

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:

$$\forall x. 0 \leq x \quad (L1)$$

$$\forall x. \forall y. x \leq y \rightarrow s(x) \leq s(y) \quad (L2)$$

$$\text{half}(0) = 0 \quad (H1)$$

$$\text{half}(s(0)) = 0 \quad (H2)$$

$$\forall x. \text{half}(s(s(x))) = s(\text{half}(x)) \quad (H3)$$

- ▶ Beweise

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

Wohlfundierte Induktion

- ▶ Wohlfundiertes Induktionsschema

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

- ▶  $<$  wohlfundierte Relation:

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

Beweis mit wohlfundierter Induktion

- ▶  $<$ -Relation

$$\forall x. 0 < s(x) \quad \forall x, y. x < y \rightarrow s(x) < s(y)$$

- ▶ Beweise  $<$  ist wohlfundiert

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

$$\frac{\left[ \begin{array}{c} \forall x. x < c \\ \text{half}(x) \leq x \\ c = 0 \end{array} \right] \quad \left[ \begin{array}{c} \forall x. x < c \\ \text{half}(x) \leq x \\ c = s(0) \end{array} \right] \quad \left[ \begin{array}{c} \forall x. x < c \\ \text{half}(x) \leq x \\ \exists u. c = s(s(u)) \end{array} \right]}{\begin{array}{c} c = 0 \vee \\ c = s(0) \vee \\ \exists u. c = s(s(u)) \end{array} \quad \left[ \begin{array}{c} \vdots \\ \text{half}(c) \leq c \\ \vdots \end{array} \right] \quad \left[ \begin{array}{c} \vdots \\ \text{half}(c) \leq c \\ \vdots \end{array} \right]}{\forall x. \text{half}(x) \leq x}$$

Mehr Information

- ▶ Besser zum beweisen wäre wenn man gleich hätte

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

- ▶ Vergleiche:
  - $\text{half}(0) = 0 \quad (H1)$
  - $\text{half}(s(0)) = 0 \quad (H2)$
  - $\forall x. \text{half}(s(s(x))) = s(\text{half}(x)) \quad (H3)$

- ▶ Generiere Induktionsschema aus rekursiven Funktionsdefinitionen

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

Weitere Beispiele

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

- ▶ Sortieren

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

$$\forall s, t. m = \min(\text{cons}(n, l)) \rightarrow \text{sort}(\text{cons}(n, l)) = \text{cons}(m, \text{sort}(\text{cons}(n, l) - m))$$

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

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

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

- ▶ Induktionsschema

$$\frac{\forall m, n. m = \min(\text{cons}(n, l)) \wedge P(\text{cons}(n, l) - m) \quad P(\text{Nil})}{\forall l. P(l)}$$

## Weitere Beispiele

► Fibonacci:

$$\begin{aligned} \text{fib}(0) &= 0 \\ \text{fib}(s(0)) &= s(0) \\ \forall n. \text{fib}(s(s(n))) &= \text{fib}(s(n)) + \text{fib}(n) \end{aligned}$$

$$\frac{\begin{array}{c} [ P(s(c)), P(c) ] \\ \vdots \\ P(0) \quad P(s(0)) \quad P(s(s(c))) \\ \hline \forall x. P(x) \end{array}}{\forall x. P(x)}$$

9 [39]

## Weitere Beispiele

► GGT:

$$\begin{aligned} \forall y. \text{ggt}(0, y) &= y \\ \forall x. \text{ggt}(s(x), 0) &= s(x) \\ \forall x, y. x \leq y &\rightarrow \text{ggt}(x, y) = \text{ggt}(x, y - x) \\ \forall x, y. \neg(x \leq y) &\rightarrow \text{ggt}(x, y) = \text{ggt}(x - y, y) \end{aligned}$$

$$\frac{\begin{array}{c} \forall y. P(0, y) \quad \forall x. P(s(x), 0) \\ \vdots \\ P(x, y) \quad P(x, y) \\ \hline \forall x, y. P(x, y) \end{array}}{\forall x, y. P(x, y)}$$

10 [39]

## Zulässige Induktionsschema

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

$$\begin{aligned} P_0 &\rightarrow f(x_1, \dots, x_n) = t_0 \\ &\vdots \\ P_n &\rightarrow f(x_1, \dots, x_n) = t_n \end{aligned}$$

$$P_i \wedge P_j \leftrightarrow \perp, \forall i \neq j$$

- **terminierend**
- Rekursive Definition nach wohlfundierter Relation garantiert Terminierung  
Für jeden **atomaren, rekursiven** Aufruf  $f(t_1, \dots, t_n)$  erzeuge Terminierungshypothese

$$P_i \rightarrow (x_1, \dots, x_n) > (t_1, \dots, t_n)$$

11 [39]

## Grenzen

$$\begin{aligned} \forall x. x < 101 &\rightarrow f(x) = f(f(x + 11)) \\ \forall x. \neg(x < 101) &\rightarrow f(x) = x - 10 \end{aligned}$$

- $f$  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.

12 [39]

$$\begin{aligned} f(99) &= f(f(110)) & f(87) &= f(f(98)) \\ &= f(100) & &= f(f(f(109))) \\ &= f(f(111)) & &= f(f(99)) \\ &= f(101) & &= f(f(f(110))) \\ &= 91 & &= f(f(100)) \\ & & &= f(f(f(111))) \\ & & &= f(f(101)) \\ & & &= f(91) \\ & & &= f(f(102)) \\ & & &= f(92) \\ & & &= f(f(103)) \\ & & &= f(93) \\ & & &\dots \text{ Pattern continues} \\ & & &= f(99) \\ & & &\text{(same as on the left)} \\ & & &= 91 \end{aligned}$$

13 [39]

## 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)

14 [39]

## 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

15 [39]

## Getypte Prädikatenlogik – Signatur

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

16 [39]

## Getypte Prädikatenlogik – Terme & Formeln

	Ungetypt	Getypt
<b>Terme</b> $Term_{\Sigma}$		$Term_{\Sigma}^{\tau_1} \cup \dots \cup Term_{\Sigma}^{\tau_n}, \tau \in \mathcal{T}$
- 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 $f(t_1, \dots, t_n) \in Term_{\Sigma}^{\tau_0}$
<b>Formeln</b> $Form_{\Sigma}$		
- 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	$\neg\psi, \varphi \wedge \psi, \varphi \vee \psi, \varphi$	$\rightarrow\psi, \varphi \leftrightarrow \psi, \dots$
- Quantoren	$\forall x.\phi \in Form_{\Sigma}, x \in X$ $\exists x.\phi \in Form_{\Sigma}, x \in X$	$\forall x_{\tau}.\phi \in Form_{\Sigma}$ $\exists x_{\tau}.\phi \in Form_{\Sigma}$

17 [39]

## 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

18 [39]

## Natürliches Schließen — Die Regeln

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

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

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

19 [39]

## Die fehlenden Schlussregeln

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

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

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

20 [39]

## Natürliches Schließen mit Quantoren

$$\frac{\phi}{\forall x.\phi} \forall I \text{ (*)} \qquad \frac{\forall x.\phi}{\phi[x^t]} \forall E \text{ (†)}$$

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

21 [39]

## Der Existenzquantor

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

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

- ▶ (\*) **Eigenvariablenbedingung:**  
x nicht frei in  $\psi$ , oder einer offeneren Vorbedingung außer  $\phi$
- ▶ (†) Ggf. Umbenennung durch Substitution

22 [39]

## 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:

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

23 [39]

## Getypte Prädikatenlogik – ND Regeln

$$\frac{\phi}{\forall x_{\tau}.\phi} \forall I \text{ (*)} \qquad \frac{\forall x_{\tau}.\phi}{\phi[x^t]} \forall E \text{ (†)}$$

- ▶ (\*) **Eigenvariablenbedingung:**  
 $x_{\tau}$  nicht frei in offenen Vorbedingungen von  $\phi$  ( $x_{\tau}$  beliebig)
- ▶ (†)  $t \in Term_{\Sigma}^{\tau}$ ; Ggf. Umbenennung durch Substitution

24 [39]

## Der Existenzquantor

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

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

- ▶ (\*) **Eigenvariablenbedingung:**  
 $x_r$  nicht frei in  $\psi$ , oder einer offenen Vorbedingung außer  $\phi$
- ▶ (\dagger)  $t \in \text{Term}_\Sigma$ ; Ggf. Umbenennung durch Substitution

25 [39]

## Regeln für die Gleichheit

- ▶ Reflexivität, Symmetrie, Transitivität:

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

- ▶ 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:

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

26 [39]

## Basic Definitions

### Definition 1 (Loose Spezifikationen)

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

### Definition 2 (Konsistenz)

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

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

27 [39]

## 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

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

wobei die  $|_\Sigma$  die Einschränkung auf Formeln aus  $\text{Form}_\Sigma$  ist

### Lemma 4

Jede konservative Erweiterung einer konsistenten Theorie ist konsistent.

28 [39]

## 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

29 [39]

## Typdefinitionen sind konservative Erweiterungen

### Definition 5 (Typdefinitionen)

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

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

30 [39]

## 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

31 [39]

## 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.

- ▶  $S \vdash \forall x_t. \forall_{i=1 \dots k} \exists y_{\tau_1^i}^1, \dots, y_{\tau_{n_i}^i}^k. x = c_i(y^1, \dots, y^{n_i})$
- ▶  $S \vdash \forall y_{\tau_1^1}^1, \dots, y_{\tau_{n_1}^1}^{n_1}. \forall z_{\tau_1^1}^1, \dots, z_{\tau_{n_1}^1}^{n_1}. c_1(y^1, \dots, y^{n_1}) = c_1(z^1, \dots, z^{n_1}) \rightarrow ((y^1 = z^1 \wedge \dots \wedge y^{n_1} = z^{n_1}) \text{ für alle } c_i)$
- ▶  $S \vdash \forall y_{\tau_1^1}^1, \dots, y_{\tau_{n_1}^1}^{n_1}. \forall z_{\tau_1^1}^1, \dots, z_{\tau_{n_1}^1}^{n_1}. c_i(y^1, \dots, y^{n_i}) = c_j(z^1, \dots, z^{n_j}) \text{ für alle } i \neq j$

32 [39]

## Kriterien für eindeutig und total

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

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

so daß beweisbar

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

33 [39]

## 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 Argument in rekursiven Funktionsaufrufen kleiner sind bezüglich der Ordnung auf den natürlichen Zahlen unter der entsprechenden Bedingung  $P_i$ .

34 [39]

## Terminierung

- ▶ Beispiele:
  - ▶  $\text{half}(x)$  eine Hypothese pro Rekursionsgleichung
  - ▶  $\text{fib}(x)$ : mehrere Hypothesen pro Rekursionsgleichung
  - ▶  $\text{gcd}(x, y)$ : 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

35 [39]

## Erweiterung um totale, terminierende Funktionen ist konservativ

### Definition 7 (Funktions- und Prädikatsdefinitionen)

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

- ▶  $\Psi$  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*

36 [39]

## 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  $\text{Rep}_t$
  - ▶ Wenn freie Erzeugtheit des neuen Typs beweisbar, dann folgt Induktionsschema direkt auf dem neuen Typ
- ▶ Damit hat man garantiert immer konsistente Spezifikationen (= Modellierung).

37 [39]

## Abgeleitete Induktionsschemata

- ▶  $\text{fib}(x)$

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

- ▶  $\text{half}(x)$

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

- ▶  $\text{gcd}(x)$

$$\frac{P(0, y) \quad x > 0 \longrightarrow P(x, 0) \quad \forall x, y. x > y \wedge P(x - y, y) \longrightarrow P(x, y) \quad \forall x, y. \neg(x > y) \wedge P(x, y - x) \longrightarrow P(x, y)}{\forall x. \forall y. P(x, y)}$$

38 [39]

## Abgeleitete 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.

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

- ▶ Fälle:

1.  $\varphi(\text{half}(0)) \rightsquigarrow \varphi(0)$
2.  $\varphi(\text{half}(s(0))) \rightsquigarrow \varphi(0)$
3.  $\varphi(\text{half}(x)) \longrightarrow \varphi(\text{half}(s(s(x)))) \rightsquigarrow \varphi(\text{half}(x)) \longrightarrow \varphi(s(\text{half}(x)))$

39 [39]