $=$, $\geq$, ...

Prädikatenlogik

Reichere Struktur

- Objekte (Elemente)
  Leute, Häuser, Zahlen, Theorien, Farben, Jahre, ...
- Relationen (Eigenschaften)
  rot, rund, prim, mehrstöckig, ...
  ist Bruder von, ist größer als, ist Teil von, hat Farbe, besitzt, ..
  $=$, $\geq$, ...
- Funktionen
  $+$, Mitte von, Vater von, Anfang von, ...

Syntax der Prädikatenlogik

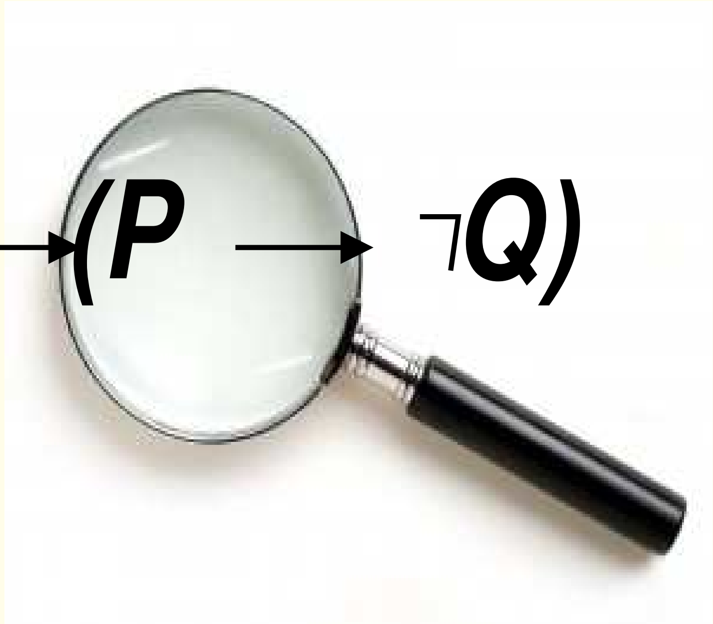
Idee

$$(P \wedge Q) \longrightarrow (P \longrightarrow \neg Q)$$

Syntax der Prädikatenlogik

Idee

$$(P \wedge Q) \longrightarrow (P \longrightarrow \neg Q)$$



Syntax der Prädikatenlogik

Idee

$(P \wedge Q) \longrightarrow$

Syntax der Prädikatenlogik

Idee

$$(P \wedge Q) \rightarrow (\mathit{even}(x) \rightarrow \neg$$

Syntax der Prädikatenlogik

Idee

$$(\mathit{even}(x) \wedge \mathit{even}(x+1)) \longrightarrow (\mathit{even}(x) \longrightarrow \neg\ \mathit{even}(x+1))$$

Syntax der Prädikatenlogik

Idee

$$(even(x) \bigwedge even(x+1)) \longrightarrow (even(x) \longrightarrow \neg\, even(x+1))$$

Semantik: Wahr in $\mathbb{N}$, für $x = 1$.

Syntax der Prädikatenlogik

Idee

$$\forall x \; (\mathit{even}(x) \wedge \mathit{even}(x+1)) \longrightarrow (\mathit{even}(x) \longrightarrow \neg \; \mathit{even}(x+1))$$

Semantik: Wahr in $\mathbb{N}$

Syntax der Prädikatenlogik: Logische Zeichen

Wie in der Aussagenlogik

$\top$ Symbol für die Formel "wahr" (Formel, die immer wahr ist)
$\bot$ Symbol für die Formel "falsch" (Formel, die immer falsch ist)

$\neg$ Negationssymbol ("nicht")

$\wedge$ Konjunktionssymbol ("und")
$\vee$ Disjunktionssymbol ("oder")
$\rightarrow$ Implikationssymbol ("wenn ... dann")
$\leftrightarrow$ Symbol für Äquivalenz ("genau dann, wenn")

( ) die beiden Klammern

Syntax der Prädikatenlogik: Logische Zeichen

Wie in der Aussagenlogik

$\top$ Symbol für die Formel "wahr" (Formel, die immer wahr ist)
$\bot$ Symbol für die Formel "falsch" (Formel, die immer falsch ist)

$\neg$ Negationssymbol ("nicht")

$\wedge$ Konjunktionssymbol ("und")
$\vee$ Disjunktionssymbol ("oder")
$\rightarrow$ Implikationssymbol ("wenn . . . dann")
$\leftrightarrow$ Symbol für Äquivalenz ("genau dann, wenn")

( ) die beiden Klammern

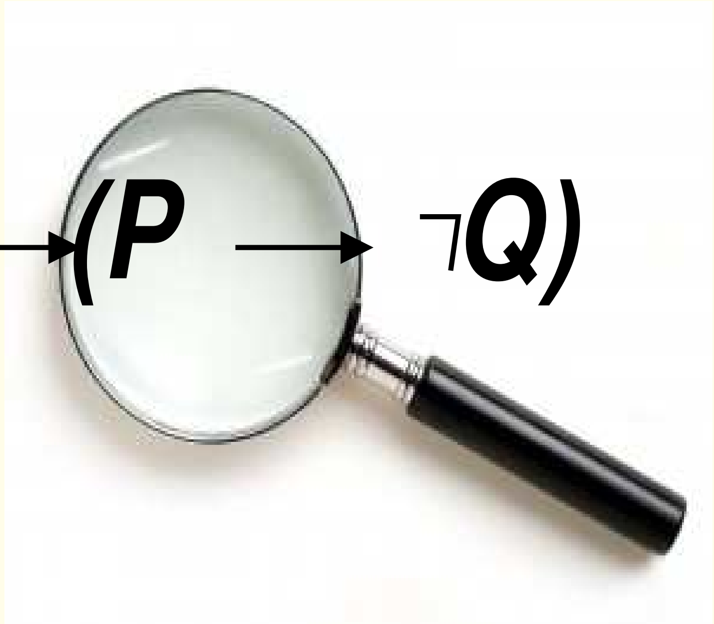
Quantoren

$\forall$ Allquantor ("für alle")
$\exists$ Existenzquantor ("es gibt")

Syntax der Prädikatenlogik

Idee

$$(P \wedge Q) \longrightarrow (P \longrightarrow \neg Q)$$



Syntax der Prädikatenlogik

Idee

$(P \wedge Q) \longrightarrow$

$\neg Q)$

Syntax der Prädikatenlogik

Idee

$(P \wedge Q) \rightarrow (\textit{even}(x) \rightarrow \neg$

Syntax der Prädikatenlogik: Vokabular

Signatur

Zweck: Festlegung der nichtlogischen Symbole

$$\Sigma = (\Omega, \Pi),$$

Syntax der Prädikatenlogik: Vokabular

Definition

Prädikatenlogische Signatur: Paar $\Sigma = (\Omega, \Pi)$ wobei:

- Ω eine Menge von Funktionssymbolen f mit Stelligkeit $n \geq 0$, geschrieben f/n,

  z.B. 2/0, koblenz/0, c/0, sqrt/1, leftLegOf/1, $+/2$, -/2

- Π Menge von Prädikatensymbolen p mit Stelligkeit $m \geq 0$, geschrieben p/m

  z.B. bruderVon/1, even/1, $> /2, \approx /2, \ldots$

Bemerkung: Das Gleichheitsprädikat $\approx$ kann (muss aber nicht) enthalten sein. Es wird infix notiert.

Syntax der Prädikatenlogik: Vokabular

Definition.

Funktionsssymbole mit Stelligkeit $n = 0$ heißen Konstante

z.B. 1, 2, koblenz, c

Syntax der Prädikatenlogik: Vokabular

Definition.

Funktionsssymbole mit Stelligkeit $n = 0$ heißen Konstante

z.B. 1, 2, koblenz, c

Definition.

Prädikatensymbole mit Stelligkeit $n = 0$ heißen Aussagenvariablen

Syntax der Prädikatenlogik: Vokabular

Variablen

Prädikatenlogik erlaubt die Formulierung abstrakter (schematischer) Aussagen. Technisches Hilfmittel hierfür sind die Variablen.

Wir nehmen an, dass

$$X$$

eine vorgegebene Menge von abzählbar unendlich vielen Symbolen ist, die wir für (die Bezeichnung von) Variablen verwenden.

Terme

Terme über Σ (bzw. Σ-Terme) mit Variablen in X werden nach folgenden syntaktischen Regeln gebildet:

$$
\begin{array}{lcll}
s, t, u, v & ::= & x & , x \in X \qquad \text{(Variable)} \\
& | & f(s_1, ..., s_n) & , f/n \in \Omega \qquad \text{(F-Terme)}
\end{array}
$$

Mit $T_\Sigma(X)$ bezeichnen wir die Menge der Σ-Terme mit Variablen in X.

Menge $T_\Sigma(X)$ der Σ-Terme mit Variablen in X:

Die kleinste Menge mit: $X \subseteq T_\Sigma(X)$

Wenn
- $f \in \Omega$,
- n ist die Stelligkeit von f
- $t_1, \ldots, t_n \in T_\Sigma(X)$

dann $f(t_1, \ldots, t_n) \in T_\Sigma(X)$

Terme

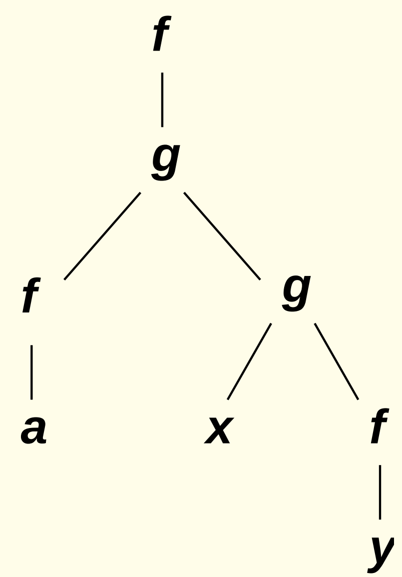
Terme sind also vollständig geklammerte Ausdrücke, die wir auch als markierte, geordnete Bäume auffassen können. Die Knoten sind mit Funktionssymbolen oder Variablen markiert. Jeder mit einem Funktionssymbol f der Stelligkeit n markierte Knoten hat genau n Unterbäume, einen für jedes Argument von f.

Beispiel:

$\Omega = \{a/0, f/1, g/2\}$

$x, y \in X$

$f(g(f(a), g(x, f(y))))$



Beispiele

$\Sigma = (\Omega, \Pi)$

$\Omega = \{Jan/0, Vater/1, Mutter/1\}$

$\{x, y, z\} \subseteq X$

Terme:

x	Jan
$Vater(x)$	$Vater(Jan)$
$Mutter(x)$	$Mutter(Jan)$
$Vater(Mutter(x))$	$Vater(Mutter(Jan))$

Beispiele

$\Sigma = (\Omega, \Pi)$

$\Omega = \{0/0, 1/0, succ/1, +/2\}$

$\{x, y, z\} \subseteq X$

Terme:

$x, y, z, 0, 1$

$succ(x), succ(0), succ(1)$

$x + y, x + z, x + 0, x + 1, (x + (succ(y) + 1)), \ldots$

Beispiele

$\Sigma = (\Omega, \Pi)$

$\Omega = \{0/0, 1/0, succ/1, +/2\}$

$\{x, y, z\} \subseteq X$

Terme:

$x, y, z, 0, 1$

$succ(x), succ(0), succ(1)$

$x + y, x + z, x + 0, x + 1, (x + (succ(y) + 1)), \ldots$ Infix

$+(x, y), +(x, z), +(x, 0), +(x, 1), +(x, +(succ(y), 1))$ Präfix

Atome

Atome (Atomare Formeln) über Σ genügen dieser Syntax:

$$\begin{array}{lcll} A, B & ::= & p(s_1, \ldots, s_m) & , \ p/m \in \Pi \\ \Big[& \mid & (s \approx t) & \text{(Gleichung)} \quad \Big] \end{array}$$

Ist $m = 0$, so handelt es sich bei p um eine Aussagenvariable. Wir verwenden insbesondere die Buchstaben P, Q, R, S, um Aussagenvariablen zu bezeichnen

Beispiele

$\Sigma = (\Omega, \Pi)$

$\Omega = \{Jan/0, Anna/0, Vater/1, Mutter/1\}$

$\Pi = \{Mann/1, Frau/1, Bruder/2, \approx/2\}$

$\{x, y, z\} \subseteq X$

Atome:

$Mann(x)$	$Mann(Jan)$	$Mann(Anna)$
$Mann(Vater(x))$	$Frau(Vater(Jan))$	$Mann(Vater(Jan))$
$Bruder(x, y)$	$Bruder(x, Jan)$	$Bruder(x, Anna)$

$Vater(Jan) \approx Vater(Anna)$

$Vater(x) \approx Mutter(y)$

Beispiele

$\Sigma = (\Omega, \Pi)$

$\Omega = \{0/0, 1/0, succ/1, +/2\}$

$\Pi = \{\leq, \approx, \text{even}, \text{odd}\}$

$\{x, y, z\} \subseteq X$

Atome:

$x \leq y, \quad z \approx 0, \quad 0 \leq 1$

$succ(x) \leq succ(0), \quad succ(1) \approx succ(0)$

$x + y \leq x + z, \quad x + 0 \approx x + 1 \quad (x + (succ(y) + 1)) \leq z, \ldots$ Infix

$\text{even}(succ(0)), \quad \text{even}(x), \quad \text{even}(x + 1), \quad \text{odd}(x + (succ(y) + y))$

Beispiele

$\Sigma = (\Omega, \Pi)$

$\Omega = \{0/0, 1/0, succ/1, +/2\}$

$\Pi = \{\leq, \approx, \text{even}, \text{odd}\}$

$\{x, y, z\} \subseteq X$

Atome:

$x \leq y, \quad z \approx 0, \quad 0 \leq 1$

$succ(x) \leq succ(0), \quad succ(1) \approx succ(0)$

$x + y \leq x + z, \quad x + 0 \approx x + 1 \quad (x + (succ(y) + 1)) \leq z, \ldots$ Infix

Präfix:

$\leq (+(x, y), +(x, z)), \quad + (x, 0) \approx +(x, 1), \quad \leq (+(x, +(succ(y), 1)), z)$

$\text{even}(succ(0)), \quad \text{even}(x), \quad \text{even}(x + 1), \quad \text{odd}(x + (succ(y) + y))$

Literale

$$
\begin{array}{lcll}
L & ::= & A & \text{(positives Literal)} \\
 & \mid & \neg A & \text{(negatives Literal)}
\end{array}
$$

Beispiele:

$$
\begin{array}{ll}
Mann(Vater(x)) & Vater(Jan) \approx Vater(Anna) \\
\neg Mann(Vater(x)) & \neg(Vater(Jan) \approx Vater(Anna))
\end{array}
$$

$$
\begin{array}{ll}
x \leq y & (x + (succ(y) + 1)) \approx z \\
\neg x \leq y & \neg((x + (succ(y) + 1)) \approx z)
\end{array}
$$

Klauseln

$$
\begin{array}{llll}
C, D & ::= & \bot & \text{(leere Klausel)} \\
 & | & L_1 \vee \ldots \vee L_k,\ k \geq 1 & \text{(nichtleere Klausel)}
\end{array}
$$

Beispiele:

$\bot$

$Mann(Vater(x)) \quad \vee \quad \neg(Vater(Jan) \approx Vater(Anna))$

$\neg Frau(Vater(x))$

$\neg(x \leq y) \quad \vee \quad (x + (succ(y) + 1)) \approx z$

$x \leq y$

$\neg(x \leq y) \quad \vee \quad \neg(x \leq y) \quad \vee \quad (x \leq y)$

Formeln

Formeln über Σ:

$$
\begin{array}{lcll}
F, G, H & ::= & \bot & \text{(Falsum)} \\
& \mid & \top & \text{(Verum)} \\
& \mid & A & \text{(atomare Formel)} \\
& \mid & \neg F & \text{(Negation)} \\
& \mid & (F \wedge G) & \text{(Konjunktion)} \\
& \mid & (F \vee G) & \text{(Disjunktion)} \\
& \mid & (F \rightarrow G) & \text{(Implikation)} \\
& \mid & (F \leftrightarrow G) & \text{(Äquivalenz)} \\
& \mid & \forall x F & \text{(Allquantifizierung)} \\
& \mid & \exists x F & \text{(Existenzquantifizierung)}
\end{array}
$$

Formeln

Menge For_Σ der Formeln über Σ mit Variablen in X:

Die kleinste Menge, die

- Alle atomaren Formeln enthält,
- $\top \in \mathsf{For}_\Sigma$, $\bot \in \mathsf{For}_\Sigma$,
- Wenn $F, G \in \mathsf{For}_\Sigma$, dann auch
  $\neg F, F \wedge G, F \vee G, F \rightarrow G, F \leftrightarrow G \in \mathsf{For}_\Sigma$,
- Wenn $F \in \mathsf{For}_\Sigma$ und $x \in X$, dann
  $\forall x F \in \mathsf{For}_\Sigma, \exists x F \in \mathsf{For}_\Sigma$

Konventionen zur Notation

- Klammereinsparungen werden nach folgenden Regeln vorgenommen:
  - $\neg \quad >_p \quad \wedge \quad >_p \quad \vee \quad >_p \quad \rightarrow \quad >_p \quad \leftrightarrow$ (Präzedenzen),
  - $\vee$ und $\wedge$ sind assoziativ und kommutativ.
- $Qx_1, \ldots, x_n\ F$ für $Qx_1 \ldots Qx_n\ F$.
- Terme und Atome in Infix-, Präfix-, Postfix- oder Mixfixnotation; Beispiele:

$s + t$	für $+(s, t)$
$s \leq t$	für $\leq (s, t)$
$-s$	für $-(s)$
0	für $0()$

Beispiel

“Alle, die in Koblenz studieren, sind schlau”

Beispiel

“Alle, die in Koblenz studieren, sind schlau”

$\Omega = \{\mathit{koblenz}/0\} \quad \Pi = \{\mathit{studiertIn}/2, \mathit{schlau}/1\}$

Beispiel

"Alle, die in Koblenz studieren, sind schlau"

$\Omega = \{koblenz/0\} \quad \Pi = \{studiertIn/2, schlau/1\}$

$$\underbrace{\forall x(\underbrace{\underbrace{studiertIn(\underbrace{x}_{\substack{Variable\\Term}}, \underbrace{koblenz}_{\substack{Konstante\\Term}})}_{Atom} \to \underbrace{schlau(\underbrace{x}_{\substack{Variable\\Term}})}_{Atom}}_{Formel})}_{Formel}$$

Beispiel

"Es gibt jemand, der in Landau studiert und schlau ist"

$\Omega = \{landau/0\} \qquad \Pi = \{studiertIn/2, schlau/1\}$

$$\underbrace{\exists x(\underbrace{\underbrace{studiertIn(\underbrace{x}_{\substack{\mathit{Variable}\\ \mathit{Term}}}, \underbrace{landau}_{\substack{\mathit{Konstante}\\ \mathit{Term}}})}_{\mathit{Atom}} \wedge \underbrace{schlau(\underbrace{x}_{\substack{\mathit{Variable}\\ \mathit{Term}}})}_{\mathit{Atom}}}_{\mathit{Formel}})}_{\mathit{Formel}}$$

Beispiele

X Variablenmenge, $x, y, z \in X$; $\quad \Sigma = \{\Omega, \Pi\}$, mit

$\Omega = \{f/2, g/3, a/0, b/0\}$ und $\Pi = \{p/2, q/1, r/0\}$

	Term	Atom	Literal	Klausel	Formel	Nichts
x	ja	nein	nein	nein	nein	nein
a	ja	nein	nein	nein	nein	nein
$f(g(a, y, b), x)$	ja	nein	nein	nein	nein	nein
$g(a, x)$	nein	nein	nein	nein	nein	ja
$p(g(a, y, b), x)$	nein	ja	ja	ja	ja	nein
$p(q(a), b)$	nein	nein	nein	nein	nein	ja
$\neg q(g(a, b, x))$	nein	nein	ja	ja	ja	nein
$\neg q(g(a, b, x)) \vee a$	nein	nein	nein	nein	nein	ja
$\neg q(g(a, b, x)) \vee p(a, a)$	nein	nein	nein	ja	ja	nein
$\forall x(p(x, f(x, x)))$	nein	nein	nein	nein	ja	nein
$\forall b(p(b, f(b, b)))$	nein	nein	nein	nein	nein	ja

Beispiel

“Alle, die in Koblenz studieren, sind schlau”

$\Omega = \{koblenz/0\} \quad \Pi = \{studiertIn/2, schlau/1\}$

$$\forall x(studiertIn(x, koblenz) \rightarrow schlau(x))$$

“Es gibt jemand, der in Landau studiert und schlau ist”

$\Omega = \{landau/0\} \quad \Pi = \{studiertIn/2, schlau/1\}$

$$\exists x(studiertIn(x, landau) \land schlau(x)$$

Formalisierung in Prädikatenlogik

Universelle Quantifizierung:

Faustregel: $\rightarrow$ ist der logische (Top-level-)Operator mit $\forall$

Häufiger Fehler: Verwendung von $\wedge$ mit $\forall$

Formalisierung in Prädikatenlogik

Universelle Quantifizierung:

Faustregel: $\rightarrow$ ist der logische (Top-level-)Operator mit $\forall$

Häufiger Fehler: Verwendung von $\wedge$ mit $\forall$

Beispiel

"Alle, die in Koblenz studieren, sind schlau."

Richtig: $\forall x(\mathit{studiertIn}(x, \mathit{koblenz}) \rightarrow \mathit{schlau}(x))$

Falsch: $\forall x(\mathit{studiertIn}(x, \mathit{koblenz}) \wedge \mathit{schlau}(x))$

"Alle studieren in Koblenz und sind schlau."

Formalisierung in Prädikatenlogik

Existenzielle Quantifizierung

Faustregel: $\wedge$ ist der logische (Top-level-)Operator mit $\exists$

Häufiger Fehler: Verwendung von $\rightarrow$ mit $\exists$

Formalisierung in Prädikatenlogik

Existenzielle Quantifizierung

Faustregel: $\wedge$ ist der logische (Top-level-)Operator mit $\exists$

Häufiger Fehler: Verwendung von $\rightarrow$ mit $\exists$

Beispiel

"Es gibt jemand, der in Landau studiert und schlau ist."

Richtig: $\exists x(\mathit{studiertIn}(x, \mathit{landau}) \wedge \mathit{schlau}(x))$

Falsch: $\exists x(\mathit{studiertIn}(x, \mathit{landau}) \rightarrow \mathit{schlau}(x))$

"Es gibt jemanden, der, falls er/sie in Landau studiert, schlau ist."
Trivial wahr, wenn es irgendjemanden gibt, der nicht in Landau studiert.

Beispiel: Tante Agatha

Jemand, der in Schloss Dreadbury wohnt, hat Tante Agatha ermordet. Agatha, ihr Butler und ihr Neffe Charles waren die einzigen Bewohner von Schloss Dreadbury. Ein Mörder hasst immer sein Opfer und ist niemals reicher als sein Opfer. Charles hasst niemanden, den Tante Agatha gehasst hat. Agatha hat jeden gehasst außer ihrem Butler. Der Butler hasst jeden, der nicht reicher ist als Tante Agatha. Der Butler hasst jeden, den Tante Agatha gehasst hat. Niemand hasst jeden. Agatha war nicht der Butler.

Wer hat Tante Agatha ermordet?

Beispiel: Tante Agatha

Jemand, der in Schloss Dreadbury wohnt, hat Tante Agatha ermordet.

Agatha, ihr Butler und ihr Neffe Charles waren die einzigen Bewohner von Schloss Dreadbury.

Beispiel: Tante Agatha

Jemand, der in Schloss Dreadbury wohnt, hat Tante Agatha ermordet.

- $\exists x \ (\text{schlossbewohner}(x) \wedge \text{ermordet}(x, \text{a}))$

Agatha, ihr Butler und ihr Neffe Charles waren die einzigen Bewohner von Schloss Dreadbury.

Beispiel: Tante Agatha

Jemand, der in Schloss Dreadbury wohnt, hat Tante Agatha ermordet.

- $\exists x\ (\mathsf{schlossbewohner}(x) \land \mathsf{ermordet}(x, \mathsf{a}))$

Agatha, ihr Butler und ihr Neffe Charles waren die einzigen Bewohner von Schloss Dreadbury.

- $\forall x\ (\mathsf{schlossbewohner}(x) \leftrightarrow (x \approx \mathsf{a} \lor x \approx \mathsf{b} \lor x \approx \mathsf{c}))$

Beispiel: Tante Agatha

Ein Mörder hasst immer sein Opfer und ist niemals reicher als sein Opfer.

Charles hasst niemanden, den Tante Agatha gehasst hat.

Agatha hat jeden gehasst außer ihrem Butler.

Beispiel: Tante Agatha

Ein Mörder hasst immer sein Opfer und ist niemals reicher als sein Opfer.

- $\forall x, y \; (\text{ermordet}(x, y) \to \text{hasst}(x, y))$
  $\forall x, y \; (\text{ermordet}(x, y) \to \neg\, \text{reicher}(x, y))$

Charles hasst niemanden, den Tante Agatha gehasst hat.

Agatha hat jeden gehasst außer ihrem Butler.

Beispiel: Tante Agatha

Ein Mörder hasst immer sein Opfer und ist niemals reicher als sein Opfer.

- $\forall x, y\ (\text{ermordet}(x, y) \rightarrow \text{hasst}(x, y))$
  $\forall x, y\ (\text{ermordet}(x, y) \rightarrow \neg\, \text{reicher}(x, y))$

Charles hasst niemanden, den Tante Agatha gehasst hat.

- $\forall x\ (\text{hasst}(\text{c}, x) \rightarrow \neg\, \text{hasst}(\text{a}, x))$

Agatha hat jeden gehasst außer ihrem Butler.

Beispiel: Tante Agatha

Ein Mörder hasst immer sein Opfer und ist niemals reicher als sein Opfer.

- $\forall x, y \; (\text{ermordet}(x, y) \rightarrow \text{hasst}(x, y))$
  $\forall x, y \; (\text{ermordet}(x, y) \rightarrow \neg \, \text{reicher}(x, y))$

Charles hasst niemanden, den Tante Agatha gehasst hat.

- $\forall x \; (\text{hasst}(\text{c}, x) \rightarrow \neg \, \text{hasst}(\text{a}, x))$

Agatha hat jeden gehasst außer ihrem Butler.

- $\forall x \; (\neg \, \text{hasst}(\text{a}, x) \leftrightarrow x \approx \text{b})$

Beispiel: Tante Agatha

Der Butler hasst jeden, der nicht reicher ist als Tante Agatha.

- $\forall x\ (\neg\, \mathsf{reicher}(x, \mathsf{a}) \to \mathsf{hasst}(\mathsf{b}, x))$

Der Butler hasst jeden, den Tante Agatha gehasst hat.

- $\forall x\ (\mathsf{hasst}(\mathsf{a}, x) \to \mathsf{hasst}(\mathsf{b}, x))$

Niemand hasst jeden.

- $\forall x\ \exists y\ (\neg\, \mathsf{hasst}(x, y))$

Agatha war nicht der Butler.

- $\neg\, \mathsf{a} \approx \mathsf{b}$

Beispiel: Arithmetik

$$
\begin{aligned}
\Sigma_{PA} &= (\Omega_{PA},\ \Pi_{PA}) \\
\Omega_{PA} &= \{0/0,\ +/2,\ */2,\ s/1\} \\
\Pi_{PA} &= \{\leq /2,\ < /2, \approx /2\}
\end{aligned}
$$

$+,\ *$ infix; $* >_p +$

Formelbeispiele über dieser Signatur sind

$$
\begin{aligned}
&\forall x, y(x \leq y \leftrightarrow \exists z(x + z \approx y)) \\
&\exists x \forall y(x + y \approx y) \\
&\forall x, y(x * s(y) \approx x * y + x) \\
&\forall x, y(s(x) \approx s(y) \rightarrow x \approx y) \\
&\forall x \exists y\ x < y
\end{aligned}
$$

Übersicht

Prädikatenlogik

- **Syntax**

  Signatur

  Terme

  Atome

  Literale

  Klauseln

  Formeln

  Formalisierung in Prädikatenlogik

  Beispiele