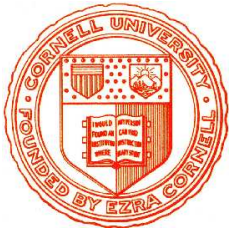
Automatisierte Logik und Programmierung

Einheit 3



Refinement Logic

1. **Inferenzsystem**
   - Aussagenlogischer Kalkül
   - Substitution und Quantorenregeln
   - Gleichheitsbehandlung
2. **Korrektheit und Vollständigkeit**
3. **Beweismethodik und Strategien**

Beweisführung durch schrittweise Verfeinerung

- **Analytischer Kalkül**
  - Zielorientiertes Vorgehen durch Top-Down Entwicklung von Beweisen
  - Inferenzregeln zerlegen Beweisziel in "leichtere" Teilaufgaben

Beweisführung durch schrittweise Verfeinerung

- **Analytischer Kalkül**
  - Zielorientiertes Vorgehen durch Top-Down Entwicklung von Beweisen
  - Inferenzregeln zerlegen Beweisziel in "leichtere" Teilaufgaben
- **Sequenzenkalkül**
  - In jedem Beweis(teil-)ziel sind alle Annahmen explizit genannt
  - Technisch: Schließen über Sequenzen (Mengen von Formeln)
  - $H_1,..,H_n \vdash C$ heißt: *Konklusion C folgt aus Hypothesen $H_1,..,H_n$*"

Beweisführung durch schrittweise Verfeinerung

- **Analytischer Kalkül**
  - Zielorientiertes Vorgehen durch Top-Down Entwicklung von Beweisen
  - Inferenzregeln zerlegen Beweisziel in "leichtere" Teilaufgaben
- **Sequenzenkalkül**
  - In jedem Beweis(teil-)ziel sind alle Annahmen explizit genannt
  - Technisch: Schließen über Sequenzen (Mengen von Formeln)
  - $H_1,..,H_n \vdash C$ heißt: *Konklusion C folgt aus Hypothesen $H_1,..,H_n$*"
- **Computergerechte Formalisierung**
  - Hypothesen dargestellt als Liste von Formeln
  - Regeln greifen auf konkrete Hypothesen durch Parameter zu

Beweisführung durch schrittweise Verfeinerung

- **Analytischer Kalkül**
  - Zielorientiertes Vorgehen durch Top-Down Entwicklung von Beweisen
  - Inferenzregeln zerlegen Beweisziel in "leichtere" Teilaufgaben
- **Sequenzenkalkül**
  - In jedem Beweis(teil-)ziel sind alle Annahmen explizit genannt
  - Technisch: Schließen über Sequenzen (Mengen von Formeln)
  - $H_1,..,H_n \vdash C$ heißt: *Konklusion C folgt aus Hypothesen $H_1,..,H_n$*"
- **Computergerechte Formalisierung**
  - Hypothesen dargestellt als Liste von Formeln
  - Regeln greifen auf konkrete Hypothesen durch Parameter zu
- **Konstruktive Auslegung**
  - Inferenzregeln zerlegen zusammengesetzte Formeln in ihre Bestandteile bis Konklusion C direkt aus einer Annahme H_i folgt (d.h. $C = H_i$)
  - Klassisches Schließen ermöglicht durch Hinzunahme einer Sonderregel

Refinement Logic: Grundkonzepte präzisiert

- **Sequenz:** $\boldsymbol{H_1, \ldots, H_n \vdash C}$
  - $\Gamma = H_1, \ldots, H_n$ Hypothesenliste,
  - H_i ist Formel oder Deklaration $x : T$ $\quad (x \in \mathcal{V}, T \in \mathcal{T})$,
  - C ist einzelne Formel (Konklusion)

Refinement Logic: Grundkonzepte präzisiert

- **Sequenz:** $\boldsymbol{H_1, \ldots, H_n \vdash C}$
  - $\Gamma = H_1, \ldots, H_n$ Hypothesenliste,
  - H_i ist Formel oder Deklaration $x : T$ $\quad (x \in \mathcal{V}, T \in \mathcal{T})$,
  - C ist einzelne Formel (Konklusion)
  - Semantik entspricht $H_1, \ldots, H_n \Rightarrow C$
  - Homomorphe Fortsetzung von Interpretationen

$$\iota(H_1, \ldots, H_n \vdash C) = \begin{cases} \mathsf{wahr} & \text{falls aus } \iota(H_1) = \mathsf{wahr} \\ & \text{und } \ldots \iota(H_n) = \mathsf{wahr} \\ & \text{immer } \iota(C) = \mathsf{wahr} \text{ folgt} \\ \mathsf{falsch} & \text{sonst} \end{cases}$$

  - Begriffe Modell, Gültigkeit, Erfüllbarkeit analog

Refinement Logic: Grundkonzepte präzisiert

- **Sequenz:** $\boldsymbol{H_1, \ldots, H_n \vdash C}$
  - $\Gamma = H_1, \ldots, H_n$ Hypothesenliste,
  - H_i ist Formel oder Deklaration $x : T$ $\quad (x \in \mathcal{V}, T \in \mathcal{T})$,
  - C ist einzelne Formel (Konklusion)
  - Semantik entspricht $H_1, \ldots, H_n \Rightarrow C$
  - Homomorphe Fortsetzung von Interpretationen

$$\iota(H_1, \ldots, H_n \vdash C) = \begin{cases} \mathsf{wahr} & \text{falls aus } \iota(H_1) = \mathsf{wahr} \\ & \text{und } \ldots \iota(H_n) = \mathsf{wahr} \\ & \text{immer } \iota(C) = \mathsf{wahr} \text{ folgt} \\ \mathsf{falsch} & \text{sonst} \end{cases}$$

  - Begriffe Modell, Gültigkeit, Erfüllbarkeit analog

- **Inferenzregel:** Abbildung von Beweisziel in Teilziele
  - Beweisziel: einzelne Sequenz, die zu beweisen ist
  - Teilziele: endliche (evtl. leere) Liste von Sequenzen, die nach Regelanwendung noch zu zeigen sind
  - Zugriff auf Hypothesen durch *Parameter*

$$\frac{\Gamma_1 \vdash C_1 \quad \ldots \quad \Gamma_n \vdash C_n}{\Gamma \vdash C} \;\texttt{BY}\; \mathit{rule}$$

$$\frac{\Gamma, A, B, \Delta \vdash C}{\Gamma, A \wedge B, \Delta \vdash C} \;\texttt{BY andE}\; i$$

Beweisregeln folgen direkt aus Semantik

- **Zerlegung einer Konjunktion in der Konklusion**
  - $A \wedge B$ ist gültig wenn A gültig ist und B gültig ist
  - Um $A \wedge B$ zu zeigen, genügt es A zu beweisen und B zu beweisen
  - Die Hypothesenliste Γ ändert sich dabei nicht

Beweisregeln folgen direkt aus Semantik

- **Zerlegung einer Konjunktion in der Konklusion**
  - $A \wedge B$ ist gültig wenn A gültig ist und B gültig ist
  - Um $A \wedge B$ zu zeigen, genügt es A zu beweisen und B zu beweisen
  - Die Hypothesenliste Γ ändert sich dabei nicht

$$\Gamma \vdash A \wedge B \quad \texttt{BY andI}$$
$$\Gamma \vdash A$$
$$\Gamma \vdash B$$

Beweisregeln folgen direkt aus Semantik

- **Zerlegung einer Konjunktion in der Konklusion**
  - $A \wedge B$ ist gültig wenn A gültig ist und B gültig ist
  - Um $A \wedge B$ zu zeigen, genügt es A zu beweisen und B zu beweisen
  - Die Hypothesenliste Γ ändert sich dabei nicht

$$\Gamma \vdash A \wedge B \quad \texttt{BY andI}$$
$$\Gamma \vdash A$$
$$\Gamma \vdash B$$

- **Zerlegung einer Konjunktion in den Hypothesen**
  - Wenn eine Aussage C aus den Annahmen A und B folgt, dann folgt sie auch aus $A \wedge B$
  - Um $A \wedge B \vdash C$ zu zeigen, genügt es $A, B \vdash C$ zu beweisen
  - Die restlichen Hypothesen ändern sich nicht
  - Beweisregel muß Position von $A \wedge B$ angeben

Beweisregeln folgen direkt aus Semantik

- **Zerlegung einer Konjunktion in der Konklusion**
  - $A \wedge B$ ist gültig wenn A gültig ist und B gültig ist
  - Um $A \wedge B$ zu zeigen, genügt es A zu beweisen und B zu beweisen
  - Die Hypothesenliste Γ ändert sich dabei nicht

$$\frac{\Gamma \vdash A \wedge B \quad \texttt{BY andI}}{\begin{array}{l}\Gamma \vdash A \\ \Gamma \vdash B\end{array}}$$

- **Zerlegung einer Konjunktion in den Hypothesen**
  - Wenn eine Aussage C aus den Annahmen A und B folgt, dann folgt sie auch aus $A \wedge B$
  - Um $A \wedge B \vdash C$ zu zeigen, genügt es $A, B \vdash C$ zu beweisen
  - Die restlichen Hypothesen ändern sich nicht
  - Beweisregel muß Position von $A \wedge B$ angeben

$$\frac{\Gamma, A \wedge B, \Delta \vdash C \quad \texttt{BY andE } i}{\Gamma, A, B, \Delta \vdash C}$$

Beweisregeln folgen direkt aus Semantik

- **Zerlegung einer Konjunktion in der Konklusion**
  - $A \wedge B$ ist gültig wenn A gültig ist und B gültig ist
  - Um $A \wedge B$ zu zeigen, genügt es A zu beweisen und B zu beweisen
  - Die Hypothesenliste Γ ändert sich dabei nicht

$$\frac{\Gamma \vdash A \wedge B \quad \texttt{BY andI}}{\Gamma \vdash A \quad \Gamma \vdash B}$$

- **Zerlegung einer Konjunktion in den Hypothesen**
  - Wenn eine Aussage C aus den Annahmen A und B folgt, dann folgt sie auch aus $A \wedge B$
  - Um $A \wedge B \vdash C$ zu zeigen, genügt es $A, B \vdash C$ zu beweisen
  - Die restlichen Hypothesen ändern sich nicht
  - Beweisregel muß Position von $A \wedge B$ angeben

$$\frac{\Gamma, A \wedge B, \Delta \vdash C \quad \texttt{BY andE } i}{\Gamma, A, B, \Delta \vdash C}$$

- **Hypothesenregel**
  - Wenn die Konklusion C in den Hypothesen vorkommt, dann ist sie bewiesen

Beweisregeln folgen direkt aus Semantik

- **Zerlegung einer Konjunktion in der Konklusion**
  - $A \wedge B$ ist gültig wenn A gültig ist und B gültig ist
  - Um $A \wedge B$ zu zeigen, genügt es A zu beweisen und B zu beweisen
  - Die Hypothesenliste Γ ändert sich dabei nicht

$$\Gamma \vdash A \wedge B \quad \texttt{BY andI}$$
$$\Gamma \vdash A$$
$$\Gamma \vdash B$$

- **Zerlegung einer Konjunktion in den Hypothesen**
  - Wenn eine Aussage C aus den Annahmen A und B folgt, dann folgt sie auch aus $A \wedge B$
  - Um $A \wedge B \vdash C$ zu zeigen, genügt es $A, B \vdash C$ zu beweisen
  - Die restlichen Hypothesen ändern sich nicht
  - Beweisregel muß Position von $A \wedge B$ angeben

$$\Gamma, A \wedge B, \Delta \vdash C \quad \texttt{BY andE } i$$
$$\Gamma, A, B, \Delta \vdash C$$

- **Hypothesenregel**
  - Wenn die Konklusion C in den Hypothesen vorkommt, dann ist sie bewiesen

$$\Gamma, C, \Delta \vdash C \quad \texttt{BY hypothesis } i$$

Beweis für Kommutativität von $\wedge$

$$A \wedge B \vdash B \wedge A$$

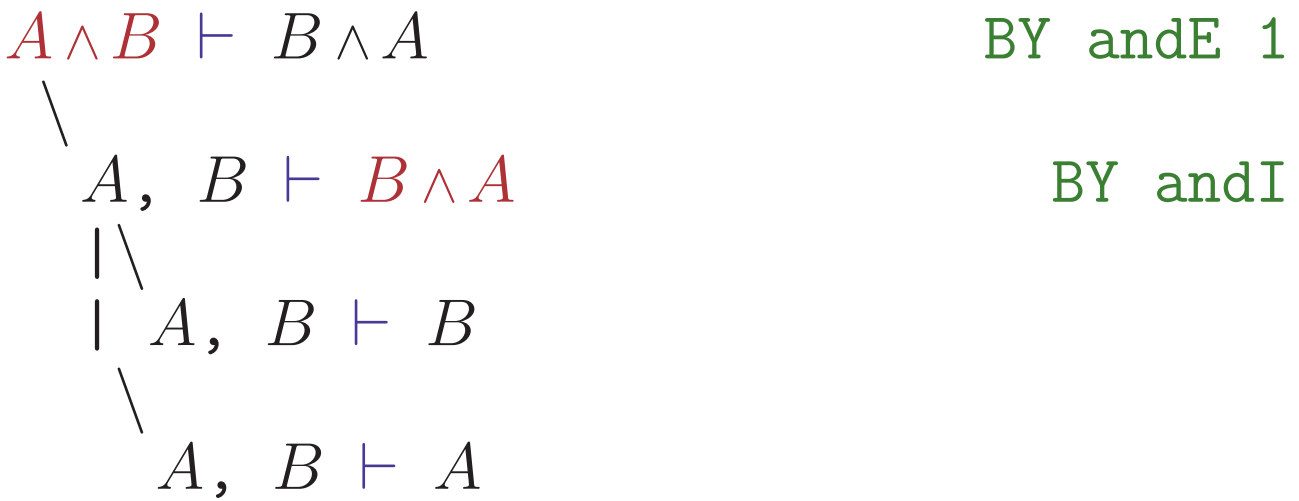
Beweis für Kommutativität von $\wedge$

$A \wedge B \vdash B \wedge A$ BY andE 1

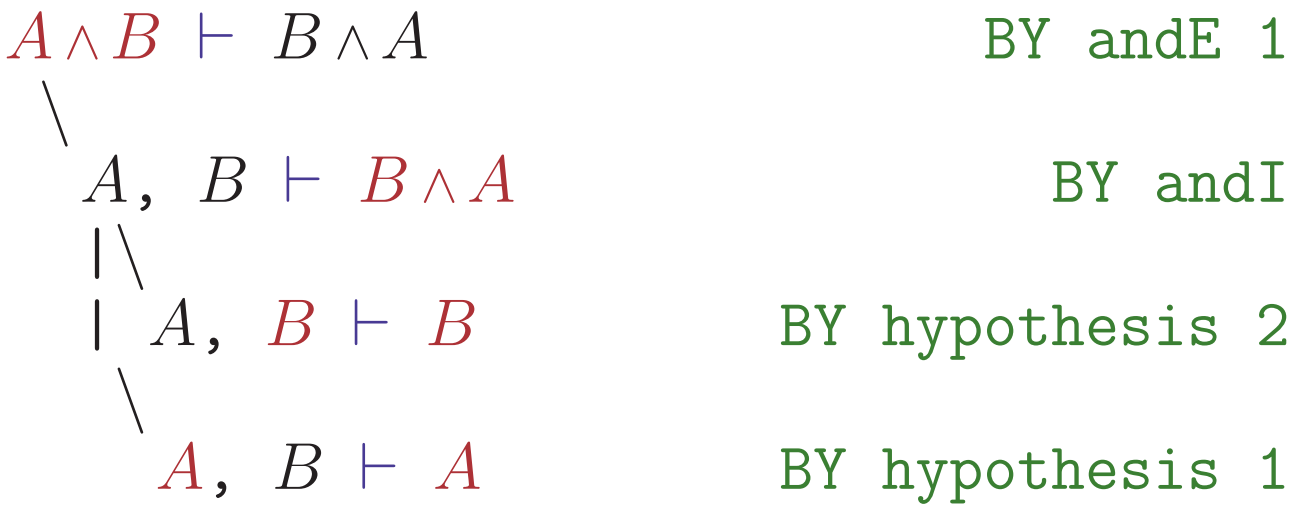
$\backslash$

$A,\ B \vdash B \wedge A$

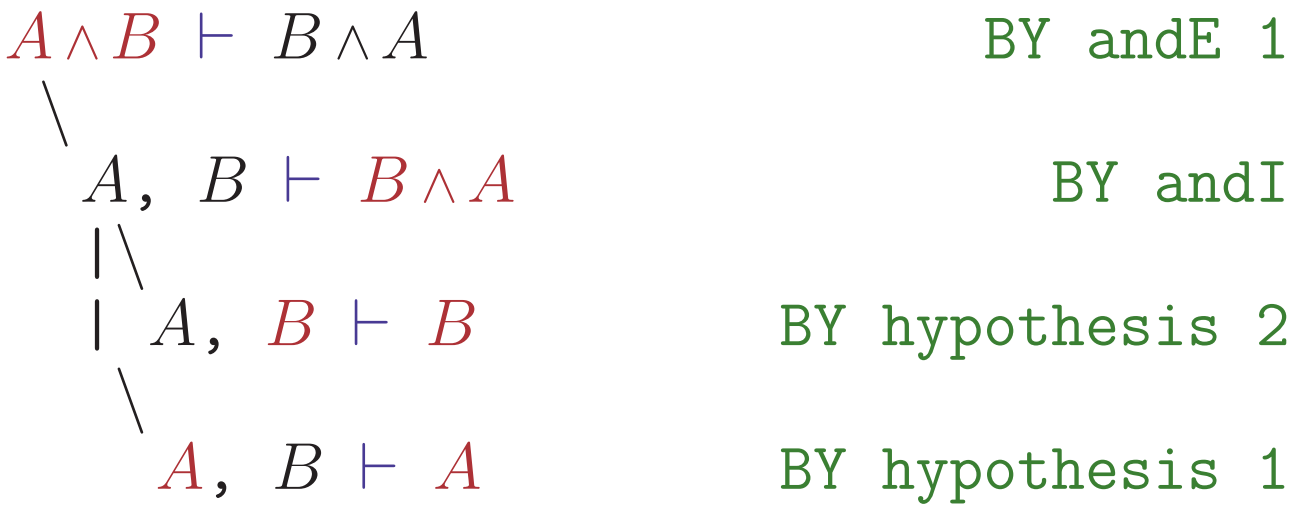
Beweis für Kommutativität von $\wedge$



Beweis für Kommutativität von $\wedge$

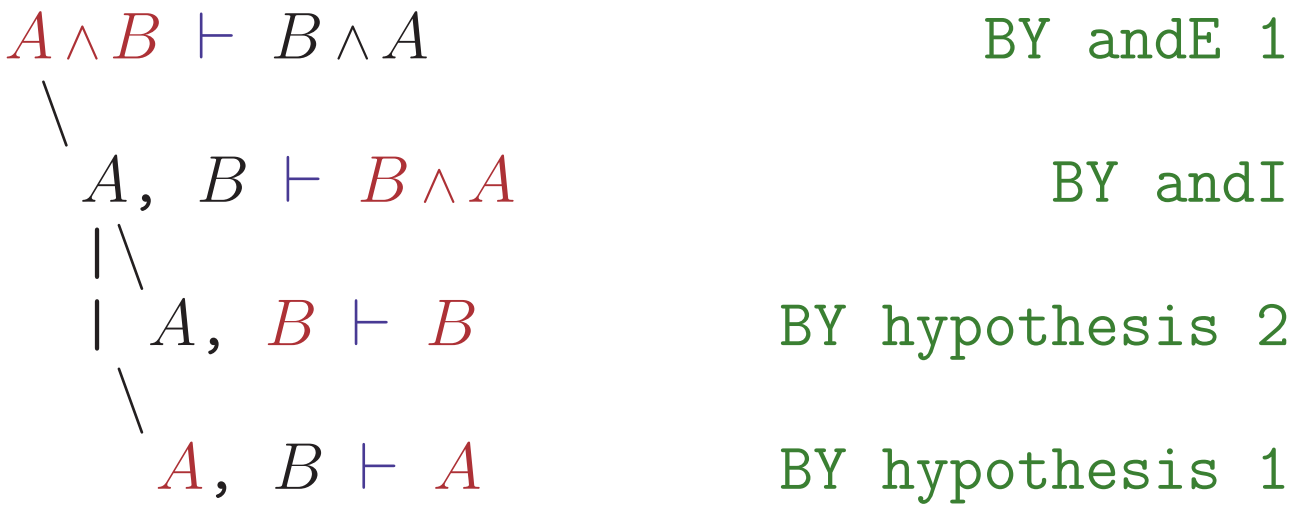


Beweis für Kommutativität von $\wedge$



- **Beweis:** Baum mit Sequenzen und Regeln als Knoten
  - Nachfolger eines Knotens sind Teilziele der Regelanwendung auf Sequenz
  - unvollständig: manche Blätter ohne Regel
  - vollständig: Regeln der Blätter ohne Teilziele

Beweis für Kommutativität von $\wedge$



- **Beweis:** Baum mit Sequenzen und Regeln als Knoten
  - Nachfolger eines Knotens sind Teilziele der Regelanwendung auf Sequenz
  - unvollständig: manche Blätter ohne Regel
  - vollständig: Regeln der Blätter ohne Teilziele
- **Theorem:** Formel C mit vollständigem Beweis für $\vdash C$

Schliessen über Disjunktionen

- **Zerlegung einer Disjunktion in der Konklusion**
  - $A \vee B$ ist gültig wenn A gültig ist oder wenn B gültig ist
  - Um $A \vee B$ zu zeigen, genügt es entweder A oder B zu beweisen
  - Mit der Wahl der Beweisregel wird

    eine Entscheidung getroffen
  - Die Hypothesenliste ändert sich nicht

Schliessen über Disjunktionen

- **Zerlegung einer Disjunktion in der Konklusion**
  - $A \vee B$ ist gültig wenn A gültig ist oder wenn B gültig ist
  - Um $A \vee B$ zu zeigen, genügt es entweder A oder B zu beweisen
  - Mit der Wahl der Beweisregel wird eine Entscheidung getroffen
  - Die Hypothesenliste ändert sich nicht

$\Gamma \vdash A \vee B$ `BY orI1`
$\Gamma \vdash A$

$\Gamma \vdash A \vee B$ `BY orI2`
$\Gamma \vdash B$

Schliessen über Disjunktionen

- **Zerlegung einer Disjunktion in der Konklusion**
  - $A \vee B$ ist gültig wenn A gültig ist oder wenn B gültig ist
  - Um $A \vee B$ zu zeigen, genügt es entweder A oder B zu beweisen
  - Mit der Wahl der Beweisregel wird eine Entscheidung getroffen
  - Die Hypothesenliste ändert sich nicht

$\Gamma \vdash A \vee B$ `BY orI1`
$\Gamma \vdash A$

$\Gamma \vdash A \vee B$ `BY orI2`
$\Gamma \vdash B$

- **Zerlegung einer Disjunktion in den Hypothesen**
  - Wenn eine Aussage C aus der Annahmen A und aus der Annahme B folgt, dann folgt sie auch aus $A \vee B$
  - Um $A \vee B \vdash C$ zu zeigen, muß man $A \vdash C$ und $B \vdash C$ beweisen
  - Die restlichen Hypothesen ändern sich nicht
  - Beweisregel muß Position von $A \vee B$ angeben

Schliessen über Disjunktionen

- **Zerlegung einer Disjunktion in der Konklusion**
  - $A \vee B$ ist gültig wenn A gültig ist oder wenn B gültig ist
  - Um $A \vee B$ zu zeigen, genügt es entweder A oder B zu beweisen
  - Mit der Wahl der Beweisregel wird eine Entscheidung getroffen
  - Die Hypothesenliste ändert sich nicht

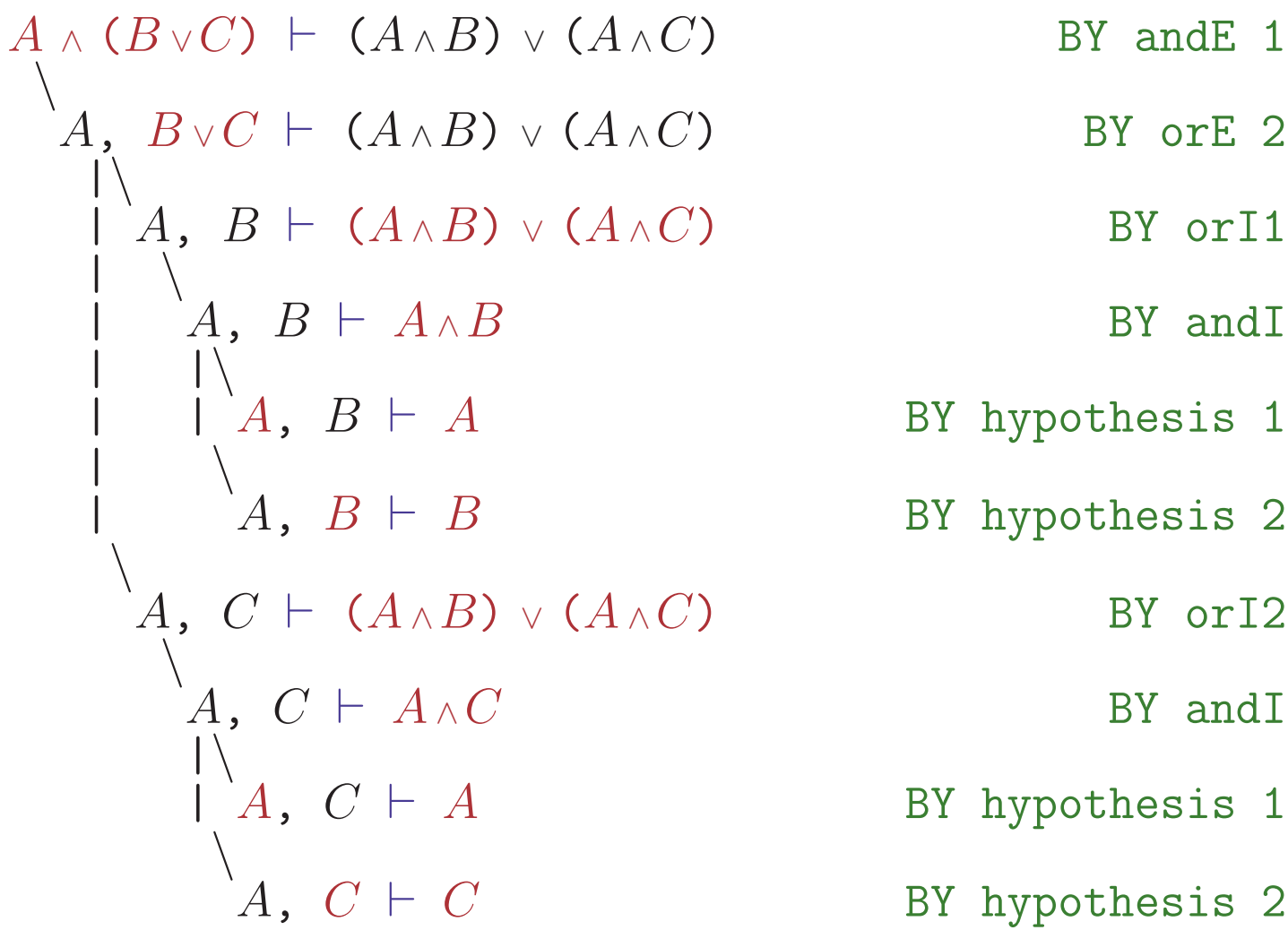
$$\frac{\Gamma \vdash A \vee B \quad \texttt{BY orI1}}{\Gamma \vdash A}$$

$$\frac{\Gamma \vdash A \vee B \quad \texttt{BY orI2}}{\Gamma \vdash B}$$

- **Zerlegung einer Disjunktion in den Hypothesen**
  - Wenn eine Aussage C aus der Annahmen A und aus der Annahme B folgt, dann folgt sie auch aus $A \vee B$
  - Um $A \vee B \vdash C$ zu zeigen, muß man $A \vdash C$ und $B \vdash C$ beweisen
  - Die restlichen Hypothesen ändern sich nicht
  - Beweisregel muß Position von $A \vee B$ angeben

$$\frac{\Gamma, A \vee B, \Delta \vdash C \quad \texttt{BY orE } i}{\begin{array}{l}\Gamma, A, \Delta \vdash C \\ \Gamma, B, \Delta \vdash C\end{array}}$$

Beweis für Distributivität von $\wedge$ und $\vee$



Implikation und Negation

- **Zerlegung einer Implikation in der Konklusion**
  - $A \Rightarrow B$ ist gültig wenn B unter der Annahme A gültig ist
  - Um $A \Rightarrow B$ zu zeigen, genügt es A anzunehmen und B zu beweisen
  - Die Beweisregel verschiebt A in die Hypothesen

Implikation und Negation

- **Zerlegung einer Implikation in der Konklusion**
  - $A \Rightarrow B$ ist gültig wenn B unter der Annahme A gültig ist
  - Um $A \Rightarrow B$ zu zeigen, genügt es A anzunehmen und B zu beweisen
  - Die Beweisregel verschiebt A in die Hypothesen

$$\frac{\Gamma \vdash A \Rightarrow B \quad \texttt{BY impI}}{\Gamma,\ A \vdash B}$$

Implikation und Negation

- **Zerlegung einer Implikation in der Konklusion**
  - $A \Rightarrow B$ ist gültig wenn B unter der Annahme A gültig ist
  - Um $A \Rightarrow B$ zu zeigen, genügt es A anzunehmen und B zu beweisen
  - Die Beweisregel verschiebt A in die Hypothesen

$$\frac{\Gamma \vdash A \Rightarrow B \quad \texttt{BY impI}}{\Gamma,\ A \vdash B}$$

- **Zerlegung einer Implikation in den Hypothesen**
  - Wenn eine Aussage C aus der Annahme B folgt und A gültig ist, dann folgt C aus $A \Rightarrow B$
  - Um $A \Rightarrow B \vdash C$ zu zeigen, muß man $B \vdash C$ und $\vdash A$ beweisen
  - Die restlichen Hypothesen ändern sich nicht
  - Beweisregel muß $A \Rightarrow B$ erhalten

Implikation und Negation

- **Zerlegung einer Implikation in der Konklusion**
  - $A \Rightarrow B$ ist gültig wenn B unter der Annahme A gültig ist
  - Um $A \Rightarrow B$ zu zeigen, genügt es A anzunehmen und B zu beweisen
  - Die Beweisregel verschiebt A in die Hypothesen

$$\frac{\Gamma \vdash A \Rightarrow B \quad \texttt{BY impI}}{\Gamma,\ A \vdash B}$$

- **Zerlegung einer Implikation in den Hypothesen**
  - Wenn eine Aussage C aus der Annahme B folgt und A gültig ist, dann folgt C aus $A \Rightarrow B$
  - Um $A \Rightarrow B \vdash C$ zu zeigen, muß man $B \vdash C$ und $\vdash A$ beweisen
  - Die restlichen Hypothesen ändern sich nicht
  - Beweisregel muß $A \Rightarrow B$ erhalten

$$\frac{\Gamma, A \Rightarrow B, \Delta \vdash C \quad \texttt{BY impE } i}{\begin{array}{l}\Gamma, A \Rightarrow B, \Delta \vdash A \\ \Gamma, \Delta, B \vdash C\end{array}}$$

Implikation und Negation

- **Zerlegung einer Implikation in der Konklusion**
  - $A \Rightarrow B$ ist gültig wenn B unter der Annahme A gültig ist
  - Um $A \Rightarrow B$ zu zeigen, genügt es A anzunehmen und B zu beweisen
  - Die Beweisregel verschiebt A in die Hypothesen

$$\frac{\Gamma \vdash A \Rightarrow B \quad \texttt{BY impI}}{\Gamma,\ A \vdash B}$$

- **Zerlegung einer Implikation in den Hypothesen**
  - Wenn eine Aussage C aus der Annahme B folgt und A gültig ist, dann folgt C aus $A \Rightarrow B$
  - Um $A \Rightarrow B \vdash C$ zu zeigen, muß man $B \vdash C$ und $\vdash A$ beweisen
  - Die restlichen Hypothesen ändern sich nicht
  - Beweisregel muß $A \Rightarrow B$ erhalten

$$\frac{\Gamma, A \Rightarrow B, \Delta \vdash C \quad \texttt{BY impE } i}{\Gamma, A \Rightarrow B, \Delta \vdash A \qquad \Gamma, \Delta, B \vdash C}$$

- **Zerlegung einer Negation**
  - $\neg A$ ist dasselbe wie $A \Rightarrow \mathsf{ff}$
  - Die Regeln für Negation entsprechen denen der Implikation
  - Jede Aussage folgt unter der Annahme ff ($\mathsf{ff} \vdash C$ ist gültig)

Currying zwischen Implikation und Konjunktion

$\vdash\ (A \wedge B \Rightarrow C)\ \Rightarrow\ (A \Rightarrow B \Rightarrow C)$ BY impI

$A \wedge B \Rightarrow C\ \vdash\ A \Rightarrow B \Rightarrow C$ BY impI

$A \wedge B \Rightarrow C,\ A\ \vdash\ B \Rightarrow C$ BY impI

$A \wedge B \Rightarrow C,\ A,\ B\ \vdash\ C$ BY impE 1

$A \wedge B \Rightarrow C,\ A,\ B\ \vdash\ A \wedge B$ BY andI

$A \wedge B \Rightarrow C,\ A,\ B\ \vdash\ A$ BY hypothesis 2

$A \wedge B \Rightarrow C,\ A,\ B\ \vdash\ B$ BY hypothesis 3

$A,\ B,\ C\ \vdash\ C$ BY hypothesis 3

Sonstige Inferenzregeln

- **Einfügen von Zwischenbehauptungen**
  - C ist gültig wenn A gilt und C aus der Annahme A folgt
  - Um C zu zeigen, kann man eine Zwischenbehauptung A beweisen und C unter der Annahme A beweisen
  - A kann eine beliebige "Schnitt"-Formel sein
  - Inferenzregel muß A als Parameter haben

Sonstige Inferenzregeln

- **Einfügen von Zwischenbehauptungen**
  - C ist gültig wenn A gilt und C aus der Annahme A folgt
  - Um C zu zeigen, kann man eine Zwischenbehauptung A beweisen und C unter der Annahme A beweisen
  - A kann eine beliebige "Schnitt"-Formel sein
  - Inferenzregel muß A als Parameter haben
  - Beweise werden signifikant kürzer, wenn A mehrfach benutzt wird

$$\frac{\Gamma \vdash A \qquad \Gamma, A \vdash C}{\Gamma \vdash C}\ \texttt{BY cut}\ A$$

Sonstige Inferenzregeln

- **Einfügen von Zwischenbehauptungen**
  - C ist gültig wenn A gilt und C aus der Annahme A folgt
  - Um C zu zeigen, kann man eine Zwischenbehauptung A beweisen und C unter der Annahme A beweisen
  - A kann eine beliebige "Schnitt"-Formel sein
  - Inferenzregel muß A als Parameter haben
  - Beweise werden signifikant kürzer, wenn A mehrfach benutzt wird

$$\frac{\Gamma \vdash C \qquad \texttt{BY cut } A}{\Gamma \vdash A \qquad \Gamma, A \vdash C}$$

- **Ausdünnen der Annahmen**
  - Hypothesen, die nicht gebraucht werden, können entfernt werden

Sonstige Inferenzregeln

- **Einfügen von Zwischenbehauptungen**
  - C ist gültig wenn A gilt und C aus der Annahme A folgt
  - Um C zu zeigen, kann man eine Zwischenbehauptung A beweisen und C unter der Annahme A beweisen
  - A kann eine beliebige "Schnitt"-Formel sein
  - Inferenzregel muß A als Parameter haben
  - Beweise werden signifikant kürzer, wenn A mehrfach benutzt wird

$$\frac{\Gamma \vdash C \qquad \texttt{BY cut } A}{\Gamma \vdash A \quad \Gamma, A \vdash C}$$

- **Ausdünnen der Annahmen**
  - Hypothesen, die nicht gebraucht werden, können entfernt werden
  - Beweise werden übersichtlicher

$$\frac{\Gamma, A, \Delta \vdash C \ \texttt{BY thin } i}{\Gamma, \Delta \vdash C}$$

Sonstige Inferenzregeln

- **Einfügen von Zwischenbehauptungen**
  - C ist gültig wenn A gilt und C aus der Annahme A folgt
  - Um C zu zeigen, kann man eine Zwischenbehauptung A beweisen und C unter der Annahme A beweisen
  - A kann eine beliebige "Schnitt"-Formel sein
  - Inferenzregel muß A als Parameter haben
  - Beweise werden signifikant kürzer, wenn A mehrfach benutzt wird

$$\frac{\Gamma \vdash C \qquad \texttt{BY cut } A}{\Gamma \vdash A \qquad \Gamma, A \vdash C}$$

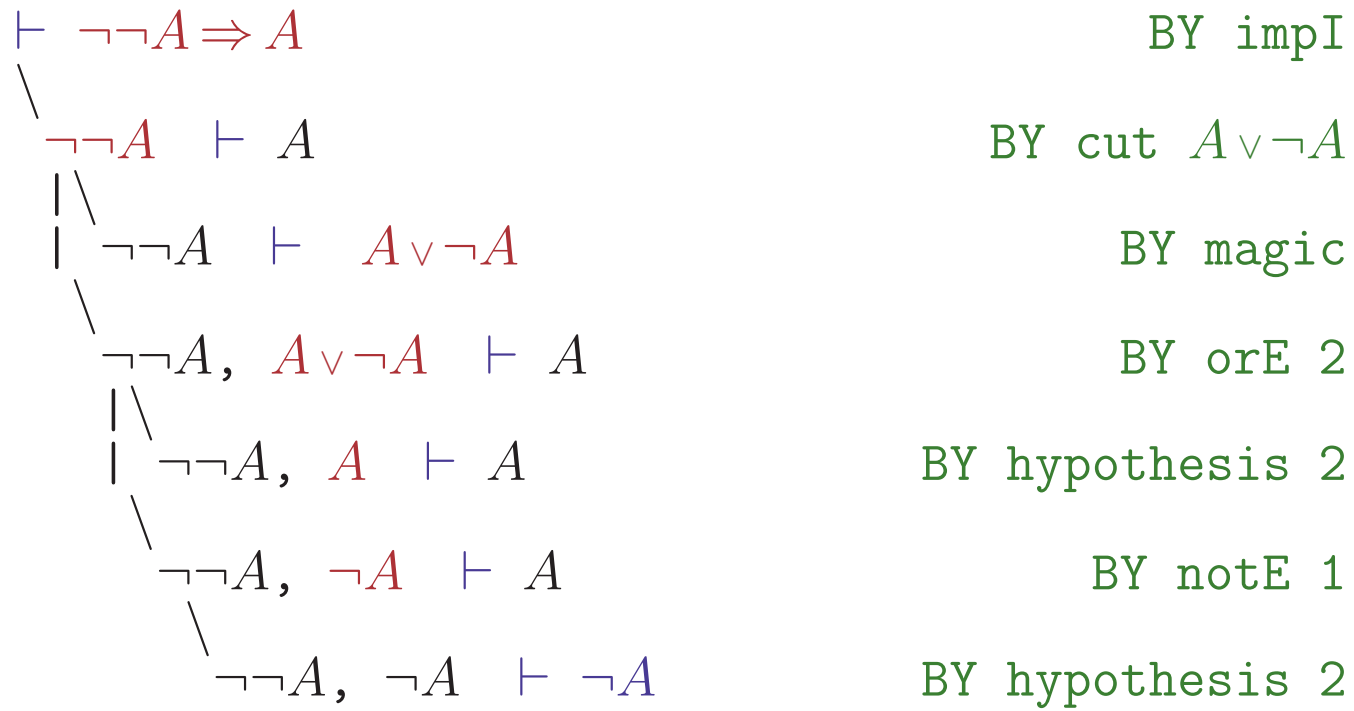
- **Ausdünnen der Annahmen**
  - Hypothesen, die nicht gebraucht werden, können entfernt werden
  - Beweise werden übersichtlicher

$$\frac{\Gamma, A, \Delta \vdash C \ \texttt{BY thin } i}{\Gamma, \Delta \vdash C}$$

- **Klassisches Schließen**
  - Das Gesetz vom Ausgeschlossenen Dritten wird als gültig postuliert

$$\Gamma \vdash A \vee \neg A \ \texttt{BY magic}$$

Klassische Behandlung der doppelten Negation



Refinement Logic: Aussagenlogische Regeln

`andE` i	$\Gamma,\ A \wedge B,\ \Delta \vdash C$ $\Gamma,\ A,\ B,\ \Delta \vdash C$	$\Gamma \vdash A \wedge B$ $\Gamma \vdash A$ $\Gamma \vdash B$	`andI`
`orE` i	$\Gamma,\ A \vee B,\ \Delta \vdash C$ $\Gamma,\ A,\ \Delta \vdash C$ $\Gamma,\ B,\ \Delta \vdash C$	$\Gamma \vdash A \vee B$ $\Gamma \vdash A$ $\Gamma \vdash B$	`orI1` `orI2`
`impE` i	$\Gamma,\ A \Rightarrow B,\ \Delta \vdash C$ $\Gamma,\ A \Rightarrow B,\ \Delta \vdash A$ $\Gamma,\ \Delta,\ B \vdash C$	$\Gamma \vdash A \Rightarrow B$ $\Gamma,\ A \vdash B$	`impI`
`notE` i	$\Gamma,\ \neg A,\ \Delta \vdash C$ $\Gamma,\ \neg A,\ \Delta \vdash A$	$\Gamma \vdash \neg A$ $\Gamma,\ A \vdash \mathsf{ff}$	`notI`
`falseE` i	$\Gamma,\ \mathsf{ff},\ \Delta \vdash C$		`magic`
`hypothesis` i	$\Gamma,\ A,\ \Delta \vdash A$	$\Gamma,\ \Delta \vdash C$ $\Gamma,\ \Delta \vdash A$ $\Gamma,\ A,\ \Delta \vdash C$	`cut` i A
`thin` i	$\Gamma,\ A,\ \Delta \vdash C$ $\Gamma,\ \Delta \vdash C$		

Formale Behandlung von Quantoren

- **Zerlegung eines Allquantors in der Konklusion**
  - $\forall x{:}T.A$ ist gültig wenn $\iota_x^u(A)$ für jeden möglichen Wert u wahr ist
  - Um $\forall x{:}T.A$ zu zeigen, nimmt man an $x{:}T$ sei beliebig aber fest und beweist dann A für dieses x
  - Der Name x ist ein Platzhalter für einen beliebigen Wert u
  - Wenn der Name x in den Hypothesen bereits vorkommt, wählt man einen neuen Namen y und ersetzt x in A durch y

Formale Behandlung von Quantoren

- **Zerlegung eines Allquantors in der Konklusion**
  - $\forall x{:}T.A$ ist gültig wenn $\iota_x^u(A)$ für jeden möglichen Wert u wahr ist
  - Um $\forall x{:}T.A$ zu zeigen, nimmt man an $x{:}T$ sei beliebig aber fest und beweist dann A für dieses x
  - Der Name x ist ein Platzhalter für einen beliebigen Wert u
  - Wenn der Name x in den Hypothesen bereits vorkommt, wählt man einen neuen Namen y und ersetzt x in A durch y
- **Zerlegung eines Existenzquantors**
  - $\exists x{:}T.A$ ist gültig wenn $\iota_x^u(A)$ für einen Wert u wahr ist
  - Um $\exists x{:}T.A$ zu zeigen, beschreibt man einen konkreten Wert durch einen Term t und beweist A für diesen Term
  - Formal muß der Name x in A durch den Term t ersetzt werden

Formale Behandlung von Quantoren

- **Zerlegung eines Allquantors in der Konklusion**
  - $\forall x{:}T.A$ ist gültig wenn $\iota_x^u(A)$ für jeden möglichen Wert u wahr ist
  - Um $\forall x{:}T.A$ zu zeigen, nimmt man an $x{:}T$ sei beliebig aber fest und beweist dann A für dieses x
  - Der Name x ist ein Platzhalter für einen beliebigen Wert u
  - Wenn der Name x in den Hypothesen bereits vorkommt, wählt man einen neuen Namen y und ersetzt x in A durch y
- **Zerlegung eines Existenzquantors**
  - $\exists x{:}T.A$ ist gültig wenn $\iota_x^u(A)$ für einen Wert u wahr ist
  - Um $\exists x{:}T.A$ zu zeigen, beschreibt man einen konkreten Wert durch einen Term t und beweist A für diesen Term
  - Formal muß der Name x in A durch den Term t ersetzt werden

Beweis braucht syntaktische Simulation von ι_x^u

Mechanismen zur syntaktischen Simulation von ι_x^u

- **Semantik von Quantoren verwendet $\boldsymbol{\iota_x^u}$**
  - $\iota(\forall x : T \,.\, A)$ und $\iota(\exists x : T \,.\, A)$ wird durch $\iota_x^u(A)$ erklärt
    - $\iota_x^u(A)$ muß für alle oder einen Wert u wahr werden

Mechanismen zur syntaktischen Simulation von ι_x^u

- **Semantik von Quantoren verwendet $\boldsymbol{\iota_x^u}$**
  - $\iota(\forall x : T \,.\, A)$ und $\iota(\exists x : T \,.\, A)$ wird durch $\iota_x^u(A)$ erklärt
    - $\iota_x^u(A)$ muß für alle oder einen Wert u wahr werden
  - ι_x^u modifiziert die Interpretation ι für die gebundene Variable x
  - Syntaktischer Beweis muß stattdessen Formel A modifizieren

Mechanismen zur syntaktischen Simulation von ι_x^u

- **Semantik von Quantoren verwendet $\boldsymbol{\iota_x^u}$**
  - $\iota(\forall x : T \,.\, A)$ und $\iota(\exists x : T \,.\, A)$ wird durch $\iota_x^u(A)$ erklärt
    - $\iota_x^u(A)$ muß für alle oder einen Wert u wahr werden
  - ι_x^u modifiziert die Interpretation ι für die gebundene Variable x
  - Syntaktischer Beweis muß stattdessen Formel A modifizieren
- **Formales Konzept: Substitution $\boldsymbol{A[t/x]}$**
  - Viele alternative Schreibweisen (sehr häufig $A\{x \backslash t\}$)
  - Vorkommen der Variablen x in A werden durch den Term t ersetzt

Mechanismen zur syntaktischen Simulation von ι_x^u

- **Semantik von Quantoren verwendet $\boldsymbol{\iota_x^u}$**
  - $\iota(\forall x : T \,.\, A)$ und $\iota(\exists x : T \,.\, A)$ wird durch $\iota_x^u(A)$ erklärt
    - $\iota_x^u(A)$ muß für alle oder einen Wert u wahr werden
  - ι_x^u modifiziert die Interpretation ι für die gebundene Variable x
  - Syntaktischer Beweis muß stattdessen Formel A modifizieren
- **Formales Konzept: Substitution $\boldsymbol{A[t/x]}$**
  - Viele alternative Schreibweisen (sehr häufig $A\{x \backslash t\}$)
  - Vorkommen der Variablen x in A werden durch den Term t ersetzt
  - Hinreichend wenn jedes Objekt des Universums durch Terme beschreibbar
    - Reelle Zahlen, Funktionenräume etc. haben zu viele Objekte
  - Allquantor ist sonst nicht vollständig repräsentierbar

Substitution $A[t/x]$ – wichtige Aspekte

Substitution $A[t/x]$ – wichtige Aspekte

- **Substitution muß Semantik erhalten**
  - Die Namen quantifizierter Variablen dürfen keine Rolle spielen
    - $\exists x : T \,.\, A(x)$ hat dieselbe Bedeutung wie $\exists y : T \,.\, A(y)$

Substitution $A[t/x]$ – wichtige Aspekte

- **Substitution muß Semantik erhalten**
  - Die Namen quantifizierter Variablen dürfen keine Rolle spielen
    - $\exists x : T \,.\, A(x)$ hat dieselbe Bedeutung wie $\exists y : T \,.\, A(y)$
  - Keine Ersetzung von x durch t in $(\exists x : \mathbb{N} \,.\, x \leq 4)[t/x]$
    - Das "äußere" x hat mit dem innerhalb des Quantors nichts zu tun

Substitution $A[t/x]$ – wichtige Aspekte

- **Substitution muß Semantik erhalten**
  - Die Namen quantifizierter Variablen dürfen keine Rolle spielen
    - $\exists x : T \,.\, A(x)$ hat dieselbe Bedeutung wie $\exists y : T \,.\, A(y)$
  - Keine Ersetzung von x durch t in $(\exists x : \mathbb{N} \,.\, x \leq 4)[t/x]$
    - Das "äußere" x hat mit dem innerhalb des Quantors nichts zu tun
  - Keine Ersetzung von x durch y in $(\exists y : \mathbb{N} \,.\, x < y)[y/x]$
    - Durch die Ersetzung würde ein ungewollter Selbstbezug entstehen

Substitution $A[t/x]$ – wichtige Aspekte

- **Substitution muß Semantik erhalten**
  - Die Namen quantifizierter Variablen dürfen keine Rolle spielen
    - $\exists x{:}T\,.\,A(x)$ hat dieselbe Bedeutung wie $\exists y{:}T\,.\,A(y)$
  - Keine Ersetzung von x durch t in $(\exists x{:}\mathbb{N}\,.\,x{\leq}4)[t/x]$
    - Das "äußere" x hat mit dem innerhalb des Quantors nichts zu tun
  - Keine Ersetzung von x durch y in $(\exists y{:}\mathbb{N}\,.\,x{<}y)[y/x]$
    - Durch die Ersetzung würde ein ungewollter Selbstbezug entstehen
- **Variablen*vorkommen* müssen identifizierbar sein**
  - Gebundenes Vorkommen x in A: x erscheint in Quantor, der A umfaßt
  - Freies Vorkommen x in A: x kommt in A vor, ohne gebunden zu sein

Substitution $A[t/x]$ – wichtige Aspekte

- **Substitution muß Semantik erhalten**
  - Die Namen quantifizierter Variablen dürfen keine Rolle spielen
    - $\exists x : T \,.\, A(x)$ hat dieselbe Bedeutung wie $\exists y : T \,.\, A(y)$
  - Keine Ersetzung von x durch t in $(\exists x : \mathbb{N} \,.\, x \leq 4)[t/x]$
    - Das "äußere" x hat mit dem innerhalb des Quantors nichts zu tun
  - Keine Ersetzung von x durch y in $(\exists y : \mathbb{N} \,.\, x < y)[y/x]$
    - Durch die Ersetzung würde ein ungewollter Selbstbezug entstehen
- **Variablen*vorkommen* müssen identifizierbar sein**
  - Gebundenes Vorkommen x in A: x erscheint in Quantor, der A umfaßt
  - Freies Vorkommen x in A: x kommt in A vor, ohne gebunden zu sein
  - Notation $\boldsymbol{A[x]}$: Ausdruck A hat mögliche freie Vorkommen von x

Substitution $A[t/x]$ – wichtige Aspekte

- **Substitution muß Semantik erhalten**
  - Die Namen quantifizierter Variablen dürfen keine Rolle spielen
    - $\exists x : T \,.\, A(x)$ hat dieselbe Bedeutung wie $\exists y : T \,.\, A(y)$
  - Keine Ersetzung von x durch t in $(\exists x : \mathbb{N} \,.\, x \leq 4)[t/x]$
    - Das "äußere" x hat mit dem innerhalb des Quantors nichts zu tun
  - Keine Ersetzung von x durch y in $(\exists y : \mathbb{N} \,.\, x < y)[y/x]$
    - Durch die Ersetzung würde ein ungewollter Selbstbezug entstehen
- **Variablen*vorkommen* müssen identifizierbar sein**
  - Gebundenes Vorkommen x in A: x erscheint in Quantor, der A umfaßt
  - Freies Vorkommen x in A: x kommt in A vor, ohne gebunden zu sein
  - Notation $\boldsymbol{A[x]}$: Ausdruck A hat mögliche freie Vorkommen von x
  - A heißt geschlossen falls A keine freien Variablen enthält

Vorkommen von Variablen präzisiert

- x die Variable x kommt frei vor; $y \neq x$ kommt nicht vor.

  ff: die Variable x kommt nicht vor

Vorkommen von Variablen präzisiert

- x — die Variable x kommt frei vor; $y \neq x$ kommt nicht vor.
  ff: — die Variable x kommt nicht vor
- $f(t_1, ..., t_n)$ — freie Vorkommen von x in t_i bleiben frei
  $t_1 = t_2$ — gebundene Vorkommen von x bleiben gebunden.
  $P(t_1, ..., t_n)$

Vorkommen von Variablen präzisiert

- x — die Variable x kommt frei vor; $y \neq x$ kommt nicht vor.
  ff: — die Variable x kommt nicht vor
- $f(t_1, ..., t_n)$ — freie Vorkommen von x in t_i bleiben frei
  $t_1 = t_2$ — gebundene Vorkommen von x bleiben gebunden.
  $P(t_1, ..., t_n)$
- $\neg A$, (A) — freie Vorkommen von x in A, B bleiben frei
  $A \wedge B$, $A \vee B$ — gebundene Vorkommen von x bleiben gebunden.
  $A \Rightarrow B$

Vorkommen von Variablen präzisiert

- x — die Variable x kommt frei vor; $y{\neq}x$ kommt nicht vor.
  ff: — die Variable x kommt nicht vor
- $f(t_1, ..., t_n)$ — freie Vorkommen von x in t_i bleiben frei
  $t_1{=}t_2$ — gebundene Vorkommen von x bleiben gebunden.
  $P(t_1, ..., t_n)$
- $\neg A$, (A) — freie Vorkommen von x in A, B bleiben frei
  $A \wedge B$, $A \vee B$ — gebundene Vorkommen von x bleiben gebunden.
  $A \Rightarrow B$
- $\forall x{:}T \,.\, A$ — beliebige Vorkommen von x in A werden gebunden
  $\exists x{:}T \,.\, A$ — Vorkommen von $y{\neq}x$ in A bleiben unverändert.

Vorkommen von Variablen präzisiert

- x — die Variable x kommt frei vor; $y{\neq}x$ kommt nicht vor.
  ff: — die Variable x kommt nicht vor
- $f(t_1, ..., t_n)$ — freie Vorkommen von x in t_i bleiben frei
  $t_1{=}t_2$ — gebundene Vorkommen von x bleiben gebunden.
  $P(t_1, ..., t_n)$
- $\neg A$, (A) — freie Vorkommen von x in A, B bleiben frei
  $A \wedge B$, $A \vee B$ — gebundene Vorkommen von x bleiben gebunden.
  $A \Rightarrow B$
- $\forall x{:}T\,.\,A$ — beliebige Vorkommen von x in A werden gebunden
  $\exists x{:}T\,.\,A$ — Vorkommen von $y{\neq}x$ in A bleiben unverändert.

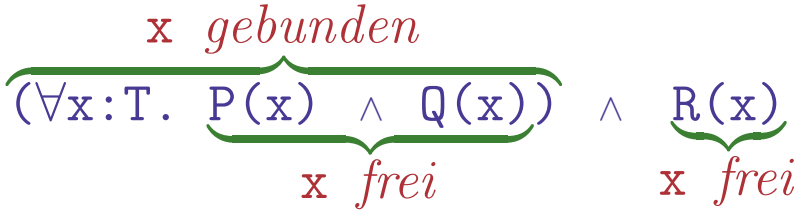
`(∀x:T. P(x) ∧ Q(x)) ∧ R(x)`

Vorkommen von Variablen präzisiert

- x — die Variable x kommt frei vor; $y{\neq}x$ kommt nicht vor.
  ff: — die Variable x kommt nicht vor
- $f(t_1, ..., t_n)$ — freie Vorkommen von x in t_i bleiben frei
  $t_1{=}t_2$ — gebundene Vorkommen von x bleiben gebunden.
  $P(t_1, ..., t_n)$
- $\neg A$, (A) — freie Vorkommen von x in A, B bleiben frei
  $A \wedge B$, $A \vee B$ — gebundene Vorkommen von x bleiben gebunden.
  $A \Rightarrow B$
- $\forall x{:}T\,.\,A$ — beliebige Vorkommen von x in A werden gebunden
  $\exists x{:}T\,.\,A$ — Vorkommen von $y{\neq}x$ in A bleiben unverändert.

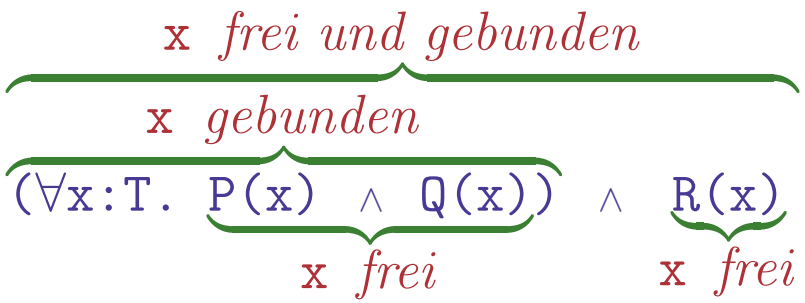
Vorkommen von Variablen präzisiert

- x — die Variable x kommt frei vor; $y{\neq}x$ kommt nicht vor.
  ff: — die Variable x kommt nicht vor
- $f(t_1, ..., t_n)$ — freie Vorkommen von x in t_i bleiben frei
  $t_1{=}t_2$ — gebundene Vorkommen von x bleiben gebunden.
  $P(t_1, ..., t_n)$
- $\neg A$, (A) — freie Vorkommen von x in A, B bleiben frei
  $A \wedge B$, $A \vee B$ — gebundene Vorkommen von x bleiben gebunden.
  $A \Rightarrow B$
- $\forall x{:}T\,.\,A$ — beliebige Vorkommen von x in A werden gebunden
  $\exists x{:}T\,.\,A$ — Vorkommen von $y{\neq}x$ in A bleiben unverändert.



Vorkommen von Variablen präzisiert

- x — die Variable x kommt frei vor; $y{\neq}x$ kommt nicht vor.
  ff: — die Variable x kommt nicht vor
- $f(t_1, ..., t_n)$ — freie Vorkommen von x in t_i bleiben frei
  $t_1{=}t_2$ — gebundene Vorkommen von x bleiben gebunden.
  $P(t_1, ..., t_n)$
- $\neg A$, (A) — freie Vorkommen von x in A, B bleiben frei
  $A \wedge B$, $A \vee B$ — gebundene Vorkommen von x bleiben gebunden.
  $A \Rightarrow B$
- $\forall x{:}T\,.\,A$ — beliebige Vorkommen von x in A werden gebunden
  $\exists x{:}T\,.\,A$ — Vorkommen von $y{\neq}x$ in A bleiben unverändert.



Substitution $A[t/x]$ formal

Endliche Abbildung $\boldsymbol{\sigma}$ von Variablen in Terme

– $\boldsymbol{\sigma = [t_1, .., t_n/x_1, .., x_n]} \mathrel{\hat{=}} \sigma(x_1){=}t_1, \ldots, \sigma(x_n){=}t_n$

– $\boldsymbol{A\sigma}$: Anwendung von σ auf den Ausdruck A

– $\boldsymbol{\tau\,\sigma}$: Komposition von τ und σ (σ idempotent falls $\sigma\,\sigma = \sigma$)

Substitution $A[t/x]$ formal

Endliche Abbildung σ von Variablen in Terme

- $\boldsymbol{\sigma = [t_1, .., t_n/x_1, .., x_n]} \mathrel{\hat{=}} \sigma(x_1){=}t_1, \ldots, \sigma(x_n){=}t_n$
- $\boldsymbol{A\sigma}$: Anwendung von σ auf den Ausdruck A
- $\boldsymbol{\tau\,\sigma}$: Komposition von τ und σ (σ idempotent falls $\sigma\,\sigma = \sigma$)

$\lfloor x \rfloor [t/x] \qquad = t$

Substitution $A[t/x]$ formal

Endliche Abbildung σ von Variablen in Terme

- $\boldsymbol{\sigma = [t_1, .., t_n/x_1, .., x_n]} \mathrel{\hat{=}} \sigma(x_1){=}t_1, \ldots, \sigma(x_n){=}t_n$
- $\boldsymbol{A\sigma}$: Anwendung von σ auf den Ausdruck A
- $\boldsymbol{\tau\,\sigma}$: Komposition von τ und σ (σ idempotent falls $\sigma\,\sigma = \sigma$)

$$\lfloor x \rfloor[t/x] = t \qquad \lfloor x \rfloor[t/y] = x \quad (y{\neq}x)$$

Substitution $A[t/x]$ formal

Endliche Abbildung σ von Variablen in Terme

- $\boldsymbol{\sigma = [t_1, .., t_n/x_1, .., x_n]} \mathrel{\hat{=}} \sigma(x_1){=}t_1, \ldots, \sigma(x_n){=}t_n$
- $\boldsymbol{A\sigma}$: Anwendung von σ auf den Ausdruck A
- $\boldsymbol{\tau\,\sigma}$: Komposition von τ und σ (σ idempotent falls $\sigma\,\sigma = \sigma$)

$$\lfloor x \rfloor [t/x] = t \qquad \lfloor x \rfloor [t/y] = x \quad (y{\neq}x)$$

$$\lfloor f(t_1, .., t_n) \rfloor \sigma = f(t_1\sigma, .., t_n\sigma)$$

$$\lfloor P(t_1, .., t_n) \rfloor \sigma = P(t_1\sigma, .., t_n\sigma) \qquad \lfloor t_1 \texttt{ = } t_2 \rfloor \sigma = t_1\sigma \texttt{ = } t_2\sigma$$

Substitution $A[t/x]$ formal

Endliche Abbildung σ von Variablen in Terme

- $\boldsymbol{\sigma = [t_1, .., t_n/x_1, .., x_n]} \mathrel{\hat{=}} \sigma(x_1){=}t_1, \ldots, \sigma(x_n){=}t_n$
- $\boldsymbol{A\sigma}$: Anwendung von σ auf den Ausdruck A
- $\boldsymbol{\tau\,\sigma}$: Komposition von τ und σ (σ idempotent falls $\sigma\,\sigma = \sigma$)

$\lfloor x \rfloor[t/x]$	$= t$	$\lfloor x \rfloor[t/y]$	$= x$ $(y{\neq}x)$
$\lfloor f(t_1, .., t_n) \rfloor\sigma$	$= f(t_1\sigma, .., t_n\sigma)$	$\lfloor \mathsf{ff} \rfloor\sigma$	$= \mathsf{ff}$
$\lfloor P(t_1, .., t_n) \rfloor\sigma$	$= P(t_1\sigma, .., t_n\sigma)$	$\lfloor t_1 \texttt{ = } t_2 \rfloor\sigma$	$= t_1\sigma \texttt{ = } t_2\sigma$

Substitution $A[t/x]$ formal

Endliche Abbildung σ von Variablen in Terme

- $\boldsymbol{\sigma = [t_1, .., t_n/x_1, .., x_n]} \mathrel{\hat{=}} \sigma(x_1){=}t_1, \ldots, \sigma(x_n){=}t_n$
- $\boldsymbol{A\sigma}$: Anwendung von σ auf den Ausdruck A
- $\boldsymbol{\tau\,\sigma}$: Komposition von τ und σ (σ idempotent falls $\sigma\,\sigma = \sigma$)

$$
\begin{array}{lll|lll}
\lfloor x \rfloor[t/x] & = t & & \lfloor x \rfloor[t/y] & = x & (y{\neq}x) \\
\lfloor f(t_1, .., t_n) \rfloor\sigma & = f(t_1\sigma, .., t_n\sigma) & & \lfloor \mathsf{ff} \rfloor\sigma & = \mathsf{ff} & \\
\lfloor P(t_1, .., t_n) \rfloor\sigma & = P(t_1\sigma, .., t_n\sigma) & & \lfloor t_1 \texttt{ = } t_2 \rfloor\sigma & = t_1\sigma \texttt{ = } t_2\sigma & \\
\lfloor \neg A \rfloor\sigma & = \neg A\sigma & & \lfloor A \wedge B \rfloor\sigma & = A\sigma \wedge B\sigma & \\
\lfloor A \vee B \rfloor\sigma & = A\sigma \vee B\sigma & & \lfloor A \Rightarrow B \rfloor\sigma & = A\sigma \Rightarrow B\sigma & \\
\lfloor (A) \rfloor\sigma & = (A\sigma) & & & &
\end{array}
$$

Substitution $A[t/x]$ formal

Endliche Abbildung σ von Variablen in Terme

- $\boldsymbol{\sigma = [t_1, .., t_n/x_1, .., x_n]} \mathrel{\hat{=}} \sigma(x_1){=}t_1, \ldots, \sigma(x_n){=}t_n$
- $\boldsymbol{A\sigma}$: Anwendung von σ auf den Ausdruck A
- $\boldsymbol{\tau\,\sigma}$: Komposition von τ und σ (σ idempotent falls $\sigma\,\sigma = \sigma$)

$\lfloor x \rfloor[t/x]$	$= t$	$\lfloor x \rfloor[t/y]$	$= x$	$(y{\neq}x)$
$\lfloor f(t_1, .., t_n) \rfloor\sigma$	$= f(t_1\sigma, .., t_n\sigma)$	$\lfloor \mathsf{ff} \rfloor\sigma$	$= \mathsf{ff}$	
$\lfloor P(t_1, .., t_n) \rfloor\sigma$	$= P(t_1\sigma, .., t_n\sigma)$	$\lfloor t_1 \texttt{ = } t_2 \rfloor\sigma$	$= t_1\sigma \texttt{ = } t_2\sigma$	
$\lfloor \neg A \rfloor\sigma$	$= \neg A\sigma$	$\lfloor A \wedge B \rfloor\sigma$	$= A\sigma \wedge B\sigma$	
$\lfloor A \vee B \rfloor\sigma$	$= A\sigma \vee B\sigma$	$\lfloor A \Rightarrow B \rfloor\sigma$	$= A\sigma \Rightarrow B\sigma$	
$\lfloor (A) \rfloor\sigma$	$= (A\sigma)$			
$\lfloor \forall x{:}T\,.\,A \rfloor[t/x]$	$= \forall x{:}T\,.\,A$	$\lfloor \exists x{:}T\,.\,A \rfloor[t/x]$	$= \exists x{:}T\,.\,A$	

Substitution $A[t/x]$ formal

Endliche Abbildung σ von Variablen in Terme

- $\boldsymbol{\sigma = [t_1, .., t_n/x_1, .., x_n]} \mathrel{\hat{=}} \sigma(x_1){=}t_1, \ldots, \sigma(x_n){=}t_n$
- $\boldsymbol{A\sigma}$: Anwendung von σ auf den Ausdruck A
- $\boldsymbol{\tau\,\sigma}$: Komposition von τ und σ (σ idempotent falls $\sigma\,\sigma = \sigma$)

$\lfloor x \rfloor[t/x]$	$= t$	$\lfloor x \rfloor[t/y]$	$= x$	$(y{\neq}x)$
$\lfloor f(t_1, .., t_n) \rfloor\sigma$	$= f(t_1\sigma, .., t_n\sigma)$	$\lfloor \mathsf{ff} \rfloor\sigma$	$= \mathsf{ff}$	
$\lfloor P(t_1, .., t_n) \rfloor\sigma$	$= P(t_1\sigma, .., t_n\sigma)$	$\lfloor t_1 \texttt{ = } t_2 \rfloor\sigma$	$= t_1\sigma \texttt{ = } t_2\sigma$	
$\lfloor \neg A \rfloor\sigma$	$= \neg A\sigma$	$\lfloor A \wedge B \rfloor\sigma$	$= A\sigma \wedge B\sigma$	
$\lfloor A \vee B \rfloor\sigma$	$= A\sigma \vee B\sigma$	$\lfloor A \Rightarrow B \rfloor\sigma$	$= A\sigma \Rightarrow B\sigma$	
$\lfloor (A) \rfloor\sigma$	$= (A\sigma)$			
$\lfloor \forall x{:}T\,.\,A \rfloor[t/x]$	$= \forall x{:}T\,.\,A$	$\lfloor \exists x{:}T\,.\,A \rfloor[t/x]$	$= \exists x{:}T\,.\,A$	
$\lfloor \forall x{:}T\,.\,A \rfloor[t/y]$	$= \lfloor \forall z{:}T\,.\,A[z/x] \rfloor[t/y]$	$\lfloor \exists x{:}T\,.\,A \rfloor[t/y]$	$= \lfloor \exists z{:}T\,.\,A[z/x] \rfloor[t/y]$	*

*: *$y{\neq}x$, y frei in A, x frei in t, z neue Variable*

Substitution $A[t/x]$ formal

Endliche Abbildung σ von Variablen in Terme

- $\boldsymbol{\sigma = [t_1, .., t_n/x_1, .., x_n]} \mathrel{\hat{=}} \sigma(x_1){=}t_1, \ldots, \sigma(x_n){=}t_n$
- $\boldsymbol{A\sigma}$: Anwendung von σ auf den Ausdruck A
- $\boldsymbol{\tau\,\sigma}$: Komposition von τ und σ (σ idempotent falls $\sigma\,\sigma = \sigma$)

$\lfloor x \rfloor[t/x]$	$= t$	$\lfloor x \rfloor[t/y]$	$= x$	$(y{\neq}x)$
$\lfloor f(t_1, .., t_n) \rfloor\sigma$	$= f(t_1\sigma, .., t_n\sigma)$	$\lfloor \mathsf{ff} \rfloor\sigma$	$= \mathsf{ff}$	
$\lfloor P(t_1, .., t_n) \rfloor\sigma$	$= P(t_1\sigma, .., t_n\sigma)$	$\lfloor t_1 \texttt{ = } t_2 \rfloor\sigma$	$= t_1\sigma \texttt{ = } t_2\sigma$	
$\lfloor \neg A \rfloor\sigma$	$= \neg A\sigma$	$\lfloor A \wedge B \rfloor\sigma$	$= A\sigma \wedge B\sigma$	
$\lfloor A \vee B \rfloor\sigma$	$= A\sigma \vee B\sigma$	$\lfloor A \Rightarrow B \rfloor\sigma$	$= A\sigma \Rightarrow B\sigma$	
$\lfloor (A) \rfloor\sigma$	$= (A\sigma)$			
$\lfloor \forall x{:}T\,.\,A \rfloor[t/x]$	$= \forall x{:}T\,.\,A$	$\lfloor \exists x{:}T\,.\,A \rfloor[t/x]$	$= \exists x{:}T\,.\,A$	
$\lfloor \forall x{:}T\,.\,A \rfloor[t/y]$	$= \lfloor \forall z{:}T\,.\,A[z/x] \rfloor[t/y]$	$\lfloor \exists x{:}T\,.\,A \rfloor[t/y]$	$= \lfloor \exists z{:}T\,.\,A[z/x] \rfloor[t/y]$	*
$\lfloor \forall x{:}T\,.\,A \rfloor[t/y]$	$= \forall x{:}T\,.\,\lfloor A[t/y] \rfloor$	$\lfloor \exists x{:}T\,.\,A \rfloor[t/y]$	$= \exists x{:}T\,.\,\lfloor A[t/y] \rfloor$	**

*: *$y{\neq}x$, y frei in A, x frei in t, z neue Variable*

**: *$y{\neq}x$, y nicht frei in A oder x nicht frei in t*

Substitution ausgewertet

$$\lfloor(\forall y{:}T.\ R(+(x,y)) \ \wedge\ \exists x{:}T.\ x{=}y) \ \wedge\ P(x)\rfloor[-(y,4)/x]$$

Substitution ausgewertet

$$\lfloor(\forall y{:}T.\ R(+(x,y)) \ \wedge\ \exists x{:}T.\ x{=}y) \ \wedge\ P(x)\rfloor[-(y,4)/x]$$

$$= \lfloor(\forall y{:}T.\ R(+(x,y)) \ \wedge\ \exists x{:}T.\ x{=}y)\rfloor[-(y,4)/x]$$

$$\wedge\ \lfloor P(x)\rfloor[-(y,4)/x]$$

Substitution ausgewertet

$$\lfloor(\forall y{:}T.\ R(+(x,y)) \ \wedge\ \exists x{:}T.\ x{=}y) \ \wedge\ P(x)\rfloor[-(y,4)/x]$$

$$= \lfloor(\forall y{:}T.\ R(+(x,y)) \ \wedge\ \exists x{:}T.\ x{=}y)\rfloor[-(y,4)/x]$$
$$\wedge\ \lfloor P(x)\rfloor[-(y,4)/x]$$

$$= (\forall z{:}T.\ \lfloor R(+(x,z)) \ \wedge\ \exists x{:}T.\ x{=}z\rfloor[-(y,4)/x])$$
$$\wedge\ P(-(y,4))$$

Substitution ausgewertet

$$\lfloor(\forall y{:}T.\ R(+(x,y)) \ \wedge\ \exists x{:}T.\ x{=}y) \ \wedge\ P(x)\rfloor[-(y,4)/x]$$

$$= \lfloor(\forall y{:}T.\ R(+(x,y)) \ \wedge\ \exists x{:}T.\ x{=}y)\rfloor[-(y,4)/x] \ \wedge\ \lfloor P(x)\rfloor[-(y,4)/x]$$

$$= (\forall z{:}T.\ \lfloor R(+(x,z)) \ \wedge\ \exists x{:}T.\ x{=}z\rfloor[-(y,4)/x]) \ \wedge\ P(-(y,4))$$

$$= (\forall z{:}T.\ \lfloor R(+(x,z))\rfloor[-(y,4)/x] \ \wedge\ \lfloor\exists x{:}T.\ x{=}z\rfloor[-(y,4)/x]) \ \wedge\ P(-(y,4))$$

Substitution ausgewertet

$$
\begin{aligned}
&\lfloor(\forall y{:}T.\ R(+(x,y)) \ \wedge\ \exists x{:}T.\ x{=}y) \ \wedge\ P(x)\rfloor[-(y,4)/x] \\
=\ &\lfloor(\forall y{:}T.\ R(+(x,y)) \ \wedge\ \exists x{:}T.\ x{=}y)\rfloor[-(y,4)/x] \\
&\wedge\ \lfloor P(x)\rfloor[-(y,4)/x] \\
=\ &(\forall z{:}T.\ \lfloor R(+(x,z)) \ \wedge\ \exists x{:}T.\ x{=}z\rfloor[-(y,4)/x]) \\
&\wedge\ P(-(y,4)) \\
=\ &(\forall z{:}T.\ \lfloor R(+(x,z))\rfloor[-(y,4)/x] \ \wedge\ \lfloor\exists x{:}T.\ x{=}z\rfloor[-(y,4)/x]) \\
&\wedge\ P(-(y,4)) \\
=\ &(\forall z{:}T.\ R(+(-(y,4),z)) \ \wedge\ \exists x{:}T.\ x{=}z) \ \wedge\ P(-(y,4))
\end{aligned}
$$

Refinement Logic: Prädikatenlogische Regeln

Simuliere $\iota_x^u(A)$ durch $\iota(A[t/x])$

– $\iota(\forall x : T \,.\, A) =$ wahr, wenn $\iota_x^u(A) =$ wahr für alle $u \in \iota(T)$

Refinement Logic: Prädikatenlogische Regeln

Simuliere $\iota_x^u(A)$ durch $\iota(A[t/x])$

– $\iota(\forall x : T \,.\, A) = \mathsf{wahr}$, wenn $\iota_x^u(A) = \mathsf{wahr}$ für alle $u \in \iota(T)$

– $\forall x : T \,.\, A$ ist gültig, wenn $A[x'/x]$ gültig ist für eine neue Variable x'

· Die Interpretation von x' ist nicht weiter festgelegt
also muß A für jede Zuordnung eines Objekts u zu x' wahr sein

Refinement Logic: Prädikatenlogische Regeln

Simuliere $\iota_x^u(A)$ durch $\iota(A[t/x])$

– $\iota(\forall x : T \,.\, A) = \mathsf{wahr}$, wenn $\iota_x^u(A) = \mathsf{wahr}$ für alle $u \in \iota(T)$

– $\forall x : T \,.\, A$ ist gültig, wenn $A[x'/x]$ gültig ist für eine neue Variable x'

  · Die Interpretation von x' ist nicht weiter festgelegt
    also muß A für jede Zuordnung eines Objekts u zu x' wahr sein

– $\iota(\exists x : T \,.\, A) = \mathsf{wahr}$, wenn $\iota(A[t/x]) = \mathsf{wahr}$ für einen Term t

Refinement Logic: Prädikatenlogische Regeln

Simuliere $\iota_x^u(A)$ durch $\iota(A[t/x])$

– $\iota(\forall x : T \,.\, A) = \mathsf{wahr}$, wenn $\iota_x^u(A) = \mathsf{wahr}$ für alle $u \in \iota(T)$

– $\forall x : T \,.\, A$ ist gültig, wenn $A[x'/x]$ gültig ist für eine neue Variable x'

  · Die Interpretation von x' ist nicht weiter festgelegt
    also muß A für jede Zuordnung eines Objekts u zu x' wahr sein

– $\iota(\exists x : T \,.\, A) = \mathsf{wahr}$, wenn $\iota(A[t/x]) = \mathsf{wahr}$ für einen Term t

– $\exists x : T \,.\, A$ ist gültig, wenn $A[t/x]$ gültig ist für einen Term t

Refinement Logic: Prädikatenlogische Regeln

Simuliere $\iota_x^u(A)$ durch $\iota(A[t/x])$

– $\iota(\forall x{:}T\,.\,A) = \mathsf{wahr}$, wenn $\iota_x^u(A) = \mathsf{wahr}$ für alle $u \in \iota(T)$

– $\forall x{:}T\,.\,A$ ist gültig, wenn $A[x'/x]$ gültig ist für eine neue Variable x'

· Die Interpretation von x' ist nicht weiter festgelegt
also muß A für jede Zuordnung eines Objekts u zu x' wahr sein

– $\iota(\exists x{:}T\,.\,A) = \mathsf{wahr}$, wenn $\iota(A[t/x]) = \mathsf{wahr}$ für einen Term t

– $\exists x{:}T\,.\,A$ ist gültig, wenn $A[t/x]$ gültig ist für einen Term t

	Elimination (links)	Introduktion (rechts)
`allE` i t	$\dfrac{\Gamma,\ \forall x{:}T\,.\,A,\ A[t/x],\ \Delta \vdash C}{\Gamma,\ \forall x{:}T\,.\,A,\ \Delta \vdash C}$	$\dfrac{\Gamma,\ x'{:}T \vdash A[x'/x]}{\Gamma \vdash \forall x{:}T\,.\,A}$ `allI` *
`exE` i **	$\dfrac{\Gamma,\ x'{:}T,\ A[x'/x],\ \Delta \vdash C}{\Gamma,\ \exists x{:}T\,.\,A,\ \Delta \vdash C}$	$\dfrac{\Gamma \vdash A[t/x]}{\Gamma \vdash \exists x{:}T\,.\,A}$ `exI` t

*: *Die Umbenennung $[x'/x]$ kann entfallen, wenn x nicht frei in Γ vorkommt*

**: *Die Umbenennung $[x'/x]$ kann entfallen, wenn x nicht frei in C, Γ, Δ vorkommt*

Beweis für Distributivität von $\wedge$ und $\forall$

$\forall x{:}T.\,P(x) \wedge Q(x) \vdash (\forall x{:}T.\,P(x)) \wedge (\forall x{:}T.\,Q(x))$ BY andI

- $\forall x{:}T.\,P(x) \wedge Q(x) \vdash \forall x{:}T.\,P(x)$ BY allI
  - $\forall x{:}T.\,P(x) \wedge Q(x),\ x{:}T \vdash P(x)$ BY allE 1 x
    - $\forall x{:}T.\,P(x) \wedge Q(x),\ x{:}T,\ P(x) \wedge Q(x) \vdash P(x)$ BY andE 3
      - $\forall x{:}T.\,P(x) \wedge Q(x),\ x{:}T,\ P(x),\ Q(x) \vdash P(x)$ BY hypothesis 3
- $\forall x{:}T.\,P(x) \wedge Q(x) \vdash \forall x{:}T.\,Q(x)$ BY allI
  - $\forall x{:}T.\,P(x) \wedge Q(x),\ x{:}T \vdash Q(x)$ BY allE 1 x
    - $\forall x{:}T.\,P(x) \wedge Q(x),\ x{:}T,\ P(x) \wedge Q(x) \vdash Q(x)$ BY andE 3
      - $\forall x{:}T.\,P(x) \wedge Q(x),\ x{:}T,\ P(x),\ Q(x) \vdash Q(x)$ BY hypothesis 4

Gleichheitsbehandlung im Sequenzenkalkül

`reflexivity`	$\Gamma \vdash t{=}t$
`symmetry`	$\Gamma \vdash t_1{=}t_2$ $\Gamma \vdash t_2{=}t_1$
`transitivity` u	$\Gamma \vdash t_1{=}t_2$ $\Gamma \vdash t_1{=}u$ $\Gamma \vdash u{=}t_2$
`substitution` $t_1{=}t_2$	$\Gamma \vdash C$ $\Gamma \vdash t_1{=}t_2$ $\Gamma \vdash C[t_2/t_1]$ (Termersetzung !)

Substitution subsumiert Regeln für Transitivität und Symmetrie

Sequenzenbeweis für Transitivität der Gleichheit

$\vdash \quad (t_1{=}u \ \wedge \ u{=}t_2) \ \Rightarrow \quad t_1{=}t_2$ BY impI

\

$t_1{=}u \ \wedge \ u{=}t_2 \ \vdash \quad t_1{=}t_2$ BY andE 1

\

$t_1{=}u, \quad u{=}t_2 \ \vdash \quad t_1{=}t_2$ BY substitution $t_1{=}u$

|\

| $t_1{=}u, \quad u{=}t_2 \ \vdash \quad t_1{=}u$ BY hypothesis 1

\

$t_1{=}u, \quad u{=}t_2 \ \vdash \quad u{=}t_2$ BY hypothesis 2

Refinement Logic: Korrektheit

Alle Theoreme der Refinement Logic sind gültig

- **C Theorem $\hat{=}$ $\vdash C$ hat vollständigen Beweis**
  - Knoten im Beweisbaum sind Regelanwendungen
  - Blätter sind Regelanwendungen ohne Teilziele (`falseE`, `hypothesis`)

Refinement Logic: Korrektheit

Alle Theoreme der Refinement Logic sind gültig

- **C Theorem $\hat{=} \vdash C$ hat vollständigen Beweis**
  - Knoten im Beweisbaum sind Regelanwendungen
  - Blätter sind Regelanwendungen ohne Teilziele (`falseE`, `hypothesis`)
- **Strukturelle Induktion über Beweisbaum**

  "Wenn $\Gamma \vdash C$ einen vollständigen Beweis hat, so ist $\Gamma \vdash C$ gültig"
  - Basisfall: Anwendung von `falseE` oder `hypothesis` liefert keine Teilziele
  - Schrittfall: Regelanwendung liefert Teilziele (mit vollständigem Beweis)

Refinement Logic: Korrektheit

Alle Theoreme der Refinement Logic sind gültig

- **C Theorem $\hat{=}$ $\vdash C$ hat vollständigen Beweis**
  - Knoten im Beweisbaum sind Regelanwendungen
  - Blätter sind Regelanwendungen ohne Teilziele (`falseE`, `hypothesis`)
- **Strukturelle Induktion über Beweisbaum**

  "Wenn $\Gamma \vdash C$ einen vollständigen Beweis hat, so ist $\Gamma \vdash C$ gültig"
  - Basisfall: Anwendung von `falseE` oder `hypothesis` liefert keine Teilziele
  - Schrittfall: Regelanwendung liefert Teilziele (mit vollständigem Beweis)

  In beiden Fällen ist nur die Korrektheit der angewandten Regel zu zeigen
  - *"$\Gamma \vdash C$ ist gültig, wenn alle Teilziele $\Gamma_i \vdash C_i$ gültig sind"*

Refinement Logic: Korrektheit

Alle Theoreme der Refinement Logic sind gültig

- **C Theorem $\hat{=} \vdash C$ hat vollständigen Beweis**
  - Knoten im Beweisbaum sind Regelanwendungen
  - Blätter sind Regelanwendungen ohne Teilziele (`falseE`, `hypothesis`)
- **Strukturelle Induktion über Beweisbaum**

  "Wenn $\Gamma \vdash C$ einen vollständigen Beweis hat, so ist $\Gamma \vdash C$ gültig"
  - Basisfall: Anwendung von `falseE` oder `hypothesis` liefert keine Teilziele
  - Schrittfall: Regelanwendung liefert Teilziele (mit vollständigem Beweis)

  In beiden Fällen ist nur die Korrektheit der angewandten Regel zu zeigen
  - *"$\Gamma \vdash C$ ist gültig, wenn alle Teilziele $\Gamma_i \vdash C_i$ gültig sind"*
- **Beispiel: Korrektheit von `impI`**

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \Rightarrow B}\ \texttt{BY impI}$$

Refinement Logic: Korrektheit

Alle Theoreme der Refinement Logic sind gültig

- **C Theorem $\hat{=} \vdash C$ hat vollständigen Beweis**
  - Knoten im Beweisbaum sind Regelanwendungen
  - Blätter sind Regelanwendungen ohne Teilziele (`falseE`, `hypothesis`)
- **Strukturelle Induktion über Beweisbaum**

  "Wenn $\Gamma \vdash C$ einen vollständigen Beweis hat, so ist $\Gamma \vdash C$ gültig"
  - Basisfall: Anwendung von `falseE` oder `hypothesis` liefert keine Teilziele
  - Schrittfall: Regelanwendung liefert Teilziele (mit vollständigem Beweis)

  In beiden Fällen ist nur die Korrektheit der angewandten Regel zu zeigen
  - *"$\Gamma \vdash C$ ist gültig, wenn alle Teilziele $\Gamma_i \vdash C_i$ gültig sind"*
- **Beispiel: Korrektheit von `impI`**
  - Sei $\Gamma, A \vdash B$ gültig, ι beliebig

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \Rightarrow B}\ \texttt{BY impI}$$

Refinement Logic: Korrektheit

Alle Theoreme der Refinement Logic sind gültig

- **C Theorem $\hat{=} \vdash C$ hat vollständigen Beweis**
  - Knoten im Beweisbaum sind Regelanwendungen
  - Blätter sind Regelanwendungen ohne Teilziele (`falseE`, `hypothesis`)
- **Strukturelle Induktion über Beweisbaum**

  "Wenn $\Gamma \vdash C$ einen vollständigen Beweis hat, so ist $\Gamma \vdash C$ gültig"
  - Basisfall: Anwendung von `falseE` oder `hypothesis` liefert keine Teilziele
  - Schrittfall: Regelanwendung liefert Teilziele (mit vollständigem Beweis)

  In beiden Fällen ist nur die Korrektheit der angewandten Regel zu zeigen
  - *"$\Gamma \vdash C$ ist gültig, wenn alle Teilziele $\Gamma_i \vdash C_i$ gültig sind"*
- **Beispiel: Korrektheit von `impI`**
  - Sei $\Gamma, A \vdash B$ gültig, ι beliebig
  - dann ist $\iota(\Gamma, A \vdash B)$=wahr

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \Rightarrow B}\ \texttt{BY impI}$$

Refinement Logic: Korrektheit

Alle Theoreme der Refinement Logic sind gültig

- **C Theorem $\hat{=} \vdash C$ hat vollständigen Beweis**
  - Knoten im Beweisbaum sind Regelanwendungen
  - Blätter sind Regelanwendungen ohne Teilziele (`falseE`, `hypothesis`)
- **Strukturelle Induktion über Beweisbaum**

  "Wenn $\Gamma \vdash C$ einen vollständigen Beweis hat, so ist $\Gamma \vdash C$ gültig"
  - Basisfall: Anwendung von `falseE` oder `hypothesis` liefert keine Teilziele
  - Schrittfall: Regelanwendung liefert Teilziele (mit vollständigem Beweis)

  In beiden Fällen ist nur die Korrektheit der angewandten Regel zu zeigen
  - *"$\Gamma \vdash C$ ist gültig, wenn alle Teilziele $\Gamma_i \vdash C_i$ gültig sind"*
- **Beispiel: Korrektheit von `impI`**
  - Sei $\Gamma, A \vdash B$ gültig, ι beliebig
  - dann ist $\iota(\Gamma, A \vdash B)$=wahr
  - also $\iota(B)$=wahr, wann immer $\iota(\Gamma)$=wahr und $\iota(A)$=wahr

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \Rightarrow B}\ \texttt{BY impI}$$

Refinement Logic: Korrektheit

Alle Theoreme der Refinement Logic sind gültig

- **C Theorem $\hat{=} \vdash C$ hat vollständigen Beweis**
  - Knoten im Beweisbaum sind Regelanwendungen
  - Blätter sind Regelanwendungen ohne Teilziele (`falseE`, `hypothesis`)
- **Strukturelle Induktion über Beweisbaum**

  "Wenn $\Gamma \vdash C$ einen vollständigen Beweis hat, so ist $\Gamma \vdash C$ gültig"
  - Basisfall: Anwendung von `falseE` oder `hypothesis` liefert keine Teilziele
  - Schrittfall: Regelanwendung liefert Teilziele (mit vollständigem Beweis)

  In beiden Fällen ist nur die Korrektheit der angewandten Regel zu zeigen
  - *"$\Gamma \vdash C$ ist gültig, wenn alle Teilziele $\Gamma_i \vdash C_i$ gültig sind"*
- **Beispiel: Korrektheit von `impI`**
  - Sei $\Gamma, A \vdash B$ gültig, ι beliebig
  - dann ist $\iota(\Gamma, A \vdash B)$=wahr
  - also $\iota(B)$=wahr, wann immer $\iota(\Gamma)$=wahr und $\iota(A)$=wahr
  - also $\iota(A \Rightarrow B)$=wahr, wann immer $\iota(\Gamma)$=wahr

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \Rightarrow B}\ \texttt{BY impI}$$

Refinement Logic: Korrektheit

Alle Theoreme der Refinement Logic sind gültig

- **C Theorem $\hat{=} \vdash C$ hat vollständigen Beweis**
  - Knoten im Beweisbaum sind Regelanwendungen
  - Blätter sind Regelanwendungen ohne Teilziele (`falseE`, `hypothesis`)
- **Strukturelle Induktion über Beweisbaum**

  "Wenn $\Gamma \vdash C$ einen vollständigen Beweis hat, so ist $\Gamma \vdash C$ gültig"
  - Basisfall: Anwendung von `falseE` oder `hypothesis` liefert keine Teilziele
  - Schrittfall: Regelanwendung liefert Teilziele (mit vollständigem Beweis)

  In beiden Fällen ist nur die Korrektheit der angewandten Regel zu zeigen
  - *"$\Gamma \vdash C$ ist gültig, wenn alle Teilziele $\Gamma_i \vdash C_i$ gültig sind"*
- **Beispiel: Korrektheit von `impI`**

  $$\frac{\Gamma \vdash A \Rightarrow B \quad \texttt{BY impI}}{\Gamma, A \vdash B}$$

  - Sei $\Gamma, A \vdash B$ gültig, ι beliebig
  - dann ist $\iota(\Gamma, A \vdash B)$=wahr
  - also $\iota(B)$=wahr, wann immer $\iota(\Gamma)$=wahr und $\iota(A)$=wahr
  - also $\iota(A \Rightarrow B)$=wahr, wann immer $\iota(\Gamma)$=wahr
  - also $\iota(\Gamma \vdash A \Rightarrow B)$=wahr $\surd$

Refinement Logic: Vollständigkeit

Alle gültigen Formeln haben einen vollständigen Beweis

- **Nicht jeder Sequenzenbeweisansatz terminiert**

  . . . auch nicht wenn Formel gültig ist

Refinement Logic: Vollständigkeit

Alle gültigen Formeln haben einen vollständigen Beweis

- **Nicht jeder Sequenzenbeweisansatz terminiert**

  . . . auch nicht wenn Formel gültig ist

$\forall x{:}T.\,P(x) \wedge Q(x) \vdash (\forall x{:}T.\,P(x)) \wedge (\forall x{:}T.\,Q(x))$ BY allE 1 x

$\forall x{:}T.\,P(x) \wedge Q(x),\ P(x) \wedge Q(x) \vdash (\forall x{:}T.\,P(x)) \wedge (\forall x{:}T.\,Q(x))$ BY andE 2

$\forall x{:}T.\,P(x) \wedge Q(x),\ P(x),\ Q(x) \vdash (\forall x{:}T.\,P(x)) \wedge (\forall x{:}T.\,Q(x))$ BY andI

$\forall x{:}T.\,P(x) \wedge Q(x),\ P(x),\ Q(x) \vdash \forall x{:}T.\,P(x)$ BY allI

$\forall x{:}T.\,P(x) \wedge Q(x),\ P(x),\ Q(x),\ x'{:}T \vdash P(x')$ BY ?????

$\forall x{:}T.\,P(x) \wedge Q(x),\ P(x),\ Q(x) \vdash \forall x{:}T.\,Q(x)$ BY allI

$\forall x{:}T.\,P(x) \wedge Q(x),\ P(x),\ Q(x),\ x'{:}T \vdash Q(x')$ BY ?????

Refinement Logic: Vollständigkeit

Alle gültigen Formeln haben einen vollständigen Beweis

- **Nicht jeder Sequenzenbeweisansatz terminiert**
  ... auch nicht wenn Formel gültig ist
- **Beschreibe systematische Beweisprozedur** (ineffizient!)
  – Methode zählt alle möglichen Substitutionen für alle Quantoren auf
  – Sequenzenbeweis ist unendlich, wenn Prozedur nicht terminieren kann

Refinement Logic: Vollständigkeit

Alle gültigen Formeln haben einen vollständigen Beweis

- **Nicht jeder Sequenzenbeweisansatz terminiert**
  ...auch nicht wenn Formel gültig ist
- **Beschreibe systematische Beweisprozedur** (ineffizient!)
  – Methode zählt alle möglichen Substitutionen für alle Quantoren auf
  – Sequenzenbeweis ist unendlich, wenn Prozedur nicht terminieren kann
- **Systematische Prozedur liefert "Entscheidung"**
  – Unendliche Sequenzenbeweise haben offene Zweige

Refinement Logic: Vollständigkeit

Alle gültigen Formeln haben einen vollständigen Beweis

- **Nicht jeder Sequenzenbeweisansatz terminiert**
  ...auch nicht wenn Formel gültig ist
- **Beschreibe systematische Beweisprozedur** (ineffizient!)
  - Methode zählt alle möglichen Substitutionen für alle Quantoren auf
  - Sequenzenbeweis ist unendlich, wenn Prozedur nicht terminieren kann
- **Systematische Prozedur liefert "Entscheidung"**
  - Unendliche Sequenzenbeweise haben offene Zweige
  - Offene Zweige in Sequenzenbeweisen sind uniform widerlegbar
    - Interpretiere jede atomare Formel links als wahr, rechts als falsch

Refinement Logic: Vollständigkeit

Alle gültigen Formeln haben einen vollständigen Beweis

- **Nicht jeder Sequenzenbeweisansatz terminiert**
  ...auch nicht wenn Formel gültig ist
- **Beschreibe systematische Beweisprozedur** (ineffizient!)
  - Methode zählt alle möglichen Substitutionen für alle Quantoren auf
  - Sequenzenbeweis ist unendlich, wenn Prozedur nicht terminieren kann
- **Systematische Prozedur liefert "Entscheidung"**
  - Unendliche Sequenzenbeweise haben offene Zweige
  - Offene Zweige in Sequenzenbeweisen sind uniform widerlegbar
    · Interpretiere jede atomare Formel links als wahr, rechts als falsch

Gültige Formeln haben (endliche) vollständige Beweise ✓

Refinement Logic: Vollständigkeit

Alle gültigen Formeln haben einen vollständigen Beweis

- **Nicht jeder Sequenzenbeweisansatz terminiert**
  ...auch nicht wenn Formel gültig ist
- **Beschreibe systematische Beweisprozedur** (ineffizient!)
  - Methode zählt alle möglichen Substitutionen für alle Quantoren auf
  - Sequenzenbeweis ist unendlich, wenn Prozedur nicht terminieren kann
- **Systematische Prozedur liefert "Entscheidung"**
  - Unendliche Sequenzenbeweise haben offene Zweige
  - Offene Zweige in Sequenzenbeweisen sind uniform widerlegbar
    · Interpretiere jede atomare Formel links als wahr, rechts als falsch

  Gültige Formeln haben (endliche) vollständige Beweise ✓
- **Details des Vollständigkeitsbeweises sind aufwendig**
  - Hintikka Konstruktion für Kalkül $\mathcal{LK}$ mit mehreren Konklusionen
  - Transformation: $\mathcal{LK}$ Beweise $\mapsto$ Sequenzenkalkül mit einer Konklusionen

Taktiken und Beweisstrategien

- **Konservative Erweiterung des Regelsystems**
  - Taktiken sind definitorische Abkürzung für komplexe Regelanwendungen

Taktiken und Beweisstrategien

- **Konservative Erweiterung des Regelsystems**
  - Taktiken sind definitorische Abkürzung für komplexe Regelanwendungen
- **Erzeugung durch einfache Verknüpfung von Regeln**

Taktiken und Beweisstrategien

- **Konservative Erweiterung des Regelsystems**
  - Taktiken sind definitorische Abkürzung für komplexe Regelanwendungen
- **Erzeugung durch einfache Verknüpfung von Regeln**
  - Formale metasprachliche Ausdrücke liefern einfache Taktiksprache (ML)

    r_1 `THEN` r_2: Führe Regel r_2 direkt nach r_1 aus

    r_1 `ORELSE` r_2: Führe Regel r_2 aus, wenn r_1 nicht anwendbar ist

    `Repeat` r: Führe Regel r solange wie möglich aus

Taktiken und Beweisstrategien

- **Konservative Erweiterung des Regelsystems**
  - Taktiken sind definitorische Abkürzung für komplexe Regelanwendungen
- **Erzeugung durch einfache Verknüpfung von Regeln**
  - Formale metasprachliche Ausdrücke liefern einfache Taktiksprache (ML)

    r_1 `THEN` r_2: Führe Regel r_2 direkt nach r_1 aus

    r_1 `ORELSE` r_2: Führe Regel r_2 aus, wenn r_1 nicht anwendbar ist

    `Repeat` r: Führe Regel r solange wie möglich aus
  - Formulierung komplexer Taktiken und Beweisstrategien ebenfalls möglich

Taktiken und Beweisstrategien

- **Konservative Erweiterung des Regelsystems**
  - Taktiken sind definitorische Abkürzung für komplexe Regelanwendungen
- **Erzeugung durch einfache Verknüpfung von Regeln**
  - Formale metasprachliche Ausdrücke liefern einfache Taktiksprache (ML)

    r_1 `THEN` r_2: Führe Regel r_2 direkt nach r_1 aus

    r_1 `ORELSE` r_2: Führe Regel r_2 aus, wenn r_1 nicht anwendbar ist

    `Repeat` r: Führe Regel r solange wie möglich aus
  - Formulierung komplexer Taktiken und Beweisstrategien ebenfalls möglich
- **Keine Vergrößerung der Beweiskraft**
  - Beweise mit elementaren Regeln durch Expansion rekonstruierbar
  - Ermöglicht gleichzeitig ein kleines Grundinferenzsystem
    und flexible Präsentation von Beweisen (kürzer, bessere Struktur)

Sequenzenkalkül: Beweismethodik

- **Kalkül garantiert Korrektheit formaler Beweise**
  - Keine Methode um Beweise zu finden
  - Systematische Beweisprozedur im Vollständigkeitsbeweis ist ineffizient

Sequenzenkalkül: Beweismethodik

- **Kalkül garantiert Korrektheit formaler Beweise**
  - Keine Methode um Beweise zu finden
  - Systematische Beweisprozedur im Vollständigkeitsbeweis ist ineffizient
- **Es gibt Leitlinien für erfolgreiche Beweissuche**
  - Versuche vorrangig Zweige abzuschließen (`falseE`, `hypothesis`)
  - Verwende Dekompositionsregeln, die Formeln äquivalent aufbrechen
  - Verwende `orE` vor `orI1` / `orI2`
  - Verwende `exE` und `allI` vor `exI` und `allE`
  - Wähle anwendbare Regel, welche die wenigsten Teilziele erzeugt

Sequenzenkalkül: Beweismethodik

- **Kalkül garantiert Korrektheit formaler Beweise**
  - Keine Methode um Beweise zu finden
  - Systematische Beweisprozedur im Vollständigkeitsbeweis ist ineffizient
- **Es gibt Leitlinien für erfolgreiche Beweissuche**
  - Versuche vorrangig Zweige abzuschließen (`falseE`, `hypothesis`)
  - Verwende Dekompositionsregeln, die Formeln äquivalent aufbrechen
  - Verwende `orE` vor `orI1` / `orI2`
  - Verwende `exE` und `allI` vor `exI` und `allE`
  - Wähle anwendbare Regel, welche die wenigsten Teilziele erzeugt

  Methodik ist als Taktik `prover` programmierbar

Sequenzenkalkül: Beweismethodik

- **Kalkül garantiert Korrektheit formaler Beweise**
  - Keine Methode um Beweise zu finden
  - Systematische Beweisprozedur im Vollständigkeitsbeweis ist ineffizient
- **Es gibt Leitlinien für erfolgreiche Beweissuche**
  - Versuche vorrangig Zweige abzuschließen (`falseE`, `hypothesis`)
  - Verwende Dekompositionsregeln, die Formeln äquivalent aufbrechen
  - Verwende `orE` vor `orI1` / `orI2`
  - Verwende `exE` und `allI` vor `exI` und `allE`
  - Wähle anwendbare Regel, welche die wenigsten Teilziele erzeugt

  Methodik ist als Taktik `prover` programmierbar
- **Beweismethodik läßt Fragen offen**
  - Auswahl der Substitution für Quantoren erfordert "Vorausschau"
  - Maschinennahe Methoden finden Substitution durch Unifikation

Sequenzenkalkül: Beweismethodik

- **Kalkül garantiert Korrektheit formaler Beweise**
  - Keine Methode um Beweise zu finden
  - Systematische Beweisprozedur im Vollständigkeitsbeweis ist ineffizient
- **Es gibt Leitlinien für erfolgreiche Beweissuche**
  - Versuche vorrangig Zweige abzuschließen (`falseE`, `hypothesis`)
  - Verwende Dekompositionsregeln, die Formeln äquivalent aufbrechen
  - Verwende `orE` vor `orI1` / `orI2`
  - Verwende `exE` und `allI` vor `exI` und `allE`
  - Wähle anwendbare Regel, welche die wenigsten Teilziele erzeugt

  Methodik ist als Taktik `prover` programmierbar
- **Beweismethodik läßt Fragen offen**
  - Auswahl der Substitution für Quantoren erfordert "Vorausschau"
  - Maschinennahe Methoden finden Substitution durch Unifikation

  Mehr dazu in der Vorlesung "Inferenzmethoden"

Prädikatenlogik – Grenzen

- **Universelle Sprache mit wenigen Vorgaben**
  - Flexibel, aber zu wenig Struktur (nur logische Konnektive)

Prädikatenlogik – Grenzen

- **Universelle Sprache mit wenigen Vorgaben**
  - Flexibel, aber zu wenig Struktur (nur logische Konnektive)
- **Keine Schlüsse über Werte von Termen möglich**
  - Interpretation von Gleichheit (z.B. `4+4=8`) ist nicht festgelegt

Prädikatenlogik – Grenzen

- **Universelle Sprache mit wenigen Vorgaben**
  - Flexibel, aber zu wenig Struktur (nur logische Konnektive)
- **Keine Schlüsse über Werte von Termen möglich**
  - Interpretation von Gleichheit (z.B. `4+4=8`) ist nicht festgelegt
- **Kein Schließen über Datentypen möglich**
  - Welche Struktur und welche Elemente hat ein Datentyp?
  - Interpretation von $\forall \texttt{x}:\mathbb{N}.\ \texttt{x=0} \vee \texttt{x} \geq \texttt{1}$ nicht festgelegt

Prädikatenlogik – Grenzen

- **Universelle Sprache mit wenigen Vorgaben**
  - – Flexibel, aber zu wenig Struktur (nur logische Konnektive)
- **Keine Schlüsse über Werte von Termen möglich**
  - – Interpretation von Gleichheit (z.B. `4+4=8`) ist nicht festgelegt
- **Kein Schließen über Datentypen möglich**
  - – Welche Struktur und welche Elemente hat ein Datentyp?
  - – Interpretation von $\forall \texttt{x}:\mathbb{N}.\ \texttt{x=0} \vee \texttt{x} \geq \texttt{1}$ nicht festgelegt
- **Erweiterung durch Axiome unpraktisch**
  - \+ alle Eigenschaften der Logik bleiben erhalten
  - – Formales Schließen mühsam (vgl. Hilberttyp-Kalküle)

Prädikatenlogik – Grenzen

- **Universelle Sprache mit wenigen Vorgaben**
  - – Flexibel, aber zu wenig Struktur (nur logische Konnektive)
- **Keine Schlüsse über Werte von Termen möglich**
  - – Interpretation von Gleichheit (z.B. `4+4=8`) ist nicht festgelegt
- **Kein Schließen über Datentypen möglich**
  - – Welche Struktur und welche Elemente hat ein Datentyp?
  - – Interpretation von $\forall \texttt{x}:\mathbb{N}.\ \texttt{x=0} \vee \texttt{x} \geq \texttt{1}$ nicht festgelegt
- **Erweiterung durch Axiome unpraktisch**
  - \+ alle Eigenschaften der Logik bleiben erhalten
  - – Formales Schließen mühsam (vgl. Hilberttyp-Kalküle)

$\Downarrow$

Erweitere Semantik und Inferenzsystem

- \+ Formales Schließen "natürlich" und einfacher
- – Mehr Theorie: Korrektheit, Vollständigkeit etc. neu zu beweisen