# Formale Grundlagen der Informatik 1
# Kapitel 19

# Prädikatenlogik
# Syntax & Semantik

Frank Heitmann
heitmann@informatik.uni-hamburg.de

23. Juni 2015

# Motivation

Die Prädikatenlogik ist eine *Erweiterung der Aussagenlogik*. Sie hat eine größere Ausdrucksstärke und erlaub eine feinere Differenzierung. Ferner sind Beziehungen/Relationen ausdrückbar.

- Terme repräsentieren Objekte
- Prädikatensymbole repräsentieren Eigenschaften und Relationen
- Formeln repräsentieren Aussagen

# Motivation

Die Prädikatenlogik ist eine *Erweiterung der Aussagenlogik*. Sie hat eine größere Ausdrucksstärke und erlaub eine feinere Differenzierung. Ferner sind Beziehungen/Relationen ausdrückbar.

- Terme repräsentieren Objekte
- Prädikatensymbole repräsentieren Eigenschaften und Relationen
- Formeln repräsentieren Aussagen

# Motivation - Beispiel

Jeder Student ist jünger als (irgend-)ein Lehrer.

- $S(x)$ - $x$ ist Student
- $L(x)$ - $x$ ist Lehrer
- $J(x, y)$ - $x$ ist jünger als $y$

$$\forall x(S(x) \Rightarrow \exists y(L(y) \land J(x, y)))$$

# Motivation - Beispiel

Jeder Student ist jünger als (irgend-)ein Lehrer.

- $S(x)$ - $x$ ist Student
- $L(x)$ - $x$ ist Lehrer
- $J(x, y)$ - $x$ ist jünger als $y$

$$\forall x(S(x) \Rightarrow \exists y(L(y) \wedge J(x, y)))$$

# Motivation - Beispiel

Jeder Student ist jünger als (irgend-)ein Lehrer.

- $S(x)$ - $x$ ist Student
- $L(x)$ - $x$ ist Lehrer
- $J(x, y)$ - $x$ ist jünger als $y$

$$\forall x(S(x) \Rightarrow \exists y(L(y) \wedge J(x, y)))$$

# Motivation - Beispiel

$$F = \forall x (P(x) \Rightarrow \exists y Q(f(x), y))$$

$$G = P(x, a) \land \forall x \exists y P(x, y)$$

# Alphabet

## Definition (Alphabet)

Das **Alphabet** der Prädikatenlogik besteht aus

- einer Menge von **Variablen**: $x, y, z, x_1, x_2, \dots$
- einer Menge von **Konstanten**: $a, b, c, a_1, a_2, \dots$
- einer Menge von **Funktionssymbolen**: $f, g, h, f_1, f_2, \dots$
- einer Menge von **Prädikatensymbolen**: $P, Q, R, P_1, P_2, \dots$
- einer Menge von **Aussagenymbolen**: $A, B, C, A_1, A_2, \dots$
- den **Junktoren**: $\neg, \wedge, \vee, \Rightarrow, \Leftrightarrow$
- den **Quantoren**: $\forall, \exists$
- den **Hilfssymbolen**: $), ($ und $,$

Jedem Funktions- und Prädikatensymbol ist dabei eindeutig eine **Stelligkeit** $k \geq 0$ zugewiesen.

# Alphabet

## Definition (Alphabet)

Das **Alphabet** der Prädikatenlogik besteht aus

- einer Menge von **Variablen**: $x, y, z, x_1, x_2, \ldots$
- einer Menge von **Konstanten**: $a, b, c, a_1, a_2, \ldots$
- einer Menge von **Funktionssymbolen**: $f, g, h, f_1, f_2, \ldots$
- einer Menge von **Prädikatensymbolen**: $P, Q, R, P_1, P_2, \ldots$
- einer Menge von **Aussagenymbolen**: $A, B, C, A_1, A_2, \ldots$
- den **Junktoren**: $\neg, \wedge, \vee, \Rightarrow, \Leftrightarrow$
- den **Quantoren**: $\forall, \exists$
- den **Hilfssymbolen**: $), ($ und $,$

Jedem Funktions- und Prädikatensymbol ist dabei eindeutig eine **Stelligkeit** $k \geq 0$ zugewiesen.

# Alphabet

## Definition (Alphabet)

Das **Alphabet** der Prädikatenlogik besteht aus

- einer Menge von **Variablen**: $x, y, z, x_1, x_2, \ldots$
- einer Menge von **Konstanten**: $a, b, c, a_1, a_2, \ldots$
- einer Menge von **Funktionssymbolen**: $f, g, h, f_1, f_2, \ldots$
- einer Menge von **Prädikatensymbolen**: $P, Q, R, P_1, P_2, \ldots$
- einer Menge von **Aussagenymbolen**: $A, B, C, A_1, A_2, \ldots$
- den **Junktoren**: $\neg, \wedge, \vee, \Rightarrow, \Leftrightarrow$
- den **Quantoren**: $\forall, \exists$
- den **Hilfssymbolen**: $), ($ und ,

Jedem Funktions- und Prädikatensymbol ist dabei eindeutig eine **Stelligkeit** $k \geq 0$ zugewiesen.

# Alphabet

## Definition (Alphabet)

Das **Alphabet** der Prädikatenlogik besteht aus

- einer Menge von **Variablen**: $x, y, z, x_1, x_2, \ldots$
- einer Menge von **Konstanten**: $a, b, c, a_1, a_2, \ldots$
- einer Menge von **Funktionssymbolen**: $f, g, h, f_1, f_2, \ldots$
- einer Menge von **Prädikatensymbolen**: $P, Q, R, P_1, P_2, \ldots$
- einer Menge von **Aussagenymbolen**: $A, B, C, A_1, A_2, \ldots$
- den **Junktoren**: $\neg, \wedge, \vee, \Rightarrow, \Leftrightarrow$
- den **Quantoren**: $\forall, \exists$
- den **Hilfssymbolen**: ), ( und ,

Jedem Funktions- und Prädikatensymbol ist dabei eindeutig eine **Stelligkeit** $k \geq 0$ zugewiesen.

# Alphabet

## Definition (Alphabet)

Das **Alphabet** der Prädikatenlogik besteht aus

- einer Menge von **Variablen**: $x, y, z, x_1, x_2, \ldots$
- einer Menge von **Konstanten**: $a, b, c, a_1, a_2, \ldots$
- einer Menge von **Funktionssymbolen**: $f, g, h, f_1, f_2, \ldots$
- einer Menge von **Prädikatensymbolen**: $P, Q, R, P_1, P_2, \ldots$
- einer Menge von **Aussagenymbolen**: $A, B, C, A_1, A_2, \ldots$
- den **Junktoren**: $\neg, \wedge, \vee, \Rightarrow, \Leftrightarrow$
- den **Quantoren**: $\forall, \exists$
- den **Hilfssymbolen**: ), ( und ,

Jedem Funktions- und Prädikatensymbol ist dabei eindeutig eine **Stelligkeit** $k \geq 0$ zugewiesen.

# Alphabet

## Definition (Alphabet)

Das **Alphabet** der Prädikatenlogik besteht aus

- einer Menge von **Variablen**: $x, y, z, x_1, x_2, \ldots$
- einer Menge von **Konstanten**: $a, b, c, a_1, a_2, \ldots$
- einer Menge von **Funktionssymbolen**: $f, g, h, f_1, f_2, \ldots$
- einer Menge von **Prädikatensymbolen**: $P, Q, R, P_1, P_2, \ldots$
- einer Menge von **Aussagenymbolen**: $A, B, C, A_1, A_2, \ldots$
- den **Junktoren**: $\neg, \wedge, \vee, \Rightarrow, \Leftrightarrow$
- den **Quantoren**: $\forall, \exists$
- den **Hilfssymbolen**: $), ($ und $,$

Jedem Funktions- und Prädikatensymbol ist dabei eindeutig eine **Stelligkeit** $k \geq 0$ zugewiesen.

# Alphabet

## Definition (Alphabet)

Das **Alphabet** der Prädikatenlogik besteht aus

- einer Menge von **Variablen**: $x, y, z, x_1, x_2, \ldots$
- einer Menge von **Konstanten**: $a, b, c, a_1, a_2, \ldots$
- einer Menge von **Funktionssymbolen**: $f, g, h, f_1, f_2, \ldots$
- einer Menge von **Prädikatensymbolen**: $P, Q, R, P_1, P_2, \ldots$
- einer Menge von **Aussagenymbolen**: $A, B, C, A_1, A_2, \ldots$
- den **Junktoren**: $\neg, \wedge, \vee, \Rightarrow, \Leftrightarrow$
- den **Quantoren**: $\forall, \exists$
- den **Hilfssymbolen**: $), ($ und $,$

Jedem Funktions- und Prädikatensymbol ist dabei eindeutig eine **Stelligkeit** $k \geq 0$ zugewiesen.

# Alphabet

## Definition (Alphabet)

Das **Alphabet** der Prädikatenlogik besteht aus

- einer Menge von **Variablen**: $x, y, z, x_1, x_2, \ldots$
- einer Menge von **Konstanten**: $a, b, c, a_1, a_2, \ldots$
- einer Menge von **Funktionssymbolen**: $f, g, h, f_1, f_2, \ldots$
- einer Menge von **Prädikatensymbolen**: $P, Q, R, P_1, P_2, \ldots$
- einer Menge von **Aussagenymbolen**: $A, B, C, A_1, A_2, \ldots$
- den **Junktoren**: $\neg, \wedge, \vee, \Rightarrow, \Leftrightarrow$
- den **Quantoren**: $\forall, \exists$
- den **Hilfssymbolen**: ), ( und ,

Jedem Funktions- und Prädikatensymbol ist dabei eindeutig eine **Stelligkeit** $k \geq 0$ zugewiesen.

# Alphabet

## Definition (Alphabet)

Das **Alphabet** der Prädikatenlogik besteht aus

- einer Menge von **Variablen**: $x, y, z, x_1, x_2, \ldots$
- einer Menge von **Konstanten**: $a, b, c, a_1, a_2, \ldots$
- einer Menge von **Funktionssymbolen**: $f, g, h, f_1, f_2, \ldots$
- einer Menge von **Prädikatensymbolen**: $P, Q, R, P_1, P_2, \ldots$
- einer Menge von **Aussagenymbolen**: $A, B, C, A_1, A_2, \ldots$
- den **Junktoren**: $\neg, \wedge, \vee, \Rightarrow, \Leftrightarrow$
- den **Quantoren**: $\forall, \exists$
- den **Hilfssymbolen**: $), ($ und ,

Jedem Funktions- und Prädikatensymbol ist dabei eindeutig eine **Stelligkeit** $k \geq 0$ zugewiesen.

# Alphabet

## Definition (Alphabet)

Das **Alphabet** der Prädikatenlogik besteht aus

- einer Menge von **Variablen**: $x, y, z, x_1, x_2, \dots$
- einer Menge von **Konstanten**: $a, b, c, a_1, a_2, \dots$
- einer Menge von **Funktionssymbolen**: $f, g, h, f_1, f_2, \dots$
- einer Menge von **Prädikatensymbolen**: $P, Q, R, P_1, P_2, \dots$
- einer Menge von **Aussagenymbolen**: $A, B, C, A_1, A_2, \dots$
- den **Junktoren**: $\neg, \wedge, \vee, \Rightarrow, \Leftrightarrow$
- den **Quantoren**: $\forall, \exists$
- den **Hilfssymbolen**: $), ($ und ,

Jedem Funktions- und Prädikatensymbol ist dabei eindeutig eine **Stelligkeit** $k \geq 0$ zugewiesen.

# Terme

## Definition (Terme der Prädikatenlogik)

Die Menge der **prädikatenlogischen Terme** wird wie folgt definiert:

1. Jede Variable ist ein Term.
2. Jede Konstante ist ein Term.
3. Ist $f$ ein Funktionssymbol der Stelligkeit $k$ und sind $t_1, \ldots, t_k$ Terme, so ist auch $f(t_1, \ldots, t_k)$ ein Term.
4. Es gibt keine anderen Terme, als die, die durch *endliche Anwendungen* der Schritte 1-3 erzeugt werden.

Der Term $t$ ist **Teilterm** des Terms $t'$, wenn $t$ beim Aufbau von $t'$ verwendet wurde.

# Terme

## Definition (Terme der Prädikatenlogik)

Die Menge der **prädikatenlogischen Terme** wird wie folgt definiert:

1. Jede Variable ist ein Term.
2. Jede Konstante ist ein Term.
3. Ist $f$ ein Funktionssymbol der Stelligkeit $k$ und sind $t_1, \dots, t_k$ Terme, so ist auch $f(t_1, \dots, t_k)$ ein Term.
4. Es gibt keine anderen Terme, als die, die durch *endliche Anwendungen* der Schritte 1-3 erzeugt werden.

Der Term $t$ ist **Teilterm** des Terms $t'$, wenn $t$ beim Aufbau von $t'$ verwendet wurde.

# Terme

## Definition (Terme der Prädikatenlogik)

Die Menge der **prädikatenlogischen Terme** wird wie folgt definiert:

1. Jede Variable ist ein Term.
2. Jede Konstante ist ein Term.
3. Ist $f$ ein Funktionssymbol der Stelligkeit $k$ und sind $t_1, \ldots, t_k$ Terme, so ist auch $f(t_1, \ldots, t_k)$ ein Term.
4. Es gibt keine anderen Terme, als die, die durch *endliche Anwendungen* der Schritte 1-3 erzeugt werden.

Der Term $t$ ist **Teilterm** des Terms $t'$, wenn $t$ beim Aufbau von $t'$ verwendet wurde.

# Terme

## Definition (Terme der Prädikatenlogik)

Die Menge der **prädikatenlogischen Terme** wird wie folgt definiert:

1. Jede Variable ist ein Term.
2. Jede Konstante ist ein Term.
3. Ist $f$ ein Funktionssymbol der Stelligkeit $k$ und sind $t_1, \ldots, t_k$ Terme, so ist auch $f(t_1, \ldots, t_k)$ ein Term.
4. Es gibt keine anderen Terme, als die, die durch *endliche Anwendungen* der Schritte 1-3 erzeugt werden.

Der Term $t$ ist **Teilterm** des Terms $t'$, wenn $t$ beim Aufbau von $t'$ verwendet wurde.

# Terme

## Definition (Terme der Prädikatenlogik)

Die Menge der **prädikatenlogischen Terme** wird wie folgt definiert:

1. Jede Variable ist ein Term.
2. Jede Konstante ist ein Term.
3. Ist $f$ ein Funktionssymbol der Stelligkeit $k$ und sind $t_1, \ldots, t_k$ Terme, so ist auch $f(t_1, \ldots, t_k)$ ein Term.
4. Es gibt keine anderen Terme, als die, die durch *endliche Anwendungen* der Schritte 1-3 erzeugt werden.

Der Term $t$ ist **Teilterm** des Terms $t'$, wenn $t$ beim Aufbau von $t'$ verwendet wurde.

# Terme

## Definition (Terme der Prädikatenlogik)

Die Menge der **prädikatenlogischen Terme** wird wie folgt definiert:

1. Jede Variable ist ein Term.
2. Jede Konstante ist ein Term.
3. Ist $f$ ein Funktionssymbol der Stelligkeit $k$ und sind $t_1, \ldots, t_k$ Terme, so ist auch $f(t_1, \ldots, t_k)$ ein Term.
4. Es gibt keine anderen Terme, als die, die durch *endliche Anwendungen* der Schritte 1-3 erzeugt werden.

Der Term $t$ ist **Teilterm** des Terms $t'$, wenn $t$ beim Aufbau von $t'$ verwendet wurde.

# Formeln

## Definition (Formeln der Prädikatenlogik)

Die Menge der **prädikatenlogischen Formeln** $\mathcal{L}_{PL}$ wird wie folgt definiert:

1. Jedes Aussagensymbol ist eine (atomare) Formel.
2. Ist $P$ ein Prädikatensymbol der Stelligkeit $k$ und sind $t_1, \ldots, t_k$ Terme, so ist $P(t_1, \ldots, t_k)$ eine (atomare) Formel.
3. Sind $F$ und $G$ Formeln, so sind auch $\neg F, (F \vee G), (F \wedge G), (F \Rightarrow G)$ und $(F \Leftrightarrow G)$ Formeln.
4. Ist $x$ eine Variable und $F$ eine Formel, so sind auch $\exists x F$ und $\forall x F$ Formeln.
5. Es gibt keine anderen Formeln, als die, die durch *endliche Anwendungen* der Schritte 1-4 erzeugt werden.

# Formeln

## Definition (Formeln der Prädikatenlogik)

Die Menge der **prädikatenlogischen Formeln** $\mathcal{L}_{PL}$ wird wie folgt definiert:

1. Jedes Aussagensymbol ist eine (atomare) Formel.
2. Ist $P$ ein Prädikatensymbol der Stelligkeit $k$ und sind $t_1, \ldots, t_k$ Terme, so ist $P(t_1, \ldots, t_k)$ eine (atomare) Formel.
3. Sind $F$ und $G$ Formeln, so sind auch $\neg F, (F \vee G), (F \wedge G), (F \Rightarrow G)$ und $(F \Leftrightarrow G)$ Formeln.
4. Ist $x$ eine Variable und $F$ eine Formel, so sind auch $\exists x F$ und $\forall x F$ Formeln.
5. Es gibt keine anderen Formeln, als die, die durch *endliche Anwendungen* der Schritte 1-4 erzeugt werden.

# Formeln

## Definition (Formeln der Prädikatenlogik)

Die Menge der **prädikatenlogischen Formeln** $\mathcal{L}_{PL}$ wird wie folgt definiert:

1. Jedes Aussagensymbol ist eine (atomare) Formel.
2. Ist $P$ ein Prädikatensymbol der Stelligkeit $k$ und sind $t_1, \ldots, t_k$ Terme, so ist $P(t_1, \ldots, t_k)$ eine (atomare) Formel.
3. Sind $F$ und $G$ Formeln, so sind auch $\neg F, (F \vee G), (F \wedge G), (F \Rightarrow G)$ und $(F \Leftrightarrow G)$ Formeln.
4. Ist $x$ eine Variable und $F$ eine Formel, so sind auch $\exists x F$ und $\forall x F$ Formeln.
5. Es gibt keine anderen Formeln, als die, die durch *endliche Anwendungen* der Schritte 1-4 erzeugt werden.

# Formeln

## Definition (Formeln der Prädikatenlogik)

Die Menge der **prädikatenlogischen Formeln** $\mathcal{L}_{PL}$ wird wie folgt definiert:

1. Jedes Aussagensymbol ist eine (atomare) Formel.
2. Ist $P$ ein Prädikatensymbol der Stelligkeit $k$ und sind $t_1, \dots, t_k$ Terme, so ist $P(t_1, \dots, t_k)$ eine (atomare) Formel.
3. Sind $F$ und $G$ Formeln, so sind auch $\neg F, (F \vee G), (F \wedge G), (F \Rightarrow G)$ und $(F \Leftrightarrow G)$ Formeln.
4. Ist $x$ eine Variable und $F$ eine Formel, so sind auch $\exists x F$ und $\forall x F$ Formeln.
5. Es gibt keine anderen Formeln, als die, die durch *endliche Anwendungen* der Schritte 1-4 erzeugt werden.

# Formeln

## Definition (Formeln der Prädikatenlogik)

Die Menge der **prädikatenlogischen Formeln** $\mathcal{L}_{PL}$ wird wie folgt definiert:

1. Jedes Aussagensymbol ist eine (atomare) Formel.
2. Ist $P$ ein Prädikatensymbol der Stelligkeit $k$ und sind $t_1, \ldots, t_k$ Terme, so ist $P(t_1, \ldots, t_k)$ eine (atomare) Formel.
3. Sind $F$ und $G$ Formeln, so sind auch $\neg F, (F \vee G), (F \wedge G), (F \Rightarrow G)$ und $(F \Leftrightarrow G)$ Formeln.
4. Ist $x$ eine Variable und $F$ eine Formel, so sind auch $\exists x F$ und $\forall x F$ Formeln.
5. Es gibt keine anderen Formeln, als die, die durch *endliche Anwendungen* der Schritte 1-4 erzeugt werden.

# Formeln

## Definition (Formeln der Prädikatenlogik)

Die Menge der **prädikatenlogischen Formeln** $\mathcal{L}_{PL}$ wird wie folgt definiert:

1. Jedes Aussagensymbol ist eine (atomare) Formel.
2. Ist $P$ ein Prädikatensymbol der Stelligkeit $k$ und sind $t_1, \ldots, t_k$ Terme, so ist $P(t_1, \ldots, t_k)$ eine (atomare) Formel.
3. Sind $F$ und $G$ Formeln, so sind auch $\neg F, (F \vee G), (F \wedge G), (F \Rightarrow G)$ und $(F \Leftrightarrow G)$ Formeln.
4. Ist $x$ eine Variable und $F$ eine Formel, so sind auch $\exists x F$ und $\forall x F$ Formeln.
5. Es gibt keine anderen Formeln, als die, die durch *endliche Anwendungen* der Schritte 1-4 erzeugt werden.

# Teilformeln

## Definition (Formeln der Prädikatenlogik (Fortsetzung))

- Eine Formel $F$ ist **Teilformel** einer Formel $H$, wenn $F$ beim Aufbau von $H$ verwendet wurde.
- Ein Term $t$ ist **Teilterm** einer Formel $H$, wenn $t$ beim Aufbau von $H$ verwendet wird oder Teilterm eines Terms ist für den dies gilt.
- In $\exists xF$ und $\forall xF$ wird $x$ als **Quantorenvariable** bezeichnet und $F$ als **Skopus** des Quantors.

# Teilformeln

**Definition (Formeln der Prädikatenlogik (Fortsetzung))**

- Eine Formel $F$ ist **Teilformel** einer Formel $H$, wenn $F$ beim Aufbau von $H$ verwendet wurde.
- Ein Term $t$ ist **Teilterm** einer Formel $H$, wenn $t$ beim Aufbau von $H$ verwendet wird oder Teilterm eines Terms ist für den dies gilt.
- In $\exists xF$ und $\forall xF$ wird $x$ als **Quantorenvariable** bezeichnet und $F$ als **Skopus** des Quantors.

# Teilformeln

**Definition (Formeln der Prädikatenlogik (Fortsetzung))**

- Eine Formel $F$ ist **Teilformel** einer Formel $H$, wenn $F$ beim Aufbau von $H$ verwendet wurde.
- Ein Term $t$ ist **Teilterm** einer Formel $H$, wenn $t$ beim Aufbau von $H$ verwendet wird oder Teilterm eines Terms ist für den dies gilt.
- In $\exists xF$ und $\forall xF$ wird $x$ als **Quantorenvariable** bezeichnet und $F$ als **Skopus** des Quantors.

# Freie und gebundene Variablen

## Definition (Freie und gebundene Variablen)

- In $\exists x F$ und $\forall x F$ wird $x$ als **Quantorenvariable** bezeichnet und $F$ als **Skopus** des Quantors.
- Eine Variable $x$, die im Skopus $F$ eines Quantors mit der Quantorenvariablen $x$ frei vorkommt, ist durch den Quantor in dieser Position **gebunden**.
- Eine Variable, die als Teilterm einer Formel $F$ auftritt und an keinen Quantor gebunden ist, ist in dieser Position **frei**.
- Eine Formel ohne freie Variablen heißt **geschlossen**.

## Beispiel

In $\forall x(P(x) \vee R(x, y)) \Rightarrow Q(x)$ sind die ersten zwei $x$ gebunden, das $y$ und das dritte $x$ sind frei.

# Freie und gebundene Variablen

## Definition (Freie und gebundene Variablen)

- In $\exists x F$ und $\forall x F$ wird $x$ als **Quantorenvariable** bezeichnet und $F$ als **Skopus** des Quantors.
- Eine Variable $x$, die im Skopus $F$ eines Quantors mit der Quantorenvariablen $x$ frei vorkommt, ist durch den Quantor in dieser Position **gebunden**.
- Eine Variable, die als Teilterm einer Formel $F$ auftritt und an keinen Quantor gebunden ist, ist in dieser Position **frei**.
- Eine Formel ohne freie Variablen heißt **geschlossen**.

## Beispiel

In $\forall x(P(x) \vee R(x, y)) \Rightarrow Q(x)$ sind die ersten zwei $x$ gebunden, das $y$ und das dritte $x$ sind frei.

# Freie und gebundene Variablen

## Definition (Freie und gebundene Variablen)

- In $\exists xF$ und $\forall xF$ wird $x$ als **Quantorenvariable** bezeichnet und $F$ als **Skopus** des Quantors.
- Eine Variable $x$, die im Skopus $F$ eines Quantors mit der Quantorenvariablen $x$ frei vorkommt, ist durch den Quantor in dieser Position **gebunden**.
- Eine Variable, die als Teilterm einer Formel $F$ auftritt und an keinen Quantor gebunden ist, ist in dieser Position **frei**.
- Eine Formel ohne freie Variablen heißt **geschlossen**.

## Beispiel

In $\forall x(P(x) \vee R(x, y)) \Rightarrow Q(x)$ sind die ersten zwei $x$ gebunden, das $y$ und das dritte $x$ sind frei.

# Freie und gebundene Variablen

## Definition (Freie und gebundene Variablen)

- In $\exists x F$ und $\forall x F$ wird $x$ als **Quantorenvariable** bezeichnet und $F$ als **Skopus** des Quantors.
- Eine Variable $x$, die im Skopus $F$ eines Quantors mit der Quantorenvariablen $x$ frei vorkommt, ist durch den Quantor in dieser Position **gebunden**.
- Eine Variable, die als Teilterm einer Formel $F$ auftritt und an keinen Quantor gebunden ist, ist in dieser Position **frei**.
- Eine Formel ohne freie Variablen heißt **geschlossen**.

## Beispiel

In $\forall x(P(x) \vee R(x, y)) \Rightarrow Q(x)$ sind die ersten zwei $x$ gebunden, das $y$ und das dritte $x$ sind frei.

# Freie und gebundene Variablen

## Definition (Freie und gebundene Variablen)

- In $\exists xF$ und $\forall xF$ wird $x$ als **Quantorenvariable** bezeichnet und $F$ als **Skopus** des Quantors.
- Eine Variable $x$, die im Skopus $F$ eines Quantors mit der Quantorenvariablen $x$ frei vorkommt, ist durch den Quantor in dieser Position **gebunden**.
- Eine Variable, die als Teilterm einer Formel $F$ auftritt und an keinen Quantor gebunden ist, ist in dieser Position **frei**.
- Eine Formel ohne freie Variablen heißt **geschlossen**.

## Beispiel

In $\forall x(P(x) \vee R(x, y)) \Rightarrow Q(x)$ sind die ersten zwei $x$ gebunden, das $y$ und das dritte $x$ sind frei.

# Induktion und Rekursion

Entsprechend den Definitionen lassen sich wieder über die Struktur der Terme und der Formeln

- induktive Beweis führen und
- Funktionen rekursiv definieren.

**Anmerkung**

Manchmal muss eine Aussage zunächst für die Terme, dann für die Formeln gezeigt werden!

# Informale Semantik

Die Idee bei der Semantik ist, die benutzten Symbole durch Elemente einer Grundmenge (z.B. Zahlen) zu interpretieren:

- freie Variablen und Konstanten durch Objekte der Grundmenge
- Funktionssymbole durch Funktionen über der Grundmenge
- Prädikatensymble durch Mengen über der Grundmenge
- Aussagensymbole durch Wahrheitswerte
- komplexe Ausdrücke auf obigem aufbauend ...

Wahrheitswerte ergeben sich dann durch diese Interpretation ...

# Informale Semantik

Die Idee bei der Semantik ist, die benutzten Symbole durch Elemente einer Grundmenge (z.B. Zahlen) zu interpretieren:

- freie Variablen und Konstanten durch Objekte der Grundmenge
- Funktionssymbole durch Funktionen über der Grundmenge
- Prädikatensymble durch Mengen über der Grundmenge
- Aussagensymbole durch Wahrheitswerte
- komplexe Ausdrücke auf obigem aufbauend ...

Wahrheitswerte ergeben sich dann durch diese Interpretation ...

# Informale Semantik

Die Idee bei der Semantik ist, die benutzten Symbole durch Elemente einer Grundmenge (z.B. Zahlen) zu interpretieren:

- freie Variablen und Konstanten durch Objekte der Grundmenge
- Funktionssymbole durch Funktionen über der Grundmenge
- Prädikatensymble durch Mengen über der Grundmenge
- Aussagensymbole durch Wahrheitswerte
- komplexe Ausdrücke auf obigem aufbauend ...

Wahrheitswerte ergeben sich dann durch diese Interpretation ...

# Informale Semantik

Die Idee bei der Semantik ist, die benutzten Symbole durch Elemente einer Grundmenge (z.B. Zahlen) zu interpretieren:

- freie Variablen und Konstanten durch Objekte der Grundmenge
- Funktionssymbole durch Funktionen über der Grundmenge
- Prädikatensymble durch Mengen über der Grundmenge
- Aussagensymbole durch Wahrheitswerte
- komplexe Ausdrücke auf obigem aufbauend ...

Wahrheitswerte ergeben sich dann durch diese Interpretation ...

# Informale Semantik

Die Idee bei der Semantik ist, die benutzten Symbole durch Elemente einer Grundmenge (z.B. Zahlen) zu interpretieren:

- freie Variablen und Konstanten durch Objekte der Grundmenge
- Funktionssymbole durch Funktionen über der Grundmenge
- Prädikatensymble durch Mengen über der Grundmenge
- Aussagensymbole durch Wahrheitswerte
- komplexe Ausdrücke auf obigem aufbauend ...

Wahrheitswerte ergeben sich dann durch diese Interpretation ...

# Informale Semantik

Die Idee bei der Semantik ist, die benutzten Symbole durch Elemente einer Grundmenge (z.B. Zahlen) zu interpretieren:

- freie Variablen und Konstanten durch Objekte der Grundmenge
- Funktionssymbole durch Funktionen über der Grundmenge
- Prädikatensymble durch Mengen über der Grundmenge
- Aussagensymbole durch Wahrheitswerte
- komplexe Ausdrücke auf obigem aufbauend ...

Wahrheitswerte ergeben sich dann durch diese Interpretation ...

# Informale Semantik

Die Idee bei der Semantik ist, die benutzten Symbole durch Elemente einer Grundmenge (z.B. Zahlen) zu interpretieren:

- freie Variablen und Konstanten durch Objekte der Grundmenge
- Funktionssymbole durch Funktionen über der Grundmenge
- Prädikatensymble durch Mengen über der Grundmenge
- Aussagensymbole durch Wahrheitswerte
- komplexe Ausdrücke auf obigem aufbauend ...

Wahrheitswerte ergeben sich dann durch diese Interpretation ...

# Strukturen

## Definition (Struktur)

Eine **Struktur** ist ein Tupel $\mathcal{A} = (U, I)$ mit

- dem Universum $U$ (auch Grundmenge oder Domäne) eine beliebige nicht-leere Menge und
- der Interpretation $I$, die eine Abbildung ist und
  - Variablen und Konstanten auf Elemente auf $U$ abbildet,
  - $k$-stellige Funktionssymbole auf $k$-stellige Funktionen über $U$
  - $k$-stellige Prädikatensymbole auf Mengen von $k$-Tupeln über $U$ und
  - Aussagensymbole auf Wahrheitswerte.

# Strukturen

## Definition (Struktur)

Eine **Struktur** ist ein Tupel $\mathcal{A} = (U, I)$ mit

- dem Universum $U$ (auch Grundmenge oder Domäne) eine beliebige nicht-leere Menge und
- der Interpretation $I$, die eine Abbildung ist und
  - Variablen und Konstanten auf Elemente auf $U$ abbildet,
  - $k$-stellige Funktionssymbole auf $k$-stellige Funktionen über $U$
  - $k$-stellige Prädikatensymbole auf Mengen von $k$-Tupeln über $U$ und
  - Aussagensymbole auf Wahrheitswerte.

# Strukturen

## Definition (Struktur)

Eine **Struktur** ist ein Tupel $\mathcal{A} = (U, I)$ mit

- dem Universum $U$ (auch Grundmenge oder Domäne) eine beliebige nicht-leere Menge und
- der Interpretation $I$, die eine Abbildung ist und
  - Variablen und Konstanten auf Elemente auf $U$ abbildet,
  - $k$-stellige Funktionssymbole auf $k$-stellige Funktionen über $U$
  - $k$-stellige Prädikatensymbole auf Mengen von $k$-Tupeln über $U$ und
  - Aussagensymbole auf Wahrheitswerte.

# Strukturen

## Definition (Struktur)

Eine **Struktur** ist ein Tupel $\mathcal{A} = (U, I)$ mit

- dem Universum $U$ (auch Grundmenge oder Domäne) eine beliebige nicht-leere Menge und
- der Interpretation $I$, die eine Abbildung ist und
  - Variablen und Konstanten auf Elemente auf $U$ abbildet,
  - $k$-stellige Funktionssymbole auf $k$-stellige Funktionen über $U$
  - $k$-stellige Prädikatensymbole auf Mengen von $k$-Tupeln über $U$ und
  - Aussagensymbole auf Wahrheitswerte.

# Strukturen

## Definition (Struktur)

Eine **Struktur** ist ein Tupel $\mathcal{A} = (U, I)$ mit

- dem Universum $U$ (auch Grundmenge oder Domäne) eine beliebige nicht-leere Menge und
- der Interpretation $I$, die eine Abbildung ist und
  - Variablen und Konstanten auf Elemente auf $U$ abbildet,
  - $k$-stellige Funktionssymbole auf $k$-stellige Funktionen über $U$
  - $k$-stellige Prädikatensymbole auf Mengen von $k$-Tupeln über $U$ und
  - Aussagensymbole auf Wahrheitswerte.

# Strukturen

## Definition (Struktur)

Eine **Struktur** ist ein Tupel $\mathcal{A} = (U, I)$ mit

- dem Universum $U$ (auch Grundmenge oder Domäne) eine beliebige nicht-leere Menge und
- der Interpretation $I$, die eine Abbildung ist und
  - Variablen und Konstanten auf Elemente auf $U$ abbildet,
  - $k$-stellige Funktionssymbole auf $k$-stellige Funktionen über $U$
  - $k$-stellige Prädikatensymbole auf Mengen von $k$-Tupeln über $U$ und
  - Aussagensymbole auf Wahrheitswerte.

# Strukturen

## Definition (Struktur)

Eine **Struktur** ist ein Tupel $\mathcal{A} = (U, I)$ mit

- dem Universum $U$ (auch Grundmenge oder Domäne) eine beliebige nicht-leere Menge und
- der Interpretation $I$, die eine Abbildung ist und
  - Variablen und Konstanten auf Elemente auf $U$ abbildet,
  - $k$-stellige Funktionssymbole auf $k$-stellige Funktionen über $U$
  - $k$-stellige Prädikatensymbole auf Mengen von $k$-Tupeln über $U$ und
  - Aussagensymbole auf Wahrheitswerte.

# Struktur - Beispiel

## Beispiel

Haben wir z.B.

$$\forall x P(g(x)) \vee Q(f(x,y),x)$$

So könnten wir z.B. nehmen:

- $U = \mathbb{N}$
- $I(g) = g'$ wobei $g' : \mathbb{N} \to \mathbb{N}$ mit $g'(n) = n + 1$
- $I(f) = f'$ wobei $f' : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ mit $f'(n,m) = n + m$
- $I(P) = \{2, 4, 6, 8, \ldots\} \subseteq U$
- $I(Q) = \{(x,y) \mid x < y\} \subseteq U \times U$
- $I(y) = 5 \in U$
- $I(x) = 3 \in U$

# Struktur - Beispiel

## Beispiel

Haben wir z.B.

$$\forall x P(g(x)) \vee Q(f(x,y),x)$$

So könnten wir z.B. nehmen:

- $U = \mathbb{N}$
- $I(g) = g'$ wobei $g' : \mathbb{N} \to \mathbb{N}$ mit $g'(n) = n + 1$
- $I(f) = f'$ wobei $f' : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ mit $f'(n,m) = n + m$
- $I(P) = \{2, 4, 6, 8, \ldots\} \subseteq U$
- $I(Q) = \{(x,y) \mid x < y\} \subseteq U \times U$
- $I(y) = 5 \in U$
- $I(x) = 3 \in U$

# Struktur - Beispiel

## Beispiel

Haben wir z.B.

$$\forall x P(g(x)) \vee Q(f(x,y),x)$$

So könnten wir z.B. nehmen:

- $U = \mathbb{N}$
- $I(g) = g'$ wobei $g' : \mathbb{N} \to \mathbb{N}$ mit $g'(n) = n + 1$
- $I(f) = f'$ wobei $f' : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ mit $f'(n,m) = n + m$
- $I(P) = \{2, 4, 6, 8, \ldots\} \subseteq U$
- $I(Q) = \{(x,y) \mid x < y\} \subseteq U \times U$
- $I(y) = 5 \in U$
- $I(x) = 3 \in U$

# Struktur - Beispiel

## Beispiel

Haben wir z.B.

$$\forall x P(g(x)) \vee Q(f(x,y),x)$$

So könnten wir z.B. nehmen:

- $U = \mathbb{N}$
- $I(g) = g'$ wobei $g' : \mathbb{N} \to \mathbb{N}$ mit $g'(n) = n + 1$
- $I(f) = f'$ wobei $f' : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ mit $f'(n,m) = n + m$
- $I(P) = \{2, 4, 6, 8, \dots\} \subseteq U$
- $I(Q) = \{(x,y) \mid x < y\} \subseteq U \times U$
- $I(y) = 5 \in U$
- $I(x) = 3 \in U$

# Struktur - Beispiel

## Beispiel

Haben wir z.B.

$$\forall x P(g(x)) \vee Q(f(x,y),x)$$

So könnten wir z.B. nehmen:

- $U = \mathbb{N}$
- $I(g) = g'$ wobei $g' : \mathbb{N} \to \mathbb{N}$ mit $g'(n) = n + 1$
- $I(f) = f'$ wobei $f' : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ mit $f'(n,m) = n + m$
- $I(P) = \{2, 4, 6, 8, \ldots\} \subseteq U$
- $I(Q) = \{(x,y) \mid x < y\} \subseteq U \times U$
- $I(y) = 5 \in U$
- $I(x) = 3 \in U$

# Struktur - Beispiel

## Beispiel

Haben wir z.B.

$$\forall x P(g(x)) \lor Q(f(x,y),x)$$

So könnten wir z.B. nehmen:

- $U = \mathbb{N}$
- $I(g) = g'$ wobei $g' : \mathbb{N} \to \mathbb{N}$ mit $g'(n) = n + 1$
- $I(f) = f'$ wobei $f' : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ mit $f'(n,m) = n + m$
- $I(P) = \{2, 4, 6, 8, \ldots\} \subseteq U$
- $I(Q) = \{(x,y) \mid x < y\} \subseteq U \times U$
- $I(y) = 5 \in U$
- $I(x) = 3 \in U$

# Struktur - Beispiel

## Beispiel

Haben wir z.B.

$$\forall x P(g(x)) \vee Q(f(x,y),x)$$

So könnten wir z.B. nehmen:

- $U = \mathbb{N}$
- $I(g) = g'$ wobei $g' : \mathbb{N} \to \mathbb{N}$ mit $g'(n) = n + 1$
- $I(f) = f'$ wobei $f' : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ mit $f'(n,m) = n + m$
- $I(P) = \{2, 4, 6, 8, \dots\} \subseteq U$
- $I(Q) = \{(x,y) \mid x < y\} \subseteq U \times U$
- $I(y) = 5 \in U$
- $I(x) = 3 \in U$

# Struktur - Beispiel

## Beispiel

Haben wir z.B.

$$\forall x P(g(x)) \lor Q(f(x,y),x)$$

So könnten wir z.B. nehmen:

- $U = \mathbb{N}$
- $I(g) = g'$ wobei $g' : \mathbb{N} \to \mathbb{N}$ mit $g'(n) = n + 1$
- $I(f) = f'$ wobei $f' : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ mit $f'(n,m) = n + m$
- $I(P) = \{2,4,6,8,\dots\} \subseteq U$
- $I(Q) = \{(x,y) \mid x < y\} \subseteq U \times U$
- $I(y) = 5 \in U$
- $I(x) = 3 \in U$

# Auswertung

## Definition (Auswertung von Termen und atomaren Formeln)

$I$ wird rekursiv zu einer **Auswertung** fortgesetzt (für die oft auch einfach $\mathcal{A}$ geschrieben wird).

1. Rekursive Definition von $\mathcal{A}$ für Terme:
   - Für jede Variable $x$ ist $\mathcal{A}(x) = I(x)$
   - Für jede Konstante $a$ ist $\mathcal{A}(a) = I(a)$
   - Für $k$ Terme $t_1, \ldots, t_k$ und ein $k$-stelliges Funktionssymbol $f$ ist $\mathcal{A}(f(t_1, \ldots, t_k)) = I(f)(\mathcal{A}(t_1), \ldots, \mathcal{A}(t_k))$
2. Rekursive Definition von $\mathcal{A}$ für atomare Formeln:
   - Für jedes Aussagensymbol $B$ ist $\mathcal{A}(B) = I(B)$
   - Für $k$ Terme $t_1, \ldots, t_k$ und ein $k$-stelliges Prädikatensymbol $P$ ist

$$\mathcal{A}(P(t_1, \ldots, t_k)) = \begin{cases} 1 & \text{, falls } (\mathcal{A}(t_1), \ldots, \mathcal{A}(t_k)) \in I(P) \\ 0 & \text{, sonst} \end{cases}$$

# Auswertung

## Definition (Auswertung von Termen und atomaren Formeln)

$I$ wird rekursiv zu einer **Auswertung** fortgesetzt (für die oft auch einfach $\mathcal{A}$ geschrieben wird).

1. Rekursive Definition von $\mathcal{A}$ für Terme:
   - Für jede Variable $x$ ist $\mathcal{A}(x) = I(x)$
   - Für jede Konstante $a$ ist $\mathcal{A}(a) = I(a)$
   - Für $k$ Terme $t_1, \ldots, t_k$ und ein $k$-stelliges Funktionssymbol $f$ ist $\mathcal{A}(f(t_1, \ldots, t_k)) = I(f)(\mathcal{A}(t_1), \ldots, \mathcal{A}(t_k))$
2. Rekursive Definition von $\mathcal{A}$ für atomare Formeln:
   - Für jedes Aussagensymbol $B$ ist $\mathcal{A}(B) = I(B)$
   - Für $k$ Terme $t_1, \ldots, t_k$ und ein $k$-stelliges Prädikatensymbol $P$ ist

$$\mathcal{A}(P(t_1, \ldots, t_k)) = \begin{cases} 1 & \text{, falls } (\mathcal{A}(t_1), \ldots, \mathcal{A}(t_k)) \in I(P) \\ 0 & \text{, sonst} \end{cases}$$

# Auswertung

## Definition (Auswertung von Termen und atomaren Formeln)

$I$ wird rekursiv zu einer **Auswertung** fortgesetzt (für die oft auch einfach $\mathcal{A}$ geschrieben wird).

1. Rekursive Definition von $\mathcal{A}$ für Terme:
   - Für jede Variable $x$ ist $\mathcal{A}(x) = I(x)$
   - Für jede Konstante $a$ ist $\mathcal{A}(a) = I(a)$
   - Für $k$ Terme $t_1, \ldots, t_k$ und ein $k$-stelliges Funktionssymbol $f$ ist $\mathcal{A}(f(t_1, \ldots, t_k)) = I(f)(\mathcal{A}(t_1), \ldots, \mathcal{A}(t_k))$
2. Rekursive Definition von $\mathcal{A}$ für atomare Formeln:
   - Für jedes Aussagensymbol $B$ ist $\mathcal{A}(B) = I(B)$
   - Für $k$ Terme $t_1, \ldots, t_k$ und ein $k$-stelliges Prädikatensymbol $P$ ist

$$\mathcal{A}(P(t_1, \ldots, t_k)) = \begin{cases} 1 & \text{, falls } (\mathcal{A}(t_1), \ldots, \mathcal{A}(t_k)) \in I(P) \\ 0 & \text{, sonst} \end{cases}$$

# Auswertung

## Definition (Auswertung von Termen und atomaren Formeln)

$I$ wird rekursiv zu einer **Auswertung** fortgesetzt (für die oft auch einfach $\mathcal{A}$ geschrieben wird).

1. Rekursive Definition von $\mathcal{A}$ für Terme:
   - Für jede Variable $x$ ist $\mathcal{A}(x) = I(x)$
   - Für jede Konstante $a$ ist $\mathcal{A}(a) = I(a)$
   - Für $k$ Terme $t_1, \ldots, t_k$ und ein $k$-stelliges Funktionssymbol $f$ ist $\mathcal{A}(f(t_1, \ldots, t_k)) = I(f)(\mathcal{A}(t_1), \ldots, \mathcal{A}(t_k))$
2. Rekursive Definition von $\mathcal{A}$ für atomare Formeln:
   - Für jedes Aussagensymbol $B$ ist $\mathcal{A}(B) = I(B)$
   - Für $k$ Terme $t_1, \ldots, t_k$ und ein $k$-stelliges Prädikatensymbol $P$ ist

$$\mathcal{A}(P(t_1, \ldots, t_k)) = \begin{cases} 1 & \text{, falls } (\mathcal{A}(t_1), \ldots, \mathcal{A}(t_k)) \in I(P) \\ 0 & \text{, sonst} \end{cases}$$

# Auswertung

## Definition (Auswertung von Termen und atomaren Formeln)

$I$ wird rekursiv zu einer **Auswertung** fortgesetzt (für die oft auch einfach $\mathcal{A}$ geschrieben wird).

1. Rekursive Definition von $\mathcal{A}$ für Terme:
   - Für jede Variable $x$ ist $\mathcal{A}(x) = I(x)$
   - Für jede Konstante $a$ ist $\mathcal{A}(a) = I(a)$
   - Für $k$ Terme $t_1, \ldots, t_k$ und ein $k$-stelliges Funktionssymbol $f$ ist $\mathcal{A}(f(t_1, \ldots, t_k)) = I(f)(\mathcal{A}(t_1), \ldots, \mathcal{A}(t_k))$
2. Rekursive Definition von $\mathcal{A}$ für atomare Formeln:
   - Für jedes Aussagensymbol $B$ ist $\mathcal{A}(B) = I(B)$
   - Für $k$ Terme $t_1, \ldots, t_k$ und ein $k$-stelliges Prädikatensymbol $P$ ist

$$\mathcal{A}(P(t_1, \ldots, t_k)) = \begin{cases} 1 & \text{, falls } (\mathcal{A}(t_1), \ldots, \mathcal{A}(t_k)) \in I(P) \\ 0 & \text{, sonst} \end{cases}$$

# Auswertung

## Definition (Auswertung von Termen und atomaren Formeln)

$I$ wird rekursiv zu einer **Auswertung** fortgesetzt (für die oft auch einfach $\mathcal{A}$ geschrieben wird).

1. Rekursive Definition von $\mathcal{A}$ für Terme:
   - Für jede Variable $x$ ist $\mathcal{A}(x) = I(x)$
   - Für jede Konstante $a$ ist $\mathcal{A}(a) = I(a)$
   - Für $k$ Terme $t_1, \ldots, t_k$ und ein $k$-stelliges Funktionssymbol $f$ ist $\mathcal{A}(f(t_1, \ldots, t_k)) = I(f)(\mathcal{A}(t_1), \ldots, \mathcal{A}(t_k))$
2. Rekursive Definition von $\mathcal{A}$ für atomare Formeln:
   - Für jedes Aussagensymbol $B$ ist $\mathcal{A}(B) = I(B)$
   - Für $k$ Terme $t_1, \ldots, t_k$ und ein $k$-stelliges Prädikatensymbol $P$ ist

$$\mathcal{A}(P(t_1, \ldots, t_k)) = \begin{cases} 1 & \text{, falls } (\mathcal{A}(t_1), \ldots, \mathcal{A}(t_k)) \in I(P) \\ 0 & \text{, sonst} \end{cases}$$

# Auswertung

## Definition (Auswertung von Termen und atomaren Formeln)

$I$ wird rekursiv zu einer **Auswertung** fortgesetzt (für die oft auch einfach $\mathcal{A}$ geschrieben wird).

1. Rekursive Definition von $\mathcal{A}$ für Terme:
   - Für jede Variable $x$ ist $\mathcal{A}(x) = I(x)$
   - Für jede Konstante $a$ ist $\mathcal{A}(a) = I(a)$
   - Für $k$ Terme $t_1, \dots, t_k$ und ein $k$-stelliges Funktionssymbol $f$ ist $\mathcal{A}(f(t_1, \dots, t_k)) = I(f)(\mathcal{A}(t_1), \dots, \mathcal{A}(t_k))$
2. Rekursive Definition von $\mathcal{A}$ für atomare Formeln:
   - Für jedes Aussagensymbol $B$ ist $\mathcal{A}(B) = I(B)$
   - Für $k$ Terme $t_1, \dots, t_k$ und ein $k$-stelliges Prädikatensymbol $P$ ist

$$\mathcal{A}(P(t_1, \dots, t_k)) = \begin{cases} 1 & \text{, falls } (\mathcal{A}(t_1), \dots, \mathcal{A}(t_k)) \in I(P) \\ 0 & \text{, sonst} \end{cases}$$

# Auswertung

## Definition (Auswertung von Termen und atomaren Formeln)

$I$ wird rekursiv zu einer **Auswertung** fortgesetzt (für die oft auch einfach $\mathcal{A}$ geschrieben wird).

1. Rekursive Definition von $\mathcal{A}$ für Terme:
   - Für jede Variable $x$ ist $\mathcal{A}(x) = I(x)$
   - Für jede Konstante $a$ ist $\mathcal{A}(a) = I(a)$
   - Für $k$ Terme $t_1, \ldots, t_k$ und ein $k$-stelliges Funktionssymbol $f$ ist $\mathcal{A}(f(t_1, \ldots, t_k)) = I(f)(\mathcal{A}(t_1), \ldots, \mathcal{A}(t_k))$
2. Rekursive Definition von $\mathcal{A}$ für atomare Formeln:
   - Für jedes Aussagensymbol $B$ ist $\mathcal{A}(B) = I(B)$
   - Für $k$ Terme $t_1, \ldots, t_k$ und ein $k$-stelliges Prädikatensymbol $P$ ist

$$\mathcal{A}(P(t_1, \ldots, t_k)) = \begin{cases} 1 & \text{, falls } (\mathcal{A}(t_1), \ldots, \mathcal{A}(t_k)) \in I(P) \\ 0 & \text{, sonst} \end{cases}$$

# Auswertung - Beispiel

## Beispiel

Ist $F = P(a, f(a))$ und $\mathcal{A} = (U, I)$ mit $U = \mathbb{N}$ und $I$ mit $I(a) = 4$ $I(P) = \{(x, y) \mid x, y \text{ sind gerade}\}$ und $I(f) = f'$ wobei $f' : \mathbb{N} \to \mathbb{N}$ mit $f'(n) = 2 \cdot n$, dann ist:

$$\begin{array}{ll} & \mathcal{A}(P(a, f(a))) = 1 \\ gdw. & (\mathcal{A}(a), \mathcal{A}(f(a))) \in I(P) \\ gdw. & (4, I(f)(\mathcal{A}(a))) \in I(P) \\ gdw. & (4, f'(4)) \in I(P) \\ gdw. & (4, 8) \in I(P) \end{array}$$

Da $(4, 8) \in I(P)$ gilt, ist $\mathcal{A}$ eine erfüllende Struktur.

# Auswertung - Beispiel

## Beispiel

Ist $F = P(a, f(a))$ und $\mathcal{A} = (U, I)$ mit $U = \mathbb{N}$ und $I$ mit $I(a) = 4$ $I(P) = \{(x, y) \mid x, y \text{ sind gerade}\}$ und $I(f) = f'$ wobei $f' : \mathbb{N} \to \mathbb{N}$ mit $f'(n) = 2 \cdot n$, dann ist:

$$
\begin{aligned}
& \mathcal{A}(P(a, f(a))) = 1 \\
\textit{gdw.} \quad & (\mathcal{A}(a), \mathcal{A}(f(a))) \in I(P) \\
\textit{gdw.} \quad & (4, I(f)(\mathcal{A}(a))) \in I(P) \\
\textit{gdw.} \quad & (4, f'(4)) \in I(P) \\
\textit{gdw.} \quad & (4, 8) \in I(P)
\end{aligned}
$$

Da $(4, 8) \in I(P)$ gilt, ist $\mathcal{A}$ eine erfüllende Struktur.

# Auswertung - Beispiel

## Beispiel

Ist $F = P(a, f(a))$ und $\mathcal{A} = (U, I)$ mit $U = \mathbb{N}$ und $I$ mit $I(a) = 4$ $I(P) = \{(x, y) \mid x, y \text{ sind gerade}\}$ und $I(f) = f'$ wobei $f' : \mathbb{N} \to \mathbb{N}$ mit $f'(n) = 2 \cdot n$, dann ist:

$$\begin{aligned}
& \mathcal{A}(P(a, f(a))) = 1 \\
gdw. \quad & (\mathcal{A}(a), \mathcal{A}(f(a))) \in I(P) \\
gdw. \quad & (4, I(f)(\mathcal{A}(a))) \in I(P) \\
gdw. \quad & (4, f'(4)) \in I(P) \\
gdw. \quad & (4, 8) \in I(P)
\end{aligned}$$

Da $(4, 8) \in I(P)$ gilt, ist $\mathcal{A}$ eine erfüllende Struktur.

# Auswertung - Beispiel

## Beispiel

Ist $F = P(a, f(a))$ und $\mathcal{A} = (U, I)$ mit $U = \mathbb{N}$ und $I$ mit $I(a) = 4$
$I(P) = \{(x, y) \mid x, y \text{ sind gerade}\}$ und $I(f) = f'$ wobei $f' : \mathbb{N} \to \mathbb{N}$ mit $f'(n) = 2 \cdot n$, dann ist:

$$\begin{array}{ll} & \mathcal{A}(P(a, f(a))) = 1 \\ gdw. & (\mathcal{A}(a), \mathcal{A}(f(a))) \in I(P) \\ gdw. & (4, I(f)(\mathcal{A}(a))) \in I(P) \\ gdw. & (4, f'(4)) \in I(P) \\ gdw. & (4, 8) \in I(P) \end{array}$$

Da $(4, 8) \in I(P)$ gilt, ist $\mathcal{A}$ eine erfüllende Struktur.

# Auswertung - Beispiel

## Beispiel

Ist $F = P(a, f(a))$ und $\mathcal{A} = (U, I)$ mit $U = \mathbb{N}$ und $I$ mit $I(a) = 4$ $I(P) = \{(x, y) \mid x, y \text{ sind gerade}\}$ und $I(f) = f'$ wobei $f' : \mathbb{N} \to \mathbb{N}$ mit $f'(n) = 2 \cdot n$, dann ist:

$$\begin{array}{ll} & \mathcal{A}(P(a, f(a))) = 1 \\ gdw. & (\mathcal{A}(a), \mathcal{A}(f(a))) \in I(P) \\ gdw. & (4, I(f)(\mathcal{A}(a))) \in I(P) \\ gdw. & (4, f'(4)) \in I(P) \\ gdw. & (4, 8) \in I(P) \end{array}$$

Da $(4, 8) \in I(P)$ gilt, ist $\mathcal{A}$ eine erfüllende Struktur.

# Auswertung - Beispiel

## Beispiel

Ist $F = P(a, f(a))$ und $\mathcal{A} = (U, I)$ mit $U = \mathbb{N}$ und $I$ mit $I(a) = 4$ $I(P) = \{(x, y) \mid x, y \text{ sind gerade}\}$ und $I(f) = f'$ wobei $f' : \mathbb{N} \to \mathbb{N}$ mit $f'(n) = 2 \cdot n$, dann ist:

$$
\begin{array}{rl}
 & \mathcal{A}(P(a, f(a))) = 1 \\
gdw. & (\mathcal{A}(a), \mathcal{A}(f(a))) \in I(P) \\
gdw. & (4, I(f)(\mathcal{A}(a))) \in I(P) \\
gdw. & (4, f'(4)) \in I(P) \\
gdw. & (4, 8) \in I(P)
\end{array}
$$

Da $(4, 8) \in I(P)$ gilt, ist $\mathcal{A}$ eine erfüllende Struktur.

# Auswertung - Beispiel

## Beispiel

Ist $F = P(a, f(a))$ und $\mathcal{A} = (U, I)$ mit $U = \mathbb{N}$ und $I$ mit $I(a) = 4$ $I(P) = \{(x, y) \mid x, y \text{ sind gerade}\}$ und $I(f) = f'$ wobei $f' : \mathbb{N} \to \mathbb{N}$ mit $f'(n) = 2 \cdot n$, dann ist:

$$\begin{array}{ll} & \mathcal{A}(P(a, f(a))) = 1 \\ gdw. & (\mathcal{A}(a), \mathcal{A}(f(a))) \in I(P) \\ gdw. & (4, I(f)(\mathcal{A}(a))) \in I(P) \\ gdw. & (4, f'(4)) \in I(P) \\ gdw. & (4, 8) \in I(P) \end{array}$$

Da $(4, 8) \in I(P)$ gilt, ist $\mathcal{A}$ eine erfüllende Struktur.

# Auswertung - Beispiel

## Beispiel

Ist $F = P(a, f(a))$ und $\mathcal{A} = (U, I)$ mit $U = \mathbb{N}$ und $I$ mit $I(a) = 4$ $I(P) = \{(x, y) \mid x, y \text{ sind gerade}\}$ und $I(f) = f'$ wobei $f' : \mathbb{N} \to \mathbb{N}$ mit $f'(n) = 2 \cdot n$, dann ist:

$$
\begin{aligned}
& \mathcal{A}(P(a, f(a))) = 1 \\
gdw. \quad & (\mathcal{A}(a), \mathcal{A}(f(a))) \in I(P) \\
gdw. \quad & (4, I(f)(\mathcal{A}(a))) \in I(P) \\
gdw. \quad & (4, f'(4)) \in I(P) \\
gdw. \quad & (4, 8) \in I(P)
\end{aligned}
$$

Da $(4, 8) \in I(P)$ gilt, ist $\mathcal{A}$ eine erfüllende Struktur.

# Auswertung

## Definition (Auswertung komplexer Formeln ohne Quantoren)

$I$ (bzw. $\mathcal{A}$) wird weiter rekursiv zu einer **Auswertung** fortgesetzt.
Die Junktoren werden dabei wie in der Aussagenlogik behandelt:
Für alle Formeln $F$ und $G$ und Strukturen $\mathcal{A}$ ist:

- $\mathcal{A}(\neg F) = 1$ genau dann, wenn $\mathcal{A}(F) = 0$
- $\mathcal{A}((F \vee G)) = 1$ gdw. $\mathcal{A}(F) = 1$ oder $\mathcal{A}(G) = 1$
- $\mathcal{A}((F \wedge G)) = 1$ gdw. $\mathcal{A}(F) = 1$ und $\mathcal{A}(G) = 1$
- $\mathcal{A}((F \Rightarrow G)) = 1$ gdw. $\mathcal{A}(F) = 0$ oder $\mathcal{A}(G) = 1$
- $\mathcal{A}((F \Leftrightarrow G)) = 1$ gdw. $\mathcal{A}(F) = \mathcal{A}(G)$

# Auswertung

## Definition (Auswertung komplexer Formeln ohne Quantoren)

$I$ (bzw. $\mathcal{A}$) wird weiter rekursiv zu einer **Auswertung** fortgesetzt. Die Junktoren werden dabei wie in der Aussagenlogik behandelt: Für alle Formeln $F$ und $G$ und Strukturen $\mathcal{A}$ ist:

- $\mathcal{A}(\neg F) = 1$ genau dann, wenn $\mathcal{A}(F) = 0$
- $\mathcal{A}((F \vee G)) = 1$ gdw. $\mathcal{A}(F) = 1$ oder $\mathcal{A}(G) = 1$
- $\mathcal{A}((F \wedge G)) = 1$ gdw. $\mathcal{A}(F) = 1$ und $\mathcal{A}(G) = 1$
- $\mathcal{A}((F \Rightarrow G)) = 1$ gdw. $\mathcal{A}(F) = 0$ oder $\mathcal{A}(G) = 1$
- $\mathcal{A}((F \Leftrightarrow G)) = 1$ gdw. $\mathcal{A}(F) = \mathcal{A}(G)$

# Auswertung - Beispiel

## Beispiel

Ist $F = P(a, f(a)) \wedge P(f(a), a)$ und $\mathcal{A} = (U, I)$ wie vorhin, dann ist

$$
\begin{array}{ll}
 & \mathcal{A}(P(a, f(a)) \wedge P(f(a), a)) = 1 \\
gdw. & (\mathcal{A}(a), \mathcal{A}(f(a))) \in I(P) \text{ und} \\
 & (\mathcal{A}(f(a)), \mathcal{A}(a)) \in I(P) \\
gdw. & \ldots \\
gdw. & (4, 8) \in I(P) \text{ und } (8, 4) \in I(P)
\end{array}
$$

# Auswertung - Beispiel

**Beispiel**

Ist $F = P(a, f(a)) \wedge P(f(a), a)$ und $\mathcal{A} = (U, I)$ wie vorhin, dann ist

$$\begin{array}{ll} & \mathcal{A}(P(a, f(a)) \wedge P(f(a), a)) = 1 \\ gdw. & (\mathcal{A}(a), \mathcal{A}(f(a))) \in I(P) \text{ und} \\ & (\mathcal{A}(f(a)), \mathcal{A}(a)) \in I(P) \\ gdw. & \ldots \\ gdw. & (4, 8) \in I(P) \text{ und } (8, 4) \in I(P) \end{array}$$

# Auswertung

## Definition ($x$-Variante von Strukturen)

Sei $\mathcal{A} = (U, I)$ eine Struktur, $d \in U$ und $x$ eine Variable. Mit $\mathcal{A}_{[x/d]} = (U, I_{[x/d]})$ bezeichnen wir die Struktur, die überall komplett mit $\mathcal{A}$ übereinstimmt, außer bei $x$. Dies wird mit $d$ interpretiert. Es gilt also für alle verfügbaren Symbole $\tau$:

$$\mathcal{A}_{[x/d]}(\tau) = I_{[x/d]}(\tau) = \begin{cases} d & \text{, falls } \tau = x \\ I(\tau) & \text{, sonst} \end{cases}$$

Die Struktur $\mathcal{A}_{[x/d]}$ wird als $x$-Variante zu $\mathcal{A}$ bezeichnet.

# Auswertung

## Definition ($x$-Variante von Strukturen)

Sei $\mathcal{A} = (U, I)$ eine Struktur, $d \in U$ und $x$ eine Variable. Mit $\mathcal{A}_{[x/d]} = (U, I_{[x/d]})$ bezeichnen wir die Struktur, die überall komplett mit $\mathcal{A}$ übereinstimmt, außer bei $x$. Dies wird mit $d$ interpretiert. Es gilt also für alle verfügbaren Symbole $\tau$:

$$\mathcal{A}_{[x/d]}(\tau) = I_{[x/d]}(\tau) = \begin{cases} d & \text{, falls } \tau = x \\ I(\tau) & \text{, sonst} \end{cases}$$

Die Struktur $\mathcal{A}_{[x/d]}$ wird als $x$-Variante zu $\mathcal{A}$ bezeichnet.

# Auswertung

## Definition ($x$-Variante von Strukturen)

Sei $\mathcal{A} = (U, I)$ eine Struktur, $d \in U$ und $x$ eine Variable. Mit $\mathcal{A}_{[x/d]} = (U, I_{[x/d]})$ bezeichnen wir die Struktur, die überall komplett mit $\mathcal{A}$ übereinstimmt, außer bei $x$. Dies wird mit $d$ interpretiert. Es gilt also für alle verfügbaren Symbole $\tau$:

$$\mathcal{A}_{[x/d]}(\tau) = I_{[x/d]}(\tau) = \begin{cases} d & \text{, falls } \tau = x \\ I(\tau) & \text{, sonst} \end{cases}$$

Die Struktur $\mathcal{A}_{[x/d]}$ wird als $x$-Variante zu $\mathcal{A}$ bezeichnet.

# Auswertung

**Definition (Auswertung komplexer Formeln mit Quantoren)**

Wir schließen die rekursive Fortsetzung von $I$ mit der Betrachtung der Quantoren ab: Für jede Variable $x$, jede Formel $F$ und jede Struktur $\mathcal{A}$ ist

$$\mathcal{A}(\forall x F) = \begin{cases} 1 & \text{, falls für alle } d \in U \text{ gilt: } \mathcal{A}_{[x/d]}(F) = 1 \\ 0 & \text{, sonst} \end{cases}$$

$$\mathcal{A}(\exists x F) = \begin{cases} 1 & \text{, falls es ein } d \in U \text{ gibt mit: } \mathcal{A}_{[x/d]}(F) = 1 \\ 0 & \text{, sonst} \end{cases}$$

# Auswertung

**Definition (Auswertung komplexer Formeln mit Quantoren)**

Wir schließen die rekursive Fortsetzung von $I$ mit der Betrachtung der Quantoren ab: Für jede Variable $x$, jede Formel $F$ und jede Struktur $\mathcal{A}$ ist

$$\mathcal{A}(\forall x F) = \begin{cases} 1 & \text{, falls für alle } d \in U \text{ gilt: } \mathcal{A}_{[x/d]}(F) = 1 \\ 0 & \text{, sonst} \end{cases}$$

$$\mathcal{A}(\exists x F) = \begin{cases} 1 & \text{, falls es ein } d \in U \text{ gibt mit: } \mathcal{A}_{[x/d]}(F) = 1 \\ 0 & \text{, sonst} \end{cases}$$

# Anmerkung zu den Quantoren

**Wichtige Anmerkung**

Bei der Auswertung einer quantifizierten Formel ist der ursprüngliche Wert der Quantorenvariable unerheblich, d.h. für z.B. $\forall x P(x)$ ist der Wert von $I(x)$ nicht erheblich (und muss auch gar nicht definiert sein).

# Auswertung - Beispiel

## Beispiel

Sei $F = \forall x(P(x) \vee Q(a, f(x)))$. Dann ist

$$\begin{array}{ll}
& \mathcal{A}(\forall x(P(x) \vee Q(a, f(x)))) = 1 \\
gdw. & \text{für alle } d \in U \text{ gilt: } \mathcal{A}_{[x/d]}(P(x) \vee Q(a, f(x))) = 1 \\
gdw. & \text{für alle } d \in U \text{ gilt: } \mathcal{A}_{[x/d]}(P(x)) = 1 \text{ oder} \\
& \mathcal{A}_{[x/d]}(Q(a, f(x))) = 1 \\
gdw. & \text{für alle } d \in U \text{ gilt: } \mathcal{A}_{[x/d]}(x) \in I(P) \text{ oder} \\
& (\mathcal{A}_{[x/d]}(a), \mathcal{A}_{[x/d]}(f(x))) \in I(Q) \\
gdw. & \text{für alle } d \in U \text{ gilt: } d \in I(P) \text{ oder} \\
& (I(a), I(f)(d)) \in I(Q)
\end{array}$$

Eine erfüllende Struktur ist z.B. durch

# Auswertung - Beispiel

## Beispiel

Sei $F = \forall x(P(x) \vee Q(a, f(x)))$. Dann ist

$$
\begin{aligned}
& \mathcal{A}(\forall x(P(x) \vee Q(a, f(x)))) = 1 \\
gdw. \quad & \text{für alle } d \in U \text{ gilt: } \mathcal{A}_{[x/d]}(P(x) \vee Q(a, f(x))) = 1 \\
gdw. \quad & \text{für alle } d \in U \text{ gilt: } \mathcal{A}_{[x/d]}(P(x)) = 1 \text{ oder} \\
& \mathcal{A}_{[x/d]}(Q(a, f(x))) = 1 \\
gdw. \quad & \text{für alle } d \in U \text{ gilt: } \mathcal{A}_{[x/d]}(x) \in I(P) \text{ oder} \\
& (\mathcal{A}_{[x/d]}(a), \mathcal{A}_{[x/d]}(f(x))) \in I(Q) \\
gdw. \quad & \text{für alle } d \in U \text{ gilt: } d \in I(P) \text{ oder} \\
& (I(a), I(f)(d)) \in I(Q)
\end{aligned}
$$

Eine erfüllende Struktur ist z.B. durch

# Auswertung - Beispiel

## Beispiel

Sei $F = \forall x(P(x) \vee Q(a, f(x)))$. Dann ist

$$
\begin{array}{ll}
 & \mathcal{A}(\forall x(P(x) \vee Q(a, f(x)))) = 1 \\
gdw. & \text{für alle } d \in U \text{ gilt: } \mathcal{A}_{[x/d]}(P(x) \vee Q(a, f(x))) = 1 \\
gdw. & \text{für alle } d \in U \text{ gilt: } \mathcal{A}_{[x/d]}(P(x)) = 1 \text{ oder} \\
 & \mathcal{A}_{[x/d]}(Q(a, f(x))) = 1 \\
gdw. & \text{für alle } d \in U \text{ gilt: } \mathcal{A}_{[x/d]}(x) \in I(P) \text{ oder} \\
 & (\mathcal{A}_{[x/d]}(a), \mathcal{A}_{[x/d]}(f(x))) \in I(Q) \\
gdw. & \text{für alle } d \in U \text{ gilt: } d \in I(P) \text{ oder} \\
 & (I(a), I(f)(d)) \in I(Q)
\end{array}
$$

Eine erfüllende Struktur ist z.B. durch

# Auswertung - Beispiel

## Beispiel

Sei $F = \forall x(P(x) \vee Q(a, f(x)))$. Dann ist

$$
\begin{array}{ll}
 & \mathcal{A}(\forall x(P(x) \vee Q(a, f(x)))) = 1 \\
gdw. & \text{für alle } d \in U \text{ gilt: } \mathcal{A}_{[x/d]}(P(x) \vee Q(a, f(x))) = 1 \\
gdw. & \text{für alle } d \in U \text{ gilt: } \mathcal{A}_{[x/d]}(P(x)) = 1 \text{ oder} \\
 & \mathcal{A}_{[x/d]}(Q(a, f(x))) = 1 \\
gdw. & \text{für alle } d \in U \text{ gilt: } \mathcal{A}_{[x/d]}(x) \in I(P) \text{ oder} \\
 & (\mathcal{A}_{[x/d]}(a), \mathcal{A}_{[x/d]}(f(x))) \in I(Q) \\
gdw. & \text{für alle } d \in U \text{ gilt: } d \in I(P) \text{ oder} \\
 & (I(a), I(f)(d)) \in I(Q)
\end{array}
$$

Eine erfüllende Struktur ist z.B. durch $U = \mathbb{N}$ und $I(P) = U$ (Rest beliebig) gegeben (warum?) oder durch $I(a) = 1$, $I(f) = f'$ mit $f' : \mathbb{N} \to \mathbb{N}$ und $f'(n) = 1$ und $I(Q) = \{(1, 1)\}$ (warum?).

# Analoge Begriffe

Viele Begriffe werden analog zur AL definiert:

- Eine Struktur $\mathcal{A}$ heißt **Modell** für $F$, wenn $\mathcal{A}(F) = 1$ gilt.
- Eine Struktur ist Modell einer Formelmenge $M$, wenn sie alle Formeln in $M$ wahr macht.
- Eine Formel $F$ heißt erfüllbar, wenn es eine Struktur $\mathcal{A}$ gibt mit $\mathcal{A}(F) = 1$
- Eine Formel $F$ heißt falsifizierbar, wenn es eine Struktur $\mathcal{A}$ gibt mit $\mathcal{A}(F) = 0$
- Die Begriffe (allgemein-)gültig, kontingent, Tautologie, Kontradiktion, Folgerbarkeit und Äquivalenz werden wie in der Aussagenlogik eingeführt. Formulierung wie *alle Strukturen* beinhalten allerdings alle Strukturen mit beliebigen Domänen und Interpretationen!

# Analoge Begriffe

Viele Begriffe werden analog zur AL definiert:

- Eine Struktur $\mathcal{A}$ heißt **Modell** für $F$, wenn $\mathcal{A}(F) = 1$ gilt.
- Eine Struktur ist Modell einer Formelmenge $M$, wenn sie alle Formeln in $M$ wahr macht.
- Eine Formel $F$ heißt erfüllbar, wenn es eine Struktur $\mathcal{A}$ gibt mit $\mathcal{A}(F) = 1$
- Eine Formel $F$ heißt falsifizierbar, wenn es eine Struktur $\mathcal{A}$ gibt mit $\mathcal{A}(F) = 0$
- Die Begriffe (allgemein-)gültig, kontingent, Tautologie, Kontradiktion, Folgerbarkeit und Äquivalenz werden wie in der Aussagenlogik eingeführt. Formulierung wie *alle Strukturen* beinhalten allerdings alle Strukturen mit beliebigen Domänen und Interpretationen!

# Analoge Begriffe

Viele Begriffe werden analog zur AL definiert:

- Eine Struktur $\mathcal{A}$ heißt **Modell** für $F$, wenn $\mathcal{A}(F) = 1$ gilt.
- Eine Struktur ist Modell einer Formelmenge $M$, wenn sie alle Formeln in $M$ wahr macht.
- Eine Formel $F$ heißt erfüllbar, wenn es eine Struktur $\mathcal{A}$ gibt mit $\mathcal{A}(F) = 1$
- Eine Formel $F$ heißt falsifizierbar, wenn es eine Struktur $\mathcal{A}$ gibt mit $\mathcal{A}(F) = 0$
- Die Begriffe (allgemein-)gültig, kontingent, Tautologie, Kontradiktion, Folgerbarkeit und Äquivalenz werden wie in der Aussagenlogik eingeführt. Formulierung wie *alle Strukturen* beinhalten allerdings alle Strukturen mit beliebigen Domänen und Interpretationen!

# Analoge Begriffe

Viele Begriffe werden analog zur AL definiert:

- Eine Struktur $\mathcal{A}$ heißt **Modell** für $F$, wenn $\mathcal{A}(F) = 1$ gilt.
- Eine Struktur ist Modell einer Formelmenge $M$, wenn sie alle Formeln in $M$ wahr macht.
- Eine Formel $F$ heißt erfüllbar, wenn es eine Struktur $\mathcal{A}$ gibt mit $\mathcal{A}(F) = 1$
- Eine Formel $F$ heißt falsifizierbar, wenn es eine Struktur $\mathcal{A}$ gibt mit $\mathcal{A}(F) = 0$
- Die Begriffe (allgemein-)gültig, kontingent, Tautologie, Kontradiktion, Folgerbarkeit und Äquivalenz werden wie in der Aussagenlogik eingeführt. Formulierung wie *alle Strukturen* beinhalten allerdings alle Strukturen mit beliebigen Domänen und Interpretationen!

# Analoge Begriffe

Viele Begriffe werden analog zur AL definiert:

- Eine Struktur $\mathcal{A}$ heißt **Modell** für $F$, wenn $\mathcal{A}(F) = 1$ gilt.
- Eine Struktur ist Modell einer Formelmenge $M$, wenn sie alle Formeln in $M$ wahr macht.
- Eine Formel $F$ heißt erfüllbar, wenn es eine Struktur $\mathcal{A}$ gibt mit $\mathcal{A}(F) = 1$
- Eine Formel $F$ heißt falsifizierbar, wenn es eine Struktur $\mathcal{A}$ gibt mit $\mathcal{A}(F) = 0$
- Die Begriffe (allgemein-)gültig, kontingent, Tautologie, Kontradiktion, Folgerbarkeit und Äquivalenz werden wie in der Aussagenlogik eingeführt. Formulierung wie *alle Strukturen* beinhalten allerdings alle Strukturen mit beliebigen Domänen und Interpretationen!

# Analoge Begriffe

Viele Begriffe werden analog zur AL definiert:

- Eine Struktur $\mathcal{A}$ heißt **Modell** für $F$, wenn $\mathcal{A}(F) = 1$ gilt.
- Eine Struktur ist Modell einer Formelmenge $M$, wenn sie alle Formeln in $M$ wahr macht.
- Eine Formel $F$ heißt erfüllbar, wenn es eine Struktur $\mathcal{A}$ gibt mit $\mathcal{A}(F) = 1$
- Eine Formel $F$ heißt falsifizierbar, wenn es eine Struktur $\mathcal{A}$ gibt mit $\mathcal{A}(F) = 0$
- Die Begriffe (allgemein-)gültig, kontingent, Tautologie, Kontradiktion, Folgerbarkeit und Äquivalenz werden wie in der Aussagenlogik eingeführt. Formulierung wie *alle Strukturen* beinhalten allerdings alle Strukturen mit beliebigen Domänen und Interpretationen!

# Beispiel

## Beispiel

Es gilt $\models \forall x P(x) \Rightarrow \exists y P(y)$.

Angenommen $\mathcal{A}$ ist eine Struktur mit $\mathcal{A}(\forall x P(x) \Rightarrow \exists y P(y)) = 0$. Dann ist $\mathcal{A}(\forall x P(x)) = 1$ und $\mathcal{A}(\exists y P(y)) = 0$. Nun kann $\mathcal{A}(\exists y P(y)) = 0$ nur gelten, wenn für jedes $d \in U$ $\mathcal{A}_{[y/d]}(P(y)) = 0$, also für jedes $d \in U$ $d \notin I(P)$ gilt. Dafür muss $I(P) = \emptyset$ sein.

Für $\mathcal{A}(\forall x P(x)) = 1$ muss für jedes $e \in U$ $\mathcal{A}_{[x/e]}(P(x)) = 1$ gelten, also muss für jedes $e \in U$ $e \in I(P)$ gelten. Dafür muss $I(P) = U$ sein.

Nach Definition darf $U$ aber nicht leer sein und die Struktur $\mathcal{A}$ kann es nicht geben!

# Beispiel

## Beispiel

Es gilt $\models \forall x P(x) \Rightarrow \exists y P(y)$.

Angenommen $\mathcal{A}$ ist eine Struktur mit $\mathcal{A}(\forall x P(x) \Rightarrow \exists y P(y)) = 0$. Dann ist $\mathcal{A}(\forall x P(x)) = 1$ und $\mathcal{A}(\exists y P(y)) = 0$. Nun kann $\mathcal{A}(\exists y P(y)) = 0$ nur gelten, wenn für jedes $d \in U$ $\mathcal{A}_{[y/d]}(P(y)) = 0$, also für jedes $d \in U$ $d \notin I(P)$ gilt. Dafür muss $I(P) = \emptyset$ sein.

Für $\mathcal{A}(\forall x P(x)) = 1$ muss für jedes $e \in U$ $\mathcal{A}_{[x/e]}(P(x)) = 1$ gelten, also muss für jedes $e \in U$ $e \in I(P)$ gelten. Dafür muss $I(P) = U$ sein.

Nach Definition darf $U$ aber nicht leer sein und die Struktur $\mathcal{A}$ kann es nicht geben!

# Beispiel

## Beispiel

Es gilt $\models \forall x P(x) \Rightarrow \exists y P(y)$.
Angenommen $\mathcal{A}$ ist eine Struktur mit $\mathcal{A}(\forall x P(x) \Rightarrow \exists y P(y)) = 0$. Dann ist $\mathcal{A}(\forall x P(x)) = 1$ und $\mathcal{A}(\exists y P(y)) = 0$. Nun kann $\mathcal{A}(\exists y P(y)) = 0$ nur gelten, wenn für jedes $d \in U$ $\mathcal{A}_{[y/d]}(P(y)) = 0$, also für jedes $d \in U$ $d \notin I(P)$ gilt. Dafür muss $I(P) = \emptyset$ sein.
Für $\mathcal{A}(\forall x P(x)) = 1$ muss für jedes $e \in U$ $\mathcal{A}_{[x/e]}(P(x)) = 1$ gelten, also muss für jedes $e \in U$ $e \in I(P)$ gelten. Dafür muss $I(P) = U$ sein.
Nach Definition darf $U$ aber nicht leer sein und die Struktur $\mathcal{A}$ kann es nicht geben!

# Beispiel

## Beispiel

Es gilt $\models \forall x P(x) \Rightarrow \exists y P(y)$.
Angenommen $\mathcal{A}$ ist eine Struktur mit $\mathcal{A}(\forall x P(x) \Rightarrow \exists y P(y)) = 0$. Dann ist $\mathcal{A}(\forall x P(x)) = 1$ und $\mathcal{A}(\exists y P(y)) = 0$. Nun kann $\mathcal{A}(\exists y P(y)) = 0$ nur gelten, wenn für jedes $d \in U$ $\mathcal{A}_{[y/d]}(P(y)) = 0$, also für jedes $d \in U$ $d \notin I(P)$ gilt. Dafür muss $I(P) = \emptyset$ sein.
Für $\mathcal{A}(\forall x P(x)) = 1$ muss für jedes $e \in U$ $\mathcal{A}_{[x/e]}(P(x)) = 1$ gelten, also muss für jedes $e \in U$ $e \in I(P)$ gelten. Dafür muss $I(P) = U$ sein.
Nach Definition darf $U$ aber nicht leer sein und die Struktur $\mathcal{A}$ kann es nicht geben!

# Beispiel

## Beispiel

Es gilt $\models \forall x P(x) \Rightarrow \exists y P(y)$.
Angenommen $\mathcal{A}$ ist eine Struktur mit $\mathcal{A}(\forall x P(x) \Rightarrow \exists y P(y)) = 0$. Dann ist $\mathcal{A}(\forall x P(x)) = 1$ und $\mathcal{A}(\exists y P(y)) = 0$. Nun kann $\mathcal{A}(\exists y P(y)) = 0$ nur gelten, wenn für jedes $d \in U$ $\mathcal{A}_{[y/d]}(P(y)) = 0$, also für jedes $d \in U$ $d \notin I(P)$ gilt. Dafür muss $I(P) = \emptyset$ sein.
Für $\mathcal{A}(\forall x P(x)) = 1$ muss für jedes $e \in U$ $\mathcal{A}_{[x/e]}(P(x)) = 1$ gelten, also muss für jedes $e \in U$ $e \in I(P)$ gelten. Dafür muss $I(P) = U$ sein.
Nach Definition darf $U$ aber nicht leer sein und die Struktur $\mathcal{A}$ kann es nicht geben!

# Beispiel

## Beispiel

Es gilt $\models \forall x P(x) \Rightarrow \exists y P(y)$.

Angenommen $\mathcal{A}$ ist eine Struktur mit $\mathcal{A}(\forall x P(x) \Rightarrow \exists y P(y)) = 0$. Dann ist $\mathcal{A}(\forall x P(x)) = 1$ und $\mathcal{A}(\exists y P(y)) = 0$. Nun kann $\mathcal{A}(\exists y P(y)) = 0$ nur gelten, wenn für jedes $d \in U$ $\mathcal{A}_{[y/d]}(P(y)) = 0$, also für jedes $d \in U$ $d \notin I(P)$ gilt. Dafür muss $I(P) = \emptyset$ sein.

Für $\mathcal{A}(\forall x P(x)) = 1$ muss für jedes $e \in U$ $\mathcal{A}_{[x/e]}(P(x)) = 1$ gelten, also muss für jedes $e \in U$ $e \in I(P)$ gelten. Dafür muss $I(P) = U$ sein.

Nach Definition darf $U$ aber nicht leer sein und die Struktur $\mathcal{A}$ kann es nicht geben!

# Beispiel

## Beispiel

Es gilt $\models \forall x P(x) \Rightarrow \exists y P(y)$.
Angenommen $\mathcal{A}$ ist eine Struktur mit $\mathcal{A}(\forall x P(x) \Rightarrow \exists y P(y)) = 0$. Dann ist $\mathcal{A}(\forall x P(x)) = 1$ und $\mathcal{A}(\exists y P(y)) = 0$. Nun kann $\mathcal{A}(\exists y P(y)) = 0$ nur gelten, wenn für jedes $d \in U$ $\mathcal{A}_{[y/d]}(P(y)) = 0$, also für jedes $d \in U$ $d \notin I(P)$ gilt. Dafür muss $I(P) = \emptyset$ sein.
Für $\mathcal{A}(\forall x P(x)) = 1$ muss für jedes $e \in U$ $\mathcal{A}_{[x/e]}(P(x)) = 1$ gelten, also muss für jedes $e \in U$ $e \in I(P)$ gelten. Dafür muss $I(P) = U$ sein.
Nach Definition darf $U$ aber nicht leer sein und die Struktur $\mathcal{A}$ kann es nicht geben!

# Beispiel

## Beispiel

Es gilt $\models \forall x P(x) \Rightarrow \exists y P(y)$.

Angenommen $\mathcal{A}$ ist eine Struktur mit $\mathcal{A}(\forall x P(x) \Rightarrow \exists y P(y)) = 0$. Dann ist $\mathcal{A}(\forall x P(x)) = 1$ und $\mathcal{A}(\exists y P(y)) = 0$. Nun kann $\mathcal{A}(\exists y P(y)) = 0$ nur gelten, wenn für jedes $d \in U$ $\mathcal{A}_{[y/d]}(P(y)) = 0$, also für jedes $d \in U$ $d \notin I(P)$ gilt. Dafür muss $I(P) = \emptyset$ sein.

Für $\mathcal{A}(\forall x P(x)) = 1$ muss für jedes $e \in U$ $\mathcal{A}_{[x/e]}(P(x)) = 1$ gelten, also muss für jedes $e \in U$ $e \in I(P)$ gelten. Dafür muss $I(P) = U$ sein.

Nach Definition darf $U$ aber nicht leer sein und die Struktur $\mathcal{A}$ kann es nicht geben!

# Beispiel

## Beispiel

Es gilt $\models \forall x P(x) \Rightarrow \exists y P(y)$.
Angenommen $\mathcal{A}$ ist eine Struktur mit $\mathcal{A}(\forall x P(x) \Rightarrow \exists y P(y)) = 0$. Dann ist $\mathcal{A}(\forall x P(x)) = 1$ und $\mathcal{A}(\exists y P(y)) = 0$. Nun kann $\mathcal{A}(\exists y P(y)) = 0$ nur gelten, wenn für jedes $d \in U$ $\mathcal{A}_{[y/d]}(P(y)) = 0$, also für jedes $d \in U$ $d \notin I(P)$ gilt. Dafür muss $I(P) = \emptyset$ sein.
Für $\mathcal{A}(\forall x P(x)) = 1$ muss für jedes $e \in U$ $\mathcal{A}_{[x/e]}(P(x)) = 1$ gelten, also muss für jedes $e \in U$ $e \in I(P)$ gelten. Dafür muss $I(P) = U$ sein.
Nach Definition darf $U$ aber nicht leer sein und die Struktur $\mathcal{A}$ kann es nicht geben!

# Beispiel

## Beispiel

Es gilt $\models \forall x P(x) \Rightarrow \exists y P(y)$.
Angenommen $\mathcal{A}$ ist eine Struktur mit $\mathcal{A}(\forall x P(x) \Rightarrow \exists y P(y)) = 0$. Dann ist $\mathcal{A}(\forall x P(x)) = 1$ und $\mathcal{A}(\exists y P(y)) = 0$. Nun kann $\mathcal{A}(\exists y P(y)) = 0$ nur gelten, wenn für jedes $d \in U$ $\mathcal{A}_{[y/d]}(P(y)) = 0$, also für jedes $d \in U$ $d \notin I(P)$ gilt. Dafür muss $I(P) = \emptyset$ sein.
Für $\mathcal{A}(\forall x P(x)) = 1$ muss für jedes $e \in U$ $\mathcal{A}_{[x/e]}(P(x)) = 1$ gelten, also muss für jedes $e \in U$ $e \in I(P)$ gelten. Dafür muss $I(P) = U$ sein.
Nach Definition darf $U$ aber nicht leer sein und die Struktur $\mathcal{A}$ kann es nicht geben!

# Beispiel

## Beispiel

Es gilt $\models \forall x P(x) \Rightarrow \exists y P(y)$.

Angenommen $\mathcal{A}$ ist eine Struktur mit $\mathcal{A}(\forall x P(x) \Rightarrow \exists y P(y)) = 0$. Dann ist $\mathcal{A}(\forall x P(x)) = 1$ und $\mathcal{A}(\exists y P(y)) = 0$. Nun kann $\mathcal{A}(\exists y P(y)) = 0$ nur gelten, wenn für jedes $d \in U$ $\mathcal{A}_{[y/d]}(P(y)) = 0$, also für jedes $d \in U$ $d \notin I(P)$ gilt. Dafür muss $I(P) = \emptyset$ sein.

Für $\mathcal{A}(\forall x P(x)) = 1$ muss für jedes $e \in U$ $\mathcal{A}_{[x/e]}(P(x)) = 1$ gelten, also muss für jedes $e \in U$ $e \in I(P)$ gelten. Dafür muss $I(P) = U$ sein.

Nach Definition darf $U$ aber nicht leer sein und die Struktur $\mathcal{A}$ kann es nicht geben!

# Beispiel

## Beispiel

Es gilt $\models \forall x P(x) \Rightarrow \exists y P(y)$.
Angenommen $\mathcal{A}$ ist eine Struktur mit $\mathcal{A}(\forall x P(x) \Rightarrow \exists y P(y)) = 0$. Dann ist $\mathcal{A}(\forall x P(x)) = 1$ und $\mathcal{A}(\exists y P(y)) = 0$. Nun kann $\mathcal{A}(\exists y P(y)) = 0$ nur gelten, wenn für jedes $d \in U$ $\mathcal{A}_{[y/d]}(P(y)) = 0$, also für jedes $d \in U$ $d \notin I(P)$ gilt. Dafür muss $I(P) = \emptyset$ sein.
Für $\mathcal{A}(\forall x P(x)) = 1$ muss für jedes $e \in U$ $\mathcal{A}_{[x/e]}(P(x)) = 1$ gelten, also muss für jedes $e \in U$ $e \in I(P)$ gelten. Dafür muss $I(P) = U$ sein.
Nach Definition darf $U$ aber nicht leer sein und die Struktur $\mathcal{A}$ kann es nicht geben!

# Zusammenfassung

Wir haben heute

- Prädikatenlogische Syntax und
- Prädikatenlogische Semantik

eingeführt.

An der Semantik erkennt man schon, dass dies viel komplizierter ist als bei der Aussagenlogik!

# Ausblick

Wie in der Aussagenlogik werden wir zuerst Syntax & Semantik einüben.

Im Anschluss ist es wieder wichtig für eine gegebene Formel herauszufinden, ob sie erfüllbar ist oder unerfüllbar. Genauso sind wieder Tests daraufhin interessant, ob $F \models G$ gilt, ob $F \equiv G$ usw. Dazu kann man wieder die Resolution benutzen.

Ausblick:

- Normalformen in der Prädikatenlogik (Klauselnormalform)
- Prädikatenlogische Resolution