Formale Systeme

Prädikatenlogik: Semantik

Prof. Dr. Bernhard Beckert | WS 2009/2010

KIT – INSTITUT FÜR THEORETISCHE INFORMATIK

Bedeutung von Formeln

Ist die Formel

$$q(x) \rightarrow \exists y(\ in(y, x) \wedge kl(y)\ ),$$

wahr?

Die Signatur $\Sigma = \{k(\ ),\ q(\ ),\ d(\ ),\ kl(\ ),\ gr(\ ),\ in(\ ,\ )\}$ liegt fest.

Die Wahrheit ist abhängig von

- einer Interpretation $(\mathcal{D}, I)$
- einer Variablenbelegung β

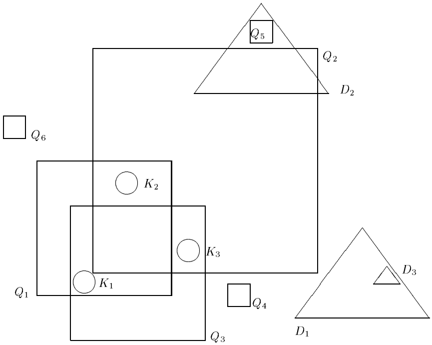
Interpretation

Definition

Es sei Σ eine Signatur der PL1.
Eine *Interpretation* $\mathcal{D}$ von Σ ist ein Paar (D, I) mit

1. D ist eine beliebige, nichtleere Menge
2. I ist eine Abbildung der Signatursymbole, die
   - jeder Konstanten c ein Element $I(c) \in D$
   - für $n \geq 1$: jedem n-stelligen Funktionssymbol f eine Funktion $I(f) : D^n \to D$
   - jedem 0-stelligen Prädikatsymbol P einen Wahrheitswert $I(P) \in \{W, F\}$
   - für $n \geq 1$: jedem n-stelligen Prädikatsymbol p eine n-stellige Relation $I(p) \subseteq D^n$ zuordnet.

Beispiel einer Interpretation (Tarski's World)



$P_\Sigma = \{k(\ ),\ q(\ ),\ d(\ ),\ kl(\ ),\ gr(\ ),\ in(\ ,\ )\}$ $D_{Bsp} = \{Q_i : 1 \leq i \leq 6\} \cup \{K_1, K_2, K_3, D_1, D_2, D_3\}$
$I_{Bsp}(q) = \{Q_i : 1 \leq i \leq 6\}$

$I_{Bsp}(k) = \{K_1, K_2, K_3\}$, $I_{Bsp}(d) = \{D_1, D_2, D_3\}$

$I_{Bsp}(in)\{(K_1, Q_1), (K_1, Q_3), (K_2, Q_1), (K_2, Q_2), (K_3, Q_2), (K_3, Q_3), (D_3, D_1), (Q_5, D_2)\}$

Variablenbelegung

Definition

Es sei (D, I) eine Interpretation von Σ.
Eine *Variablenbelegung* (oder kurz *Belegung* über D) ist eine Funktion

$$\beta : Var \to D.$$

Zu β, $x \in Var$ und $d \in D$ definieren wir die *Modifikation* von β an der Stelle x zu d:

$$\beta_x^d(y) = \begin{cases} d & \text{falls } y = x \\ \beta(y) & \text{falls } y \neq x \end{cases}$$

Auswertung von Termen

Definition

Es sei (D, I) eine Interpretation von Σ und β eine Variablenbelegung über D.
Wir definieren eine Funktion $val_{D,I,\beta}$, mit

$$val_{D,I,\beta}(t) \in D \text{ für } t \in Term_\Sigma$$
$$val_{D,I,\beta}(A) \in \{W, F\} \text{ für } A \in For_\Sigma$$

$val_{D,I,\beta}$ auf $Term_\Sigma$:
$val_{D,I,\beta}(x) = \beta(x)$ für $x \in Var$
$val_{D,I,\beta}(f(t_1, \ldots, t_n)) = (I(f))(val_{D,I,\beta}(t_1), \ldots, val_{D,I,\beta}(t_n))$

Auswertung von Formeln

Definition

1. $val_{D,I,\beta}(\mathbf{1}) = W$
   $val_{D,I,\beta}(\mathbf{0}) = F$
   $val_{D,I,\beta}(s \doteq t) := \begin{cases} W \text{ falls } val_{D,I,\beta}(s) = val_{D,I,\beta}(t) \\ F \text{ sonst} \end{cases}$
   $val_{D,I,\beta}(P) := I(P)$ für 0-stellige Prädikate P

   $val_{D,I,\beta}(p(t_1, \ldots, t_n)) :=$

   $\begin{cases} W & \text{falls } (val_{D,I,\beta}(t_1), \ldots, val_{D,I,\beta}(t_n)) \in I(p) \\ F & \text{sonst} \end{cases}$

Auswertung von Formeln

Definition

2. $val_{D,I,\beta}(X)$ für $X \in \{\neg A, A \wedge B, A \vee B, A \rightarrow B, A \leftrightarrow B\}$ wie in der Aussagenlogik.
   Ferner:
   $val_{D,I,\beta}(\forall x A) :=$

   $$\begin{cases} W \text{ falls für alle } d \in D : val_{D,I,\beta_x^d}(A) = W \\ F \text{ sonst} \end{cases}$$

   $val_{D,I,\beta}(\exists x A) :=$

   $$\begin{cases} W \text{ falls ein } d \in D \text{ existiert mit } val_{D,I,\beta_x^d}(A) = W \\ F \text{ sonst} \end{cases}$$

Beispiel

Wir wollen die Formel

$$q(x) \to \exists y(\ in(y,x) \land kl(y)\ ),$$

in der Interpretation $(\mathcal{D}_{Bsp}, I_{Bsp})$ aus der Abbildung mit der Variablenbelegung $\beta(x) = Q_1$ auswerten.

Formel links von $\to$
$val_{\mathcal{D}_{Bsp},\beta}(x) = Q_1 \ \in I(q)$, also $val_{\mathcal{D}_{Bsp},\beta}(q(x)) = \ W$.

Formel rechts von $\to$
$val_{\mathcal{D}_{Bsp},\beta}(\exists y(\ in(y,x) \land kl(y)) = \ W$

Wähle K_1 als Belegung für y.

Beispiel

Die weitere Auswertung führt zu

$$val_{\mathcal{D}_{Bsp},\beta_y^{K_1}}((\ in(y,x) \wedge kl(y)) = \ W$$

weil $(K_1, Q_1) \in I_{Bsp}(in)$ und $K_1 \in I_{Bsp}(kl)$

Insgesamt
$val_{\mathcal{D}_{Bsp},\beta}(q(x) \rightarrow \exists y(\ in(y,x) \wedge kl(y)) = \ W$

Arithmetische Strukturen

Signatur $\Sigma_{arith} = \{+, *, \leq\}$

Die mathematischen ganzen Zahlen

$$\mathcal{Z} = (\mathbb{Z}, +_{\mathcal{Z}}, *_{\mathcal{Z}}, \leq_{\mathcal{Z}}).$$

Die ganzen Zahlen in Java

$$\mathcal{Z}_{Jint} = (\mathbb{Z}_{Jint}, +_{Jint}, *_{Jint}, \leq_{Jint}).$$

wobei:

$$\begin{array}{lcl} \mathbb{Z}_{Jint} & = & [minInt, maxInt] \quad = \quad [-2147483648, 2147483647] \\ n +_{Jint} m & = & \text{nächste Folie} \\ n *_{Jint} m & = & \text{nächste Folie} \\ n \leq_{Jint} m & \Leftrightarrow & n \leq_{\mathcal{Z}} m \end{array}$$

Java Arithmetik

Für $n, m \in [minInt, maxInt]$ gilt

$$n +_{Jint} m = \begin{cases} n +_{\mathcal{Z}} m & \text{falls } n +_{\mathcal{Z}} m \in [minInt, maxInt] \\ minInt -_{\mathcal{Z}} 1 +_{\mathcal{Z}} ((n +_{\mathcal{Z}} m) -_{\mathcal{Z}} maxInt) & \\ \qquad\qquad \text{falls } n +_{\mathcal{Z}} m > maxInt & \\ maxInt + 1 + ((n +_{\mathcal{Z}} m) - minInt) & \\ \qquad\qquad \text{falls } n +_{\mathcal{Z}} m < minInt & \end{cases}$$

Z.B.

$$maxInt +_{Jint} 1 = minInt \text{ und } minInt -_{Jint} 1 = maxInt$$

Entsprechend für $*_{Jint}$.

Vergleich von $\mathcal{Z}$ und $\mathcal{Z}_{Jint}$

Formel ϕ	$\mathcal{Z} \models \phi$	$\mathcal{Z}_{jint} \models \phi$
$\forall x \exists y (x < y)$	ja	nein
$\forall x \forall y ((x + 1) * y = x * y + y)$	ja	ja
$\exists x (0 < x \wedge x + 1 < 0)$	nein	ja

Koinzidenzlemma

Theorem

$\mathcal{D}$ *sei Interpretation,* β, γ *Variablenbelegungen*

1. *Gilt für den Term t* $\beta(x) = \gamma(x)$ *für alle* $x \in Var(t)$*, dann* $val_{\mathcal{D},\beta}(t) = val_{\mathcal{D},\gamma}(t)$.
2. *Gilt für die Formel A* $\beta(x) = \gamma(x)$ *für alle* $x \in Frei(A)$*, dann* $val_{\mathcal{D},\beta}(A) = val_{\mathcal{D},\gamma}(A)$.
3. *Ist* $A \in For_\Sigma$ *geschlossen, dann gilt* $val_{\mathcal{D},\beta}(A) = val_{\mathcal{D},\gamma}(A)$

Beweis: Durch strukturelle Induktion unter Ausnutzung der Definition von *val*.

Substitutionslemma für Terme

Theorem

Σ sei eine Signatur, $\mathcal{D}$ eine Interpretation für Σ,
β eine Belegung, σ eine Substitution und $t \in Term_\Sigma$.

Dann gilt

$$val_{\mathcal{D},\beta}(\sigma(t)) = val_{\mathcal{D},\beta'}(t).$$

wobei

$$\beta'(x) = val_{\mathcal{D},\beta}(\sigma(x))$$

für alle $x \in Var$.

Beweis

Strukturelle Induktion nach t.

Beweis

Strukturelle Induktion nach t.

$t = x \in Var$:

$$\begin{array}{lcll} val_{\mathcal{D},\beta}(\sigma(x)) & = & \beta'(x) & \text{Def. von } \beta' \\ & = & val_{\mathcal{D},\beta'}(x) & \text{Def. von } val(x) \end{array}$$

Beweis
Induktionsschritt

$$t = f(t_1, \ldots, t_n):$$

$$\begin{aligned} val_{\mathcal{D},\beta}(\sigma(f(t_1, \ldots, t_n))) = \\ = \quad val_{\mathcal{D},\beta}(f(\sigma(t_1), \ldots, \sigma(t_n))) \end{aligned}$$

Beweis
Induktionsschritt

$$t = f(t_1, \ldots, t_n):$$

$$\begin{aligned} val_{\mathcal{D},\beta}(\sigma(f(t_1, \ldots, t_n))) = & \\ = \quad & val_{\mathcal{D},\beta}(f(\sigma(t_1), \ldots, \sigma(t_n))) \end{aligned}$$

Beweis
Induktionsschritt

$t = f(t_1, \ldots, t_n)$:

$$
\begin{aligned}
val_{\mathcal{D},\beta}(\sigma(f(t_1, \ldots, t_n))) = \\
&= val_{\mathcal{D},\beta}(f(\sigma(t_1), \ldots, \sigma(t_n))) \\
&= I(f)(val_{\mathcal{D},\beta}(\sigma(t_1)), \ldots, val_{\mathcal{D},\beta}(\sigma(t_n)))
\end{aligned}
$$

Beweis
Induktionsschritt

$$t = f(t_1, \ldots, t_n):$$

$$\begin{aligned}
val_{\mathcal{D},\beta}(\sigma(f(t_1, \ldots, t_n))) = & \\
= \quad & val_{\mathcal{D},\beta}(f(\sigma(t_1), \ldots, \sigma(t_n))) \\
= \quad & I(f)(val_{\mathcal{D},\beta}(\sigma(t_1)), \ldots, val_{\mathcal{D},\beta}(\sigma(t_n)))
\end{aligned}$$

Beweis
Induktionsschritt

$$t = f(t_1, \dots, t_n):$$

$$val_{\mathcal{D},\beta}(\sigma(f(t_1, \dots, t_n))) =$$

$$\begin{aligned}
&= val_{\mathcal{D},\beta}(f(\sigma(t_1), \dots, \sigma(t_n))) \\
&= I(f)(val_{\mathcal{D},\beta}(\sigma(t_1)), \dots, val_{\mathcal{D},\beta}(\sigma(t_n))) \\
&= I(f)(val_{\mathcal{D},\beta'}(t_1), \dots, val_{\mathcal{D},\beta'}(t_n)) \\
&\quad \text{(nach Induktionsannahme)}
\end{aligned}$$

Beweis
Induktionsschritt

$$t = f(t_1, \ldots, t_n):$$

$$val_{\mathcal{D},\beta}(\sigma(f(t_1, \ldots, t_n))) =$$

$$\begin{aligned}
&= val_{\mathcal{D},\beta}(f(\sigma(t_1), \ldots, \sigma(t_n))) \\
&= I(f)(val_{\mathcal{D},\beta}(\sigma(t_1)), \ldots, val_{\mathcal{D},\beta}(\sigma(t_n))) \\
&= I(f)(val_{\mathcal{D},\beta'}(t_1), \ldots, val_{\mathcal{D},\beta'}(t_n))
\end{aligned}$$

(nach Induktionsannahme)

Beweis
Induktionsschritt

$$t = f(t_1, \ldots, t_n):$$
$$val_{\mathcal{D},\beta}(\sigma(f(t_1, \ldots, t_n))) =$$

$$\begin{aligned}
&= val_{\mathcal{D},\beta}(f(\sigma(t_1), \ldots, \sigma(t_n))) \\
&= I(f)(val_{\mathcal{D},\beta}(\sigma(t_1)), \ldots, val_{\mathcal{D},\beta}(\sigma(t_n))) \\
&= I(f)(val_{\mathcal{D},\beta'}(t_1), \ldots, val_{\mathcal{D},\beta'}(t_n)) \\
&\quad \text{(nach Induktionsannahme)} \\
&= val_{\mathcal{D},\beta'}(f(t_1, \ldots, t_n)).
\end{aligned}$$

Quiz

Es bezeichne F die Formel

$$p(x,z) \wedge \exists y(p(x,y) \wedge p(z,y) \to p(x,y))$$

Welche der folgenden Subsitutionen ist kollisionsfrei für F?

σ_1	$\{x/a, z/b\}$	*kollisionsfrei*
σ_2	$\{x/(x+z), z/(x+z)\}$	*kollisionsfrei*
σ_3	$\{x/(x+y), z/a\}$	*Kollision*
σ_4	$\{x/y\}$	*Kollision*
σ_5	$\{x/z\}$	*kollisionsfrei*

Substitutionslemma für Formeln

Theorem

Σ sei eine Signatur, $\mathcal{D}$ eine Interpretation für Σ,
β eine Belegung, $A \in For_\Sigma$ und
σ eine für A kollisionsfreie Substitution.

Dann gilt:

$$val_{\mathcal{D},\beta}(\sigma(A)) = val_{\mathcal{D},\beta'}(A),$$

wobei

$$\beta'(x) = val_{\mathcal{D},\beta}(\sigma(x))$$

für alle $x \in Var$.

Beweis

Induktion nach A.
Exemplarisch: Schritt von A nach $\exists x A$.
Notation: val_β abkürzend für $val_{\mathcal{D},\beta}$.
Außerdem: $\sigma_x(x) = x,\ \sigma_x(y) = \sigma(y)$ für $y \neq x$.

$$
\begin{array}{lll}
val_\beta(\sigma(\exists x A)) = W & \text{gdw} & val_\beta(\exists x \sigma_x(A)) = W \\
 & & \text{Anwendung von } \sigma \\
 & \text{gdw} & val_{\beta_x^d}(\sigma_x(A)) = W \text{ für ein } d \in D \\
 & & \text{Def. von } val \\
 & \text{gdw} & val_{(\beta_x^d)''}(A) = W \qquad \textcolor{red}{\text{Ind.Vor}} \\
 & & \text{wo } (\beta_x^d)''(y) = val_{\beta_x^d}(\sigma_x(y)) \text{ für all } y. \\
 & \text{gdw} & val_{(\beta')_x^d}(A) = W \qquad \textcolor{red}{\text{Lücke}} \\
 & \text{gdw} & val_{\beta'}(\exists x A) = W \\
 & & \text{Def. von } val
\end{array}
$$

Beweis

Der Beweis wird vollständig geführt sein, wenn wir die Lücke

$$(\beta_x^d)'' = (\beta')_x^d$$

schließen können. Wir müssen für jede Variable $y \in \mathit{Frei}(A)$ zeigen $(\beta_x^d)''(y) = (\beta')_x^d(y)$.

$y = x$:

$$\begin{array}{lcll}
(\beta_x^d)''(x) & = & val_{\beta_x^d}(\sigma_x(x)) & \text{Def. von } (\beta_x^d)'' \\
& = & val_{\beta_x^d}(x) & \text{Def. von } \sigma_x \\
& = & \beta_x^d(x) & \text{Def. von } \mathit{val} \text{ für Variable} \\
& = & d & \text{Def. der modifizierten Belegung} \\
& = & (\beta')_x^d(x) & \text{Def. der modifizierten Belegung}
\end{array}$$

Beweis (Forts)

$y \neq x$, y frei in A:

$$
\begin{array}{rcll}
(\beta_x^d)''(y) & = & val_{\beta_x^d}(\sigma_x(y)) & \text{Def. von } (\beta_x^d)'' \\
 & = & val_{\beta_x^d}(\sigma(y)) & \text{Def. von } \sigma_x \\
 & = & val_{\beta}(\sigma(y)) & \text{da } x \text{ nicht in } \sigma(y) \text{ vorkommt} \\
 & & & \text{Kollisionsfreiheit von } \sigma \\
 & = & \beta'(y) & \text{Def. von } \beta' \\
 & = & (\beta')_x^d(y) & \text{Def. der modifizierten Belegung}
\end{array}
$$

Sir Anthony Hoare

Sir C.A.R.Hoare
Studied philosophy at Oxford U.
Graduate at Moscow State U. 1959
Programmer for Elliott Brothers, 1960
Prof. of CS at Queen's U. Belfast, 1968
An axiomatic basis for computer programimg
Communications ACM, 1969
Oxford U. Programming Research, 1977
Microsoft Research, Cambridge, now

Hoare-Kalkül und Substitutionslemma

Die Zuweisungsregel im Hoare-Kalkül lautet:

$$\{\{x/s\}A\}\ \mathrm{x := s}\ \{A\}$$

wobei die Substitution $\{x/s\}$ kollisionsfrei sein muß.
Die Zuweisungsregel besagt, daß

- ausgehend von einem Zustand, in dem die Formel $\{x/s\}A$ wahr ist,
- nach Ausführung der Programmstücks $x := s$
- ein Zustand erreicht wird, in dem die Formel A gilt.

Hoare-Kalkül und Substitutionslemma

Hintergrund-Interpretation $\mathcal{H}$.

Programmzustand$=$ Variablenbelegung β.

Gelte $val_{\mathcal{H},\beta}(\{x/s\}A) = W$

Nach der Zuweisung $x := s$ wird ein Zustand β' erreicht

$$\beta'(y) := \begin{cases} val_{\mathcal{H},\beta}(s) \text{ falls } x = y \\ \beta(y) \text{ sonst} \end{cases}$$

Die Regel behauptet $val_{\mathcal{H},\beta'}(A) = W$.

Das ist gerade die Aussage des Substitutionslemmas für die Formel A ist und die Substitution $\sigma = \{x/s\}$.

Anwendung des Substitutionslemmas

Theorem

Sei Σ eine Signatur, $\mathcal{D}$ eine Interpretation für Σ, β eine Belegung und σ eine für A kollisionsfreie Substitution mit $\sigma(y) = y$ für alle Variablen $y \neq x$, dann gilt:

- $val_{\mathcal{D},\beta}(\forall xA \to \sigma(A)) = W$
- $val_{\mathcal{D},\beta}(\sigma(A) \to \exists xA) = W$.

Beweis

Wir nehmen an, daß $val_{\mathcal{D},\beta}(\forall x A) = W$ gilt, d.h.

$$val_{\mathcal{D},\beta_x^d}(A) = W \text{ für alle } d \in D.$$

Zu zeigen ist

$$val_{\mathcal{D},\beta}(\sigma(A)) = W$$

Nach dem Substitutionslemma ist das gleichbedeutend mit

$$val_{\mathcal{D},\beta'}(A) = W$$

wobei

$$\beta'(y) = val_{\mathcal{D},\beta}(\sigma(y)) = \begin{cases} \beta(y) \text{ falls } x \neq y \\ val_{\mathcal{D},\beta}(\sigma(x)) \text{ falls y = x} \end{cases}$$

Also $\beta' = \beta_x^d$ für $d = val_{\mathcal{D},\beta}(\sigma(x))$.

Die zweite Aussage läßt sich analog beweisen.

Der Modellbegriff

Den Modell und Folgerungsbegriff definieren wir nur für Formeln und Formelmengen ohne freie Variablen

Das ist mit Abstand der häufigste Anwendungsfall

Der Fall mit freien Variablen wird ausführlich in den Übungsaufgaben im Skript behandelt

Definition

- Eine Interpretation $\mathcal{D}$ über Σ nennen wir ein **Modell** einer Formel A ohne freie Variablen über Σ, wenn $val_{\mathcal{D}}(A) = W$.
- $\mathcal{D}$ heißt **Modell** einer Formelmenge M ohne freie Variablen, wenn für jede Formel $B \in M$ gilt $val_{\mathcal{D}}(B) = W$.

Der logische Folgerungsbegriff

Definition

Es sei $M \subseteq For_\Sigma, A \in For_\Sigma$, beide ohne freie Variablen.

$$M \models_\Sigma A \quad :\Leftrightarrow$$

Jedes Modell von M ist auch Modell von A.

Lies: **Aus** M **folgt** A (über Σ).

Kurznotationen:

$\models$ statt $\models_\Sigma$, $\models A$ für $\emptyset \models A$, $B \models A$ für $\{B\} \models A$.

Bemerkungen zum Modellbegriff

$$M \models A \quad \text{gdw} \quad M \cup \{\neg A\}$$

hat kein Modell

Allgemeingültigkeit

Definition

$A \in For_\Sigma$ heißt

- **allgemeingültig** gdw $\models A$
- **erfüllbar** gdw $\neg A$ ist nicht allgemeingültig.

Allgemeingültigkeit

Theorem

1. *Die folgenden Aussagen sind äquivalent:*
   1. *A allgemeingültig*
   2. *Jede Interpretation $\mathcal{D}$ ist Modell von A.*
   3. $val_{\mathcal{D}}(A) = W$ *für alle* $\mathcal{D}$.
2. *Die folgenden Aussagen sind äquivalent:*
   1. *A erfüllbar*
   2. *Es gibt* $\mathcal{D}$ *mit* $val_{\mathcal{D}}(A) = W$

Beispiele für allgemeingültige Formeln

1. $\neg\forall x A \leftrightarrow \exists x \neg A$,
2. $\neg\exists x A \leftrightarrow \forall x \neg A$
3. $\forall x \forall y A \leftrightarrow \forall y \forall x A$,
4. $\exists x \exists y A \leftrightarrow \exists y \exists x A$
5. $\forall x (A \wedge B) \leftrightarrow \forall x A \wedge \forall x B$
6. $\exists x (A \vee B) \leftrightarrow \exists x A \vee \exists x B$
7. $\forall \vec{y}(A \wedge QxB \leftrightarrow Qx(A \wedge B))$,
   falls $x \notin Frei(A)$ und $\vec{y}$ alle freie Variablen in $A \wedge QxB$ sind.
8. $\forall \vec{y}(QxA \wedge B \leftrightarrow Qx(A \wedge B))$,
   falls $x \notin Frei(B)$ und $\vec{y}$ alle freie Variablen in $A \wedge QxB$ sind.
9. $\forall \vec{y}(A \vee QxB \leftrightarrow Qx(A \vee B))$,
   falls $x \notin Frei(A)$ und $\vec{y}$ alle freie Variablen in $A \wedge QxB$ sind.
10. $\forall \vec{y}(QxA \vee B \leftrightarrow Qx(A \vee B))$,
    falls $x \notin Frei(B)$ und $\vec{y}$ alle freie Variablen in $A \wedge QxB$ sind.

Beweisbeispiel

Voraussetzung: $x \notin \mathit{Frei}(A)$
Für alle $\mathcal{D}, \beta$ ist zu zeigen:

$$val_{\mathcal{D},\beta}(A \rightarrow \forall x B) = val_{\mathcal{D},\beta}(\forall x(A \rightarrow B))$$

Falls $val_{\mathcal{D},\beta}(A \rightarrow \forall x B) = W$, dann folgt unmittelbar aus der Definition von *val* $val_{\mathcal{D},\beta}(\forall x(A \rightarrow B)) = W$ (Übung).

Sei jetzt $val_{D,I,\beta}(\forall x(A \rightarrow B)) = W$, d. h. für alle $d \in D$:
$(val_{\mathcal{D},\beta_x^d}(A) = W \Rightarrow val_{D,I,\beta_x^d}(B) = W)$. (*)

Angenommen, es wäre $val_{D,I,\beta}(A \rightarrow \forall x B) = F$. Dann gilt also

$$val_{\mathcal{D},\beta}(A) = W \quad \text{und} \quad val_{\mathcal{D},\beta}(\forall x B) = F$$

es gibt also ein $e \in D$ mit $val_{\mathcal{D},\beta_x^e}(B) = F$.

Wegen $x \notin \mathit{Frei}(A)$ gilt auch $val_{\mathcal{D},\beta_x^e}(A) = W$. Aus $(*)$ folgt somit der Widerspruch
$val_{\mathcal{D},\beta_x^e}(B) = W$

Beispiel für ein Ableitbarkeitsproblem

$$\begin{array}{lcl} \forall x \forall y \forall z (r(x,y) \land r(y,z) \rightarrow r(x,z)) & & \\ \forall x \forall y (r(x,y) \rightarrow r(y,x)) & \models & \forall x r(x,x) \\ \forall x \exists y (r(x,y) & & \end{array}$$

$$\begin{array}{lcl} \text{Transitivität} & & \\ \text{Symmetrie} & \models & \text{Reflexivität} \\ \text{Endlosigkeit} & & \end{array}$$

Die Antwort ist

JA

2. Beispiel für ein Ableitbarkeitsproblem

$$\neg\exists x(a < x \land c(x) \land \forall y(a \leq y < x \rightarrow b(y))$$

$$\models$$

$$\exists x(a < x \land \neg c(x) \land \forall y(a \leq y < x \rightarrow \neg b(y))$$

Gegenbeispiel:

$$\begin{array}{ccccc} a & & p_1 & & p_2 \\ \cdot & < & \cdot & < & \cdot \\ & & & & \\ b(a) & & \neg b(p_1) & & \neg b(p_2) \\ \neg c(a) & & \neg c(p_1) & & c(p_2) \end{array}$$