

# Automatisierte Logik und Programmierung

## Einheit 3

### Formale Logik (Teil 2)



1. Formalisierung in Aussagenlogik
2. Evidenz für logische Aussagen
3. Evidenzkonstruktion mit Refinement Logik
4. Formale Prädikatenlogik
5. Metamathematik der Refinement Logik

## Das übliche Verständnis des Begriffs “Logik”

- **Ermöglicht Formulierung universeller Zusammenhänge**

... und ihre Anwendung auf Individuen

*“Jeder Mensch ist sterblich.*

$((\forall x)(Human(x) \Rightarrow Mortal(x))$

*Sokrates ist ein Mensch.*

$\wedge Human(sokrates))$

*Also ist Sokrates sterblich”*

$\Rightarrow Mortal(sokrates)$

- **Unterstützt unterspezifizierte Aussagen und Funktionen**

*“Studierende, die mindestens 120 Leistungspunkte erworben haben,  
können ein Thema für die Bachelorarbeit bekommen”*

$(\forall st) (lp(s) \geq 120 \Rightarrow (\exists t) (BA(t) \wedge Bekommt(s, t)))$

- **Erweiterung der Aussagenlogik**

– Syntax wird ergänzt um Variablen, Funktionen, und Quantoren

– Neue Konzepte: Bindungsbereich, Variablenvorkommen und Substitution

## • Erlaubte Symbole

- **Variablen**  $x, y, z, x_0, y_0, \dots$
- **Funktionssymbole**  $f, g, h, a, b, c, f_0, g_0, \dots$  (mit Stelligkeit,  $a, b, c$  oft nullstellig)
- **Prädikatssymbole**  $P, Q, R, P_0, Q_0, R_0, \dots$  (mit Stelligkeit)
- **Logische Symbole**  $f, \neg, \wedge, \vee, \Rightarrow, \forall, \exists$  und Klammern

## • Terme: Syntax für individuelle Objekte

- Variablen und nullstellige Funktionen (**Konstante**) sind (atomare) Terme
- Sind  $t_1, \dots, t_n$  Terme und  $f$   $n$ -stellige Funktion, dann ist  $f(t_1, \dots, t_n)$  Term

## • Formeln: Syntax für Aussagen

- $f$  und nullstellige Prädikate (**Aussagenvariablen**) sind (atomare) Formeln
- $P(t_1, \dots, t_n)$  ist (atomare) Formel ( $t_1, \dots, t_n$  Terme,  $P$   $n$ -stelliges Prädikat)
- Sind  $A$  und  $B$  Formeln, dann auch  $\neg A$ ,  $(A \Rightarrow B)$ ,  $(A \wedge B)$ ,  $(A \vee B)$
- Ist  $B$  Formel und  $x$  eine Variable, dann sind  $(\forall x)B$  und  $(\exists x)B$  Formeln

Bindungsbereich des Quantors (**Scope**) ist die kürzeste Formel, die auf den Quantor folgt

Später: alternative Notationen  $\forall x. B$  und  $\exists x. B$  und Konventionen, Klammern zu sparen

- **Evidenz für Gültigkeit von Formeln**

- Formuliert als Terme in erweiterter  $\lambda$ -Notation (Einheit 5)
- Konstruktion von Evidenz folgt induktivem Aufbau der Syntax

- **Evidenz für atomare Formeln**

- $f$  hat keine Evidenz  $[f] = \{\}$
- $A = P(t_1, ..t_n)$  steht für unbekannte Aussagen  $[A]$  **unspezifiziert**

- **“Aussagenlogische” Evidenzkonstruktion wie zuvor**

- Implikation  $[A \Rightarrow B] = [A] \rightarrow [B]$
- Konjunktion  $[A \wedge B] = [A] \times [B]$
- Disjunktion  $[A \vee B] = [A] + [B]$
- Negation  $[\neg A] = [A] \rightarrow \{\}$

**Wie konstruiert man Evidenz für quantifizierte Formeln?**

# EVIDENZ FÜR UNIVERSELLE QUANTIFIKATION

- $(\forall x)B$ : “Für alle  $x$  gilt  $B$ ”  $[(\forall x)B] = x : \mathbb{U} \rightarrow [B]$ 
  - Für jede Instanz von  $x$  muß eine Evidenz  $b$  für  $B$  konstruiert werden
  - Evidenz für  $(\forall x)B$  muß Funktion  $f$  sein mit  $f(x) : [B]$  für alle  $x$
  - Eingabe  $x$  für  $f$  stammt aus einem **Universum** von Objekten  $\mathbb{U}$
  - Ausgabotyp  $[B]$  von  $f$  kann von konkretem Eingabewert  $x : \mathbb{U}$  abhängen  
z.B.  $B = (P a \Rightarrow P x)$  hat genau dann Evidenz, wenn  $x$  mit  $a$  instantiiert
  - Typ der Evidenzen für  $(\forall x)B$  ist ein “abhängiger” Funktionenraum

# EVIDENZ FÜR UNIVERSELLE QUANTIFIKATION

- $(\forall x)B$ : “Für alle  $x$  gilt  $B$ ”  $[(\forall x)B] = x : \mathbb{U} \rightarrow [B]$ 
  - Für jede Instanz von  $x$  muß eine Evidenz  $b$  für  $B$  konstruiert werden
  - Evidenz für  $(\forall x)B$  muß Funktion  $f$  sein mit  $f(x) : [B]$  für alle  $x$
  - Eingabe  $x$  für  $f$  stammt aus einem **Universum** von Objekten  $\mathbb{U}$
  - Ausgabotyp  $[B]$  von  $f$  kann von konkretem Eingabewert  $x : \mathbb{U}$  abhängen  
z.B.  $B = (P a \Rightarrow P x)$  hat genau dann Evidenz, wenn  $x$  mit  $a$  instantiiert
  - Typ der Evidenzen für  $(\forall x)B$  ist ein “abhängiger” Funktionenraum
- **Konkrete Evidenz für  $(\forall x)(P x \Rightarrow P x)$**

# EVIDENZ FÜR UNIVERSELLE QUANTIFIKATION

- **$(\forall x)B$ : “Für alle  $x$  gilt  $B$ ”**  $[(\forall x)B] = x : \mathbb{U} \rightarrow [B]$ 
  - Für jede Instanz von  $x$  muß eine Evidenz  $b$  für  $B$  konstruiert werden
  - Evidenz für  $(\forall x)B$  muß Funktion  $f$  sein mit  $f(x) : [B]$  für alle  $x$
  - Eingabe  $x$  für  $f$  stammt aus einem **Universum** von Objekten  $\mathbb{U}$
  - Ausgabotyp  $[B]$  von  $f$  kann von konkretem Eingabewert  $x : \mathbb{U}$  abhängen  
z.B.  $B = (P a \Rightarrow P x)$  hat genau dann Evidenz, wenn  $x$  mit  $a$  instantiiert
  - Typ der Evidenzen für  $(\forall x)B$  ist ein “abhängiger” Funktionenraum
- **Konkrete Evidenz für  $(\forall x)(P x \Rightarrow P x)$** 
  - Evidenz ist Funktion  $f : (x:\mathbb{U} \rightarrow ([P x] \rightarrow [P x]))$ , wobei für alle  $x$  gilt  
 $f(x) = g_x : [P x] \rightarrow [P x]$  und  $q_x(p) : [P x]$ , falls  $p : [P x]$
  - Einfachste Lösung ist  $g_x(p) = p$ , also  $f = \lambda x. (\lambda p. p)$

# EVIDENZ FÜR UNIVERSELLE QUANTIFIKATION

- **$(\forall x)B$ : “Für alle  $x$  gilt  $B$ ”**  $[(\forall x)B] = x : \mathbb{U} \rightarrow [B]$ 
  - Für jede Instanz von  $x$  muß eine Evidenz  $b$  für  $B$  konstruiert werden
  - Evidenz für  $(\forall x)B$  muß Funktion  $f$  sein mit  $f(x) : [B]$  für alle  $x$
  - Eingabe  $x$  für  $f$  stammt aus einem **Universum** von Objekten  $\mathbb{U}$
  - Ausgabotyp  $[B]$  von  $f$  kann von konkretem Eingabewert  $x : \mathbb{U}$  abhängen  
z.B.  $B = (P a \Rightarrow P x)$  hat genau dann Evidenz, wenn  $x$  mit  $a$  instantiiert
  - Typ der Evidenzen für  $(\forall x)B$  ist ein “abhängiger” Funktionenraum
- **Konkrete Evidenz für  $(\forall x)(P x \Rightarrow P x)$** 
  - Evidenz ist Funktion  $f : (x:\mathbb{U} \rightarrow ([P x] \rightarrow [P x]))$ , wobei für alle  $x$  gilt  
 $f(x) = g_x : [P x] \rightarrow [P x]$  und  $q_x(p) : [P x]$ , falls  $p : [P x]$
  - Einfachste Lösung ist  $g_x(p) = p$ , also  $f = \lambda x. (\lambda p. p)$
- **Konkrete Evidenz für  $((\forall x)P x) \Rightarrow P a$**



# EVIDENZ FÜR UNIVERSELLE QUANTIFIKATION

- **$(\forall x)B$ : “Für alle  $x$  gilt  $B$ ”**  $[(\forall x)B] = x : \mathbb{U} \rightarrow [B]$ 
  - Für jede Instanz von  $x$  muß eine Evidenz  $b$  für  $B$  konstruiert werden
  - Evidenz für  $(\forall x)B$  muß Funktion  $f$  sein mit  $f(x) : [B]$  für alle  $x$
  - Eingabe  $x$  für  $f$  stammt aus einem **Universum** von Objekten  $\mathbb{U}$
  - Ausgabebetyp  $[B]$  von  $f$  kann von konkretem Eingabewert  $x : \mathbb{U}$  abhängen  
z.B.  $B = (P a \Rightarrow P x)$  hat genau dann Evidenz, wenn  $x$  mit  $a$  instantiiert
  - Typ der Evidenzen für  $(\forall x)B$  ist ein “abhängiger” Funktionenraum
- **Konkrete Evidenz für  $(\forall x)(P x \Rightarrow P x)$** 
  - Evidenz ist Funktion  $f : (x:\mathbb{U} \rightarrow ([P x] \rightarrow [P x]))$ , wobei für alle  $x$  gilt  
 $f(x) = g_x : [P x] \rightarrow [P x]$  und  $q_x(p) : [P x]$ , falls  $p : [P x]$
  - Einfachste Lösung ist  $g_x(p) = p$ , also  $f = \lambda x. (\lambda p. p)$
- **Konkrete Evidenz für  $((\forall x)P x) \Rightarrow P a$** 
  - Evidenz ist Funktion  $f : (x:\mathbb{U} \rightarrow [P x]) \rightarrow [P a]$  mit  $f(h) = x_h : [P a]$   
für alle  $h : (x:\mathbb{U} \rightarrow [P x])$ .
  - Einfachste Lösung ist Anwendung von  $h$  auf Konstante  $a$   $f = \lambda h. h(a)$

# EVIDENZ FÜR EXISTENTIELLE QUANTIFIKATION

- $(\exists x)B$ : “Es gibt ein  $x$ , für das  $B$  gilt”  $[(\exists x)B] = x : \mathbb{U} \times [B]$ 
  - Um Evidenz für  $(\exists x)B$  zu konstruieren, braucht man  $b : [B]$  für ein  $x : \mathbb{U}$
  - Formel  $B$  kann von Wahl des konkreten Wertes für  $x$  abhängen
  - Typ der Evidenzen für  $(\exists x)B$  ist ein “abhängiger” Produktraum

# EVIDENZ FÜR EXISTENTIELLE QUANTIFIKATION

- $(\exists x)B$ : “Es gibt ein  $x$ , für das  $B$  gilt”  $[(\exists x)B] = x : \mathbb{U} \times [B]$ 
  - Um Evidenz für  $(\exists x)B$  zu konstruieren, braucht man  $b : [B]$  für ein  $x : \mathbb{U}$
  - Formel  $B$  kann von Wahl des konkreten Wertes für  $x$  abhängen
  - Typ der Evidenzen für  $(\exists x)B$  ist ein “abhängiger” Produktraum
- **Konkrete Evidenz für  $P a \Rightarrow ((\exists x)P x)$**

# EVIDENZ FÜR EXISTENTIELLE QUANTIFIKATION

- **$(\exists x)B$** : “Es gibt ein  $x$ , für das  $B$  gilt”  $[(\exists x)B] = x : \mathbb{U} \times [B]$ 
  - Um Evidenz für  $(\exists x)B$  zu konstruieren, braucht man  $b : [B]$  für ein  $x : \mathbb{U}$
  - Formel  $B$  kann von Wahl des konkreten Wertes für  $x$  abhängen
  - Typ der Evidenzen für  $(\exists x)B$  ist ein “abhängiger” Produktraum
- **Konkrete Evidenz für  $P a \Rightarrow ((\exists x)P x)$** 
  - Evidenz ist Funktion  $f : (p:[P a] \rightarrow (x : \mathbb{U} \times [P x]))$  mit  $f(p) = (x, p')$  für alle  $p : [P a]$ , wobei  $x : \mathbb{U}$  und  $p'$  Evidenz für  $P x$
  - Einfachste Lösung ist  $x = a$  and  $p' = p$   $f = \lambda p. (a, p)$

# EVIDENZ FÜR EXISTENTIELLE QUANTIFIKATION

- **$(\exists x)B$ : “Es gibt ein  $x$ , für das  $B$  gilt”**  $[(\exists x)B] = x : \mathbb{U} \times [B]$ 
  - Um Evidenz für  $(\exists x)B$  zu konstruieren, braucht man  $b : [B]$  für ein  $x : \mathbb{U}$
  - Formel  $B$  kann von Wahl des konkreten Wertes für  $x$  abhängen
  - Typ der Evidenzen für  $(\exists x)B$  ist ein “abhängiger” Produktraum
- **Konkrete Evidenz für  $P a \Rightarrow ((\exists x)P x)$** 
  - Evidenz ist Funktion  $f : (p:[P a] \rightarrow (x : \mathbb{U} \times [P x]))$  mit  $f(p) = (x, p')$  für alle  $p : [P a]$ , wobei  $x : \mathbb{U}$  und  $p'$  Evidenz für  $P x$
  - Einfachste Lösung ist  $x = a$  and  $p' = p$   $f = \lambda p. (a, p)$
- **Konkrete Evidenz für  $((\exists x)P x) \Rightarrow ((\exists y)P y)$**

# EVIDENZ FÜR EXISTENTIELLE QUANTIFIKATION

- **$(\exists x)B$ : “Es gibt ein  $x$ , für das  $B$  gilt”**  $[(\exists x)B] = x : \mathbb{U} \times [B]$ 
  - Um Evidenz für  $(\exists x)B$  zu konstruieren, braucht man  $b : [B]$  für ein  $x : \mathbb{U}$
  - Formel  $B$  kann von Wahl des konkreten Wertes für  $x$  abhängen
  - Typ der Evidenzen für  $(\exists x)B$  ist ein “abhängiger” Produktraum
- **Konkrete Evidenz für  $Pa \Rightarrow ((\exists x)Px)$** 
  - Evidenz ist Funktion  $f : (p:[Pa] \rightarrow (x : \mathbb{U} \times [Px]))$  mit  $f(p) = (x, p')$  für alle  $p : [Pa]$ , wobei  $x : \mathbb{U}$  und  $p'$  Evidenz für  $Px$
  - Einfachste Lösung ist  $x = a$  and  $p' = p$   $f = \lambda p. (a, p)$
- **Konkrete Evidenz für  $((\exists x)Px) \Rightarrow ((\exists y)Py)$** 
  - Evidenz ist Funktion  $f$  mit  $f(z) = (y, p')$  für alle  $z : (x : \mathbb{U} \times [Px])$ , wobei  $y : \mathbb{U}$  und  $p'$  Evidenz für  $Py$
  - $z$  muß ein Paar  $(a, p)$  mit  $a : \mathbb{U}$  und  $p : [Pa]$
  - Einfachste Lösung:  $x = a = z.1$  and  $p' = p = z.2$ , also  $f = \lambda z. (z.1, z.2)$
  - Wegen  $z = (z.1, z.2)$  kann Evidenz vereinfacht werden zu  $f = \lambda z. z$

## EVIDENZ FÜR KOMPLEXERE FORMELN

- $((\forall x)(P x \wedge Q x)) \Rightarrow ((\forall x)P x \wedge (\forall x)Q x)$

## EVIDENZ FÜR KOMPLEXERE FORMELN

- $((\forall x)(P x \wedge Q x)) \Rightarrow ((\forall x)P x \wedge (\forall x)Q x)$ 
  - Evidenz ist Funktion  $f$  so daß für alle  $h : (x:\mathbb{U} \rightarrow [P x] \times [Q x])$  gilt  
 $f(h) = (g_p, g_q) : (x:\mathbb{U} \rightarrow [P x]) \times (x:\mathbb{U} \rightarrow [Q x])$
  - $g_p$  und  $g_q$  nehmen ein  $x:\mathbb{U}$  und erzeugen Elemente von  $[P x]$  bzw.  $[Q x]$
  - Einfachste Lösung ist  $g_p(x) = h(x).1$  und  $g_q(x) = h(x).2$   
 $f = \lambda h. (\lambda x. h(x).1, \lambda x. h(x).2)$



## EVIDENZ FÜR KOMPLEXERE FORMELN

- $((\forall x)(P x \wedge Q x)) \Rightarrow ((\forall x)P x \wedge (\forall x)Q x)$ 
  - Evidenz ist Funktion  $f$  so daß für alle  $h : (x:\mathbb{U} \rightarrow [P x] \times [Q x])$  gilt  
 $f(h) = (g_p, g_q) : (x:\mathbb{U} \rightarrow [P x]) \times (x:\mathbb{U} \rightarrow [Q x])$
  - $g_p$  und  $g_q$  nehmen ein  $x:\mathbb{U}$  und erzeugen Elemente von  $[P x]$  bzw.  $[Q x]$
  - Einfachste Lösung ist  $g_p(x) = h(x).1$  und  $g_q(x) = h(x).2$   
 $f = \lambda h. (\lambda x. h(x).1, \lambda x. h(x).2)$
- **Evidenz für**  $\neg((\forall x)\neg(P x)) \Rightarrow (\exists x)P x$

# EVIDENZ FÜR KOMPLEXERE FORMELN

- $((\forall x)(P x \wedge Q x)) \Rightarrow ((\forall x)P x \wedge (\forall x)Q x)$

- Evidenz ist Funktion  $f$  so daß für alle  $h : (x:\mathbb{U} \rightarrow [P x] \times [Q x])$  gilt

$$f(h) = (g_p, g_q) : (x:\mathbb{U} \rightarrow [P x]) \times (x:\mathbb{U} \rightarrow [Q x])$$

- $g_p$  und  $g_q$  nehmen ein  $x:\mathbb{U}$  und erzeugen Elemente von  $[P x]$  bzw.  $[Q x]$

- Einfachste Lösung ist  $g_p(x) = h(x).1$  und  $g_q(x) = h(x).2$

$$f = \lambda h. (\lambda x. h(x).1, \lambda x. h(x).2)$$

- **Evidenz für**  $\neg((\forall x)\neg(P x)) \Rightarrow (\exists x)P x$

- Evidenz ist Funktion  $f$  so daß für alle  $h : (x:\mathbb{U} \rightarrow ([P x] \rightarrow \{\})) \rightarrow \{\}$  gilt

$$f(h) = (x, p) \text{ mit } p : [P x]$$

- Zur Konstruktion von  $f(h)$  benötigt man Kenntnisse über  $P$  und Objekte  $x$ , für die  $P x$  gilt

- Es gibt keinen allgemeinen Weg,  $x$  oder  $p$  aus  $h$  zu konstruieren, solange das Prädikatssymbol  $P$  unspezifiziert ist

**Keine universelle Evidenz**

# EVIDENZSEMANTIK – ZUSAMMENFASSUNG

Aussage $A$	Evidenztyp $[A]$	Evidenzkonstruktion	Dekompositionsterm
$A \Rightarrow B$	$[A] \rightarrow [B]$	$\lambda a.b$	$f(a)$
$A \wedge B$	$[A] \times [B]$	$(a, b)$	$x.1, x.2$
$A \vee B$	$[A] + [B]$	$\text{inl}(a), \text{inr}(b)$	$\text{case } x \text{ of } \text{inl}(a) \rightarrow s$   $\text{inr}(b) \rightarrow t$
$\neg A$	$[A] \rightarrow \{\}$	$\lambda a.b$	$f(a)$
f	$\{\}$	–	–
$(\forall x)B$	$x : \mathbb{U} \rightarrow [B]$	$\lambda a.b$	$f(a)$
$(\exists x)B$	$x : \mathbb{U} \times [B]$	$(a, b)$	$x.1, x.2$

- **Formeln korrespondieren mit Datentyp ihrer Evidenzen**
  - Es gibt Terme, um Evidenz zu konstruieren oder zu zerlegen
  - Beide sind invers zueinander und ermöglichen “Rechnen” mit Evidenz (Einheit 5)
- **Evidenzterme bilden eine Programmiersprache**
  - Sprache umfasst Terme der Prädikatenlogik
  - Prädikatenlogik kann nur Programme ohne Schleifen spezifizieren
  - Mehr Ausdruckskraft benötigt Gleichheit, Zahlen, Induktion, ...

- **Erweiterung der aussagenlogische Regeln**

- Regel **axiom** ist auf atomare prädikatenlogische Formeln anwendbar
- Unveränderte Regeln für  $A \Rightarrow B$ ,  $A \wedge B$ ,  $A \vee B$ ,  $\neg A$
- Konstruierte Evidenz ändert sich ebenfalls nicht

- **Behandlung von Quantoren**

- Um  $(\forall x)B$  zu zeigen, muß man  $B$  für jede Instanz von  $x$  zeigen  
Hierzu wählt man  $x' : \mathbb{U}$  beliebig aber fest und zeigt  $B$  für  $x'$  statt  $x$
- Um  $(\exists x)B$  zu zeigen, muß man eine  $B$  für eine Instanz von  $x$  zeigen  
Hierzu gibt man ein Objekt  $a$  an und zeigt  $B$  für  $a$  statt  $x$
- Regeln benötigen “syntaktische Instantiierung” von Variablen

- **Formales Konzept: Substitution  $B[t/x]$**

- Ersetzen der Variablen  $x$  in Formel  $B$  durch Term  $t$   
Unvollständiger Ersatz für Instantiierung, wenn Universum überabzählbar
- Substitution muß Verständnis von “Für alle” und “es gibt” erhalten  
 $(\forall x)P x$  und  $(\forall y)P y$  bedeuten dasselbe
- Nur **ungebundene** Variablen dürfen ersetzt werden

## VORKOMMEN VON VARIABLEN IN FORMELN

- **Vorkommen der Variablen  $x$  in Formel  $B$ , informal**
  - **Gebunden**:  $x$  erscheint im Scope eines Quantors  $(\forall x)$  oder  $(\exists x)$
  - **Frei**:  $x$  kommt in  $B$  vor, ohne gebunden zu sein
  - $B$  heißt **geschlossen** falls  $B$  keine freien Variablen enthält

# VORKOMMEN VON VARIABLEN IN FORMELN

- **Vorkommen der Variablen  $x$  in Formel  $B$ , informal**
  - **Gebunden**:  $x$  erscheint im Scope eines Quantors  $(\forall x)$  oder  $(\exists x)$
  - **Frei**:  $x$  kommt in  $B$  vor, ohne gebunden zu sein
  - $B$  heißt **geschlossen** falls  $B$  keine freien Variablen enthält

- **Präzise, induktive Definition**

$x$	die Variable $x$ kommt frei vor; $y \neq x$ kommt nicht vor
$f$	die Variable $x$ kommt nicht vor
$f(t_1, \dots, t_n)$	freie Vorkommen von $x$ in $t_i$ bleiben frei
$P(t_1, \dots, t_n)$	gebundene Vorkommen von $x$ bleiben gebunden.
$\neg A, A \Rightarrow B$	freie Vorkommen von $x$ in $A, B$ bleiben frei
$A \wedge B, A \vee B$	gebundene Vorkommen von $x$ bleiben gebunden.
$(\forall x)B$	beliebige Vorkommen von $x$ in $B$ werden gebunden
$(\exists x)B$	Vorkommen von $y \neq x$ in $B$ bleiben unverändert

# VORKOMMEN VON VARIABLEN IN FORMELN

- **Vorkommen der Variablen  $x$  in Formel  $B$ , informal**
  - **Gebunden**:  $x$  erscheint im Scope eines Quantors  $(\forall x)$  oder  $(\exists x)$
  - **Frei**:  $x$  kommt in  $B$  vor, ohne gebunden zu sein
  - $B$  heißt **geschlossen** falls  $B$  keine freien Variablen enthält

- **Präzise, induktive Definition**

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

$f$  die Variable  $x$  kommt nicht vor

$f(t_1, \dots, t_n)$  freie Vorkommen von  $x$  in  $t_i$  bleiben frei

$P(t_1, \dots, t_n)$  gebundene Vorkommen von  $x$  bleiben gebunden.

$\neg A, A \Rightarrow B$  freie Vorkommen von  $x$  in  $A, B$  bleiben frei

$A \wedge B, A \vee B$  gebundene Vorkommen von  $x$  bleiben gebunden.

$(\forall x)B$  beliebige Vorkommen von  $x$  in  $B$  werden gebunden

$(\exists x)B$  Vorkommen von  $y \neq x$  in  $B$  bleiben unverändert

$$(\forall x)(P(x) \wedge Q(x)) \wedge R(x)$$

# VORKOMMEN VON VARIABLEN IN FORMELN

- **Vorkommen der Variablen  $x$  in Formel  $B$ , informal**
  - **Gebunden**:  $x$  erscheint im Scope eines Quantors  $(\forall x)$  oder  $(\exists x)$
  - **Frei**:  $x$  kommt in  $B$  vor, ohne gebunden zu sein
  - $B$  heißt **geschlossen** falls  $B$  keine freien Variablen enthält

- **Präzise, induktive Definition**

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

$f$  die Variable  $x$  kommt nicht vor

$f(t_1, \dots, t_n)$  freie Vorkommen von  $x$  in  $t_i$  bleiben frei

$P(t_1, \dots, t_n)$  gebundene Vorkommen von  $x$  bleiben gebunden.

$\neg A, A \Rightarrow B$  freie Vorkommen von  $x$  in  $A, B$  bleiben frei

$A \wedge B, A \vee B$  gebundene Vorkommen von  $x$  bleiben gebunden.

$(\forall x)B$  beliebige Vorkommen von  $x$  in  $B$  werden gebunden

$(\exists x)B$  Vorkommen von  $y \neq x$  in  $B$  bleiben unverändert

$$(\forall x) \underbrace{(P(x) \wedge Q(x))}_{x \text{ frei}} \wedge \underbrace{R(x)}_{x \text{ frei}}$$



# VORKOMMEN VON VARIABLEN IN FORMELN

- **Vorkommen der Variablen  $x$  in Formel  $B$ , informal**
  - **Gebunden**:  $x$  erscheint im Scope eines Quantors  $(\forall x)$  oder  $(\exists x)$
  - **Frei**:  $x$  kommt in  $B$  vor, ohne gebunden zu sein
  - $B$  heißt **geschlossen** falls  $B$  keine freien Variablen enthält

- **Präzise, induktive Definition**

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

$f$  die Variable  $x$  kommt nicht vor

$f(t_1, \dots, t_n)$  freie Vorkommen von  $x$  in  $t_i$  bleiben frei

$P(t_1, \dots, t_n)$  gebundene Vorkommen von  $x$  bleiben gebunden.

$\neg A, A \Rightarrow B$  freie Vorkommen von  $x$  in  $A, B$  bleiben frei

$A \wedge B, A \vee B$  gebundene Vorkommen von  $x$  bleiben gebunden.

$(\forall x)B$  beliebige Vorkommen von  $x$  in  $B$  werden gebunden

$(\exists x)B$  Vorkommen von  $y \neq x$  in  $B$  bleiben unverändert

$$\overbrace{(\forall x)(\underbrace{P(x) \wedge Q(x)}_{x \text{ frei}})}_{x \text{ gebunden}} \wedge \underbrace{R(x)}_{x \text{ frei}}$$

# VORKOMMEN VON VARIABLEN IN FORMELN

- **Vorkommen der Variablen  $x$  in Formel  $B$ , informal**
  - **Gebunden**:  $x$  erscheint im Scope eines Quantors  $(\forall x)$  oder  $(\exists x)$
  - **Frei**:  $x$  kommt in  $B$  vor, ohne gebunden zu sein
  - $B$  heißt **geschlossen** falls  $B$  keine freien Variablen enthält

- **Präzise, induktive Definition**

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

$f$  die Variable  $x$  kommt nicht vor

$f(t_1, \dots, t_n)$  freie Vorkommen von  $x$  in  $t_i$  bleiben frei

$P(t_1, \dots, t_n)$  gebundene Vorkommen von  $x$  bleiben gebunden.

$\neg A, A \Rightarrow B$  freie Vorkommen von  $x$  in  $A, B$  bleiben frei

$A \wedge B, A \vee B$  gebundene Vorkommen von  $x$  bleiben gebunden.

$(\forall x)B$  beliebige Vorkommen von  $x$  in  $B$  werden gebunden

$(\exists x)B$  Vorkommen von  $y \neq x$  in  $B$  bleiben unverändert

*$x$  frei und gebunden*

*$x$  gebunden*

$(\forall x)(\underbrace{P(x) \wedge Q(x)}_{x \text{ frei}}) \wedge \underbrace{R(x)}_{x \text{ frei}}$

# SUBSTITUTION $B[t/x]$ FORMAL

## Endliche Abbildung $\sigma$ von Variablen in Terme

- $\sigma = [t_1, \dots, t_n / x_1, \dots, x_n] \hat{=} \sigma(x_1)=t_1, \dots, \sigma(x_n)=t_n$
- $A\sigma$ : Anwendung von  $\sigma$  auf den Ausdruck  $A$   $\tau$  und  $\sigma$

$\llbracket x \rrbracket [t/x] = t$	$\llbracket x \rrbracket [t/y] = x$	$(y \neq x)$
$\llbracket f(t_1, \dots, t_n) \rrbracket \sigma = f(t_1\sigma, \dots, t_n\sigma)$	$\llbracket f \rrbracket \sigma = f$	
$\llbracket P(t_1, \dots, t_n) \rrbracket \sigma = P(t_1\sigma, \dots, t_n\sigma)$		
$\llbracket \neg A \rrbracket \sigma = \neg A\sigma$	$\llbracket A \wedge B \rrbracket \sigma = A\sigma \wedge B\sigma$	
$\llbracket A \vee B \rrbracket \sigma = A\sigma \vee B\sigma$	$\llbracket A \Rightarrow B \rrbracket \sigma = A\sigma \Rightarrow B\sigma$	
$\llbracket (\forall x) B \rrbracket [t/x] = (\forall x) B$	$\llbracket (\exists x) B \rrbracket [t/x] = (\exists x) B$	
$\llbracket (\forall x) B \rrbracket [t/y] = \llbracket (\forall z) B[z/x] \rrbracket [t/y]$	$\llbracket (\exists x) B \rrbracket [t/y] = \llbracket (\exists z) B[z/x] \rrbracket [t/y]$	*
$\llbracket (\forall x) B \rrbracket [t/y] = (\forall x) . \llbracket B[t/y] \rrbracket$	$\llbracket (\exists x) B \rrbracket [t/y] = (\exists x) \llbracket B[t/y] \rrbracket$	**

\*:  $y \neq x$ ,  $y$  frei in  $B$ ,  $x$  frei in  $t$ ,  $z$  neue Variable

\*\* :  $y \neq x$ ,  $y$  nicht frei in  $B$  oder  $x$  nicht frei in  $t$

# REGELN FÜR ALLQUANTOR

## ● Allquantor auf rechter Seite einer Sequenz

- Um  $H \vdash (\forall x)B$  zu zeigen, muß man  $B$  für jede Instanz von  $x$  zeigen  
Einziger Weg ist generischer Beweis, der nicht von Instanz abhängt
- Wähle neue Variable  $x'$  und beweise  $B[x'/x]$

$$\boxed{\begin{array}{l} H \vdash (\forall x)B \quad \text{ev} = \lambda x'. b \\ H, x':\mathbb{U} \vdash B[x'/x] \quad \text{ev} = b \end{array}} \quad \text{allR}$$

- Im Teilziel wird generische Evidenz  $b$  für  $B[x'/x]$  und alle  $x'$  konstruiert
- Evidenz für  $(\forall x)B$  muß Funktion  $\lambda x'. b$  sein

# REGELN FÜR ALLQUANTOR

## • Allquantor auf rechter Seite einer Sequenz

- Um  $H \vdash (\forall x)B$  zu zeigen, muß man  $B$  für jede Instanz von  $x$  zeigen
- Einziger Weg ist generischer Beweis, der nicht von Instanz abhängt
- Wähle neue Variable  $x'$  und beweise  $B[x'/x]$

$$\boxed{\begin{array}{l} H \vdash (\forall x)B \quad \text{ev} = \lambda x'. b \\ H, x':\mathbb{U} \vdash B[x'/x] \quad \text{ev} = b \end{array}} \quad \text{allR}$$

- Im Teilziel wird generische Evidenz  $b$  für  $B[x'/x]$  und alle  $x'$  konstruiert
- Evidenz für  $(\forall x)B$  muß Funktion  $\lambda x'. b$  sein

## • Allquantor auf linker Seite einer Sequenz

- Um  $C$  unter Annahme  $(\forall x)B$  zu zeigen, darf man jede Instanz von  $x$  verwenden, also die Annahme  $B[t/x]$  für beliebige Terme  $t$  ergänzen

$$\boxed{\begin{array}{l} H, f:(\forall x)B, H' \vdash C \quad \text{ev} = c[f(t)/b] \\ H, f:(\forall x)B, b:B[t/x], H' \vdash C \quad \text{ev} = c \end{array}} \quad \text{allL } t$$

- Regel nimmt an, daß Evidenz  $c$  aus Evidenz  $b:B[t/x]$  konstruierbar ist
- Anwendung von  $\lambda b.c$  auf  $f(t)$  liefert Evidenz für  $C$  im Hauptziel

# ANWENDUNG DER ALLQUANTORREGELN

- **Beweis für  $(\forall x)(Px \Rightarrow Px)$**

$\vdash (\forall x)(Px \Rightarrow Px)$	ev = $\lambda x. (\lambda p. p)$	BY <b>allR</b>
1 $x:\mathbb{U} \vdash Px \Rightarrow Px$	ev = $\lambda p. p$	BY <b>impliesR</b>
1.1 $x:\mathbb{U}, p:Px \vdash Px$	ev = $p$	BY <b>axiom</b>

# ANWENDUNG DER ALLQUANTORREGELN

- **Beweis für  $(\forall x)(Px \Rightarrow Px)$**

$\vdash (\forall x)(Px \Rightarrow Px)$	$ev = \lambda x. (\lambda p. p)$	BY	allR
1 $x:\mathbb{U} \vdash Px \Rightarrow Px$	$ev = \lambda p. p$	BY	impliesR
1.1 $x:\mathbb{U}, p:Px \vdash Px$	$ev = p$	BY	axiom

- **Beweis für  $((\forall x)Px) \Rightarrow Pa$**

$\vdash ((\forall x)Px) \Rightarrow Pa$	$ev = \lambda f. f(a)$	BY	impliesR
1 $f:(\forall x)Px \vdash Pa$	$ev = p[f(a)/p]$	BY	allL $a$
1.1 $f:(\forall x)Px, p:Pa \vdash Pa$	$ev = p$	BY	axiom

# ANWENDUNG DER ALLQUANTORREGELN

## • Beweis für $(\forall x)(Px \Rightarrow Px)$

$\vdash (\forall x)(Px \Rightarrow Px)$	$ev = \lambda x. (\lambda p. p)$	BY	allR
1 $x:\mathbb{U} \vdash Px \Rightarrow Px$	$ev = \lambda p. p$	BY	impliesR
1.1 $x:\mathbb{U}, p:Px \vdash Px$	$ev = p$	BY	axiom

## • Beweis für $((\forall x)Px) \Rightarrow Pa$

$\vdash ((\forall x)Px) \Rightarrow Pa$	$ev = \lambda f. f(a)$	BY	impliesR
1 $f:(\forall x)Px \vdash Pa$	$ev = p[f(a)/p]$	BY	allL a
1.1 $f:(\forall x)Px, p:Pa \vdash Pa$	$ev = p$	BY	axiom

## • Beweis für $((\forall x)Px) \Rightarrow (Pa \wedge Pb)$

$\vdash ((\forall x)Px) \Rightarrow (Pa \wedge Pb)$	$ev = \lambda f. (f(a), f(b))$	BY	impliesR
1 $f:(\forall x)Px \vdash Pa \wedge Pb$	$ev = (f(a), f(b))$	BY	allL a
1.1 $f:(\forall x)Px, p_a:Pa \vdash Pa \wedge Pb$	$ev = (p_a, f(b))$	BY	allL b
1.1.1 $f:(\forall x)Px, p_a:Pa, p_b:Pb \vdash Pa \wedge Pb$	$ev = (p_a, p_b)$	BY	andR
1.1.1.1 $f:(\forall x)Px, p_a:Pa, p_b:Pb \vdash Pa$	$ev = p_a$	BY	axiom
1.1.1.2 $f:(\forall x)Px, p_a:Pa, p_b:Pb \vdash Pb$	$ev = p_b$	BY	axiom



# REGELN FÜR EXISTENZQUANTOR

- **Existenzquantor auf rechter Seite einer Sequenz**

- Um  $H \vdash (\exists x)B$  zu zeigen, muß  $B[t/x]$  für einen Term  $t$  gezeigt werden

$$\boxed{\begin{array}{l} H \vdash (\exists x)B \quad \text{ev} = (t, b) \\ H \vdash B[t/x] \quad \text{ev} = b \end{array}} \quad \text{exR } t$$

- Regel setzt Term  $t$  und Evidenz  $b : B[t/x]$  zur Evidenz  $(t, b)$  zusammen

# REGELN FÜR EXISTENZQUANTOR

## ● Existenzquantor auf rechter Seite einer Sequenz

- Um  $H \vdash (\exists x)B$  zu zeigen, muß  $B[t/x]$  für einen Term  $t$  gezeigt werden

$$\boxed{\begin{array}{l} H \vdash (\exists x)B \quad \text{ev} = (t, b) \\ H \vdash B[t/x] \quad \text{ev} = b \end{array}} \quad \text{exR } t$$

- Regel setzt Term  $t$  und Evidenz  $b : B[t/x]$  zur Evidenz  $(t, b)$  zusammen

## ● Existenzquantor auf linker Seite einer Sequenz

- Um  $C$  unter Annahme  $(\exists x)B$  zu beweisen, muß man  $C$  unter Annahme  $B$  für eine beliebige Instanz von  $x$ , also generisch, zeigen können
- Wähle neue Variable  $x'$  und verwende Annahme  $B[x'/x]$

$$\boxed{\begin{array}{l} H, z:(\exists x)B, H' \vdash C \quad \text{ev} = c[z.1, z.2/x', b] \\ H, x':\mathbb{U}, b:B[x'/x], H' \vdash C \quad \text{ev} = c \end{array}} \quad \text{exL}$$

- Label  $z$  für  $(\exists x)B$  entspricht Paar aus Variablen  $x'$  und Label  $b:B$
- Evidenz  $c$  hängt im Teilziel von  $x'$  und  $b$  ab
- Im Hauptziel muß  $x'$  durch  $z.1$  und  $b$  durch  $z.2$  ersetzt werden

# ANWENDUNG DER EXISTENZQUANTORREGELN

- Beweis für  $Pa \Rightarrow ((\exists x)Px)$

$\vdash Pa \Rightarrow ((\exists x)Px)$	$ev = \lambda p.(a, p)$	BY <code>impliesR</code>
1 $p:Pa \vdash (\exists x)Px$	$ev = (a, p)$	BY <code>exR a</code>
1.1 $p:Pa \vdash Pa$	$ev = p$	BY <code>axiom</code>

# ANWENDUNG DER EXISTENZQUANTORREGELN

## • Beweis für $Pa \Rightarrow ((\exists x)Px)$

$\vdash Pa \Rightarrow ((\exists x)Px)$	$ev = \lambda p.(a, p)$	BY <code>impliesR</code>
1 $p:Pa \vdash (\exists x)Px$	$ev = (a, p)$	BY <code>exR a</code>
1.1 $p:Pa \vdash Pa$	$ev = p$	BY <code>axiom</code>

## • Beweis für $((\exists x)Px) \Rightarrow ((\exists y)Py)$

$\vdash ((\exists x)Px) \Rightarrow ((\exists y)Py)$	$ev = \lambda z.(z.1, z.2)$	BY <code>impliesR</code>
1 $z:(\exists x)Px \vdash (\exists y)Py$	$ev = (z.1, z.2)$	BY <code>exL</code>
1.1 $x:\mathbb{U}, p:Px \vdash (\exists y)Py$	$ev = (x, p)$	BY <code>exR x</code>
1.1.1 $x:\mathbb{U}, p:Px \vdash Px$	$ev = p$	BY <code>axiom</code>

– Evidenz  $(z.1, z.2)$  kann zu  $z$  vereinfacht werden

# ANWENDUNG DER EXISTENZQUANTORREGELN

## • Beweis für $Pa \Rightarrow ((\exists x)Px)$

$\vdash Pa \Rightarrow ((\exists x)Px)$	$ev = \lambda p.(a, p)$	BY <code>impliesR</code>
1 $p:Pa \vdash (\exists x)Px$	$ev = (a, p)$	BY <code>exR a</code>
1.1 $p:Pa \vdash Pa$	$ev = p$	BY <code>axiom</code>

## • Beweis für $((\exists x)Px) \Rightarrow ((\exists y)Py)$

$\vdash ((\exists x)Px) \Rightarrow ((\exists y)Py)$	$ev = \lambda z.(z.1, z.2)$	BY <code>impliesR</code>
1 $z:(\exists x)Px \vdash (\exists y)Py$	$ev = (z.1, z.2)$	BY <code>exL</code>
1.1 $x:\mathbb{U}, p:Px \vdash (\exists y)Py$	$ev = (x, p)$	BY <code>exR x</code>
1.1.1 $x:\mathbb{U}, p:Px \vdash Px$	$ev = p$	BY <code>axiom</code>

– Evidenz  $(z.1, z.2)$  kann zu  $z$  vereinfacht werden

– Reihenfolge der Regelanwendungen wichtig für erfolgreichen Beweis

$\vdash ((\exists x)Px) \Rightarrow ((\exists y)Py)$	BY <code>impliesR</code>
1 $z:(\exists x)Px \vdash (\exists y)Py$	BY <code>exR x</code>
1.1 $z:(\exists x)Px \vdash Px$	BY <code>exL</code>
1.1.1 $x':\mathbb{U}, p:Px' \vdash Px$	BY <code>???</code>

# EIN KOMPLEXERER BEWEIS

$\vdash ((\forall x)(Px \wedge Qx)) \Rightarrow ((\forall x)Px \wedge (\forall x)Qx)$	BY <code>impliesR</code>
1. $f:(\forall x)(Px \wedge Qx) \vdash ((\forall x)Px \wedge (\forall x)Qx)$	BY <code>andR</code>
1.1. $f:(\forall x)(Px \wedge Qx) \vdash (\forall x)Px$	BY <code>allR</code>
1.1.1. $x:\mathbb{U}, f:(\forall x)(Px \wedge Qx) \vdash Px$	BY <code>allL x</code>
1.1.1.1. $x:\mathbb{U}, f:(\forall x)(Px \wedge Qx), z:(Px \wedge Qx) \vdash Px$	BY <code>andL</code>
1.1.1.1.1. $x:\mathbb{U}, f:(\forall x)(Px \wedge Qx), p:Px, q:Qx \vdash Px$	BY <code>axiom</code>
1.2. $f:(\forall x)(Px \wedge Qx) \vdash (\forall x)Qx$	BY <code>allR</code>
1.2.1. $x:\mathbb{U}, f:(\forall x)(Px \wedge Qx) \vdash Qx$	BY <code>allL x</code>
1.2.1.1. $x:\mathbb{U}, f:(\forall x)(Px \wedge Qx), z:(Px \wedge Qx) \vdash Qx$	BY <code>andL</code>
1.2.1.1.1. $x:\mathbb{U}, f:(\forall x)(Px \wedge Qx), p:Px, q:Qx \vdash Qx$	BY <code>axiom</code>

Konstruierte Evidenz ist  $\lambda f. (\lambda x.(f x).1, \lambda x.(f x).2)$

# REFINEMENT LOGIK – ZUSAMMENFASSUNG

Links			Rechts		
$H, f:A \Rightarrow B, H' \vdash C$	$ev=c[f(a)/b]$	<b>impliesL</b>	$H \vdash A \Rightarrow B$	$ev=\lambda a.b$	<b>impliesR</b>
$H, f:A \Rightarrow B, H' \vdash A$	$ev=a$		$H, a:A \vdash B$	$ev=b$	
$H, b:B, H' \vdash C$	$ev=c$				
$H, x:A \wedge B, H' \vdash C$	$ev=c[x.1, x.2/a, b]$	<b>andL</b>	$H \vdash A \wedge B$	$ev=(a, b)$	<b>andR</b>
$H, a:A, b:B, H' \vdash C$	$ev=c$		$H \vdash A$	$ev=a$	
			$H \vdash B$	$ev=b$	
$H, x:A \vee B, H' \vdash C$	$ev=\text{case } x \text{ of } \text{inl}(a) \rightarrow c_1$	<b>orL</b>	$H \vdash A \vee B$	$ev=\text{inl}(a)$	<b>orR1</b>
$H, a:A, H' \vdash C$	$ev=c_1$	$\text{inr}(b) \rightarrow c_2$	$H \vdash A$	$ev=a$	
$H, b:B, H' \vdash C$	$ev=c_2$		$H \vdash A \vee B$	$ev=\text{inr}(b)$	<b>orR2</b>
			$H \vdash B$	$ev=b$	
$H, f:\neg A, H' \vdash C$	$ev=\text{any}(f(a))$	<b>notL</b>	$H \vdash \neg A$	$ev=\lambda a.b$	<b>notR</b>
$H, f:\neg A, H' \vdash A$	$ev=a$		$H, a:A \vdash \mathbf{f}$	$ev=b$	
			$H, a:A, H' \vdash A$	$ev=a$	<b>axiom</b>
$H, f:(\forall x)B, H' \vdash C$	$ev=c[f(t)/b]$	<b>allL</b>	$H \vdash (\forall x)B$	$ev=\lambda x'.b$	<b>allR</b>
$H, f:(\forall x)B, b:B[t/x], H' \vdash C$	$ev=c$		$H, x':\mathbb{U} \vdash B[x'/x]$	$ev=b$	
$H, z:(\exists x)B, H' \vdash C$	$ev=c[z.1, z.2/x', b]$	<b>exL</b>	$H \vdash (\exists x)B$	$ev=(t, b)$	<b>exR</b>
$H, x':\mathbb{U}, b:B[x'/x], H' \vdash C$	$ev=c$		$H \vdash B[t/x]$	$ev=b$	$t$

*t ist ein beliebiger Term, x' eine neue Variable*

# SINNVOLLE ZUSATZREGELN

## ● Einfügen von Zwischenbehauptungen

- $C$  ist gültig, wenn  $C$  aus der Annahme  $A$  folgt und  $A$  gültig ist

$$\begin{array}{l} H \vdash C \quad \text{ev} = (\lambda x.c)(a) \\ H \vdash A \quad \text{ev} = a \\ H, x:A \vdash C \quad \text{ev} = c \end{array} \quad \text{cut } A$$

- $A$  kann eine beliebige “Schnitt”-Formel sein
- Beweise werden signifikant kürzer, wenn  $A$  mehrfach benutzt wird
- Evidenz  $(\lambda x.c)(a)$  wird nicht evaluiert (Gefahr der Termaufblähung)

## ● Ausdünnen von Annahmen

- Hypothesen, die nicht gebraucht werden, können entfernt werden

$$\begin{array}{l} H, a:A, H' \vdash C \quad \text{ev} = c \\ H, H' \vdash C \quad \text{ev} = c \end{array} \quad \text{thin } A$$

- Sinnvoll, um Hypothesenliste übersichtlich zu halten
- Evidenz  $c$  hängt im Hauptziel nicht vom Label  $a$  ab



## Aussagen über den logischen Kalkül

- **Wichtig für Analyse der Eigenschaften des Kalküls**
  - Ist Refinement Logik **korrekt**? (sind beweisene Formeln gültig?)
  - Ist Refinement Logik **vollständig**? (sind gültige Formeln beweisbar?)
  - Welche Formeln sind entscheidbar?
  - Was ist die Komplexität von Beweissuche?
- **Hilfreich für Implementierung und Automatisierung**
  - Beschreibung der Struktur der Grundelemente des Kalküls
  - Daten- und Zugriffsstrukturen für Formeln, Beweise, Evidenz, ...
  - Algorithmen zur Anwendung von Regeln und Evidenzkonstruktion
  - Benutzerdefinierbare (“konservative”) Erweiterung von Objektsprache und Regelsystem durch Definitionen und Beweisstrategien

## Präsentation von Kalkülen hat zwei Sprachebenen

- **Objektsprache:**

- Sprache des Kalküls, in dem formalisiert wird
- Formale Sprache mit präzise definierter Syntax
- Konkretes Element z.B.  $(\exists x)(P x) \vee Q x \Rightarrow (\neg((\forall x)(\neg P x \wedge \neg Q x)))$

- **Metasprache:**

- Sprache, um Aussagen über den Kalkül zu machen
  - Beschreibung von Syntax, Semantik, Eigenschaften des Kalküls
- Natürliche, oft stark schematisierte Sprache
- Enthält Objektsprache, angereichert um syntaktische Metavariablen
- Konkretes Element z.B. *aus*  $(\exists x)(A \vee B)$  *folgt*  $\neg((\forall x)(\neg A \wedge \neg B))$

- **Unterscheidung zuweilen durch Fonts / Farben**

- Ansonsten aus Kontext eindeutig erkennbar

## Metasprachliche Präzisierung der verwendeten Konzepte

- **Kalkül verwendet Objekt-Logik / Evidenz als Parameter**
  - Objektsprache ist (bisher) Sprache der Prädikatenlogik
  - Evidenz formuliert als Terme in erweiterter  $\lambda$ -Notation
  - Semantik der Logik erklärt, wann Terme Evidenz für Formeln sind
- **Sequenz (Ziel)  $H \vdash C$** 
  - $H = x_1:A_1, \dots, x_n:A_n$  **Hypothesenliste** ( $x_i$  verschieden),  $C$  **Konklusionsformel**
  - $x_i:A_i$  **Deklaration** “ $x_i$  ist Evidenz für  $A_i$ ” ( $x$  Term-Variable,  $A$  Formel)
  - **Initiaalsequenz**: Sequenz  $\vdash C$  ohne Hypothesen
- **Evidenzterm für  $x_1:A_1, \dots, x_n:A_n \vdash C$** 
  - Term  $e$  mit freien Variablen aus den  $x_i$
  - $e$  ist Evidenz für  $C$ , wenn alle  $x_i$  mit Evidenz für  $A_i$  instantiiert werden

# REFINEMENT LOGIK ALS FORMALER KALKÜL (II)

## ● (Verfeinerungs-)Regel (*dec, val*)

- *dec* Dekomposition: Abbildung von Sequenzen in Liste von Sequenzen (vom Beweisziel in Liste der Teilziele)
- *val* Validierung: Abbildung von Liste von Sequenzen und Termen in Term
- Regeln werden als Regelschemata dargestellt mit Metavariablen als Platzhaltern für Formeln und Evidenzterme
- Konkrete Regeln entstehen hieraus durch Instantiierung der Metavariablen

## ● Beweise

- Jede Sequenz  $S = H \vdash C$  ist unvollständiger Beweis mit Wurzel  $S$
- $(S, r, [\pi_1, \dots, \pi_n])$  ist Beweis mit Wurzelsequenz  $S$ , wenn  $\pi_i$  Beweise für alle Sequenzen  $S_i$  sind, die durch Anwendung von  $r$  auf  $S$  entstehen
- vollständiger Beweis: Beweis ohne unvollständige Teilbeweise
- Theorem: vollständiger Beweis, dessen Wurzel eine Initialsequenz ist

## ● Extraktterm $ext(\pi)$ eines vollständigen Beweises

- $ext(S, (dec, val), [\pi_1, \dots, \pi_n]) = val([S, (S_1, ext(\pi_1)), \dots, (S_n, ext(\pi_n))])$ ,  
wobei  $S_i$  Wurzeln der  $\pi_i$  Definition gilt auch für  $\pi = (S, (dec, val), [])$

- **Gültigkeit von Sequenzen und Formeln**

- Eine Sequenz ist **gültig**, wenn es einen Evidenzterm für sie gibt

- ↳ Eine Formel  $C$  ist gültig, wenn die Initialsequenz  $\vdash C$  gültig ist

- **Korrektheit einer Regel  $r = (dec, val)$**

- Sind  $S_i = H_i \vdash C_i$  Ergebnis der Anwendung von  $dec$  auf  $S = H \vdash C$

- und  $c_i$  Evidenzterme für  $S_i$ , dann ist  $val(S, (S_i, c_i))$  Evidenzterm für  $S$

- **Refinement Logik ist korrekt**

- Alle Regeln der Refinement Logik sind korrekt

- Alle Theoreme der Refinement Logik haben gültige Formeln als Wurzeln

- ↳ Alle beweisbaren Formeln sind gültig

- **Refinement Logik ist vollständig**

- Für jede gültige Formel  $C$  gibt es ein Theorem mit Wurzel  $\vdash C$

- ↳ Alle gültigen Formeln sind beweisbar

# DEFINITORISCHE ERWEITERUNG

- **Konservative Erweiterung der Objektsprache**

- Neues Konstrukte sind **definitorische Abkürzung** für existierende objektsprachliche Ausdrücke (ggf. mit Parametern)
- Beispiel: **Äquivalenz** in der Prädikatenlogik

$$A \Leftrightarrow B \equiv (A \Rightarrow B) \wedge (B \Rightarrow A)$$

- Bedeutung ergibt sich aus Semantik bestehender Konstrukte
- Auffalten der Definition in Beweisen ersetzt linke durch rechte Seite

$$H \vdash C \quad \text{ev} = c$$

$$H \vdash C[rhs/lhs] \quad \text{ev} = c \quad \text{unfold } name$$

- **Erlaubt kleinen Grundformalismus**

- Einfache Syntax, Semantik und Inferenzsystem
- Eigenschaften leicht beweisbar

- **Erhöht Flexibilität des Formalismus**

- Erlaubt **freiere Syntax** und **umfangreiche formale Sprache**

## ● Konservative Erweiterung des Regelsystems

- Taktiken sind Abkürzungen für komplexe Regelanwendungen
- Einfache Taktiksprache unterstützt Verknüpfung von Regeln

$r_1$  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

- Ermöglicht kürzere und besser strukturierte Beweise

$\vdash P \Rightarrow (Q \Rightarrow (P \wedge Q))$ 1. $p:P, q:Q \vdash P \wedge Q$	BY Repeat impliesR BY andR THEN axiom
--	--

$\vdash ((P \vee Q) \wedge ((P \Rightarrow R) \wedge (Q \Rightarrow R))) \Rightarrow R$ 1. $z:P \vee Q, g:P \Rightarrow R, h:Q \Rightarrow R \vdash R$ 1.1. $p:P, g:P \Rightarrow R, h:Q \Rightarrow R \vdash R$ 1.2. $q:Q, g:P \Rightarrow R, h:Q \Rightarrow R \vdash R$	BY impliesR THEN Repeat andL BY orL BY impliesL $g$ THEN axiom BY impliesL $h$ THEN axiom
---	--

- Elementarer Beweis und Extraktterm durch Expansion rekonstruierbar

## ● Taktiksprache formalisiert Metasprache der Logik (ML)

- Unterstützt Formulierung komplexer Taktiken und Strategien  
 durch Analyse des Beweisziels für gezielte Regelauswahl (ALuP II)

- **Kalkül garantiert Korrektheit formaler Beweise**

- Kalkül ist selbst keine Methode um Beweise zu finden

- **Es gibt Leitlinien für erfolgreiche Beweissuche**

- Versuche vorrangig Zweige abzuschließen ( **axiom** )

- Verwende Dekompositionsregeln, die Formeln äquivalent aufbrechen

- Verwende **orL** vor **orR1** / **orR2**

- Verwende **exL** und **allR** vor **exR** und **allL**

- Wähle anwendbare Regel, welche die wenigsten Teilziele erzeugt

Methodik ist als Taktik programmierbar

(ALuP II)

- **Beweismethodik läßt Fragen offen**

- Auswahl der Substitution für Quantoren erfordert “Vorausschau”

- Maschinennahe Methoden finden Substitution durch **Unifikation**

Mehr dazu in der Vorlesung “Inferenzmethoden”



- **Semantikdefinition durch Interpretation**

- Interpretation von Variablen, Funktionen, Prädikaten in Zielsprache
- Homomorphe Fortsetzung der Interpretation auf komplexe Formeln
- Gültigkeit ist “Wahrheit” unter allen möglichen Interpretationen

Benötigt Erklärung der Bedeutung der Zielsprache

- **Alternative Beweiskalküle**

- Axiom-orientierte **Frege–Hilbert–Kalküle**  
Sehr mächtig, nur eine Regel, aufwendige Beweissuche
- **Natürliches Schließen, Sequenzkalküle**  
Synthetische Vorgehensweise, gut für Beweispräsentation
- (Analytische) **Tableaux-Kalküle**  
kompakte, evidenzlose Version der Refinement Logik
- Maschinennahe **Resolutions-/Konnektionskalküle**  
Gut geeignet für automatische Beweissuche, schwer lesbare Beweise

- **Ausdrucksstärkere Logiken**

- Gleichheit, Sorten, Arithmetik, Logik höherer Stufe