Proseminar Perlen der Informatik
Professor Tobias Nipkow, PhD

HOL

Stefan Richter
22.2.2000

Betreut von Markus Wenzel

Inhab : Jedes Element von $\boldsymbol{U}$ ist eine nichtleere Menge.

Sub : Wenn X $\in \boldsymbol{U}$ und $\emptyset \neq Y \subseteq X$, dann Y $\in \boldsymbol{U}$.

Prod : Aus X $\in \boldsymbol{U}$ und Y $\in \boldsymbol{U}$ folgt $X \times Y \in \boldsymbol{U}$. [a]

Pow : Wenn $X \in \boldsymbol{U}$, dann ist auch die Potenzmenge $\mathcal{P}(X) = \{Y : Y \subseteq X\}$ Element von $\boldsymbol{U}$.

Infty : $\boldsymbol{U}$ enthält eine ausgewiesene unendliche Menge **I**.

Choice : Es existiert ein ausgewiesenes Element ch $\in \Pi_{X \in \boldsymbol{U}} X$.

---

[a] Dabei wird das kartesische Produkt $X \times Y$ dargestellt durch $\{(x, y) = \{\{x\}, \{x, y\}\} \ : \ x \in X \wedge y \in Y\}$.

Aus diesen Regeln lassen sich weitere Eigenschaften ableiten, die für die HOL-Semantik wichtig sind:

Da sich Funktionen als Teilmengen von kartesischen Produkten darstellen lassen, folgt aus **Sub**, **Prod** und **Pow** :

Fun : Wenn $(X \in \boldsymbol{U} \wedge Y \in \boldsymbol{U})$, dann $(X \to Y \in \boldsymbol{U})$.

Durch Anwendung von **Sub** auf eine beliebige Menge in $\boldsymbol{U}$ (**Infty** garantiert die Existenz einer solchen Menge) erhält man ein- bzw. zweielementige Mengen, von denen jeweils eine bestimmte ausgewählt werde:

Unit : $\boldsymbol{U}$ enthält die ausgezeichnete einelementige Menge $1 = \{0\}$.

Bool : $\boldsymbol{U}$ enthält die ausgewählte zweielementige Menge $2 = \{0, 1\}$.

Typen in HOL:

$$\sigma ::= \alpha \mid c \mid (\sigma_1, \ldots, \sigma_n)op \mid \sigma_1 \to \sigma_2$$

Typvariablen stehen für beliebige Mengen im Universum.

Typkonstanten haben die Form $(\sigma_1, \ldots, \sigma_n)op$ und stehen für Funktionen auf Typen. Spezielle:

atomare Typen (Basistypen)
Funktionstypen

Terme in HOL:

$$t_\sigma ::= x_\sigma \;\mid\; c_\sigma \;\mid\; (t_{\sigma'\to\sigma}\; t'_{\sigma'})_\sigma$$

$$t_{\alpha\to\beta} ::= (\lambda x_\alpha . t_\beta)_{\alpha\to\beta}$$

Variablen sind Paare x_σ in Indexnotation, wobei x ein Name und σ ein Typ sei.

Konstanten c_σ sind genau dann gültige Terme, wenn c für ein Element einer Menge steht, von deren Typ σ eine Instanz ist.

$\boldsymbol{\lambda}$-Abstraktionen $(\lambda x_{\sigma_1} . t_{\sigma_2})_{\sigma_1\to\sigma_2}$ bezeichnen totale Funktionen $v_{\sigma_1} \to t[v/x]_{\sigma_2}$, wenn x eine Variable vom Typ σ_1 und t ein gültiger Term vom Typ σ_2 ist.

Funktionsapplikationen $(t_{\sigma'\to\sigma}\; t'_{\sigma'})_\sigma$ bezeichnen das Ergebnis der Anwendung der Funktion $t_{\sigma'\to\sigma}$ auf das Argument $t'_{\sigma'}$

Primitive Konstanten:

$\Rightarrow\colon 2 \to 2 \to 2$

$$(a \Rightarrow b) = \begin{cases} 0 & \text{falls } a = 1 \text{ und } b = 0 \\ 1 & \text{sonst} \end{cases}$$

$=_X\colon X \to X \to 2$

$$(x =_X y) = \begin{cases} 1 & \text{falls } x = y \\ 0 & \text{sonst} \end{cases}$$

$\varepsilon : \ (X \to 2) \to X$

$$ch_X(f) = \begin{cases} ch(f^{-1}\{1\}) & \text{falls } f^{-1}\{1\} \neq \emptyset \\ ch(X) & \text{sonst} \end{cases}$$

Assumption introduction

$$\frac{}{t \vdash t}$$

Reflexitivity

$$\frac{}{\vdash t = t}$$

Beta – conversion

$$\frac{}{\vdash (\lambda x.t_1)t_2 = t_1[t_2/x]}$$

Substitution

$$\frac{\Gamma_1 \vdash t_1 = t_1' \quad \cdots \quad \Gamma_n \vdash t_n = t_n' \qquad \Gamma \vdash t[t_1, \ldots, t_n]}{\Gamma_1 \cup \cdots \cup \Gamma_n \cup \Gamma \vdash t[t_1', \ldots, t_n']}$$

Abstraction

$$\frac{\Gamma \vdash t_1 = t_2}{\Gamma \vdash (\lambda x.t_1) = (\lambda x.t_2)}$$

- Wenn x nicht frei in Γ

Type instantiation

$$\frac{\Gamma \vdash t}{\Gamma \vdash t[\sigma_1, \ldots, \sigma_n / \alpha_1, \ldots, \alpha_n]}$$

- Wo $t[\sigma_1, \ldots, \sigma_n / \alpha_1, \ldots, \alpha_n]$ das Ergebnis der parallelen Ersetzung der Typvariablen $\alpha_1, \ldots, \alpha_n$ durch Typen $\sigma_1, \ldots, \sigma_n$ in t ist, mit der Einschränkung, daß keine der Typvariablen $\alpha_1, \ldots, \alpha_n$ in Γ vorkommt.

Discharging an assumption

$$\frac{\Gamma \vdash t_2}{\Gamma \backslash \{t_1\} \vdash t_1 \Rightarrow t_2}$$

Modus Ponens

$$\frac{\Gamma_1 \vdash t_1 \Rightarrow t_2 \qquad \Gamma_2 \vdash t_1}{\Gamma_1 \cup \Gamma_2 \vdash t_2}$$

Die Theorie LOG:

$\vdash \mathsf{T}_{bool} = ((\lambda x_{bool}.x) = (\lambda x_{bool}.x))$

$\vdash \forall_{(\alpha\to bool)\to bool} = \lambda P_{\alpha\to bool}.P = (\lambda x.\mathsf{T})$

$\vdash \exists_{(\alpha\to bool)\to bool} = \lambda P_{\alpha\to bool}.P(\varepsilon P)$

$\vdash \mathsf{F}_{bool} = \forall b_{bool}.b$

$\vdash \neg_{bool\to bool} = \lambda b.b \Rightarrow \mathsf{F}$

$\vdash \wedge_{bool\to bool\to bool} = \lambda b_1 b_2.\forall b.(b_1 \Rightarrow (b_2 \Rightarrow b)) \Rightarrow b$

$\vdash \vee_{bool\to bool\to bool} = \lambda b_1 b_2.\forall b.(b_1 \Rightarrow b) \Rightarrow ((b_2 \Rightarrow b) \Rightarrow b)$

$\vdash \mathsf{One\_One}_{(\alpha\to\beta)\to bool} = \lambda f_{\alpha\to\beta}.\forall x_1 x_2.(fx_1 = fx_2) \Rightarrow (x_1 = x_2)$

$\vdash \mathsf{Onto}_{(\alpha\to\beta)\to bool} = \lambda f_{\alpha\to\beta}.\forall y.\exists x.y = fx$

$\vdash \mathsf{Type\_Definition}_{(\alpha\to bool)\to(\beta\to\alpha)\to bool} =$
$\lambda P_{\alpha\to bool}\ rep_{\beta\to\alpha}.\mathsf{One\_One}\ rep \wedge (\forall x.Px = (\exists y.x = rep\ y))$

$\vdash \mathsf{Inv} = \lambda f_{\alpha\to\beta}.\lambda y.\varepsilon x.y = fx$

Die Theorie INIT fügt nur die folgenden fünf Axiome der Theorie LOG hinzu:

BOOL_CASES_AX	$\vdash \forall b.(b = \mathsf{T}) \vee (b = \mathsf{F})$
IMP_ANTYSIM_AX	$\vdash \forall b_1 b_2.(b_1 \Rightarrow b_2) \Rightarrow (b_2 \Rightarrow b_1) \Rightarrow (b_1 = b_2)$
ETA_AX	$\vdash \forall f_{\alpha\to\beta}.(\lambda x.fx) = f$
SELECT_AX	$\vdash \forall P_{\alpha\to bool} x.Px \Rightarrow P(\varepsilon P)$
INFINITY_AX	$\vdash \exists f_{ind\to ind}.\mathsf{One\_One}\ f \wedge \neg(\mathsf{Onto} f)$

Eine Konstantendefinition ist eine Formel der Form $c_\sigma = t_\sigma$, so daß:

1. c ein neuer Name ist
2. t ein geschlossener gültiger Term ist
3. alle Typvariablen in t auch in σ vorkommen

Eine Typdefinition beinhaltet:

1. Spezifizierung eines existierenden *Repräsentationstyps*
2. Angabe einer Teilmenge davon durch ein *Teilmengenprädikat*
3. Beweis, daß die Teilmenge nicht leer ist
4. Forderung von Isomophie mittels einer *Repräsentationsfunktion*

Typ-Definition PROD:

$\vdash$ Mk_Pair $= \lambda ab.\ \lambda xy.\ (x = a) \wedge (y = b)$

$\vdash$ Is_Pair $= \lambda f.\ \exists ab.\ f =$ Mk_Pair $a\ b.$

$\vdash$ Type_Definition Rep_Pair

$\vdash$ Abs_Pair $=$ Inv Rep_Pair

$\vdash$ Pair $= \lambda\ x_\alpha\ y_\beta.$ Abs_Pair (Mk_Pair $x\ y$)

$\vdash$ Fst $= \lambda p_{(\alpha,\beta)prod}.\ \varepsilon x.\ \exists y.\ p = (x, y)$

$\vdash$ Snd $= \lambda p_{(\alpha,\beta)prod}.\ \varepsilon y.\ \exists x.\ p = (x, y).$

$\vdash \forall\ x\ y.$ Fst$(x, y) = x$

$\vdash \forall\ x\ y.$ Snd$(x, y) = y$

$\vdash \forall\ p_{(\alpha,\beta)prod}.\ p =$ (Fst p, Snd p)