

# Logik für Informatiker

## 3. Prädikatenlogik

### Teil 2

23.05.2019

Viorica Sofronie-Stokkermans

Universität Koblenz-Landau

e-mail: [sofronie@uni-koblenz.de](mailto:sofronie@uni-koblenz.de)

# Prädikatenlogik

---

## Syntax

### 1. Logische Symbole:

1.1: Wie in der Aussagenlogik:  $\top, \perp; \neg; \vee, \wedge, \rightarrow, \leftrightarrow$

1.2: Quantoren:  $\forall, \exists$ .

### 2. Nichtlogische Symbole: Signatur $\Sigma = (\Omega, \Pi)$ ,

2.1:  $\Omega$  Menge von Funktionssymbolen. **Notation:**  $f/n$ :  $f$  hat Stelligkeit  $n \geq 0$ ,

2.2:  $\Pi$  Menge von Prädikatensymbolen. **Notation:**  $p/m$ :  $p$  hat Stelligkeit  $m \geq 0$ .

(Das Gleichheitsprädikat  $\approx$  kann (muss aber nicht) enthalten sein.)

Funktionssymbole mit Stelligkeit  $n = 0$  heißen Konstanten.

Prädikatensymbole mit Stelligkeit  $n = 0$  heißen Aussagenvariablen.

### 3. Variablen: $X$ vorgegebene Menge von abzählbar unendlich vielen Symbolen, die wir für (die Bezeichnung von) **Variablen** verwenden.

# Terme

---

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

**Menge  $T_\Sigma(X)$  der  $\Sigma$ -Terme:**

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

Wenn •  $f \in \Omega$ ,

•  $n$  ist die Stelligkeit von  $f$

•  $t_1, \dots, t_n \in T_\Sigma(X)$

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

# Atome

---

Atome (Atomare Formeln) über  $\Sigma$  genügen dieser Syntax:

$$A, B ::= p(s_1, \dots, s_m) \quad , p/m \in \Pi$$
$$\left[ \quad \mid \quad (s \approx t) \quad \text{(Gleichung)} \quad \right]$$

wobei  $s_1, \dots, s_m, s, t \in T_\Sigma(X)$ .

Ist  $m = 0$ , so handelt es sich bei  $p$  um eine **Aussagenvariable**.

## Atome:

Formeln der Form  $p(s_1, \dots, s_m)$ , wobei  $p/m \in \Pi$  und  $s_1, \dots, s_m \in T_\Sigma(X)$ .

# Literale

---

$L ::= A$  (positives Literal)  
|  $\neg A$  (negatives Literal)

## Beispiele:

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

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

# Klauseln

---

$C, D ::= \perp$  (leere Klausel)  
|  $L_1 \vee \dots \vee L_k, k \geq 1$  (nichtleere Klausel)

## Beispiele:

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

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

$x \leq y$

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

# Formeln

---

Formeln über  $\Sigma$ :

$F, G, H$	$::=$	$\perp$	(Falsum)
		$\top$	(Verum)
		$A$	(atomare Formel)
		$\neg F$	(Negation)
		$(F \wedge G)$	(Konjunktion)
		$(F \vee G)$	(Disjunktion)
		$(F \rightarrow G)$	(Implikation)
		$(F \leftrightarrow G)$	(Äquivalenz)
		$\forall x F$	(Allquantifizierung)
		$\exists x F$	(Existenzquantifizierung)

# Formeln

---

## Menge $\text{For}_\Sigma$ der Formeln über $\Sigma$ :

Die kleinste Menge, die

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

# Konventionen zur Notation

---

- Klammereinsparungen werden nach folgenden Regeln vorgenommen:
  - $\neg$   $>_p$   $\wedge$   $>_p$   $\vee$   $>_p$   $\rightarrow$   $>_p$   $\leftrightarrow$  (Präzedenzen),
  - $\vee$  und  $\wedge$  sind assoziativ und kommutativ.

- $Q_{x_1, \dots, x_n} F$  für  $Q_{x_1} \dots Q_{x_n} F$ .

- 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  $0()$

# Beispiele

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

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

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

# 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:**  $\forall x(\text{studiertIn}(x, \text{koblenz}) \rightarrow \text{schlau}(x))$

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

“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:**  $\exists x(\text{studiertIn}(x, \text{landau}) \wedge \text{schlau}(x))$

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

“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.

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

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.

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

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

- ▶  $\forall x (\text{schlossbewohner}(x) \leftrightarrow (x \approx a \vee x \approx b \vee x \approx c))$

# 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.

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

Charles hasst niemanden, den Tante Agatha gehasst hat.

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.

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

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.

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

Agatha hat jeden gehasst außer ihrem Butler.

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

# Beispiel: Tante Agatha

---

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

▶  $\forall x (\neg \text{reicher}(x, a) \rightarrow \text{hasst}(b, x))$

Der Butler hasst jeden, den Tante Agatha gehasst hat.

▶  $\forall x (\text{hasst}(a, x) \rightarrow \text{hasst}(b, x))$

Niemand hasst jeden.

▶  $\forall x \exists y (\neg \text{hasst}(x, y))$

Agatha war nicht der Butler.

▶  $\neg a \approx b$

# Beispiel: Arithmetik

---

$$\Sigma_{PA} = (\Omega_{PA}, \Pi_{PA})$$

$$\Omega_{PA} = \{0/0, +/2, */2, s/1\}$$

$$\Pi_{PA} = \{\leq /2, < /2, \approx /2\}$$

$$+, * \text{ infix}; * >_p +$$

Formelbeispiele über dieser Signatur sind

$$\forall x, y (x \leq y \leftrightarrow \exists z (x + z \approx y))$$

$$\exists x \forall y (x + y \approx y)$$

$$\forall x, y (x * s(y) \approx x * y + x)$$

$$\forall x, y (s(x) \approx s(y) \rightarrow x \approx y)$$

$$\forall x \exists y x < y$$

# Gebundene und freie Variablen

---

# Gebundene und freie Variablen

---

## Definitionen:

- In  $QxF$ ,  $Q \in \{\exists, \forall\}$ , heißt  $F$  der **Bindungsbereich** des Quantors  $Qx$ .
- Ein Auftreten einer Variablen  $x$  heißt **gebunden**, wenn es zum Bindungsbereich eines Quantors  $Qx$  gehört.
- Alle anderen Auftreten von Variablen heißen **frei**.

Formeln ohne freie Variablen heißen **Satzformen**.

Variablenfreie Formeln heißen **Grundformeln**.

# Beispiel

---

$$\forall y \quad (\forall x \quad p(x) \rightarrow q(x, y))$$

*Bindungsbereich*

*Bind.*

# Beispiel

---

$$p(z) \rightarrow \forall x (q(x, z) \wedge \exists y r(y, z))$$

# Beispiel

---

$$p(z) \rightarrow \forall x (q(x, z) \wedge \exists z r(y, z))$$

- $x$  gebunden
- $y$  frei
- $z$  frei und gebunden

# Substitution eines Termes für eine Variable

---

# Substitution eines Termes für eine Variable

---

Mit  $F[s/x]$  bezeichnen wir das Resultat der Substitution aller freien Auftreten von  $x$  in  $F$  durch den Term  $s$ .  $F[s/x]$  sei durch strukturelle Induktion über den Aufbau von  $F$  wie folgt definiert:

$$x[s/x] = s$$

$$x'[s/x] = x' ; \text{ falls } x' \neq x$$

$$f(s_1, \dots, s_n)[s/x] = f(s_1[s/x], \dots, s_n[s/x])$$

$$\perp[s/x] = \perp$$

$$\top[s/x] = \top$$

$$p(s_1, \dots, s_n)[s/x] = p(s_1[s/x], \dots, s_n[s/x])$$

$$(u \approx v)[s/x] = (u[s/x] \approx v[s/x])$$

$$\neg F[s/x] = \neg(F[s/x])$$

$$(F \rho G)[s/x] = (F[s/x] \rho G[s/x]) ; \text{ für alle binären Junktoren } \rho \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$$

$$(QyF)[s/x] = Qz((F[z/y])[s/x]) ; \text{ } z \text{ neue Variable}$$

# Beispiel

---

## Terme:

$$\Sigma = (\Omega, \Pi), \text{ mit } \Omega = \{f/1, g/2, a/0\}, \Pi = \{p/2, \approx /2\}$$

$$x, y, z, u \in X$$

$$\begin{aligned} g(f(x), g(f(x), z)) \ [g(y, u)/x] \\ = g(f(g(y, u)), g(f(g(y, u)), z)) \end{aligned}$$

# Beispiel

---

## Atome:

$$\Sigma = (\Omega, \Pi), \text{ mit } \Omega = \{f/1, g/2, a/0\}, \Pi = \{p/2, \approx /2\}$$

$$x, y, z, u \in X$$

$$\begin{aligned} p(g(f(x), g(f(x), z)), y) [g(y, u)/x] \\ = p(g(f(g(y, u)), g(f(g(y, u)), z)), y) \end{aligned}$$

# Beispiel

---

## Formeln ohne Quantoren:

$$\Sigma = (\Omega, \Pi), \text{ mit } \Omega = \{f/1, g/2, a/0\}, \Pi = \{p/2, \approx /2\}$$

$$x, y, z, u \in X$$

$$\begin{aligned} p(g(f(x), g(f(x), z)), y) \wedge (g(x, y) \approx f(g(z, z))) & [g(y, u)/x] \\ = p(g(f(g(y, u)), g(f(g(y, u)), z)), y) \wedge (g(g(y, u), y) \approx f(g(z, z))) \end{aligned}$$

# Formeln mit Quantoren: Problematik

---

$$(QyF)[s/x] = Qz((F[z/y])[s/x]) ; \quad z \text{ neue Variable}$$

Der Grund für die Umbenennung der gebundenen Variablen  $y$  in eine neue „unbenutzte“ Variable  $z$  ist die Vermeidung des Einfangens freier Variablen in  $s$ .

Sollte  $y$  in  $s$  auftreten, wären sonst diese Auftreten nach erfolgter Substitution gebunden.

# Beispiel

---

## Formeln mit Quantoren:

$\Sigma = (\Omega, \Pi)$ , mit  $\Omega = \{f/1, g/2, a/0\}$ ,  $\Pi = \{p/2, \approx /2\}$

$(\forall x(p(x, g(f(x), y))) \wedge \exists z(x \approx g(y, z))) [g(y, z)/x]$

# Beispiel

---

## Formeln mit Quantoren:

$$\Sigma = (\Omega, \Pi), \text{ mit } \Omega = \{f/1, g/2, a/0\}, \Pi = \{p/2, \approx /2\}$$

$$\begin{aligned} & ( \forall x(p(x, g(f(x), y))) \wedge \exists z(x \approx g(y, z)) ) [g(y, z)/x] \\ & = \forall x(p(x, g(f(x), y)))[g(y, z)/x] \wedge \exists z(x \approx g(y, z))[g(y, z)/x] \end{aligned}$$

# Beispiel

---

## Formeln mit Quantoren:

$$\Sigma = (\Omega, \Pi), \text{ mit } \Omega = \{f/1, g/2, a/0\}, \Pi = \{p/2, \approx /2\}$$

$$\begin{aligned} & ( \forall x(p(x, g(f(x), y))) \wedge \exists z(x \approx g(y, z)) ) [g(y, z)/x] \\ & = \forall x(p(x, g(f(x), y))) [g(y, z)/x] \wedge \exists z(x \approx g(y, z)) [g(y, z)/x] \\ & = \forall v(p(x, g(f(x), y)) [v/x]) [g(y, z)/x] \wedge \exists u((x \approx g(y, z)) [u/z]) [g(y, z)/x] \end{aligned}$$

# Beispiel

---

## Formeln mit Quantoren:

$$\Sigma = (\Omega, \Pi), \text{ mit } \Omega = \{f/1, g/2, a/0\}, \Pi = \{p/2, \approx /2\}$$

$$\begin{aligned} & ( \forall x(p(x, g(f(x), y))) \wedge \exists z(x \approx g(y, z)) ) \quad [g(y, z)/x] \\ & = \forall x(p(x, g(f(x), y))) [g(y, z)/x] \wedge \exists z(x \approx g(y, z)) [g(y, z)/x] \\ & = \forall v(p(x, g(f(x), y)) [v/x]) [g(y, z)/x] \wedge \exists u((x \approx g(y, z)) [u/z]) [g(y, z)/x] \\ & = \forall v(p(v, g(f(v), y))) [g(y, z)/x] \wedge \exists u((x \approx g(y, u)) [g(y, z)/x]) \end{aligned}$$

# Beispiel

---

## Formeln mit Quantoren:

$$\Sigma = (\Omega, \Pi), \text{ mit } \Omega = \{f/1, g/2, a/0\}, \Pi = \{p/2, \approx /2\}$$

$$\begin{aligned} & ( \forall x(p(x, g(f(x), y))) \wedge \exists z(x \approx g(y, z)) ) [g(y, z)/x] \\ &= \forall x(p(x, g(f(x), y))) [g(y, z)/x] \wedge \exists z(x \approx g(y, z)) [g(y, z)/x] \\ &= \forall v(p(x, g(f(x), y)) [v/x]) [g(y, z)/x] \wedge \exists u(((x \approx g(y, z)) [u/z]) [g(y, z)/x]) \\ &= \forall v(p(v, g(f(v), y))) [g(y, z)/x] \wedge \exists u((x \approx g(y, u)) [g(y, z)/x]) \\ &= \forall v(p(v, g(f(v), y))) \wedge \exists u(g(y, z) \approx g(y, u)) \end{aligned}$$

# Substitution allgemein

---

**Definition.** **Substitutionen** sind Abbildungen

$$\sigma : X \rightarrow T_{\Sigma}(X),$$

so dass der **Bereich** von  $\sigma$ , d.h. die Menge

$$\text{dom}(\sigma) = \{x \in X \mid \sigma(x) \neq x\},$$

endlich ist. Die Menge der **eingeführten Variablen**, d.h. der Variablen die in einem der Terme  $\sigma(x)$ , für  $x \in \text{dom}(\sigma)$ , auftreten, wird mit  $\text{codom}(\sigma)$  bezeichnet.

Substitutionen schreiben wir auch als  $[s_1/x_1, \dots, s_n/x_n]$ ,  $x_i$  pw. verschieden, und meinen dann die Abbildung

$$[s_1/x_1, \dots, s_n/x_n](y) = \begin{cases} s_i, & \text{falls } y = x_i \\ y, & \text{sonst} \end{cases}$$

Ab jetzt schreiben wir für die Applikation  $\sigma(x)$  von Substitutionen  $x\sigma$ .

# Anwendung einer Substitution

---

„Homomorphe“ Fortsetzung von  $\sigma$  auf Terme und Formeln:

$$f(s_1, \dots, s_n)\sigma = f(s_1\sigma, \dots, s_n\sigma)$$

$$\perp\sigma = \perp$$

$$\top\sigma = \top$$

$$p(s_1, \dots, s_n)\sigma = p(s_1\sigma, \dots, s_n\sigma)$$

$$(u \approx v)\sigma = (u\sigma \approx v\sigma)$$

$$\neg F\sigma = \neg(F\sigma)$$

$$(F\rho G)\sigma = (F\sigma\rho G\sigma); \text{ f\u00fcr alle bin\u00e4ren Junktoren } \rho$$

$$(QxF)\sigma = Qz((F[z/x])\sigma); \text{ wobei } z \text{ „neue“ Variable}$$

Ab\u00e4nderung einer Substitution  $\sigma$  an  $x$  zu  $t$ :

$$\sigma[x \rightarrow t](y) = \begin{cases} t, & \text{falls } y = x \\ \sigma(y), & \text{sonst} \end{cases}$$

# Beispiel

---

## Anwendung einer Substitution auf Terme

$$\Sigma = (\Omega, \Pi), \text{ mit } \Omega = \{f/1, g/2, a/0\}, \Pi = \{p/2, \approx /2\}$$

$$\begin{aligned} g(f(x), g(f(x), z)) & [g(y, u)/x, f(x)/y, a/z] \\ &= g(f(g(y, u)), g(f(g(y, u)), a)) \end{aligned}$$

# Beispiel

---

## Anwendung einer Substitution auf Atome

$$\Sigma = (\Omega, \Pi), \text{ mit } \Omega = \{f/1, g/2, a/0\}, \Pi = \{p/2, \approx /2\}$$

$$\begin{aligned} p(g(f(x), g(f(x), z)), y) [g(y, u)/x, f(x)/y, a/z] \\ = p(g(f(g(y, u)), g(f(g(y, u)), a)), f(x)) \end{aligned}$$

# Beispiel

---

## Anwendung einer Substitution auf Formeln ohne Quantoren

$$\Sigma = (\Omega, \Pi), \text{ mit } \Omega = \{f/1, g/2, a/0\}, \Pi = \{p/2, \approx /2\}$$

$$\begin{aligned} & p(g(f(x), g(f(x), z)), y) \wedge (g(x, y) \approx f(g(z, z))) [g(y, u)/x, f(x)/y, a/z] \\ & = p(g(f(g(y, u)), g(f(g(y, u)), a)), f(x)) \wedge (g(g(y, u), f(x)) \approx f(g(a, a))) \end{aligned}$$

# Beispiel

---

## Anwendung einer Substitution auf Formeln mit Quantoren

$$\Sigma = (\Omega, \Pi), \text{ mit } \Omega = \{f/1, g/2, a/0\}, \Pi = \{p/2, \approx /2\}$$

$$(\forall x(p(x, g(f(x), y))) \wedge \exists z(x \approx g(y, z)))[g(y, z)/x, f(x)/y, a/z]$$

# Beispiel

---

## Anwendung einer Substitution auf Formeln mit Quantoren

$$\Sigma = (\Omega, \Pi), \text{ mit } \Omega = \{f/1, g/2, a/0\}, \Pi = \{p/2, \approx /2\}$$

$$\begin{aligned} & (\forall x(p(x, g(f(x), y))) \wedge \exists z(x \approx g(y, z)))[g(y, z)/x, f(x)/y, a/z] \\ &= \forall x(p(x, g(f(x), y)))[g(y, z)/x, f(x)/y, a/z] \wedge \\ & \quad \exists z(x \approx g(y, z))[g(y, z)/x, f(x)/y, a/z] \end{aligned}$$

# Beispiel

---

## Anwendung einer Substitution auf Formeln mit Quantoren

$$\Sigma = (\Omega, \Pi), \text{ mit } \Omega = \{f/1, g/2, a/0\}, \Pi = \{p/2, \approx /2\}$$

$$(\forall x(p(x, g(f(x), y))) \wedge \exists z(x \approx g(y, z)))[g(y, z)/x, f(x)/y, a/z]$$

$$= \forall x(p(x, g(f(x), y)))[g(y, z)/x, f(x)/y, a/z] \wedge \\ \exists z(x \approx g(y, z))[g(y, z)/x, f(x)/y, a/z]$$

$$= \forall v((p(x, g(f(x), y))[v/x])[g(y, z)/x, f(x)/y, a/z]) \wedge \\ \exists u(((x \approx g(y, z))[u/z])[g(y, z)/x, f(x)/y, a/z])$$

# Beispiel

---

## Anwendung einer Substitution auf Formeln mit Quantoren

$$\Sigma = (\Omega, \Pi), \text{ mit } \Omega = \{f/1, g/2, a/0\}, \Pi = \{p/2, \approx /2\}$$

$$\begin{aligned} & (\forall x(p(x, g(f(x), y))) \wedge \exists z(x \approx g(y, z)))[g(y, z)/x, f(x)/y, a/z] \\ &= \forall x(p(x, g(f(x), y))[g(y, z)/x, f(x)/y, a/z] \wedge \\ & \quad \exists z(x \approx g(y, z))[g(y, z)/x, f(x)/y, a/z] \\ &= \forall v((p(x, g(f(x), y))[v/x])[g(y, z)/x, f(x)/y, a/z]) \wedge \\ & \quad \exists u(((x \approx g(y, z))[u/z])[g(y, z)/x, f(x)/y, a/z]) \\ &= \forall v(p(v, g(f(v), f(x)))) \wedge \exists u((x \approx g(y, u))[g(y, z)/x, f(x)/y, a/z]) \end{aligned}$$

# Beispiel

---

## Anwendung einer Substitution auf Formeln mit Quantoren

$$\Sigma = (\Omega, \Pi), \text{ mit } \Omega = \{f/1, g/2, a/0\}, \Pi = \{p/2, \approx /2\}$$

$$\begin{aligned} & (\forall x(p(x, g(f(x), y))) \wedge \exists z(x \approx g(y, z)))[g(y, z)/x, f(x)/y, a/z] \\ &= \forall x(p(x, g(f(x), y))[g(y, z)/x, f(x)/y, a/z] \wedge \\ & \quad \exists z(x \approx g(y, z))[g(y, z)/x, f(x)/y, a/z] \\ &= \forall v((p(x, g(f(x), y))[v/x])[g(y, z)/x, f(x)/y, a/z]) \wedge \\ & \quad \exists u(((x \approx g(y, z))[u/z])[g(y, z)/x, f(x)/y, a/z]) \\ &= \forall v(p(v, g(f(v), f(x)))) \wedge \exists u((x \approx g(y, u))[g(y, z)/x, f(x)/y, a/z]) \\ &= \forall v(p(v, g(f(v), f(x)))) \wedge \exists u(g(y, z) \approx g(f(x), u)) \end{aligned}$$

# Bis jetzt

---

## Prädikatenlogik

- Syntax
  - Signatur
  - Terme
  - Formeln
  - Substitutionen

# Jetzt

---

## Prädikatenlogik

- Syntax
- Semantik

# Semantik

---

Semantik geben bedeutet für logische Systeme, einen Begriff von Wahrheit für Formeln zu definieren. Das hier für die Prädikatenlogik zu definierende Konzept geht auf Tarski zurück.

In der **klassischen Logik** (zurückgehend auf Aristoteles) gibt es „nur“ die zwei Wahrheitswerte „wahr“ und „falsch“, die wir mit 1 und 0 bezeichnen.

# Strukturen

---

## Definition.

Eine  $\Sigma$ -Struktur (bzw.  $\Sigma$ -Interpretation bzw.  $\Sigma$ -Modell) ist ein Tripel

$$\mathcal{A} = (U, (f_{\mathcal{A}} : U^n \rightarrow U)_{f/n \in \Omega}, (p_{\mathcal{A}} \subseteq U^m)_{p/m \in \Pi})$$

wobei  $U \neq \emptyset$  eine Menge, genannt **Universum** von  $\mathcal{A}$ .

Oft identifizieren wir  $U$  mit  $\mathcal{A}$ , wenn die Interpretation der Funktions- und Prädikatensymbole eindeutig aus dem Kontext hervorgeht.

Mit  $\Sigma$ -Str bezeichnen wir die Menge aller  $\Sigma$ -Strukturen.