

Formale Modellierung
Vorlesung 7 vom 23.05.13: FOL mit Induktion und Rekursion

Serge Autexier & Christoph Lüth

Universität Bremen

Sommersemester 2013

Fahrplan

- ▶ Teil I: Formale Logik
 - ▶ Einführung
 - ▶ Aussagenlogik: Syntax und Semantik, Natürliches Schließen
 - ▶ Konsistenz & Vollständigkeit der Aussagenlogik
 - ▶ Prädikatenlogik (FOL): Syntax und Semantik
 - ▶ Konsistenz & Vollständigkeit von FOL
 - ▶ FOL mit induktiven Datentypen
 - ▶ FOL mit Induktion und Rekursion
 - ▶ Die Gödel-Theoreme
 - ▶ Weitere Datentypen: Mengen, Multimengen, Punkte
- ▶ Teil II: Spezifikation und Verifikation
- ▶ Teil III: Schluß

Das Tagesmenü

- ▶ Axiomatische Definition von Theorien ist gefährlich
- ▶ Prädikatenlogik mit mehreren Typen
- ▶ Konservative Erweiterungen als sicheres Theorie Definitionsprinzip
 - ▶ Typdefinitionen
 - ▶ Wohlfundierte rekursive Funktionen/Prädikate

Natürliches Schließen — Die Regeln

$$\frac{\phi \quad \psi}{\phi \wedge \psi} \wedge I \qquad \frac{\phi \wedge \psi}{\phi} \wedge E_L \quad \frac{\phi \wedge \psi}{\psi} \wedge E_R$$

$$\frac{[\phi] \quad \vdots \quad \psi}{\phi \rightarrow \psi} \rightarrow I \qquad \frac{\phi \quad \phi \rightarrow \psi}{\psi} \rightarrow E$$

$$\frac{\perp}{\phi} \perp \qquad \frac{[\phi \rightarrow \perp] \quad \vdots \quad \perp}{\phi} \text{raa}$$

Die fehlenden Schlußregeln

$$\frac{[\phi] \quad \vdots \quad \perp}{\neg \phi} \neg I \qquad \frac{\phi \quad \neg \phi}{\perp} \neg E$$

$$\frac{\phi}{\phi \vee \psi} \vee I_L \quad \frac{\psi}{\phi \vee \psi} \vee I_R \qquad \frac{[\phi] \quad \vdots \quad \sigma \quad [\psi] \quad \vdots \quad \sigma}{\phi \vee \psi} \vee E$$

$$\frac{\phi \rightarrow \psi \quad \psi \rightarrow \phi}{\phi \leftrightarrow \psi} \leftrightarrow I \quad \frac{\phi \quad \phi \leftrightarrow \psi}{\psi} \leftrightarrow E_L \quad \frac{\psi \quad \phi \leftrightarrow \psi}{\phi} \leftrightarrow E_R$$

Natürliches Schließen mit Quantoren

$$\frac{\phi}{\forall x. \phi} \forall I \quad (*) \qquad \frac{\forall x. \phi}{\phi[x/t]} \forall E \quad (\dagger)$$

- ▶ (*) **Eigenvariablenbedingung:**
x nicht frei in offenen Vorbedingungen von ϕ (x beliebig)
- ▶ (\dagger) Ggf. Umbenennung durch Substitution
- ▶ Gegenbeispiele für verletzte Seitenbedingungen

Der Existenzquantor

$$\exists x. \phi \stackrel{\text{def}}{=} \neg \forall x. \neg \phi$$

$$\frac{\phi[x/t]}{\exists x. \phi} \exists I \quad (\dagger) \qquad \frac{[\phi] \quad \vdots \quad \psi}{\exists x. \phi} \exists E \quad (*)$$

- ▶ (*) **Eigenvariablenbedingung:**
x nicht frei in ψ , oder einer offeneren Vorbedingung außer ϕ
- ▶ (\dagger) Ggf. Umbenennung durch Substitution

Regeln für die Gleichheit

- ▶ Reflexivität, Symmetrie, Transitivität:

$$\frac{}{x = x} \text{ refl} \qquad \frac{x = y}{y = x} \text{ sym} \qquad \frac{x = y \quad y = z}{x = z} \text{ trans}$$

- ▶ Kongruenz:

$$\frac{x_1 = y_1, \dots, x_n = y_n}{f(x_1, \dots, x_n) = f(y_1, \dots, y_n)} \text{ cong}$$

- ▶ Substitutivität:

$$\frac{x_1 = y_1, \dots, x_m = y_m \quad P(x_1, \dots, x_m)}{P(y_1, \dots, y_m)} \text{ subst}$$

Motivation

- ▶ Typen müssen nicht-leere Trägermengen haben Korrektheit
- ▶ Neue Typen axiomatisch zu spezifizieren gefährlich
- ▶ Konservative Erweiterungen
 - ▶ Typdefinitionen sind konservative Erweiterungen
 - ▶ Terminierende totale rekursive Funktionen/Prädikate sind konservative Erweiterungen

9 [25]

Getypte Prädikatenlogik – Signatur

	Ungetypt	Getypt
Signatur Σ		
- Typen \mathcal{T}	-	$i, \mathbb{N}, \mathbb{Z}$
- Funktionssymbole \mathcal{F}	$f, ar(f) = n$	$f : \tau_1 \times \dots \times \tau_n \rightarrow \tau_0, \tau_i \in \mathcal{T}$
- Prädikatssymbole \mathcal{P}	$P, ar(P) = n$ $\doteq, ar(\doteq) = 2$	$P : \tau_1 \times \dots \times \tau_n, \tau_i \in \mathcal{T}$ $\doteq_\tau : \tau \times \tau, \tau \in \mathcal{T}$
Variablen X	abz. unendlich	abz. unendlich X_τ für jedes $\tau \in \mathcal{T}$ $x_i, x_{\mathbb{N}}, x_{\mathbb{Z}}, \dots$

10 [25]

Getypte Prädikatenlogik – Terme & Formeln

	Ungetypt	Getypt
Terme $Term_\Sigma$		$Term_\Sigma^{\tau_1} \cup \dots \cup Term_\Sigma^{\tau_n}, \tau \in \mathcal{T}$ $Term_\Sigma^{\tau_0}, \tau \in \mathcal{T}$
- Variablen	$x \in Term_\Sigma, x \in X$	$x \in Term_\Sigma^\tau, x \in X_\tau$
- Funktionen	$f \in \mathcal{F}$ mit $ar(f) = n$ und $t_1, \dots, t_n \in Term_\Sigma$, dann $f(t_1, \dots, t_n) \in Term_\Sigma$	$f : \tau_1 \times \dots \times \tau_n \rightarrow \tau_0 \in \mathcal{F}$ und $t_i \in Term_\Sigma^{\tau_i}, 1 \leq i \leq n$, dann $f(t_1, \dots, t_n) \in Term_\Sigma^{\tau_0}$
Formeln $Form_\Sigma$		
- Atome	$P \in \mathcal{P}$ mit $ar(P) = n$ und $t_1, \dots, t_n \in Term_\Sigma$, dann $P(t_1, \dots, t_n) \in Form_\Sigma$	$P : \tau_1 \times \dots \times \tau_n \in \mathcal{P}$ und $t_i \in Term_\Sigma^{\tau_i}, 1 \leq i \leq n$, dann $P(t_1, \dots, t_n) \in Form_\Sigma$
- PL Konnective	$\neg\psi, \varphi \wedge \psi, \varphi \vee \psi, \varphi \rightarrow \psi, \varphi \leftrightarrow \psi, \dots$	$\neg\psi, \varphi \leftrightarrow \psi, \dots$
- Quantoren	$\forall x. \phi \in Form_\Sigma, x \in X$ $\exists x. \phi \in Form_\Sigma, x \in X$	$\forall x_\tau. \phi \in Form_\Sigma$ $\exists x_\tau. \phi \in Form_\Sigma$

11 [25]

Getypte Prädikatenlogik – ND Regeln

$$\frac{\phi}{\forall x_\tau. \phi} \forall I \quad (*) \qquad \frac{\forall x_\tau. \phi}{\phi[x]} \forall E \quad (\dagger)$$

- ▶ (*) **Eigenvariablenbedingung:**
 x_τ nicht frei in offenen Vorbedingungen von ϕ (x_τ beliebig)
- ▶ (\dagger) $t \in Term_\Sigma^\tau$; Ggf. Umbenennung durch Substitution

12 [25]

Der Existenzquantor

$$\exists x_\tau. \phi \stackrel{def}{=} \neg \forall x_\tau. \neg \phi$$

$$\frac{\phi[x]}{\exists x_\tau. \phi} \exists I \quad (\dagger) \qquad \frac{\begin{array}{c} [\phi] \\ \vdots \\ \exists x_\tau. \phi \end{array} \psi}{\psi} \exists E \quad (*)$$

- ▶ (*) **Eigenvariablenbedingung:**
 x_τ nicht frei in ψ , oder einer offeneren Vorbedingung außer ϕ
- ▶ (\dagger) $t \in Term_\Sigma^\tau$; Ggf. Umbenennung durch Substitution

13 [25]

Regeln für die Gleichheit

- ▶ Reflexivität, Symmetrie, Transitivität:

$$\frac{}{x =_t x} \text{ refl} \qquad \frac{x =_t y \quad y =_t x}{y =_t x} \text{ sym} \qquad \frac{x =_t y \quad y =_t z}{x =_t z} \text{ trans}$$

- ▶ Kongruenz:

$$\frac{x_1 =_t y_1, \dots, x_n =_t y_n}{f(x_1, \dots, x_n) =_t f(y_1, \dots, y_n)} \text{ cong}$$

- ▶ Substitutivität:

$$\frac{x_1 =_t y_1, \dots, x_m =_t y_m \quad P(x_1, \dots, x_m)}{P(y_1, \dots, y_m)} \text{ subst}$$

14 [25]

Basic Definitions

Definition 1 (Loose Spezifikationen)

Sei $\Sigma = (\mathcal{T}, \mathcal{F}, \mathcal{P})$ eine getypte Signatur und $\Phi \in Form_\Sigma$. Dann ist $S = (\Sigma, \Phi)$ eine **lose Spezifikation**.

Die Theorie einer Spezifikation S ist $Th(S) := \{\varphi \in Form_\Sigma \mid \Phi \vdash \varphi\}$.

Definition 2 (Konsistenz)

Eine lose Spezifikation S ist **konsistent** wenn \perp nicht beweisbar in S : $\perp \notin Th(S)$.

- ▶ Insbesondere müssen dann **alle** Typen nicht-leere Trägermengen haben

15 [25]

Spezifikations Erweiterungen

Definition 3 (Erweiterungen)

Eine Spezifikation $S' = (\Sigma', \Phi')$ ist eine **Erweiterung** einer Spezifikation $S = (\Sigma, \Phi)$ genau dann wenn

- ▶ $\Sigma \subseteq \Sigma'$
- ▶ $\Phi \subseteq \Phi'$

S' ist eine **konservative Erweiterung** von S genau dann wenn

$$Th(S) = Th(S')|_\Sigma$$

wobei die $|_\Sigma$ die Einschränkung auf Formeln aus $Term_\Sigma$ ist

Lemma 4

Jede konservative Erweiterung einer konsistenten Theorie ist konsistent.

16 [25]

Typdefinition

- Spezifiziere **nicht-leere** Teilmenge eines gegebenen Typs r
- Deklariere neuen Typ t mit Trägermenge isomorph zu Werten in spezifizierter Teilmenge
- Isomorphie wird durch inverse Funktionen $\text{Abs}_t : r \rightarrow t, \text{Rep}_t : t \rightarrow r$ axiomatisch beschrieben

17 [25]

Typdefinitionen sind Erweiterungen

Definition 5 (Typdefinitionen)

Sei $S = ((\mathcal{T}, \mathcal{F}, \mathcal{P}), \Phi)$ eine Spezifikation, $r \in \mathcal{T}$ und $P \in \text{Form}_{\Sigma}$ mit genau einer freien Variable vom Typ r . Dann ist eine Erweiterung $S' = ((\mathcal{T}', \mathcal{F}', \mathcal{P}'), \Phi')$ eine **Typdefinition** für einen Typ $t \notin \mathcal{T}$ gdw.

- $\mathcal{T}' = \mathcal{T} \cup \{t\}$
- $\mathcal{F}' = \mathcal{F} \cup \{\text{Abs}_t : r \rightarrow t, \text{Rep}_t : t \rightarrow r\}$
- $\mathcal{P}' = \mathcal{P} \cup \{=: t \times t\}$
- $\Phi' = \Phi \cup \{ \forall x_t. \text{Abs}_t(\text{Rep}_t(x)) =_t x, \forall x_r. P(x_r) \rightarrow \text{Rep}_t(\text{Abs}_t(x)) =_r x \}$
- Man kann beweisen $S \vdash \exists x_r. P(x)$ (bzw. es gilt $\exists x_r. P(x) \in \text{Th}(S)$)

18 [25]

Terminierende, totale Funktionen

- Spezifiziere Funktionen/Prädikate die beweisbar total, eindeutig und terminierend sind
- Theorie-Erweiterungen um beweisbar total, eindeutig und terminierende Funktionen/Prädikate sind konservative Erweiterungen
- Syntaktische Kriterien für eindeutige und totale Deklarationen
- Beweisverfahren für terminierende Funktionen

19 [25]

Frei Erzeugte Typen

Definition 6 (Frei Erzeugte Typen)

Sei $S = ((\mathcal{T}, \mathcal{F}, \mathcal{P}), \Phi)$ eine Spezifikation, $t \in \mathcal{T}$ and $c_i : \tau_1^i \times \dots \times \tau_{n_i}^i \rightarrow t \in \mathcal{F}, 1 \leq i \leq k$. Dann ist t **frei erzeugt** in S durch **Konstruktoren** c_1, \dots, c_k gdw.

- $S \vdash \forall x_t. \bigvee_{i=1 \dots k} \exists y_{\tau_1^i}^1, \dots, y_{\tau_{n_i}^i}^k. x = c_i(y^1, \dots, y^{n_i})$
- $S \vdash \forall y_{\tau_1^i}^1, \dots, y_{\tau_{n_i}^i}^{n_i}. \forall z_{\tau_1^i}^1, \dots, z_{\tau_{n_i}^i}^{n_i}. c_i(y^1, \dots, y^{n_i}) = c_i(z^1, \dots, z^{n_i}) \rightarrow ((y^1 = z^1 \wedge \dots \wedge y^{n_i} = z^{n_i}))$ für alle c_i
- $S \vdash \forall y_{\tau_1^i}^1, \dots, y_{\tau_{n_i}^i}^{n_i}. \forall z_{\tau_1^j}^1, \dots, z_{\tau_{n_j}^j}^{n_j}. c_i(y^1, \dots, y^{n_i}) = c_j(z^1, \dots, z^{n_j})$ für alle $i \neq j$

20 [25]

Kriterien für eindeutig und total

- Sei t Typ
- Definitionsgleichungen für Funktion f sind Menge von bedingten geschlossene Gleichungen der Form

$$\begin{aligned} \forall x_{1\tau_1} \dots x_{n\tau_n} \dots P_0 \rightarrow f(x_1, \dots, x_n) = t_0 \\ \vdots \\ \forall x_{1\tau_1} \dots x_{n\tau_n} \dots P_n \rightarrow f(x_1, \dots, x_n) = t_n \end{aligned}$$

so daß beweisbar

- $S \vdash \forall x_{1\tau_1} \dots x_{n\tau_n}. P_i \wedge P_j \leftrightarrow \perp, \forall i \neq j$
- $S \vdash \forall x_{1\tau_1} \dots x_{n\tau_n}. P_1 \vee \dots \vee P_n$

21 [25]

Terminierungsbeweise – Idee

- Die natürlichen Zahlen sind frei erzeugt über 0 und s:
- Jedem Grundterm über \mathbb{N} kann eine große zugeordnet werden über die Anzahl der Konstruktoren.
- Zeige für rekursiv definierte Funktionen auf \mathbb{N} , dass die rekursiven Argumente in rekursiven Funktionsaufrufen kleiner sind bezüglich der Ordnung auf den natürlichen Zahlen unter der entsprechenden Bedingung P_i .

22 [25]

Terminierung

- Beispiele:
 - half(x) eine Hypothese pro Rekursionsgleichung
 - fib(x): mehrere Hypothesen pro Rekursionsgleichung
 - gcd(x, y): lexicographische Ordnung
- Beweise alle Hypothesen im Kalkül. Terminierung gilt **relativ** zur Terminierung der anderen involvierten Funktionen und Prädikate.
- Analog für Prädikate auf \mathbb{N} mit bedingten Äquivalenzen
- Allgemeine Typen:** für frei erzeugte Datentypen kann Abbildung in natürliche Zahlen definiert werden, die die Anzahl der Konstruktoren zählt. Damit lässt sich das Terminierungsverfahren auf all frei erzeugten Datentypen erweitern

23 [25]

Erweiterung um Totale, Terminierende Funktionen is Konservativ

Definition 7 (Funktions- und Prädikatsdefinitionen)

Sei $S = ((\mathcal{T}, \mathcal{F}, \mathcal{P}), \Phi)$ eine Spezifikation, $f : \tau_1 \times \dots \times \tau_n \rightarrow \tau_0 \notin \mathcal{F}$ ($\tau_i \in \mathcal{T}$) und $\Psi \in \text{Form}_{\Sigma \cup \{f\}}$. Dann ist eine Erweiterung $S' = ((\mathcal{T}, \mathcal{F}', \mathcal{P}), \Phi')$ eine **Funktionsdefinition** gdw.

- Ψ ist eine eindeutig und totale Definition für f
- f ist terminierend und alle in der Definition von f vorkommenden Funktionen und Prädikate sind terminierend
- $\Phi' = \Phi \cup \Psi$
- $\mathcal{F}' = \mathcal{F} \cup \{f : \tau_1 \times \dots \times \tau_n \rightarrow \tau_0\}$

Analog für **Prädikatsdefinitionen**.

Lemma 8

Funktionsdefinitionen bzw. Prädikatsdefinitionen sind konservativ

24 [25]

Sicheres Spezifikationsprinzip

- ▶ Beginne mit Basistheorie mit \mathbb{N} und wohlfundiertem Induktionsschemata für \mathbb{N} (getypte Prädikatenlogik mit Typ \mathbb{N} und Induktionsschemata!)
 - ▶ \mathbb{N} hat beweisbar nicht-leere Trägermenge
- ▶ Erweitere nur konservativ um
 - ▶ totale, terminierende Funktionen und Prädikate
 - ▶ Typdefinitionen (ausgehend von \mathbb{N})
 - ▶ Erbt Induktionsprinzip über Umweg über \mathbb{N}
- ▶ Erlaubt Definition von Konstruktoren für neue Typen
 - ▶ Terminierung: Abbildung der Termgröße auf \mathbb{N} mittels geschachtelter Anwendung von Rep_t
 - ▶ Wenn freie Erzeugtheit des neuen Typs beweisbar, dann folgt Induktionsschema direkt auf dem neuen Typ
- ▶ Damit hat man garantiert immer konsistente Spezifikationen (= Modellierung).