Formale Modellierung
Vorlesung 8 vom 07.06.14: FOL mit Induktion und Rekursion

Serge Autexier & Christoph Lüth

Universität Bremen

Sommersemester 2014

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
  - Beschreibungslogiken
  - FOL mit induktiven Datentypen
  - FOL mit Induktion und Rekursion
  - Die Unvollständigkeitssätze von Gödel
- Teil II: Spezifikation und Verifikation

Das Tagesmenü

- Beweis von Eigenschaften von Funktionen mit FOL-ND
  - Wohlfundierte Induktion und rekursive Funktionen
- Axiomatische Definition von Theorien ist gefährlich
- Prädikatenlogik mit mehreren Typen
- Konservative Erweiterungen als sicheres Theorie Definitionsprinzip
  - Typdefinitionen
  - Wohlfundierte rekursive Funktionen/Prädikate
- Terminierende Funktionen und abgeleitete Induktionsschemata

Mehr Beweise

- Definiere $\leq$ und half:

$$\forall x.0 \leq x \tag{L1}$$
$$\forall x.\forall y.x \leq y \longrightarrow \mathsf{s}(x) \leq \mathsf{s}(y) \tag{L2}$$

$$\mathsf{half}(0) = 0 \tag{H1}$$
$$\mathsf{half}(\mathsf{s}(0)) = 0 \tag{H2}$$
$$\forall x.\, \mathsf{half}(\mathsf{s}(\mathsf{s}(x))) = \mathsf{s}(\mathsf{half}(x)) \tag{H3}$$

- Beweise

$$(\text{Presburger})(\text{L1})(\text{L2})(\text{H1})(\text{H2})(\text{H3}) \vdash \forall x.\, \mathsf{half}(x) \leq x$$

Wohlfundierte Induktion

- Wohlfundiertes Induktionsschema

$$(\forall y.(\forall x.x < y \wedge P(x)) \Rightarrow P(y)) \longrightarrow \forall x.P(x)$$

- $<$ wohlfundierte Relation:

$$\forall X \subseteq \mathbb{N}.X \neq \emptyset \longrightarrow \exists x \in X.\forall y \in X.\neg(y < x)$$

Beweis mit wohlfundierter Induktion

- $<$-Relation

$$\forall x.0 < \mathsf{s}(x) \qquad \forall x, y.x < y \longrightarrow \mathsf{s}(x) < \mathsf{s}(y)$$

- Beweise $<$ ist wohlfundiert
-

$$\frac{\begin{array}{c}\left[\ \forall x.x < c \wedge P(x)\ \right] \\ \vdots \\ P(c)\end{array}}{\forall x.P(x)}$$

Beweis mit wohlfundierter Induktion

- $<$-Relation

$$\forall x.0 < \mathsf{s}(x) \qquad\qquad \forall x, y.x < y \longrightarrow \mathsf{s}(x) < \mathsf{s}(y)$$

- Beweise $<$ ist wohlfundiert
-

$$\dfrac{\left[\begin{array}{c} \forall x.x < c \\ \mathsf{half}(x) \leq x \\ c = 0 \end{array}\right] \;\|\; \left[\begin{array}{c} \forall x.x < c \\ \mathsf{half}(x) \leq x \\ c = \mathsf{s}(0) \end{array}\right] \;\|\; \left[\begin{array}{c} \forall x.x < c \\ \mathsf{half}(x) \leq x \\ \exists u.c = \mathsf{s}(\mathsf{s}(u)) \end{array}\right]}{\begin{array}{l} c = 0 \vee \\ c = \mathsf{s}(0) \vee \\ \exists u.c = \mathsf{s}(\mathsf{s}(u)) \end{array} \quad \begin{array}{c} \vdots \\ \mathsf{half}(c) \leq c \end{array} \qquad \begin{array}{c} \vdots \\ \mathsf{half}(c) \leq c \end{array} \qquad \begin{array}{c} \vdots \\ \mathsf{half}(c) \leq c \end{array}}{\forall x.\, \mathsf{half}(x) \leq x}$$

Mehr Information

- Besser zum beweisen wäre wenn man gleich hätte

$$\dfrac{\begin{array}{c}\left[\ \mathsf{half}(c) \leq c\ \right] \\ \vdots \\ \mathsf{half}(0) \leq 0 \quad \mathsf{half}(\mathsf{s}(0)) \leq \mathsf{s}(0) \quad \mathsf{half}(\mathsf{s}(\mathsf{s}(c))) \leq \mathsf{s}(\mathsf{s}(c))\end{array}}{\forall x.\, \mathsf{half}(x) \leq x}$$

Mehr Information

- Besser zum beweisen wäre wenn man gleich hätte

$$\frac{\begin{array}{c}\left[\ \mathsf{half}(c) \leq c\ \right] \\ \vdots \\ \mathsf{half}(0) \leq 0 \quad \mathsf{half}(\mathsf{s}(0)) \leq \mathsf{s}(0) \quad \mathsf{half}(\mathsf{s}(\mathsf{s}(c))) \leq \mathsf{s}(\mathsf{s}(c))\end{array}}{\forall x.\, \mathsf{half}(x) \leq x}$$

- Vergleiche:

$$\mathsf{half}(0) = 0 \tag{H1}$$
$$\mathsf{half}(\mathsf{s}(0)) = 0 \tag{H2}$$
$$\forall x.\, \mathsf{half}(\mathsf{s}(\mathsf{s}(x))) = \mathsf{s}(\mathsf{half}(x)) \tag{H3}$$

Mehr Information

- Besser zum beweisen wäre wenn man gleich hätte

$$\dfrac{\begin{array}{c}\left[\ \mathsf{half}(c) \leq c\ \right] \\ \vdots \\ \mathsf{half}(0) \leq 0 \quad \mathsf{half}(\mathsf{s}(0)) \leq \mathsf{s}(0) \quad \mathsf{half}(\mathsf{s}(\mathsf{s}(c))) \leq \mathsf{s}(\mathsf{s}(c))\end{array}}{\forall x.\, \mathsf{half}(x) \leq x}$$

- Vergleiche:

$$\mathsf{half}(0) = 0 \tag{H1}$$
$$\mathsf{half}(\mathsf{s}(0)) = 0 \tag{H2}$$
$$\forall x.\, \mathsf{half}(\mathsf{s}(\mathsf{s}(x))) = \mathsf{s}(\mathsf{half}(x)) \tag{H3}$$

- Generiere Induktionschema aus rekursiven Funktionsdefinitionen

$$\dfrac{\begin{array}{c}\left[\ P(c)\ \right] \\ \vdots \\ P(0) \quad P(\mathsf{s}(0)) \quad P(\mathsf{s}(\mathsf{s}(c)))\end{array}}{\forall x.P(x)}$$

Weitere Beispiele

$$\mathbb{LIST} := \mathsf{Nil} \mid \mathsf{cons}(\mathbb{N}, \mathbb{LIST})$$

- Sortieren

$\forall x.\, \mathsf{sort}(\mathsf{Nil}) = \mathsf{Nil}$

$\forall s, t.m = \min(\mathsf{cons}(n, l))$

$\quad \longrightarrow \mathsf{sort}(\mathsf{cons}(n, l)) = \mathsf{cons}(m, \mathsf{sort}(\mathsf{cons}(n, l) - m))$

$\forall n.\, \min(\mathsf{cons}(n, \mathsf{Nil})) = n$

$\forall n, l.\, \min(\mathsf{cons}(m, l)) < n \longrightarrow \min(\mathsf{cons}(n, \mathsf{cons}(m, l))) = \min(\mathsf{cons}(m, l))$

$\forall n, l. \neg(\min(\mathsf{cons}(m, l)) < n) \longrightarrow \min(\mathsf{cons}(n, \mathsf{cons}(m, l))) = n$

- Induktionsschema

$$\frac{\begin{array}{ll} & \forall m, n.m = \min(\mathsf{cons}(n, l)) \wedge P(\mathsf{cons}(n, l) - m) \\ P(\mathsf{Nil}) & \longrightarrow P(\mathsf{cons}(n, l)) \end{array}}{\forall l.P(l)}$$

Weitere Beispiele

- Fibonacci:

$$\mathsf{fib}(0) = 0$$
$$\mathsf{fib}(\mathsf{s}(0)) = s(0)$$
$$\forall n.\, \mathsf{fib}(s(s(n))) = \mathsf{fib}(s(n)) + \mathsf{fib}(n)$$

$$\frac{P(0) \quad P(\mathsf{s}(0)) \qquad \begin{array}{c}\left[\; P(\mathsf{s}(c)), P(c) \;\right] \\ \vdots \\ P(\mathsf{s}(\mathsf{s}(c)))\end{array}}{\forall x.P(x)}$$

Weitere Beispiele

- GGT:

$$\forall y.\, \mathrm{ggt}(0, y) = y$$
$$\forall x.\, \mathrm{ggt}(\mathsf{s}(x), 0) = \mathsf{s}(x)$$
$$\forall x, y. x \leq y \longrightarrow \mathrm{ggt}(x, y) = \mathrm{ggt}(x, y - x)$$
$$\forall x, y. \neg(x \leq y) \longrightarrow \mathrm{ggt}(x, y) = \mathrm{ggt}(x - y, y)$$

$$\frac{\forall y. P(0, y) \quad \forall x. P(\mathsf{s}(x), 0) \quad \begin{bmatrix} x \leq y \\ P(x, y - x) \end{bmatrix} \quad \begin{bmatrix} \neg(x \leq y) \\ P(x - y, x) \end{bmatrix}}{\begin{matrix} \vdots & \vdots \\ P(x, y) & P(x, y) \end{matrix}}{\forall x, y. P(x, y)}$$

Zulässige Induktionsschema

- Wann darf man die Rekursionsstruktur verwenden?
- Definierte Funktion muß...
  - eindeutig definiert sein und ...

$$P_0 \longrightarrow f(x_1, \ldots, x_n) = t_0$$
$$\vdots$$
$$P_n \longrightarrow f(x_1, \ldots, x_n) = t_n$$

    $P_i \wedge P_j \longleftrightarrow \bot, \forall i \neq j$
  - terminierend
- Rekursive Definition nach wohlfundierter Relation garantiert Terminierung
  Für jeden atomaren, rekursiven Aufruf $f(t_1, \ldots, t_n)$ erzeuge Terminierungshypothese

$$P_i \longrightarrow (x_1, \ldots, x_n) > (t_1, \ldots, t_n)$$

Grenzen

$$\forall x.x < 101 \longrightarrow f(x) = f(f(x + 11))$$
$$\forall x.\neg(x < 101) \longrightarrow f(x) = x - 10$$

Grenzen

$$\forall x.x < 101 \longrightarrow f(x) = f(f(x + 11))$$
$$\forall x.\neg(x < 101) \longrightarrow f(x) = x - 10$$

- f terminiert immer
- f ist

$$f(x) := \begin{cases} x - 10 & \text{if } x > 100 \\ 91 & \text{if } x \leq 100 \end{cases}$$

- Definition der geeigneten wohlfundierten Relation extrem schwierig.

```
f(99) = f(f(110))
      = f(100)
      = f(f(111))
      = f(101)
      = 91
```

```
f(87) = f(f(98))
      = f(f(f(109)))
      = f(f(99))
      = f(f(f(110)))
      = f(f(100))
      = f(f(f(111)))
      = f(f(101))
      = f(91)
      = f(f(102))
      = f(92)
      = f(f(103))
      = f(93)
        .... Pattern continues
      = f(99)
        (same as on the left)
      = 91
```

Zusammenfassung

- Strukturelle Induktionsschema
  - Einfach, aber zum Beweisen zu rigide
- Wohlfundiertes Induktionsschema
  - Mächtig und flexibel, wenig Hilfestellung beim Beweisen
- Wohlfundierte Relation aus Rekursionsstruktur terminierender Funktionen
  - Angepasst an Beweisproblem und vorhandene Definitionsgleichungen
  - Terminierungsbeweis notwendig (einfache Fälle automatisierbar, i.A. unentscheidbar)

Definition von Theorien

- Was alles schiefgehen kann und wie man das vermeidet
- Axiomatische Definition von Theorien ist gefährlich
- Bisher können wir Listen nicht unterscheiden von nat. Zahlen, von binären Bäumen, etc.
- Wir brauchen so etwas wie Typen für die jeweiligen Objekte, die disjunkt voneinander sind

Getypte Prädikatenlogik – Signatur

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

Getypte Prädikatenlogik – Terme & Formeln

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

Motivation

- Typen müssen nicht-leere Trägermengen haben

Motivation

- Typen müssen nicht-leere Trägermengen haben — Korrektheit
- Neue Typen axiomatisch zu spezifizieren gefährlich

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

Natürliches Schließen — Die Regeln

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

$$\frac{\begin{array}{c}[\phi]\\ \vdots \\ \psi\end{array}}{\phi \longrightarrow \psi}\ \longrightarrow I \qquad \frac{\phi \quad \phi \longrightarrow \psi}{\psi}\ \longrightarrow E$$

$$\frac{\bot}{\phi}\ \bot \qquad \frac{\begin{array}{c}[\phi \longrightarrow \bot]\\ \vdots \\ \bot\end{array}}{\phi}\ \text{raa}$$

Die fehlenden Schlußregeln

$$\dfrac{\begin{array}{c}[\phi]\\ \vdots\\ \bot\end{array}}{\neg\phi}\ \neg I \qquad \dfrac{\phi \quad \neg\phi}{\bot}\ \neg E$$

$$\dfrac{\phi}{\phi \vee \psi}\ \vee I_L \qquad \dfrac{\psi}{\phi \vee \psi}\ \vee I_R \qquad \dfrac{\phi \vee \psi \qquad \begin{array}{c}[\phi]\\ \vdots\\ \sigma\end{array} \qquad \begin{array}{c}[\psi]\\ \vdots\\ \sigma\end{array}}{\sigma}\ \vee E$$

$$\dfrac{\phi \longrightarrow \psi \quad \psi \longrightarrow \phi}{\phi \longleftrightarrow \psi}\ \longleftrightarrow I \qquad \dfrac{\phi \quad \phi \longleftrightarrow \psi}{\psi}\ \longleftrightarrow E_L \qquad \dfrac{\psi \quad \phi \longleftrightarrow \psi}{\phi}\ \longleftrightarrow E_R$$

Natürliches Schließen mit Quantoren

$$\frac{\phi}{\forall x.\phi}\ \forall I\ \ (*) \qquad\qquad \frac{\forall x.\phi}{\phi\left[\begin{smallmatrix} t \\ x \end{smallmatrix}\right]}\ \forall E\ \ (\dagger)$$

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

Der Existenzquantor

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

$$\frac{\phi\left[\begin{smallmatrix} t \\ x \end{smallmatrix}\right]}{\exists x.\phi}\ \exists I\ \ (\dagger) \qquad\qquad \frac{\exists x.\phi \qquad \begin{array}{c}[\phi] \\ \vdots \\ \psi\end{array}}{\psi}\ \exists E\ \ (*)$$

- (*) Eigenvariablenbedingung:
  x nicht frei in ψ, oder einer offenenen Vorbedingung außer ϕ
- (†) 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:

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

- Substitutivität:

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

Getypte Prädikatenlogik – ND Regeln

$$\frac{\phi}{\forall x_\tau.\phi}\ \forall I\ \ (*) \qquad\qquad \frac{\forall x_\tau.\phi}{\phi\left[\begin{smallmatrix} t \\ x \end{smallmatrix}\right]}\ \forall E\ \ (\dagger)$$

- (*) Eigenvariablenbedingung:
  x_τ nicht frei in offenen Vorbedingungen von ϕ (x_τ beliebig)

- ($\dagger$) $t \in \mathcal{T}erm^\tau_\Sigma$; Ggf. Umbenennung durch Substitution

Der Existenzquantor

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

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

- (*) Eigenvariablenbedingung:
  x_τ nicht frei in ψ, oder einer offenenen Vorbedingung außer ϕ
- (†) $t \in \mathcal{T}erm^\tau_\Sigma$; Ggf. Umbenennung durch Substitution

Regeln für die Gleichheit

- Reflexivität, Symmetrie, Transitivität:

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

- Kongruenz:

$$\frac{x_1 =_{t_1} y_1, \ldots, x_n =_{t_n} y_n}{f(x_1, \ldots, x_n) =_t f(y_1, \ldots, y_n)}\ \text{cong}$$

- Substitutivität:

$$\frac{x_1 =_{t_1} y_1, \ldots, x_m =_{t_m} y_m \quad P(x_1, \ldots, x_m)}{P(y_1, \ldots, y_m)}\ \text{subst}$$

Basic Definitions

Definition 1 (Loose Spezifikationen)

Sei $\Sigma = (\mathcal{T}, \mathcal{F}, \mathcal{P})$ eine getypte Signature und $\Phi \in \mathcal{F}orm_\Sigma$. Dann ist $S = (\Sigma, \Phi)$ eine lose Spezifikation.
Die Theorie einer Spezifikation S ist $\mathrm{Th}(S) := \{\varphi \in \mathcal{F}orm_\Sigma | \Phi \vdash \varphi\}$.

Definition 2 (Konsistenz)

Eine lose Spezifikation S ist konsistent wenn $\bot$ nicht beweisbar in S: $\bot \notin \mathrm{Th}(S)$.

- Insbesondere müssen dann alle Typen nicht-leere Trägermengen haben

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' is eine konservative Erweiterung von S genau dann wenn

$$\mathsf{Th}(S) = \mathsf{Th}(S')_{|\Sigma}$$

wobei die $|\Sigma$ die Einschränkung auf Formeln aus $\mathcal{F}orm_{\Sigma}$ ist

Lemma 4

Jede konservative Erweiterung einer konsistenten Theorie ist konsistent.

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 $\mathrm{Abs}_t : r \to t, \mathrm{Rep}_t : t \to r$ axiomatisch Beschrieben

Typdefinitionen sind konservative Erweiterungen

Definition 5 (Typdefinitionen)

Sei $S = ((\mathcal{T}, \mathcal{F}, \mathcal{P}), \Phi)$ eine Spezifikation, $r \in \mathcal{T}$ und $P \in \mathcal{F}orm_{\Sigma}$ 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.

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

Teminierende, 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

Frei Erzeugte Typen

Definition 6 (Frei Erzeugte Typen)

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

- $S \vdash \forall x_t . \bigvee_{i=1\ldots k} \exists y^1_{\tau_1^i}, \ldots y^{n_k}_{\tau^i_{n_i}} . x = c_i(y^1, \ldots, y^{n_i})$
- $S \vdash \forall y^1_{\tau_1^i}, \ldots y^{n_i}_{\tau^i_{n_i}} . \forall z^1_{\tau_1^i}, \ldots z^{n_i}_{\tau^i_{n_i}} . c_i(y^1, \ldots, y^{n_i}) = c_i(z^1, \ldots, z^{n_i}) \longrightarrow ((y^1 = z^1 \wedge \cdots \wedge y^{n_i} = y^{n_i}))$ für alle c_i
- $S \vdash \forall y^1_{\tau_1^i}, \ldots y^{n_i}_{\tau^i_{n_i}} . \forall z^1_{\tau_1^j}, \ldots z^{n_j}_{\tau^j_{n_j}} . c_i(y^1, \ldots, y^{n_i}) = c_j(z^1, \ldots, z^{n_j})$ für alle $i \neq j$

Kriterien für eindeutig und total

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

$$\forall x_{1\tau_1} \ldots x_{n\tau_n} \ldots . P_0 \longrightarrow f(x_1, \ldots, x_n) = t_0$$
$$\vdots$$
$$\forall x_{1\tau_1} \ldots x_{n\tau_n} \ldots . P_n \longrightarrow f(x_1, \ldots, x_n) = t_n$$

  so daß beweisbar

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

Terminierungsbeweise – Idee

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

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

Erweiterung um totale, terminierende Funktionen ist konservativ

Definition 7 (Funktions- und Prädikatsdefinitionen)

Sei $S = ((\mathcal{T}, \mathcal{F}, \mathcal{P}), \Phi)$ eine Spezifikation, $f : \tau_1 \times \cdots \times \tau_n \to \tau_0 \notin \mathcal{F}$ ($\tau_i \in \mathcal{T}$) und $\Psi \in \mathcal{F}orm_{\Sigma \cup \{f\}}$. 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
- $\Phi' = \Phi \cup \Psi$
- $\mathcal{F}' = \mathcal{F} \cup \{f : \tau_1 \times \cdots \times \tau_n \to \tau_0\}$

Analog für Prädikatsdefinitionen.

Lemma 8

Funktionsdefinitionen bzw. Prädikatsdefinitionen sind konservativ

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).

Abgeleitete Induktionsschemata

- $\mathrm{fib}(x)$

$$\frac{P(0) \quad P(s(0)) \quad \forall x.P(x) \wedge P(\mathrm{s}(x)) \longrightarrow P(\mathrm{s}(\mathrm{s}(x)))}{\forall x.P(x)}$$

- $\mathrm{half}(x)$

$$\frac{P(0) \quad P(s(0)) \quad \forall x.P(x) \longrightarrow P(\mathrm{s}(\mathrm{s}(x)))}{\forall x.P(x)}$$

- $\mathrm{gcd}(x)$

$$\frac{\begin{array}{l} P(0,y) \\ x > 0 \longrightarrow P(x,0) \\ \forall x,y.x > y \wedge P(x-y,y) \longrightarrow P(x,y) \\ \forall x,y.\neg(x > y) \wedge P(x,y-x) \longrightarrow P(x,y) \end{array}}{\forall x.\forall y.P(x,y)}$$

Abgeleitetete Induktionsschemata besser zum Beweisen

- Abgeleitete Induktionsschemata erzeugen Fälle, in denen die Rekursionsgleichungen der Funktion/Prädikate direkt anwendbar sind
- Abgeleitete Induktionsschemata hilfreich wenn Induktion über Variablen gemacht wird, die als Argument der entsprechenden Funktion vorkommen.

$$\forall x.\varphi(\mathsf{half}(x))$$

- Fälle:
  1. $\varphi(\mathsf{half}(0)) \rightsquigarrow \varphi(0)$
  2. $\varphi(\mathsf{half}(\mathsf{s}(0))) \rightsquigarrow \varphi(0)$
  3. $\varphi(\mathsf{half}(x)) \longrightarrow \varphi(\mathsf{half}(\mathsf{s}(\mathsf{s}(x)))) \rightsquigarrow \varphi(\mathsf{half}(x)) \longrightarrow \varphi(\mathsf{s}(\mathsf{half}(x)))$