

Inferenzmethoden

Einheit 17

Logik höherer Stufe



1. Syntax und Semantik
2. Simulation mathematischer Konstrukte
3. Beweisführung in Logik höherer Stufe

LOGIK HÖHERER STUFE

- **Logik erster Stufe hat nur einfache Variablen**
 - Keine Quantifizierung über Funktions- oder Prädikatensymbole erlaubt

LOGIK HÖHERER STUFE

- **Logik erster Stufe hat nur einfache Variablen**
 - Keine Quantifizierung über Funktions- oder Prädikatensymbole erlaubt
- **Konstrukte höherer Stufe kommen überall vor**
 - Funktionen (2. Stufe): “Bestimme $x+2$ bei Eingabe x ” $\lambda x.x+2$

LOGIK HÖHERER STUFE

- **Logik erster Stufe hat nur einfache Variablen**

- Keine Quantifizierung über Funktions- oder Prädikatensymbole erlaubt

- **Konstrukte höherer Stufe kommen überall vor**

- Funktionen (2. Stufe): “Bestimme $x+2$ bei Eingabe x ” $\lambda x.x+2$

- Induktionsprinzip: $\forall P.P(0) \wedge (\forall y:\mathbb{N}.P(y) \Rightarrow P(y+1)) \Rightarrow \forall x:\mathbb{N}.P(x)$

LOGIK HÖHERER STUFE

- **Logik erster Stufe hat nur einfache Variablen**

- Keine Quantifizierung über Funktions- oder Prädikatensymbole erlaubt

- **Konstrukte höherer Stufe kommen überall vor**

- Funktionen (2. Stufe): “Bestimme $x+2$ bei Eingabe x ” $\lambda x.x+2$

- Induktionsprinzip: $\forall P.P(0) \wedge (\forall y:\mathbb{N}.P(y) \Rightarrow P(y+1)) \Rightarrow \forall x:\mathbb{N}.P(x)$

- Zwischenwertsatz: $\forall f:\mathbb{R} \rightarrow \mathbb{R}.\forall a < b:\mathbb{R}.(f(a) > 0 \wedge f(b) < 0) \Rightarrow \exists x:\mathbb{R}.f(x)=0$

LOGIK HÖHERER STUFE

- **Logik erster Stufe hat nur einfache Variablen**

- Keine Quantifizierung über Funktions- oder Prädikatensymbole erlaubt

- **Konstrukte höherer Stufe kommen überall vor**

- Funktionen (2. Stufe): “Bestimme $x+2$ bei Eingabe x ” $\lambda x.x+2$

- Induktionsprinzip: $\forall P.P(0) \wedge (\forall y:\mathbb{N}.P(y) \Rightarrow P(y+1)) \Rightarrow \forall x:\mathbb{N}.P(x)$

- Zwischenwertsatz: $\forall f:\mathbb{R} \rightarrow \mathbb{R}.\forall a < b:\mathbb{R}.(f(a) > 0 \wedge f(b) < 0) \Rightarrow \exists x:\mathbb{R}.f(x)=0$

- Funktionale (3. Stufe): Ableitungsoperator $\lambda f.df/dx$

LOGIK HÖHERER STUFE

- **Logik erster Stufe hat nur einfache Variablen**

- Keine Quantifizierung über Funktions- oder Prädikatensymbole erlaubt

- **Konstrukte höherer Stufe kommen überall vor**

- Funktionen (2. Stufe): “Bestimme $x+2$ bei Eingabe x ” $\lambda x. x+2$

- Induktionsprinzip: $\forall P. P(0) \wedge (\forall y:\mathbb{N}. P(y) \Rightarrow P(y+1)) \Rightarrow \forall x:\mathbb{N}. P(x)$

- Zwischenwertsatz: $\forall f:\mathbb{R} \rightarrow \mathbb{R}. \forall a < b:\mathbb{R}. (f(a) > 0 \wedge f(b) < 0) \Rightarrow \exists x:\mathbb{R}. f(x) = 0$

- Funktionale (3. Stufe): Ableitungsoperator $\lambda f. df/dx$

- Quantifizierung über Funktionale, ...

LOGIK HÖHERER STUFE

- **Logik erster Stufe hat nur einfache Variablen**

- Keine Quantifizierung über Funktions- oder Prädikatensymbole erlaubt

- **Konstrukte höherer Stufe kommen überall vor**

- Funktionen (2. Stufe): “Bestimme $x+2$ bei Eingabe x ” $\lambda x.x+2$

- Induktionsprinzip: $\forall P.P(0) \wedge (\forall y:\mathbb{N}.P(y) \Rightarrow P(y+1)) \Rightarrow \forall x:\mathbb{N}.P(x)$

- Zwischenwertsatz: $\forall f:\mathbb{R}\rightarrow\mathbb{R}.\forall a < b:\mathbb{R}.(f(a)>0 \wedge f(b)<0) \Rightarrow \exists x:\mathbb{R}.f(x)=0$

- Funktionale (3. Stufe): Ableitungsoperator $\lambda f.df/dx$

- Quantifizierung über Funktionale, ...

- **Logik höherer Stufe hat keine Einschränkungen**

- Extrem einfache Syntax: $x, \lambda x.t, f(a), \forall x P, P \Rightarrow Q$

LOGIK HÖHERER STUFE

- **Logik erster Stufe hat nur einfache Variablen**

- Keine Quantifizierung über Funktions- oder Prädikatensymbole erlaubt

- **Konstrukte höherer Stufe kommen überall vor**

- Funktionen (2. Stufe): “Bestimme $x+2$ bei Eingabe x ” $\lambda x.x+2$

- Induktionsprinzip: $\forall P.P(0) \wedge (\forall y:\mathbb{N}.P(y) \Rightarrow P(y+1)) \Rightarrow \forall x:\mathbb{N}.P(x)$

- Zwischenwertsatz: $\forall f:\mathbb{R} \rightarrow \mathbb{R}.\forall a < b:\mathbb{R}.(f(a) > 0 \wedge f(b) < 0) \Rightarrow \exists x:\mathbb{R}.f(x)=0$

- Funktionale (3. Stufe): Ableitungsoperator $\lambda f.df/dx$

- Quantifizierung über Funktionale, ...

- **Logik höherer Stufe hat keine Einschränkungen**

- Extrem einfache Syntax: $x, \lambda x.t, f(a), \forall x P, P \Rightarrow Q$

- Fester Auswertungsmechanismus (β -)Reduktion: $(\lambda x.t)(a) \longrightarrow t[a/x]$

LOGIK HÖHERER STUFE

- **Logik erster Stufe hat nur einfache Variablen**
 - Keine Quantifizierung über Funktions- oder Prädikatensymbole erlaubt
- **Konstrukte höherer Stufe kommen überall vor**
 - Funktionen (2. Stufe): “Bestimme $x+2$ bei Eingabe x ” $\lambda x.x+2$
 - Induktionsprinzip: $\forall P.P(0) \wedge (\forall y:\mathbb{N}.P(y) \Rightarrow P(y+1)) \Rightarrow \forall x:\mathbb{N}.P(x)$
 - Zwischenwertsatz: $\forall f:\mathbb{R} \rightarrow \mathbb{R}.\forall a < b:\mathbb{R}.(f(a) > 0 \wedge f(b) < 0) \Rightarrow \exists x:\mathbb{R}.f(x)=0$
 - Funktionale (3. Stufe): Ableitungsoperator $\lambda f.df/dx$
 - Quantifizierung über Funktionale, ...
- **Logik höherer Stufe hat keine Einschränkungen**
 - Extrem einfache Syntax: $x, \lambda x.t, f(a), \forall x P, P \Rightarrow Q$
 - Fester Auswertungsmechanismus (β -)Reduktion: $(\lambda x.t)(a) \longrightarrow t[a/x]$
- **Logik höherer Stufe ist minimale Grundagentheorie**
 - Keine Abstützung auf Mengentheorie erforderlich
 - Reduktion erklärt Wert von Ausdrücken

LOGIK HÖHERER STUFE

- **Logik erster Stufe hat nur einfache Variablen**
 - Keine Quantifizierung über Funktions- oder Prädikatensymbole erlaubt
- **Konstrukte höherer Stufe kommen überall vor**
 - Funktionen (2. Stufe): “Bestimme $x+2$ bei Eingabe x ” $\lambda x.x+2$
 - Induktionsprinzip: $\forall P.P(0) \wedge (\forall y:\mathbb{N}.P(y) \Rightarrow P(y+1)) \Rightarrow \forall x:\mathbb{N}.P(x)$
 - Zwischenwertsatz: $\forall f:\mathbb{R}\rightarrow\mathbb{R}.\forall a < b:\mathbb{R}.(f(a)>0 \wedge f(b)<0) \Rightarrow \exists x:\mathbb{R}.f(x)=0$
 - Funktionale (3. Stufe): Ableitungsoperator $\lambda f.df/dx$
 - Quantifizierung über Funktionale, ...
- **Logik höherer Stufe hat keine Einschränkungen**
 - Extrem einfache Syntax: $x, \lambda x.t, f(a), \forall x P, P \Rightarrow Q$
 - Fester Auswertungsmechanismus (β -)Reduktion: $(\lambda x.t)(a) \longrightarrow t[a/x]$
- **Logik höherer Stufe ist minimale Grundagentheorie**
 - Keine Abstützung auf Mengentheorie erforderlich
 - Reduktion erklärt Wert von Ausdrücken
 - Mathematische Konzepte werden nicht über Axiome erklärt sondern als definitorische Abkürzung für komplexe Terme (logizistischer Ansatz)

- **Alphabet für erlaubte Symbole**
 - \mathcal{V} : Variablensymbole

- **Alphabet für erlaubte Symbole**
 - \mathcal{V} : Variablensymbole
- **Terme**
 - Variablen $x \in \mathcal{V}$

- **Alphabet für erlaubte Symbole**

- \mathcal{V} : Variablensymbole

- **Terme**

- Variablen $x \in \mathcal{V}$

- $\lambda x.t$, wobei $x \in \mathcal{V}$ und t Term

- $f t$, wobei t und f Terme

- (t) , wobei t Term

λ -Abstraktion

Applikation

- **Alphabet für erlaubte Symbole**

- \mathcal{V} : Variablensymbole

- **Terme**

- Variablen $x \in \mathcal{V}$

- $\lambda x.t$, wobei $x \in \mathcal{V}$ und t Term

λ -Abstraktion

- $f t$, wobei t und f Terme

Applikation

- (t) , wobei t Term

- $(P \Rightarrow Q)$, wobei P, Q Terme

- $(\forall x P)$, wobei $x \in \mathcal{V}$ und P Term

- **Alphabet für erlaubte Symbole**

- \mathcal{V} : Variablensymbole

- **Terme**

- Variablen $x \in \mathcal{V}$

- $\lambda x.t$, wobei $x \in \mathcal{V}$ und t Term

λ -Abstraktion

- $f t$, wobei t und f Terme

Applikation

- (t) , wobei t Term

- $(P \Rightarrow Q)$, wobei P, Q Terme

- $(\forall x P)$, wobei $x \in \mathcal{V}$ und P Term

- **Konventionen**

- Applikation bindet stärker als λ -Abstraktion

- Applikation ist links-assoziativ:

$$f t_1 t_2 \hat{=} (f t_1) t_2$$

● Alphabet für erlaubte Symbole

– \mathcal{V} : Variablensymbole

● Terme

– Variablen $x \in \mathcal{V}$

– $\lambda x.t$, wobei $x \in \mathcal{V}$ und t Term

λ -Abstraktion

– $f t$, wobei t und f Terme

Applikation

– (t) , wobei t Term

– $(P \Rightarrow Q)$, wobei P, Q Terme

– $(\forall x P)$, wobei $x \in \mathcal{V}$ und P Term

● Konventionen

– Applikation bindet stärker als λ -Abstraktion

– Applikation ist links-assoziativ:

$$f t_1 t_2 \hat{=} (f t_1) t_2$$

– Notation $f(t_1 \dots t_n)$ entspricht iterierter Applikation $f t_1 \dots t_n$

LOGIK HÖHERER STUFE – (WERT-)SEMANTIK

- **α -Konversion:** Umbenennung gebundener Variablen
 - Ersetze Teilterm der Gestalt $\lambda x.t$ durch $\lambda z.t[z/x]$ (z neue Variable)
 - Ersetze Teilterm der Gestalt $\forall x P$ durch $\forall z P[z/x]$

LOGIK HÖHERER STUFE – (WERT-)SEMANTIK

- **α -Konversion:** Umbenennung gebundener Variablen
 - Ersetze Teilterm der Gestalt $\lambda x.t$ durch $\lambda z.t[z/x]$ (z neue Variable)
 - Ersetze Teilterm der Gestalt $\forall x P$ durch $\forall z P[z/x]$
 - Terme t und u sind **kongruent** (**α -konvertibel**), wenn sie auseinander durch endlich viele Umbenennungen gebundener Variablen entstehen

LOGIK HÖHERER STUFE – (WERT-)SEMANTIK

- **α -Konversion:** Umbenennung gebundener Variablen
 - Ersetze Teilterm der Gestalt $\lambda x.t$ durch $\lambda z.t[z/x]$ (z neue Variable)
 - Ersetze Teilterm der Gestalt $\forall x P$ durch $\forall z P[z/x]$
 - Terme t und u sind **kongruent** (**α -konvertibel**), wenn sie auseinander durch endlich viele Umbenennungen gebundener Variablen entstehen
- **(β) -Reduktion:** Auswertung von Termen
 - Ersetze Teilterm der Gestalt $(\lambda x.t)(s)$ (**Redex**) durch $t[s/x]$ (**Kontraktum**)

LOGIK HÖHERER STUFE – (WERT-)SEMANTIK

- **α -Konversion:** Umbenennung gebundener Variablen
 - Ersetze Teilterm der Gestalt $\lambda x.t$ durch $\lambda z.t[z/x]$ (z neue Variable)
 - Ersetze Teilterm der Gestalt $\forall x P$ durch $\forall z P[z/x]$
 - Terme t und u sind **kongruent** (**α -konvertibel**), wenn sie auseinander durch endlich viele Umbenennungen gebundener Variablen entstehen
- **(β) -Reduktion:** Auswertung von Termen
 - Ersetze Teilterm der Gestalt $(\lambda x.t)(s)$ (**Redex**) durch $t[s/x]$ (**Kontraktum**)
 - t ist **reduzierbar** auf u ($t \xrightarrow{*} u$), wenn u aus t durch endlich viele Reduktionen und Umbenennungen entsteht ($\hat{=}$ **Termersetzung**)

LOGIK HÖHERER STUFE – (WERT-)SEMANTIK

- **α -Konversion:** Umbenennung gebundener Variablen
 - Ersetze Teilterm der Gestalt $\lambda x.t$ durch $\lambda z.t[z/x]$ (z neue Variable)
 - Ersetze Teilterm der Gestalt $\forall x P$ durch $\forall z P[z/x]$
 - Terme t und u sind **kongruent** (α -konvertibel), wenn sie auseinander durch endlich viele Umbenennungen gebundener Variablen entstehen
- **(β) -Reduktion:** Auswertung von Termen
 - Ersetze Teilterm der Gestalt $(\lambda x.t)(s)$ (**Redex**) durch $t[s/x]$ (**Kontraktum**)
 - t ist **reduzierbar** auf u ($t \xrightarrow{*} u$), wenn u aus t durch endlich viele Reduktionen und Umbenennungen entsteht ($\hat{=}$ **Termersetzung**)
- **(Semantische) Gleichheit $t = s$**
 - Es gibt einen Term u mit $t \xrightarrow{*} u$ und $s \xrightarrow{*} u$

LOGIK HÖHERER STUFE – (WERT-)SEMANTIK

- **α -Konversion:** Umbenennung gebundener Variablen
 - Ersetze Teilterm der Gestalt $\lambda x.t$ durch $\lambda z.t[z/x]$ (z neue Variable)
 - Ersetze Teilterm der Gestalt $\forall x P$ durch $\forall z P[z/x]$
 - Terme t und u sind **kongruent** (α -konvertibel), wenn sie auseinander durch endlich viele Umbenennungen gebundener Variablen entstehen
- **(β) -Reduktion:** Auswertung von Termen
 - Ersetze Teilterm der Gestalt $(\lambda x.t)(s)$ (**Redex**) durch $t[s/x]$ (**Kontraktum**)
 - t ist **reduzierbar** auf u ($t \xrightarrow{*} u$), wenn u aus t durch endlich viele Reduktionen und Umbenennungen entsteht ($\hat{=}$ **Termersetzung**)
- **(Semantische) Gleichheit $t = s$**
 - Es gibt einen Term u mit $t \xrightarrow{*} u$ und $s \xrightarrow{*} u$
- **Normalform von t :** Wert eines Terms
 - Irreduzibler (Redex-freier) Term u mit $t = u$
 - Der Wert eines Terms ist eindeutig ($\xrightarrow{*}$ ist konfluent)
 - Nicht jeder Term hat einen Wert (**keine starke Normalisierbarkeit**)

LOGIK HÖHERER STUFE – (WERT-)SEMANTIK

- **α -Konversion:** Umbenennung gebundener Variablen
 - Ersetze Teilterm der Gestalt $\lambda x.t$ durch $\lambda z.t[z/x]$ (z neue Variable)
 - Ersetze Teilterm der Gestalt $\forall x P$ durch $\forall z P[z/x]$
 - Terme t und u sind **kongruent** (α -konvertibel), wenn sie auseinander durch endlich viele Umbenennungen gebundener Variablen entstehen
- **(β) -Reduktion:** Auswertung von Termen
 - Ersetze Teilterm der Gestalt $(\lambda x.t)(s)$ (**Redex**) durch $t[s/x]$ (**Kontraktum**)
 - t ist **reduzierbar** auf u ($t \xrightarrow{*} u$), wenn u aus t durch endlich viele Reduktionen und Umbenennungen entsteht ($\hat{=}$ **Termersetzung**)
- **(Semantische) Gleichheit $t = s$**
 - Es gibt einen Term u mit $t \xrightarrow{*} u$ und $s \xrightarrow{*} u$
- **Normalform von t :** Wert eines Terms
 - Irreduzibler (Redex-freier) Term u mit $t = u$
 - Der Wert eines Terms ist eindeutig ($\xrightarrow{*}$ ist konfluent)
 - Nicht jeder Term hat einen Wert (**keine starke Normalisierbarkeit**)
- **Semantik von \forall und \Rightarrow analog zur Prädikatenlogik**

DEFINITION LOGISCHER STANDARDKONZEPTE

Junktoren werden zu benutzerdefinierten Funktionen

Konjunktion $\wedge \equiv$

Disjunktion $\vee \equiv$

Falschheit $\perp \equiv$

Negation $\neg \equiv$

Existenzquantor $\exists x A \equiv$

Gleichheit $\doteq \equiv$

DEFINITION LOGISCHER STANDARDKONZEPTE

Junktoren werden zu benutzerdefinierten Funktionen

Konjunktion $\wedge \equiv \lambda A.\lambda B.\forall P (A \Rightarrow (B \Rightarrow P)) \Rightarrow P$

Disjunktion $\vee \equiv$

Falschheit $\perp \equiv$

Negation $\neg \equiv$

Existenzquantor $\exists x A \equiv$

Gleichheit $\doteq \equiv$

DEFINITION LOGISCHER STANDARDKONZEPTE

Junktoren werden zu benutzerdefinierten Funktionen

Konjunktion $\wedge \equiv \lambda A. \lambda B. \forall P (A \Rightarrow (B \Rightarrow P)) \Rightarrow P$

Disjunktion $\vee \equiv \lambda A. \lambda B. \forall P ((A \Rightarrow P) \Rightarrow (B \Rightarrow P)) \Rightarrow P$

Falschheit $\perp \equiv$

Negation $\neg \equiv$

Existenzquantor $\exists x A \equiv$

Gleichheit $\doteq \equiv$

DEFINITION LOGISCHER STANDARDKONZEPTE

Junktoren werden zu benutzerdefinierten Funktionen

Konjunktion $\wedge \equiv \lambda A. \lambda B. \forall P (A \Rightarrow (B \Rightarrow P)) \Rightarrow P$

Disjunktion $\vee \equiv \lambda A. \lambda B. \forall P ((A \Rightarrow P) \Rightarrow (B \Rightarrow P)) \Rightarrow P$

Falschheit $\perp \equiv \forall P P$ (*intuitionistisch*)

Negation $\neg \equiv$

Existenzquantor $\exists x A \equiv$

Gleichheit $\doteq \equiv$

DEFINITION LOGISCHER STANDARDKONZEPTE

Junktoren werden zu benutzerdefinierten Funktionen

Konjunktion $\wedge \equiv \lambda A. \lambda B. \forall P (A \Rightarrow (B \Rightarrow P)) \Rightarrow P$

Disjunktion $\vee \equiv \lambda A. \lambda B. \forall P ((A \Rightarrow P) \Rightarrow (B \Rightarrow P)) \Rightarrow P$

Falschheit $\perp \equiv \forall P P$ (*intuitionistisch*)

Negation $\neg \equiv \lambda A. \forall P (A \Rightarrow P)$

Existenzquantor $\exists x A \equiv$

Gleichheit $\doteq \equiv$

DEFINITION LOGISCHER STANDARDKONZEPTE

Junktoren werden zu benutzerdefinierten Funktionen

Konjunktion $\wedge \equiv \lambda A. \lambda B. \forall P (A \Rightarrow (B \Rightarrow P)) \Rightarrow P$

Disjunktion $\vee \equiv \lambda A. \lambda B. \forall P ((A \Rightarrow P) \Rightarrow (B \Rightarrow P)) \Rightarrow P$

Falschheit $\perp \equiv \forall P P$ (*intuitionistisch*)

Negation $\neg \equiv \lambda A. \forall P (A \Rightarrow P)$

Existenzquantor $\exists x A \equiv \forall P (\forall x (A \Rightarrow P)) \Rightarrow P$

Gleichheit $\doteq \equiv$

DEFINITION LOGISCHER STANDARDKONZEPTE

Junktoren werden zu benutzerdefinierten Funktionen

Konjunktion $\wedge \equiv \lambda A. \lambda B. \forall P (A \Rightarrow (B \Rightarrow P)) \Rightarrow P$

Disjunktion $\vee \equiv \lambda A. \lambda B. \forall P ((A \Rightarrow P) \Rightarrow (B \Rightarrow P)) \Rightarrow P$

Falschheit $\perp \equiv \forall P P$ (*intuitionistisch*)

Negation $\neg \equiv \lambda A. \forall P (A \Rightarrow P)$

Existenzquantor $\exists x A \equiv \forall P (\forall x (A \Rightarrow P)) \Rightarrow P$

Gleichheit $\doteq \equiv \lambda x. \lambda y. \forall P (P(x) \Rightarrow P(y))$

DEFINITION VON ZAHLEN: CHURCH NUMERALS

$$\bar{n} \quad \equiv \quad \lambda f. \lambda x. f^n x \quad \equiv \quad \lambda f. \lambda x. \underbrace{f (f \dots (fx) \dots)}_{n\text{-mal}}$$

DEFINITION VON ZAHLEN: CHURCH NUMERALS

$$\begin{aligned} \bar{n} &\equiv \lambda f. \lambda x. f^n x && \equiv \lambda f. \lambda x. \underbrace{f (f \dots (f x) \dots)}_{n\text{-mal}} \\ s &\equiv \lambda n. \lambda f. \lambda x. n f (f x) \end{aligned}$$

DEFINITION VON ZAHLEN: CHURCH NUMERALS

$$\bar{n} \equiv \lambda f. \lambda x. f^n x \quad \equiv \lambda f. \lambda x. \underbrace{f (f \dots (f x) \dots)}_{n\text{-mal}}$$

$$s \equiv \lambda n. \lambda f. \lambda x. n f (f x)$$

$$\text{add} \equiv \lambda m. \lambda n. \lambda f. \lambda x. m f (n f x)$$

DEFINITION VON ZAHLEN: CHURCH NUMERALS

$$\bar{n} \quad \equiv \quad \lambda f. \lambda x. f^n x \quad \equiv \quad \lambda f. \lambda x. \underbrace{f (f \dots (f x) \dots)}_{n\text{-mal}}$$

$$s \quad \equiv \quad \lambda n. \lambda f. \lambda x. n f (f x)$$

$$\text{add} \quad \equiv \quad \lambda m. \lambda n. \lambda f. \lambda x. m f (n f x)$$

$$\text{mul} \quad \equiv \quad \lambda m. \lambda n. \lambda f. \lambda x. m (n f) x$$

DEFINITION VON ZAHLEN: CHURCH NUMERALS

$$\begin{aligned}\bar{n} &\equiv \lambda f.\lambda x. f^n x && \equiv \lambda f.\lambda x. \underbrace{f (f \dots (f x) \dots)}_{n\text{-mal}} \\ s &\equiv \lambda n.\lambda f.\lambda x. n f (f x) \\ \text{add} &\equiv \lambda m.\lambda n.\lambda f.\lambda x. m f (n f x) \\ \text{mul} &\equiv \lambda m.\lambda n.\lambda f.\lambda x. m (n f) x \\ \text{exp} &\equiv \lambda m.\lambda n.\lambda f.\lambda x. n m f x\end{aligned}$$

DEFINITION VON ZAHLEN: CHURCH NUMERALS

$$\begin{aligned}\bar{n} &\equiv \lambda f.\lambda x. f^n x && \equiv \lambda f.\lambda x. \underbrace{f (f \dots (f x) \dots)}_{n\text{-mal}} \\ s &\equiv \lambda n.\lambda f.\lambda x. n f (f x) \\ \text{add} &\equiv \lambda m.\lambda n.\lambda f.\lambda x. m f (n f x) \\ \text{mul} &\equiv \lambda m.\lambda n.\lambda f.\lambda x. m (n f) x \\ \text{exp} &\equiv \lambda m.\lambda n.\lambda f.\lambda x. n m f x \\ \text{zero} &\equiv \lambda n. n (\lambda n.F) \top\end{aligned}$$

DEFINITION VON ZAHLEN: CHURCH NUMERALS

$$\begin{aligned}
 \bar{n} &\equiv \lambda f. \lambda x. f^n x && \equiv \lambda f. \lambda x. \underbrace{f (f \dots (f x) \dots)}_{n\text{-mal}} \\
 s &\equiv \lambda n. \lambda f. \lambda x. n f (f x) \\
 \text{add} &\equiv \lambda m. \lambda n. \lambda f. \lambda x. m f (n f x) \\
 \text{mul} &\equiv \lambda m. \lambda n. \lambda f. \lambda x. m (n f) x \\
 \text{exp} &\equiv \lambda m. \lambda n. \lambda f. \lambda x. n m f x \\
 \text{zero} &\equiv \lambda n. n (\lambda n. F) T \\
 p &\equiv \lambda n. (n (\lambda f x. \langle s, \text{let } \langle f, x \rangle = f x \text{ in } f x \rangle) \langle \lambda z. \bar{0}, \bar{0} \rangle).2
 \end{aligned}$$

DEFINITION VON ZAHLEN: CHURCH NUMERALS

$$\bar{n} \equiv \lambda f. \lambda x. f^n x \quad \equiv \lambda f. \lambda x. \underbrace{f (f \dots (f x) \dots)}_{n\text{-mal}}$$

$$s \equiv \lambda n. \lambda f. \lambda x. n f (f x)$$

$$\text{add} \equiv \lambda m. \lambda n. \lambda f. \lambda x. m f (n f x)$$

$$\text{mul} \equiv \lambda m. \lambda n. \lambda f. \lambda x. m (n f) x$$

$$\text{exp} \equiv \lambda m. \lambda n. \lambda f. \lambda x. n m f x$$

$$\text{zero} \equiv \lambda n. n (\lambda n. F) T$$

$$p \equiv \lambda n. (n (\lambda f x. \langle s, \text{let } \langle f, x \rangle = f x \text{ in } f x \rangle) \langle \lambda z. \bar{0}, \bar{0} \rangle).2$$

$$N \equiv \lambda x. \forall P (\forall y P(y) \Rightarrow P(s y)) \Rightarrow P(\bar{0}) \Rightarrow P(x)$$

$$N(x) \hat{=} x \in \mathbb{N}$$

DEFINITION VON ZAHLEN: CHURCH NUMERALS

$$\bar{n} \equiv \lambda f. \lambda x. f^n x \quad \equiv \lambda f. \lambda x. \underbrace{f (f \dots (f x) \dots)}_{n\text{-mal}}$$

$$s \equiv \lambda n. \lambda f. \lambda x. n f (f x)$$

$$\text{add} \equiv \lambda m. \lambda n. \lambda f. \lambda x. m f (n f x)$$

$$\text{mul} \equiv \lambda m. \lambda n. \lambda f. \lambda x. m (n f) x$$

$$\text{exp} \equiv \lambda m. \lambda n. \lambda f. \lambda x. n m f x$$

$$\text{zero} \equiv \lambda n. n (\lambda n. F) T$$

$$p \equiv \lambda n. (n (\lambda f x. \langle s, \text{let } \langle f, x \rangle = f x \text{ in } f x \rangle) \langle \lambda z. \bar{0}, \bar{0} \rangle). 2$$

$$N \equiv \lambda x. \forall P (\forall y P(y) \Rightarrow P(s y)) \Rightarrow P(\bar{0}) \Rightarrow P(x)$$

$$N(x) \hat{=} x \in \mathbb{N}$$

Abkürzungen

$$T \equiv \lambda x. \lambda y. x$$

$$F \equiv \lambda x. \lambda y. y$$

$$\text{if } b \text{ then } s \text{ else } t \equiv b s t$$

$$\langle s, t \rangle \equiv \lambda p. p s t$$

$$\text{let } \langle x, y \rangle = \text{pair in } t \equiv \text{pair } (\lambda x. \lambda y. t)$$

CHURCH NUMERALS: AUSWERTUNG VON FUNKTIONEN

$$s \bar{n} \quad \equiv \quad (\lambda n. \lambda f. \lambda x. n f (f x)) (\lambda f. \lambda x. f^n x)$$

CHURCH NUMERALS: AUSWERTUNG VON FUNKTIONEN

$$\begin{aligned} s \bar{n} &\equiv (\lambda n. \lambda f. \lambda x. n f (f x)) (\lambda f. \lambda x. f^n x) \\ &\longrightarrow \lambda f. \lambda x. (\lambda f. \lambda x. f^n x) f (f x) \end{aligned}$$

CHURCH NUMERALS: AUSWERTUNG VON FUNKTIONEN

$$\begin{aligned} s \bar{n} &\equiv (\lambda n. \lambda f. \lambda x. n f (f x)) (\lambda f. \lambda x. f^n x) \\ &\longrightarrow \lambda f. \lambda x. (\lambda f. \lambda x. f^n x) f (f x) \\ &\longrightarrow \lambda f. \lambda x. (\lambda x. f^n x) (f x) \end{aligned}$$

CHURCH NUMERALS: AUSWERTUNG VON FUNKTIONEN

$$\begin{aligned} s \bar{n} &\equiv (\lambda n. \lambda f. \lambda x. n f (f x)) (\lambda f. \lambda x. f^n x) \\ &\longrightarrow \lambda f. \lambda x. (\lambda f. \lambda x. f^n x) f (f x) \\ &\longrightarrow \lambda f. \lambda x. (\lambda x. f^n x) (f x) \\ &\longrightarrow \lambda f. \lambda x. f^n (f x) \end{aligned}$$

CHURCH NUMERALS: AUSWERTUNG VON FUNKTIONEN

$$\begin{aligned} s \bar{n} &\equiv (\lambda n. \lambda f. \lambda x. n f (f x)) (\lambda f. \lambda x. f^n x) \\ &\longrightarrow \lambda f. \lambda x. (\lambda f. \lambda x. f^n x) f (f x) \\ &\longrightarrow \lambda f. \lambda x. (\lambda x. f^n x) (f x) \\ &\longrightarrow \lambda f. \lambda x. f^n (f x) \\ &\longrightarrow \lambda f. \lambda x. f^{n+1} x \end{aligned}$$

CHURCH NUMERALS: AUSWERTUNG VON FUNKTIONEN

$$\begin{aligned} s \bar{n} &\equiv (\lambda n. \lambda f. \lambda x. n f (f x)) (\lambda f. \lambda x. f^n x) \\ &\longrightarrow \lambda f. \lambda x. (\lambda f. \lambda x. f^n x) f (f x) \\ &\longrightarrow \lambda f. \lambda x. (\lambda x. f^n x) (f x) \\ &\longrightarrow \lambda f. \lambda x. f^n (f x) \\ &\longrightarrow \lambda f. \lambda x. f^{n+1} x \qquad \equiv \overline{n+1} \end{aligned}$$

CHURCH NUMERALS: AUSWERTUNG VON FUNKTIONEN

$$\begin{aligned} s \bar{n} &\equiv (\lambda n. \lambda f. \lambda x. n f (f x)) (\lambda f. \lambda x. f^n x) \\ &\longrightarrow \lambda f. \lambda x. (\lambda f. \lambda x. f^n x) f (f x) \\ &\longrightarrow \lambda f. \lambda x. (\lambda x. f^n x) (f x) \\ &\longrightarrow \lambda f. \lambda x. f^n (f x) \\ &\longrightarrow \lambda f. \lambda x. f^{n+1} x \qquad \equiv \overline{n+1} \end{aligned}$$

$$\text{add } \bar{m} \bar{n} \equiv (\lambda m. \lambda n. \lambda f. \lambda x. m f (n f x)) \bar{m} \bar{n}$$

CHURCH NUMERALS: AUSWERTUNG VON FUNKTIONEN

$$\begin{aligned} s \bar{n} &\equiv (\lambda n. \lambda f. \lambda x. n f (f x)) (\lambda f. \lambda x. f^n x) \\ &\longrightarrow \lambda f. \lambda x. (\lambda f. \lambda x. f^n x) f (f x) \\ &\longrightarrow \lambda f. \lambda x. (\lambda x. f^n x) (f x) \\ &\longrightarrow \lambda f. \lambda x. f^n (f x) \\ &\longrightarrow \lambda f. \lambda x. f^{n+1} x \qquad \equiv \overline{n+1} \end{aligned}$$

$$\begin{aligned} \text{add } \bar{m} \bar{n} &\equiv (\lambda m. \lambda n. \lambda f. \lambda x. m f (n f x)) \bar{m} \bar{n} \\ &\longrightarrow (\lambda n. \lambda f. \lambda x. \bar{m} f (n f x)) \bar{n} \end{aligned}$$

CHURCH NUMERALS: AUSWERTUNG VON FUNKTIONEN

$$\begin{aligned} s \bar{n} &\equiv (\lambda n. \lambda f. \lambda x. n f (f x)) (\lambda f. \lambda x. f^n x) \\ &\longrightarrow \lambda f. \lambda x. (\lambda f. \lambda x. f^n x) f (f x) \\ &\longrightarrow \lambda f. \lambda x. (\lambda x. f^n x) (f x) \\ &\longrightarrow \lambda f. \lambda x. f^n (f x) \\ &\longrightarrow \lambda f. \lambda x. f^{n+1} x \qquad \equiv \overline{n+1} \end{aligned}$$

$$\begin{aligned} \text{add } \bar{m} \bar{n} &\equiv (\lambda m. \lambda n. \lambda f. \lambda x. m f (n f x)) \bar{m} \bar{n} \\ &\longrightarrow (\lambda n. \lambda f. \lambda x. \bar{m} f (n f x)) \bar{n} \\ &\longrightarrow \lambda f. \lambda x. \bar{m} f (\bar{n} f x) \end{aligned}$$

CHURCH NUMERALS: AUSWERTUNG VON FUNKTIONEN

$$\begin{aligned} s \bar{n} &\equiv (\lambda n. \lambda f. \lambda x. n f (f x)) (\lambda f. \lambda x. f^n x) \\ &\longrightarrow \lambda f. \lambda x. (\lambda f. \lambda x. f^n x) f (f x) \\ &\longrightarrow \lambda f. \lambda x. (\lambda x. f^n x) (f x) \\ &\longrightarrow \lambda f. \lambda x. f^n (f x) \\ &\longrightarrow \lambda f. \lambda x. f^{n+1} x \qquad \equiv \overline{n+1} \end{aligned}$$

$$\begin{aligned} \text{add } \bar{m} \bar{n} &\equiv (\lambda m. \lambda n. \lambda f. \lambda x. m f (n f x)) \bar{m} \bar{n} \\ &\longrightarrow (\lambda n. \lambda f. \lambda x. \bar{m} f (n f x)) \bar{n} \\ &\longrightarrow \lambda f. \lambda x. \bar{m} f (\bar{n} f x) \\ &\equiv \lambda f. \lambda x. (\lambda f. \lambda x. f^m x) f (\bar{n} f x) \end{aligned}$$

CHURCH NUMERALS: AUSWERTUNG VON FUNKTIONEN

$$\begin{aligned} s \bar{n} &\equiv (\lambda n. \lambda f. \lambda x. n f (f x)) (\lambda f. \lambda x. f^n x) \\ &\longrightarrow \lambda f. \lambda x. (\lambda f. \lambda x. f^n x) f (f x) \\ &\longrightarrow \lambda f. \lambda x. (\lambda x. f^n x) (f x) \\ &\longrightarrow \lambda f. \lambda x. f^n (f x) \\ &\longrightarrow \lambda f. \lambda x. f^{n+1} x \qquad \equiv \overline{n+1} \end{aligned}$$

$$\begin{aligned} \text{add } \bar{m} \bar{n} &\equiv (\lambda m. \lambda n. \lambda f. \lambda x. m f (n f x)) \bar{m} \bar{n} \\ &\longrightarrow (\lambda n. \lambda f. \lambda x. \bar{m} f (n f x)) \bar{n} \\ &\longrightarrow \lambda f. \lambda x. \bar{m} f (\bar{n} f x) \\ &\equiv \lambda f. \lambda x. (\lambda f. \lambda x. f^m x) f (\bar{n} f x) \\ &\longrightarrow \lambda f. \lambda x. (\lambda x. f^m x) (\bar{n} f x) \end{aligned}$$

CHURCH NUMERALS: AUSWERTUNG VON FUNKTIONEN

$$\begin{aligned} s \bar{n} &\equiv (\lambda n. \lambda f. \lambda x. n f (f x)) (\lambda f. \lambda x. f^n x) \\ &\longrightarrow \lambda f. \lambda x. (\lambda f. \lambda x. f^n x) f (f x) \\ &\longrightarrow \lambda f. \lambda x. (\lambda x. f^n x) (f x) \\ &\longrightarrow \lambda f. \lambda x. f^n (f x) \\ &\longrightarrow \lambda f. \lambda x. f^{n+1} x \qquad \equiv \overline{n+1} \end{aligned}$$

$$\begin{aligned} \text{add } \bar{m} \bar{n} &\equiv (\lambda m. \lambda n. \lambda f. \lambda x. m f (n f x)) \bar{m} \bar{n} \\ &\longrightarrow (\lambda n. \lambda f. \lambda x. \bar{m} f (n f x)) \bar{n} \\ &\longrightarrow \lambda f. \lambda x. \bar{m} f (\bar{n} f x) \\ &\equiv \lambda f. \lambda x. (\lambda f. \lambda x. f^m x) f (\bar{n} f x) \\ &\longrightarrow \lambda f. \lambda x. (\lambda x. f^m x) (\bar{n} f x) \\ &\longrightarrow \lambda f. \lambda x. f^m (\bar{n} f x) \end{aligned}$$

CHURCH NUMERALS: AUSWERTUNG VON FUNKTIONEN

$$\begin{aligned}
 s \bar{n} &\equiv (\lambda n. \lambda f. \lambda x. n f (f x)) (\lambda f. \lambda x. f^n x) \\
 &\longrightarrow \lambda f. \lambda x. (\lambda f. \lambda x. f^n x) f (f x) \\
 &\longrightarrow \lambda f. \lambda x. (\lambda x. f^n x) (f x) \\
 &\longrightarrow \lambda f. \lambda x. f^n (f x) \\
 &\longrightarrow \lambda f. \lambda x. f^{n+1} x \qquad \equiv \overline{n+1}
 \end{aligned}$$

$$\begin{aligned}
 \text{add } \bar{m} \bar{n} &\equiv (\lambda m. \lambda n. \lambda f. \lambda x. m f (n f x)) \bar{m} \bar{n} \\
 &\longrightarrow (\lambda n. \lambda f. \lambda x. \bar{m} f (n f x)) \bar{n} \\
 &\longrightarrow \lambda f. \lambda x. \bar{m} f (\bar{n} f x) \\
 &\equiv \lambda f. \lambda x. (\lambda f. \lambda x. f^m x) f (\bar{n} f x) \\
 &\longrightarrow \lambda f. \lambda x. (\lambda x. f^m x) (\bar{n} f x) \\
 &\longrightarrow \lambda f. \lambda x. f^m (\bar{n} f x) \\
 &\equiv \lambda f. \lambda x. f^m ((\lambda f. \lambda x. f^n x) f x)
 \end{aligned}$$

CHURCH NUMERALS: AUSWERTUNG VON FUNKTIONEN

$$\begin{aligned}
 s \bar{n} &\equiv (\lambda n. \lambda f. \lambda x. n f (f x)) (\lambda f. \lambda x. f^n x) \\
 &\longrightarrow \lambda f. \lambda x. (\lambda f. \lambda x. f^n x) f (f x) \\
 &\longrightarrow \lambda f. \lambda x. (\lambda x. f^n x) (f x) \\
 &\longrightarrow \lambda f. \lambda x. f^n (f x) \\
 &\longrightarrow \lambda f. \lambda x. f^{n+1} x \qquad \equiv \overline{n+1}
 \end{aligned}$$

$$\begin{aligned}
 \text{add } \bar{m} \bar{n} &\equiv (\lambda m. \lambda n. \lambda f. \lambda x. m f (n f x)) \bar{m} \bar{n} \\
 &\longrightarrow (\lambda n. \lambda f. \lambda x. \bar{m} f (n f x)) \bar{n} \\
 &\longrightarrow \lambda f. \lambda x. \bar{m} f (\bar{n} f x) \\
 &\equiv \lambda f. \lambda x. (\lambda f. \lambda x. f^m x) f (\bar{n} f x) \\
 &\longrightarrow \lambda f. \lambda x. (\lambda x. f^m x) (\bar{n} f x) \\
 &\longrightarrow \lambda f. \lambda x. f^m (\bar{n} f x) \\
 &\equiv \lambda f. \lambda x. f^m ((\lambda f. \lambda x. f^n x) f x) \\
 &\longrightarrow \lambda f. \lambda x. f^m ((\lambda x. f^n x) x)
 \end{aligned}$$

CHURCH NUMERALS: AUSWERTUNG VON FUNKTIONEN

$$\begin{aligned}
 s \bar{n} &\equiv (\lambda n. \lambda f. \lambda x. n f (f x)) (\lambda f. \lambda x. f^n x) \\
 &\longrightarrow \lambda f. \lambda x. (\lambda f. \lambda x. f^n x) f (f x) \\
 &\longrightarrow \lambda f. \lambda x. (\lambda x. f^n x) (f x) \\
 &\longrightarrow \lambda f. \lambda x. f^n (f x) \\
 &\longrightarrow \lambda f. \lambda x. f^{n+1} x \qquad \equiv \overline{n+1}
 \end{aligned}$$

$$\begin{aligned}
 \text{add } \bar{m} \bar{n} &\equiv (\lambda m. \lambda n. \lambda f. \lambda x. m f (n f x)) \bar{m} \bar{n} \\
 &\longrightarrow (\lambda n. \lambda f. \lambda x. \bar{m} f (n f x)) \bar{n} \\
 &\longrightarrow \lambda f. \lambda x. \bar{m} f (\bar{n} f x) \\
 &\equiv \lambda f. \lambda x. (\lambda f. \lambda x. f^m x) f (\bar{n} f x) \\
 &\longrightarrow \lambda f. \lambda x. (\lambda x. f^m x) (\bar{n} f x) \\
 &\longrightarrow \lambda f. \lambda x. f^m (\bar{n} f x) \\
 &\equiv \lambda f. \lambda x. f^m ((\lambda f. \lambda x. f^n x) f x) \\
 &\longrightarrow \lambda f. \lambda x. f^m ((\lambda x. f^n x) x) \\
 &\longrightarrow \lambda f. \lambda x. f^m (f^n x)
 \end{aligned}$$

CHURCH NUMERALS: AUSWERTUNG VON FUNKTIONEN

$$\begin{aligned}
 s \bar{n} &\equiv (\lambda n. \lambda f. \lambda x. n f (f x)) (\lambda f. \lambda x. f^n x) \\
 &\longrightarrow \lambda f. \lambda x. (\lambda f. \lambda x. f^n x) f (f x) \\
 &\longrightarrow \lambda f. \lambda x. (\lambda x. f^n x) (f x) \\
 &\longrightarrow \lambda f. \lambda x. f^n (f x) \\
 &\longrightarrow \lambda f. \lambda x. f^{n+1} x \qquad \equiv \overline{n+1}
 \end{aligned}$$

$$\begin{aligned}
 \text{add } \bar{m} \bar{n} &\equiv (\lambda m. \lambda n. \lambda f. \lambda x. m f (n f x)) \bar{m} \bar{n} \\
 &\longrightarrow (\lambda n. \lambda f. \lambda x. \bar{m} f (n f x)) \bar{n} \\
 &\longrightarrow \lambda f. \lambda x. \bar{m} f (\bar{n} f x) \\
 &\equiv \lambda f. \lambda x. (\lambda f. \lambda x. f^m x) f (\bar{n} f x) \\
 &\longrightarrow \lambda f. \lambda x. (\lambda x. f^m x) (\bar{n} f x) \\
 &\longrightarrow \lambda f. \lambda x. f^m (\bar{n} f x) \\
 &\equiv \lambda f. \lambda x. f^m ((\lambda f. \lambda x. f^n x) f x) \\
 &\longrightarrow \lambda f. \lambda x. f^m ((\lambda x. f^n x) x) \\
 &\longrightarrow \lambda f. \lambda x. f^m (f^n x) \\
 &\longrightarrow \lambda f. \lambda x. f^{m+n} x
 \end{aligned}$$

CHURCH NUMERALS: AUSWERTUNG VON FUNKTIONEN

$$\begin{aligned}
 s \bar{n} &\equiv (\lambda n. \lambda f. \lambda x. n f (f x)) (\lambda f. \lambda x. f^n x) \\
 &\longrightarrow \lambda f. \lambda x. (\lambda f. \lambda x. f^n x) f (f x) \\
 &\longrightarrow \lambda f. \lambda x. (\lambda x. f^n x) (f x) \\
 &\longrightarrow \lambda f. \lambda x. f^n (f x) \\
 &\longrightarrow \lambda f. \lambda x. f^{n+1} x \qquad \equiv \overline{n+1}
 \end{aligned}$$

$$\begin{aligned}
 \text{add } \bar{m} \bar{n} &\equiv (\lambda m. \lambda n. \lambda f. \lambda x. m f (n f x)) \bar{m} \bar{n} \\
 &\longrightarrow (\lambda n. \lambda f. \lambda x. \bar{m} f (n f x)) \bar{n} \\
 &\longrightarrow \lambda f. \lambda x. \bar{m} f (\bar{n} f x) \\
 &\equiv \lambda f. \lambda x. (\lambda f. \lambda x. f^m x) f (\bar{n} f x) \\
 &\longrightarrow \lambda f. \lambda x. (\lambda x. f^m x) (\bar{n} f x) \\
 &\longrightarrow \lambda f. \lambda x. f^m (\bar{n} f x) \\
 &\equiv \lambda f. \lambda x. f^m ((\lambda f. \lambda x. f^n x) f x) \\
 &\longrightarrow \lambda f. \lambda x. f^m ((\lambda x. f^n x) x) \\
 &\longrightarrow \lambda f. \lambda x. f^m (f^n x) \\
 &\longrightarrow \lambda f. \lambda x. f^{m+n} x \qquad \equiv \overline{m+n}
 \end{aligned}$$

BEWEISFÜHRUNG IN LOGIK HÖHERER STUFE

Extensionsverfahren ähnlich wie bei Prädikatenlogik

BEWEISFÜHRUNG IN LOGIK HÖHERER STUFE

Extensionsverfahren ähnlich wie bei Prädikatenlogik

- **Erweiterte Matrixcharakterisierung der Gültigkeit**
 - F ist gültig gdw. alle Pfade durch F komplementär

Extensionsverfahren ähnlich wie bei Prädikatenlogik

- **Erweiterte Matrixcharakterisierung der Gültigkeit**
 - F ist gültig gdw. alle Pfade durch F komplementär
 - Komplementarität mit erweitertem Substitutionsbegriff
 - Prädikats- und Funktionssymbole dürfen ersetzt werden
 - Konnektierte Terme müssen nur semantisch gleich sein

Extensionsverfahren ähnlich wie bei Prädikatenlogik

- **Erweiterte Matrixcharakterisierung der Gültigkeit**
 - F ist gültig gdw. alle Pfade durch F komplementär
 - Komplementarität mit erweitertem Substitutionsbegriff
 - Prädikats- und Funktionssymbole dürfen ersetzt werden
 - Konnektierte Terme müssen nur semantisch gleich sein
- **Erweitertes Beweissuchverfahren** → TPS (Andrews)
 - Konnektionen-orientiertes Pfadüberprüfungsverfahren

Extensionsverfahren ähnlich wie bei Prädikatenlogik

- **Erweiterte Matrixcharakterisierung der Gültigkeit**
 - F ist gültig gdw. alle Pfade durch F komplementär
 - Komplementarität mit erweitertem Substitutionsbegriff
 - Prädikats- und Funktionssymbole dürfen ersetzt werden
 - Konnektierte Terme müssen nur semantisch gleich sein
- **Erweitertes Beweissuchverfahren** → TPS (Andrews)
 - Konnektionen-orientiertes Pfadüberprüfungsverfahren
 - Komplementaritätstest mit Unifikation höherer Stufe
 - i.A. unentscheidbar (!) und erheblich komplizierteres Verfahren → Huet

Extensionsverfahren ähnlich wie bei Prädikatenlogik

- **Erweiterte Matrixcharakterisierung der Gültigkeit**
 - F ist gültig gdw. alle Pfade durch F komplementär
 - Komplementarität mit erweitertem Substitutionsbegriff
 - Prädikats- und Funktionssymbole dürfen ersetzt werden
 - Konnektierte Terme müssen nur semantisch gleich sein
- **Erweitertes Beweissuchverfahren** → TPS (Andrews)
 - Konnektionen-orientiertes Pfadüberprüfungsverfahren
 - Komplementaritätstest mit Unifikation höherer Stufe
 - i.A. unentscheidbar (!) und erheblich komplizierteres Verfahren → Huet
- **Beweise sind i.A. sehr kurz und elegant**
 - Aber erheblich schwerer zu finden

BEWEIS DER GRUNDEIGENSCHAFTEN DER GLEICHHEIT

$$\dot{=} \equiv \lambda x.\lambda y.\forall P(P(x) \Rightarrow P(y))$$

- Reflexivität: $a \dot{=} a$

$$\left[Pa^T \quad Pa^F \right]$$

BEWEIS DER GRUNDEIGENSCHAFTEN DER GLEICHHEIT

$$\dot{=} \equiv \lambda x. \lambda y. \forall P (P(x) \Rightarrow P(y))$$

- **Reflexivität:** $a \dot{=} a$

$$\left[\overset{\text{red arc}}{Pa^T \quad Pa^F} \right]$$

- **Kommutativität:** $a \dot{=} b \Rightarrow b \dot{=} a$

$$\left[\overset{\text{red arc}}{Xa^F \quad Pb^T \quad Pa^F} \right]$$

$$[\lambda z. \neg Pz / X]$$

BEWEIS DER GRUNDEIGENSCHAFTEN DER GLEICHHEIT

$$\dot{=} \equiv \lambda x.\lambda y.\forall P(P(x) \Rightarrow P(y))$$

- **Reflexivität:** $a \dot{=} a$

$$\left[\overbrace{Pa^T \quad Pa^F} \right]$$

- **Kommutativität:** $a \dot{=} b \Rightarrow b \dot{=} a$

$$\left[\begin{array}{ccc} \overbrace{Xa^F \quad Pb^T \quad Pa^F} \\ Xb^T \end{array} \right]$$

$[\lambda z.\neg Pz/X]$

- **Transitivität:** $a \dot{=} b \wedge b \dot{=} c \Rightarrow a \dot{=} c$

$$\left[\begin{array}{cccc} \overbrace{Xa^F \quad Yb^F \quad Pa^T \quad Pc^F} \\ Xb^T \quad Yc^T \end{array} \right]$$

$[P/X, P/Y]$

BEWEIS DER GRUNDEIGENSCHAFTEN DER GLEICHHEIT

$$\dot{=} \equiv \lambda x. \lambda y. \forall P (P(x) \Rightarrow P(y))$$

- Reflexivität: $a \dot{=} a$

$$\left[\begin{array}{cc} \text{---} & \\ Pa^T & Pa^F \end{array} \right]$$

- Kommutativität: $a \dot{=} b \Rightarrow b \dot{=} a$

$$\left[\begin{array}{ccc} \text{---} & & \\ Xa^F & Pb^T & Pa^F \\ Xb^T & & \end{array} \right]$$

$[\lambda z. \neg Pz / X]$

- Transitivität: $a \dot{=} b \wedge b \dot{=} c \Rightarrow a \dot{=} c$

$$\left[\begin{array}{cccc} \text{---} & & & \\ Xa^F & Yb^F & Pa^T & Pc^F \\ Xb^T & Yc^T & & \end{array} \right]$$

$[P/X, P/Y]$

- Substitutivität: $Pa \wedge a \dot{=} b \Rightarrow Pb$

$$\left[\begin{array}{ccc} \text{---} & & \\ Pa^T & Xa^F & Pb^F \\ & Xb^T & \end{array} \right]$$

$[P/X]$

INTUITIONISTISCHE LOGIK HÖHERER STUFE

- **Unterstützt formale Manipulation von Algorithmen**
 - Synthese aus Spezifikationen, Optimierung, Verifikation, ...
 - Einheitliche Sprache für Spezifikation, Programmierung, Deduktion...

INTUITIONISTISCHE LOGIK HÖHERER STUFE

- **Unterstützt formale Manipulation von Algorithmen**
 - Synthese aus Spezifikationen, Optimierung, Verifikation, ...
 - Einheitliche Sprache für Spezifikation, Programmierung, Deduktion...
- **(Zur Zeit noch) viele verschiedene Formulierungen**
 - Martin-Löf'sche Typentheorie (Computational Type Theory)
 - Kalkül der Konstruktionen
 - System F
 - LCF (Logik berechenbarer Funktionen)
 -

INTUITIONISTISCHE LOGIK HÖHERER STUFE

- **Unterstützt formale Manipulation von Algorithmen**
 - Synthese aus Spezifikationen, Optimierung, Verifikation, ...
 - Einheitliche Sprache für Spezifikation, Programmierung, Deduktion...
- **(Zur Zeit noch) viele verschiedene Formulierungen**
 - Martin-Löf'sche Typentheorie (Computational Type Theory)
 - Kalkül der Konstruktionen
 - System F
 - LCF (Logik berechenbarer Funktionen)
 - \vdots
- **Beweissysteme interaktiv mit taktischer Steuerung**
 - **AUTOMATH** (historischer Vorläufer)
 - **Nuprl, MetaPRL** (Computational Type Theory)
 - **Alf / Agda, ...** (Martin-Löf Typentheorie)
 - **Legu, Coq** (Kalkül der Konstruktionen)
 - **Cambridge LCF**

INTUITIONISTISCHE LOGIK HÖHERER STUFE

- **Unterstützt formale Manipulation von Algorithmen**
 - Synthese aus Spezifikationen, Optimierung, Verifikation, ...
 - Einheitliche Sprache für Spezifikation, Programmierung, Deduktion...
- **(Zur Zeit noch) viele verschiedene Formulierungen**
 - Martin-Löf'sche Typentheorie (Computational Type Theory)
 - Kalkül der Konstruktionen
 - System F
 - LCF (Logik berechenbarer Funktionen)
 - \vdots
- **Beweissysteme interaktiv mit taktischer Steuerung**
 - **AUTOMATH** (historischer Vorläufer)
 - **Nuprl, MetaPRL** (Computational Type Theory)
 - **Alf / Agda, ...** (Martin-Löf Typentheorie)
 - **Legu, Coq** (Kalkül der Konstruktionen)
 - **Cambridge LCF**
- **Viele erfolgreiche Anwendungen**