

Proofs as Programs Summer School
Eugene Oregon June - July 2002

Type Systems

Herman Geuvers

Nijmegen University, NL

Lecture 4: Higher Order Logic

The original motivation of Church to introduce simple type theory was:

to define higher order (predicate) logic

In $\lambda \rightarrow$ we add the following

- **prop** as a basic type

- \Leftarrow : $\text{prop} \rightarrow \text{prop}$

- $\forall \sigma$: $(\sigma \rightarrow \text{prop}) \rightarrow \text{prop}$ (for each type σ)

This defines the language of higher order logic.

- **Induction**

$$\begin{aligned} \forall N \rightarrow \text{prop} (\lambda P : N \rightarrow \text{prop} . (P_0)) &\Leftrightarrow \forall N (\lambda x : N . (P x)) \\ &\Leftrightarrow \forall N (\lambda x : N . (P x)) \end{aligned}$$

Notation:

$$\begin{aligned} \forall P : N \rightarrow \text{prop} (P_0) &\Leftrightarrow (\forall x : N . (P x)) \\ &\Leftrightarrow \forall x : N . (P x) \end{aligned}$$

- **Higher order predicates/functions**

transitive closure of a relation R

$$\begin{aligned} \lambda R : A \rightarrow A \rightarrow \text{prop} . \lambda x, y : A . \\ (\forall Q : A \rightarrow A \rightarrow \text{prop} . (\text{trans}(Q) \Rightarrow (R \subseteq Q) \Rightarrow Q x y)) \end{aligned}$$

of type

$$(A \rightarrow A \rightarrow \text{prop}) \rightarrow (A \rightarrow A \rightarrow \text{prop})$$

Derivation rules for Higher Order Logic (following Church)

- Natural deduction style.
- Rules are 'on top' of the simple type theory.
- Judgements are of the form

$\Delta \vdash \phi$

$\Delta = \psi_1, \dots, \psi_n$

Γ is a λ -context

$\Gamma \vdash \phi : \text{prop}, \dots, \Gamma \vdash \psi_1 : \text{prop}, \dots, \Gamma \vdash \psi_n : \text{prop}$
 Γ is usually left implicit: $\Delta \vdash \phi$

$$\phi \vDash \exists x \psi \quad \frac{\phi \vdash \nabla}{\phi \vdash \nabla} \quad (\text{conversion})$$

$$\sigma : t \text{ f!} \quad \frac{[x/t]\phi \vdash \nabla}{\phi \cdot \sigma : x \Delta \vdash \nabla} \quad (\nabla\text{-elimination})$$

$$(\nabla)\forall \text{ f! } \sigma : x \text{ f!} \quad \frac{\phi \cdot \sigma : x \Delta \vdash \nabla}{\phi \vdash \nabla} \quad (\nabla\text{-introduction})$$

$$\frac{\phi \vdash \nabla}{\phi \vdash \nabla \quad \phi \Leftarrow \phi \vdash \nabla} \quad (\Leftarrow\text{-elimination})$$

$$\frac{\phi \Leftarrow \phi \vdash \nabla}{\phi \vdash \phi \cup \nabla} \quad (\Leftarrow\text{-introduction})$$

$$\nabla \in \phi \text{ f!} \quad \underline{\nabla \vdash \phi} \quad (\text{axiom})$$

Church has additional things that we will not consider now:

- **Negation** connective with rules

- Classical logic

$$\frac{\Delta \vdash \neg \phi}{\Delta \vdash \neg \neg \phi}$$

- Define other connectives in terms of \Rightarrow , \forall , \neg (classically).

- **Choice** operator $\iota_{\sigma} : (\sigma \rightarrow \text{prop}) \rightarrow \sigma$

- Rule for ι :

$$\frac{\Delta \vdash \exists x:\sigma. P \ x}{\Delta \vdash P(\iota_{\sigma} P)}$$

This (Church' original higher order logic) is basically the logic of the theorem prover HOL (Gordon, Melham, Harrison) and of Isabelle-HOL (Paulson, Nipkow).

We will here restrict to the basic **constructive** core (\forall , \Rightarrow) of

HOL.

Conversion rule:

$$\frac{\Delta \vdash AP:N \rightarrow \text{prop} \cdot (\dots Pc \dots)}{\Delta \vdash (\dots (\lambda y:N. y < 0) c \dots)} \text{A-elim}$$

$$\frac{\Delta \vdash (\dots (\lambda y:N. y < 0) c \dots)}{\Delta \vdash (\dots c > 0 \dots)} \text{conv}$$

Definability of other connectives (constructively):

$$\top := \forall a:\text{prop}. a$$

$$\phi \vee \psi := \forall a:\text{prop}. (\phi \Rightarrow a) \Rightarrow (\psi \Rightarrow a)$$

$$\phi \wedge \psi := \forall a:\text{prop}. (\phi \Rightarrow a) \wedge (\psi \Rightarrow a)$$

$$\exists x:\sigma. \phi := \forall a:\text{prop}. (\forall x:\sigma. \phi) \Rightarrow a$$

Idea:

The definition of a connective is an encoding of the **elimination** rule.

Existential quantifier

$$\exists x:\sigma.\phi := \forall a:\text{prop}.\phi \Rightarrow a \Rightarrow \sigma$$

Derivations for the elimination and introduction rules.

$$\frac{C}{\exists x:\sigma.\phi} \text{ } \frac{C}{x \notin \text{FV}(C, \text{ass.})}$$

Existential quantifier

$$\exists x:\sigma.\varphi := \forall a:\text{prop}.\left(\forall x:\sigma.\varphi \Rightarrow a\right) \Rightarrow a$$

Derivations for the elimination and introduction rules.

$$\frac{\frac{\frac{\frac{\mathcal{C}}{C \Leftarrow \phi.\sigma.x\Delta}}{C} \quad \vdots \quad [\phi]}{C \Leftarrow \phi.\sigma.x\Delta} \quad \frac{\mathcal{C}}{C \Leftarrow (C \Leftarrow \phi.\sigma.x\Delta)}}{C \Leftarrow \phi.\sigma.x\Delta} \quad (\text{ass.}) \quad \forall x \notin \text{FV}(C, \text{ass.})}{\frac{\mathcal{C}}{C \Leftarrow \phi.\sigma.x\Delta} \quad \vdots \quad [\phi]}{C \Leftarrow \phi.\sigma.x\Delta} \quad \exists x:\sigma.\varphi$$

$$\frac{\frac{\frac{\phi.\sigma.x\exists}{x \Leftarrow (x \Leftarrow \phi.\sigma.x\Delta)}}{x} \quad \frac{x \Leftarrow [x/t]\phi \quad [x/t]\phi}{x \Leftarrow \phi.\sigma.x\Delta}}{x \Leftarrow (x \Leftarrow \phi.\sigma.x\Delta)}$$

$$\frac{\phi.\sigma.x\exists}{[x/t]\phi}$$

Equality is definable in higher order logic:

t and q terms are equal if they share the same properties
(Leibniz equality)

Definition in HOL (for $t, q : A$):

$$t =_A q := \forall P:A \rightarrow \text{prop}. (Pt \Leftrightarrow Pq)$$

• This equality is reflexive and transitive (easy)

Equality is definable in higher order logic:
 t and q terms are equal if they share the same properties
 (Leibniz equality)

Definition in HOL (for $t, q : A$):

$$t =_A q := \forall P: A \rightarrow \text{prop}. (Pt \Rightarrow Pq)$$

- This equality is **reflexive** and **transitive** (easy)

- It is also **symmetric**(i) Trick: take $\lambda y:A. y =_A t$ for P .

$$\frac{\frac{\Delta \vdash t =_A q}{\Delta \vdash A \rightarrow \text{prop}. (Pt \Rightarrow Pq)}}{\Delta \vdash t =_A q} \quad \frac{\Delta \vdash t =_A t \quad \Delta \vdash t =_A t}{\dots}$$

$$\phi \vDash \phi \quad \frac{\phi \vdash \nabla}{\phi \vdash \nabla} \quad \text{(conversion)}$$

$$\sigma : t \quad \frac{[x/t]\phi \vdash \nabla}{\phi \cdot \sigma : x \Delta \vdash \nabla} \quad \text{(\nabla-elimination)}$$

$$(\nabla) \wedge \exists \sigma : x \quad \frac{\phi \cdot \sigma : x \Delta \vdash \nabla}{\phi \vdash \nabla} \quad \text{(\nabla-introduction)}$$

$$\frac{\phi \vdash \nabla}{\phi \vdash \nabla \quad \phi \Leftarrow \phi \vdash \nabla} \quad \text{(\Leftarrow-elimination)}$$

$$\frac{\phi \Leftarrow \phi \vdash \nabla}{\phi \vdash \phi \cup \nabla} \quad \text{(\Leftarrow-introduction)}$$

$$\nabla \in \phi \quad \underline{\phi \vdash \nabla} \quad \text{(axiom)}$$

Why not introduce a λ -term notation for the derivations?

This gives a type theory λ HOL

- No 'lifting' of prop to the type level
- Let prop be a new 'universe' of propositional types.
- Direct encoding (deep embedding) of HOL into the type theory λ HOL

$\phi \text{ } \mathcal{E} = \phi \text{ } \text{!}$

$$\frac{\phi : \mathcal{W} \vdash \nabla}{\phi : \mathcal{M} \vdash \nabla}$$

(conversion)

 $\phi : \text{!} \text{ } \sigma$

$$\frac{[x/t]\phi : \text{!} \mathcal{W} \vdash \nabla}{\phi : \sigma : \Delta : \mathcal{M} \vdash \nabla}$$

(A-elimination)

 $(\nabla) \text{ } \text{!} \text{ } \sigma : x \text{ } \text{!}$

$$\frac{\phi : \sigma : x \Delta : \mathcal{M} \cdot \sigma : x \mathcal{Y} \vdash \nabla}{\phi : \mathcal{M} \vdash \nabla}$$

(A-introduction)

$$\frac{\phi \mathcal{N} \mathcal{W} \vdash \nabla}{\phi : \mathcal{N} \vdash \nabla \quad \phi \Leftarrow \phi : \mathcal{W} \vdash \nabla}$$

(⇒-elimination)

$$\frac{\phi \Leftarrow \phi : \mathcal{W} \cdot \phi : x \mathcal{Y} \vdash \nabla}{\phi : \mathcal{M} \vdash \phi : x \nabla}$$

(⇒-introduction)

 $\nabla \in \phi : x \text{ } \text{!}$

$$\underline{\phi : x \vdash \nabla}$$

(axiom)

Some of the typing rules are **parameterized**

$\{\text{Prop, Type, Type}'\}$ is the set of **sorts**, \mathcal{S} .

$T ::= \text{Prop} \mid \text{Type} \mid \text{Type}' \mid \text{Var} \mid (\Pi \text{Var}:T.T) \mid (\lambda \text{Var}:T.T) \mid \text{TT}$

Pseudoterms:

We put these levels together into one type theory **λHOL** .

NB Many rules, many **similar** rules.

- The type theory for the **proof-terms** of HOL
- The (simple) type theory describing the **language** of HOL

Now we have **two** 'levels' of type theories

(axiom) $\vdash \text{Prop} : \text{Type}$ $\vdash \text{Type}' : \text{Type}'$

(var)
$$\frac{\Gamma \vdash A : s \quad \Gamma, x:A \vdash x : A}{\Gamma \vdash A : s} \text{ (weak)}$$
$$\frac{\Gamma, x:A \vdash M : C}{\Gamma \vdash A : s \quad \Gamma \vdash M : C}$$

(II)
$$\frac{\Gamma \vdash A : s_1 \quad \Gamma, x:A \vdash B : s_2 \quad \Gamma \vdash \Pi x:A. B : s_2}{\Gamma \vdash A : s_1 \quad \Gamma, x:A \vdash B : s_2} \text{ if } (s_1, s_2) \in \{ (\text{Type}, \text{Type}), (\text{Prop}, \text{Prop}), (\text{Type}, \text{Prop}) \}$$

(λ)
$$\frac{\Gamma, x:A \vdash M : B \quad \Gamma \vdash \Pi x:A. B : s}{\Gamma \vdash \lambda x:A. M : \Pi x:A. B}$$

(app)
$$\frac{\Gamma \vdash M : \Pi x:A. B \quad \Gamma \vdash N : A}{\Gamma \vdash M N : B[N/x]}$$

(conv)
$$\frac{\Gamma \vdash M : A \quad \Gamma \vdash B : s}{\Gamma \vdash M : B} \text{ if } A =_{\beta} B$$

(II)
$$\frac{\Gamma \vdash A : s_1 \quad \Gamma, x:A \vdash B : s_2 \quad \text{if } (s_1, s_2) \in \{ (\text{Type}, \text{Type}), (\text{Prop}, \text{Prop}), (\text{Type}, \text{Prop}) \}}{\Gamma \vdash A : s_1 \quad \Gamma, x:A \vdash B : s_2} \text{IIx:A.B} : s_2$$

• The combination **(Type, Type)** forms the **function types** $A \rightarrow B$ for $A, B : \text{Type}$.

This comprises the **unary predicate types** and **binary relations types**: $A \rightarrow \text{Prop}$ and $A \rightarrow A \rightarrow \text{Prop}$.

Also: **higher order predicate types** like $(A \rightarrow A \rightarrow \text{Prop}) \rightarrow \text{Prop}$. NB A Π -type formed by **(Type, Type)** is always an \rightarrow -type.

• **(Prop, Prop)** forms the **propositional types** $\phi \rightarrow \psi$ for $\phi, \psi : \text{Prop}$; **implicational formulas**.

NB A Π -type formed by **(Type, Type)** is always an \rightarrow -type.

• **(Type, Prop)** forms the **dependent propositional type** $\text{II}x:A.\phi$ for $A : \text{Type}$, $\phi : \text{Prop}$; **universally quantified formulas**.

Example: Deriving **irreflexivity** from **anti-symmetry**

Rel := $\lambda X:\text{Type}. X \rightarrow X \rightarrow \text{Prop}$

AntiSym := $\lambda X:\text{Type}.\lambda R:(\text{Rel } X).\forall x, y:X.(Rxy) \Rightarrow (Ryx) \Rightarrow \perp$

Irrefl := $\lambda X:\text{Type}.\lambda R:(\text{Rel } X).\forall x:X.(Rxx) \Rightarrow \perp$

Derivation in HOL:

$$\frac{\frac{\frac{\frac{\frac{\perp}{\forall y^A Rxy \Rightarrow Ryx \Rightarrow \perp}}{Rxx \Rightarrow Rxx \Rightarrow \perp}}{[Rxx]}{Rxx \Rightarrow \perp}}{\perp}}{\forall x^A Rxx \Rightarrow \perp}}{[Rxx]}$$

Derivation in HOL, with terms:

$$\frac{}{z : \forall x^A y^A R x y \Rightarrow R y x \Rightarrow \perp}$$

$$\frac{}{z x : \forall y^A R x y \Rightarrow R y x \Rightarrow \perp}$$

$$\frac{}{z x x : R x x \Rightarrow \perp} [q : R x x]$$

$$\frac{}{z x x q : R x x \Rightarrow \perp}$$

$$[q : R x x]$$

$$\frac{}{z x x q q : \perp}$$

$$\frac{}{\lambda q : (R x x). z x x q q : R x x \Rightarrow \perp}$$

$$\frac{}{\lambda x : A. \lambda q : (R x x). z x x q q : \forall x^A R x x \Rightarrow \perp}$$

Typing judgement in λ HOL:

$$A : \text{Type}, R : A \rightarrow A \rightarrow \text{Prop}, z : \Pi x, y : A. (R x y \rightarrow R y x \rightarrow \perp) \vdash$$

$$\lambda x : A. \lambda q : (R x x). z x x q q : (\Pi x : A. R x x \rightarrow \perp)$$

Question: is the type theory λ HOL really isomorphic with HOL?

Yes: Disambiguation Lemma Given

$\Gamma \vdash M : T$ in λ HOL

there is a permutation of $\Gamma : \Gamma^D, \Gamma^L, \Gamma^P$ such that

1. $\Gamma^D, \Gamma^L, \Gamma^P \vdash M : A$

2. Γ^D consists only of declarations $A : \text{Type}$

3. Γ^L consists only of declarations $x : \sigma$ with $\Gamma^D \vdash \sigma : \text{Type}$

4. Γ^P consists only of declarations $z : \phi$ with $\Gamma^D, \Gamma^L \vdash \phi : \text{Prop}$

Properties of λ HOL.

- **Uniqueness of types**
If $\Gamma \vdash M : A$ and $\Gamma \vdash M : B$, then $A =_{\beta} B$.

- **Subject Reduction**
If $\Gamma \vdash M : \sigma$ and $M \xrightarrow{\beta} N$, then $\Gamma \vdash N : \sigma$.

- **Strong Normalization**

If $\Gamma \vdash M : \sigma$, then all β -reductions from M terminate.

Proof of SN is a **higher order** extension of the one for λ_2 (using the saturated sets).

Decidability Questions:

$\Gamma \vdash M : \sigma ?$ TCP
 $\Gamma \vdash M : ?$ TSP
 $\Gamma \vdash ? : \sigma$ TIP

For λHOL :

• TIP is **undecidable**

• TCP/TSP: simultaneously.

The type checking algorithm is close to the one for λP . (In λP we had a judgement of **correct** context; this form of judgement could also be introduced for λHOL)

$\text{Type}_{\text{prop}, y: B}(x) =$ if $\text{Type}_{\text{prop}}(B) \in \{\text{Prop}, \text{Type}, \text{Type}'\}$ and $x:A \in \Gamma$
then A else **'false'**,

$\text{Type}_{>}(\text{Prop}) = \text{Type}$

$\text{Type}_{>}(\text{Type}) = \text{Type}'$

$\text{Type}_{\text{prop}, y: B}(\text{Prop}) =$ if $\text{Type}_{\text{prop}}(B) \in \{\text{Prop}, \text{Type}, \text{Type}'\}$ then Type

$\text{Type}_{\text{prop}, y: B}(\text{Type}) =$ if $\text{Type}_{\text{prop}}(B) \in \{\text{Prop}, \text{Type}, \text{Type}'\}$ then Type'

$\text{Type}_{\text{F}}(MN) =$ if $\text{Type}_{\text{F}}(M) = C$ and $\text{Type}_{\text{F}}(N) = D$
 then if $C \twoheadrightarrow_{\beta} \Pi x:A.B$ and $A =_{\beta} D$
 then $B[N/x]$ else 'false',
 else 'false',

$\text{Type}_{\text{F}}(\lambda x:A.M) =$ if $\text{Type}_{\text{F},x:A}(M) = B$

then
 if $\text{Type}_{\text{F}}(\Pi x:A.B) \in \{\text{Prop}, \text{Type}, \text{Type}'\}$
 then $\Pi x:A.B$ else 'false'

else 'false',

$\text{Type}_{\text{F}}(\Pi x:A.B) =$ if $\text{Type}_{\text{F}}(A) = s_1$ and $\text{Type}_{\text{F},x:A}(B) = s_2$

and $(s_1, s_2) \in \{(\text{Type}, \text{Type}), (\text{Prop}, \text{Prop}), (\text{Type}, \text{Prop})\}$

then s

else 'false'