Subtyping in Dependently-Typed Higher-Order Logic

Colin M. Rothgang[1]
IMDEA software institute, Madrid, Spain

Florian Rabe
Computer Science, University Erlangen-Nürnberg, Germany

Abstract

The recently introduced dependent typed higher-order logic (DHOL) offers an interesting compromise between expressiveness and automation support. It sacrifices the decidability of its type-system in order to significantly extend its expressiveness over standard HOL. It retains proof automation support via a sound and complete translation to HOL.

We leverage this design to extend DHOL with refinement and quotient types. Both of these are type operators commonly requested by practitioners, but they are very difficult to retrofit into a logic designed for decidable typing. In DHOL, however, adding them is not only possible but simple and elegant. In particular, we realize both as special cases of subtyping, i.e., the associated canonical operations are identity maps that do not require costly changes in representation. We rigorously work out the syntax and semantics of the extended language, including the proof of soundness and completeness.

2012 ACM Subject Classification Theory of computation

Keywords and phrases higher-order logic, dependent types, refinement types, quotient types, subtyping, automated reasoning

Digital Object Identifier 10.4230/LIPIcs...

1 Introduction and Related Work

Motivation Recently we introduced dependently-typed higher-order logic (DHOL) [14]. It can be seen as an extension of HOL [4, 8] that uses dependent function types $\Pi x{:}A.\ B$ instead of simple function types $A \to B$. It is designed to stay as simple and as close to HOL as possible while meeting the frequent user demand of supporting dependent types. Contrary to typical formulations of dependent type theory such as Martin-Löf type theory [10] and implementations in proof assistants [12, 6, 7], DHOL does not employ a sophisticated treatment of equality that keeps typing decidable. Instead, it uses a straightforward formulation of equality at the cost of making typing undecidable.

Concretely, DHOL uses a type bool of propositions in the style of HOL, and equality $s =_A t{:}\mathsf{bool}$ of typed terms is a proposition, whose truth may depend on axioms in the theory or assumptions in the context. Equality $A \equiv B$ of types is a judgment with a straightforward congruence rule: if a dependent type constructor is applied to equal arguments, it produces equal types. Thus, equality of types and all typing judgments depend on term equality and are undecidable. To obtain practical tool support, DHOL reduces every typing judgment to

[1] junior researcher

a series of proof obligations, and [14] gives a sound and complete translation to HOL that allows using existing automated theorem provers (ATPs) for HOL to discharge these.

The subtle interaction between dependent types and decidability of typing is well-known, and logic designers have traditionally shied away from undecidable typing. Indeed, only a few major systems for dependent types have embraced it: PVS [13] and Mizar [3], although based on very different foundations, feature dependent functions and refinement types in a way similar to our work. Nuprl [5] uses a very expressive type theory that features refinement and quotient types similar to ours. With ATPs becoming ever stronger, this approach of accepting undecidable typing in order to obtain simpler languages is becoming more appealing: For example, after our publication of DHOL, it took the ATP community only one year to build a native ATP for DHOL [11].

Contribution In the present paper, we leverage that DHOL's meta-theory and infrastructure are in place to deal with undecidable typing: we extend DHOL with refinement and quotient types. Both are inherently undecidable and therefore often difficult to add to languages designed to keep typing decidable. We extend the DHOL→HOL-translation accordingly and prove soundness and completeness for the extended language.

Refinement types $A|_p$ consist of all objects of type A that satisfy the predicate $p{:}A \to \mathsf{bool}$. They correspond to comprehension in set theory. We had already sketched this extension of DHOL in [14]. A major advantage of this approach to is that it allows leveraging subtyping to move between types without a change in representation. For example, we have the subtyping statement $A|_p \prec: A$ and the injection $A|_p \to A$ is a no-op, whereas the usual approach in dependent type theory (i.e., representing $A|_p$ as $\Sigma x{:}A.B$) requires projecting out the first component to move between the types. Similarly, we have $A \to B \prec: (A|_p) \to B$, whereas the usual approach in set theory (i.e., representing a function $A \to B$ as a set of $A \times B$ pairs) requires restricting the function to a smaller domain to move between the types.

Quotient types ${}^A\!/_r$, intuitively, consist of all equivalence classes of objects of type A relative to the equivalence relation $r{:}A \to A \to \mathsf{bool}$. But we again leverage subtyping to obtain a more efficient representation: We use every object of type A as an object of type ${}^A\!/_r$ and adjust the equality $=_{{}^A\!/_r}$ to obtain the quotient semantics. Thus, the projection $A \to {}^A\!/_r$ is a no-op and $A \prec: {}^A\!/_r$. The usual approach in set theory, on the other hand, (i.e., using equivalence classes as elements of the quotient) requires a change of representation. Similarly, the usual approach in dependent type theory (i.e., using setoids to represent quotients) requires explicit operations to represent the elements of the quotient.

The statement $A \prec: {}^A\!/_r$ may look odd. It is sound because we use a different equality relation at the two types: $x =_A y$ implies $x =_{{}^A\!/_r} y$ but not the other way round. We hold that our approach is not only justified by mathematical practice but provides an elegant formalization of it. Indeed, wherever possible, practitioners use elements of A as if they were elements of the quotient and avoid using equivalence classes, often to the point that readers do not even notice anymore that they are technically working in a quotient, e.g., in group presentations or field extensions. But in formal systems, this approach has been adopted only occasionally, e.g., in Nurpl's quotients [5] or in Quotient Haskell in [2].

Together, this yields the subtype hierarchy of refinements and quotients of type A as in $A|_{\lambda x:A.\ \mathsf{false}} \prec: \ldots \prec: A|_r \prec: \ldots \prec: A|_{\lambda x:A.\ \mathsf{true}} \equiv A \equiv {}^A\!/_{=_A} \prec: \ldots \prec: {}^A\!/_r \prec: \ldots \prec: {}^A\!/_{\lambda x,\, y:A.\ \mathsf{true}}$ ranging from the initial objects in the category of types, which are empty, to the terminal objects, which are singleton types.

Overview We give a self-contained definition of grammar, judgments, and inference system of DHOL in Sect. 2. Then we introduce our subtyping framework in Sect. 3, refinements in Sect. 4, and quotients in Sect. 5. We develop the meta-theory in Sect. 6 and Sect. 7, describing normalizing resp. soundness/completeness. We sketch an application to formalizing typed set theory, which partially motivated this paper, in Sect. 8, and we conclude in Sect. 9.

2 Preliminaries: Dependently Type Higher-Order Logic

2.1 Syntax

The grammar of DHOL [14] is given below. A theory true consists of dependent type declarations $\mathsf{a}{:}\Pi x_1{:}A_1.\ \ldots\ \Pi x_n{:}A_n.\ \mathsf{tp}$, which are applied to arguments to obtain base types $\mathsf{a}\ t_1\ \ldots\ t_n$. Additionally, a theory declares typed constants $\mathsf{c}{:}A$ and axioms $\triangleright F$. Contexts declare typed variables $x{:}A$ and local assumptions $\triangleright F$ (but no new types).

$$
\begin{array}{llll}
T & ::= & \circ \mid T,\ \mathsf{a}{:}(\Pi x{:}A.\ )^*\mathsf{tp} \mid T,\ \mathsf{c}{:}A \mid T,\ \triangleright F & \text{theories} \\
\Gamma & ::= & . \mid \Gamma,\ x{:}A \mid \Gamma,\ \triangleright F & \text{contexts} \\
A, B & ::= & \mathsf{a}\ t^* \mid \Pi x{:}A.\ B \mid \mathsf{bool} & \text{types} \\
s, t, F, G & ::= & \mathsf{c} \mid x \mid \lambda x{:}A.\ t \mid s\ t \mid s =_A t \mid F \Rightarrow G & \text{terms (including propositions)}
\end{array}
$$

DHOL arises in a straightforward way from HOL by adding dependent function types $\Pi x{:}A.\ B$, whose functions map each argument $x{:}A$ to a result in $B(x)$. We write this type as $A \to B$ if x does not occur free in B. Dependent function types come with terms $\lambda x{:}A.\ t$ for function construction and $s\ t$ for function application.

Following typical HOL-style [1], we use a minimal set of connectives, essentially defining all connectives and quantifiers from the equality connective $s =_A t$. Critically, we use a single axiomatic equality $s =_A t$ in the style of FOL and HOL combine it with a straightforward congruence rule for base types: our rules below derive the type equality $\mathsf{a}\ s_1\ \ldots\ s_n \equiv \mathsf{a}\ t_1\ \ldots\ t_n$ if each term s_i is equal to t_i. This makes type equality and thus typing undecidable.

Because of this undecidability, and contrary to HOL, we need *dependent* binary connectives: in an implication $F \Rightarrow G$, the well-formedness of G may depend on the truth of F. This cannot be defined from equality alone, which is why we make dependent implication an additional primitive. Dependent conjunction and disjunction are definable and behave accordingly. Another consequence of undecidable well-formedness is that the well-formedness of a declaration in a theory/context may depend on previous axioms/assumptions. Therefore, our theories/contexts are lists in which declarations and axioms/assumptions may alternate.

DHOL is a conservative extension of HOL. We can recover HOL as the fragment of DHOL in which all base types a have arity 0. Then all function types are simple, typing is decidable, and thus all axioms/assumptions can be collected into a set.

▶ **Example 1** (Lists). As a running example, we consider a formalization of lists over some type $\mathtt{obj}$, both plain lists list and lists llist n with fixed length. It is given in Fig. 1. Now for example, the statement of associativity of lconc is only well-typed if we have previously stated the associativity of plus.

$\mathtt{nat}$: tp, $\mathtt{zero}$: $\mathtt{nat}$, $\mathtt{succ}$: $\mathtt{nat} \to \mathtt{nat}$, $\mathtt{plus}$: $\mathtt{nat} \to \mathtt{nat} \to \mathtt{nat}$,
$\mathtt{obj}$: tp, $\mathtt{list}$: tp, $\mathtt{nil}$: $\mathtt{list}$, $\mathtt{cons}$: $\mathtt{obj} \to \mathtt{list} \to \mathtt{list}$, $\mathtt{conc}$: $\mathtt{list} \to \mathtt{list} \to \mathtt{list}$,
$\mathtt{llist}$: $\mathtt{nat} \to \mathsf{tp}$, $\mathtt{lnil}$: $\mathtt{llist}\ \mathtt{zero}$,
$\mathtt{lcons}$: $\Pi n{:}\mathtt{nat}.\ \mathtt{obj} \to \mathtt{llist}\ n \to \mathtt{llist}\ (\mathtt{succ}\ n)$,
$\mathtt{lconc}$: $\Pi m, n{:}\mathtt{nat}.\ \mathtt{llist}\ m \to \mathtt{llist}\ n \to \mathtt{llist}\ (\mathtt{plus}\ m\ n)$

Figure 1 Lists in DHOL as used in Ex. 1

Name	Judgment	Intuition
theories	$\vdash T\ \mathsf{Thy}$	T is well-formed theory
contexts	$\vdash_T \Gamma\ \mathsf{Ctx}$	Γ is well-formed context
types	$\Gamma \vdash_T A\ \mathsf{tp}$	A is well-formed type
typing	$\Gamma \vdash_T t{:}A$	t is a well-formed term of well-formed type A
validity	$\Gamma \vdash_T F$	well-formed Boolean F is derivable
equality of types	$\Gamma \vdash_T A \equiv B$	well-formed types A and B are equal

Figure 2 DHOL Judgments

2.2 Inference System

DHOL uses the judgments given in Fig. 2 and the rules listed in Fig. 3. Note that while equality of terms is a Boolean term and thus equality of terms is a special case of validity, equality of types is not a Boolean and is a separate judgment. In particular, users cannot state axioms that identify types, and the only type equality is given by the congruence rules. Also note how the typing rule for implication allows using the truth of F when checking G.

The rules are straightforward and induce a type-checking algorithm in the usual way. In particular, type equality is checked structurally and reduced to a set of term equalities, which must be discharged by an ATP.

2.3 Translation to HOL

We obtain a semantics of DHOL and a practical ATP workflow via a sound and complete translation to HOL [1, 8]. HOL can be obtained as the fragment of DHOL where dependent types take no arguments and thus all function types are simple. The translation is *dependency erasure*: the identity translation except for erasing all arguments of base types, i.e., translating dependent types $\mathtt{a}\ t_1\ \ldots\ t_n$ to simple types $\mathtt{a}$, effectively "'merging"' all instances of dependent types into a larger simple type. The general structure is given in Fig. 4 and the concrete definition in Fig. 5.

Typing and equality are preserved by generating a partial equivalence relation (PER) $\mathtt{A}^*$ for every type A. In general, a PER r on type U is a symmetric and transitive relation on U. This is equivalent to r being an equivalence relation on a subtype of U. The intuition behind our translation is that the DHOL-type A corresponds in HOL to the quotient of the appropriate subtype of $\overline{A}$ by the equivalence $\mathtt{A}^*$. All terms are translated to their HOL analogue except that equality is translated to the respective PER: $\overline{s =_A t} = \mathtt{A}^*\ \overline{s}\ \overline{t}$. In particular, the predicate $\mathtt{A}^*\ \overline{t}\ \overline{t}$ captures whether t represents a term of type A. For n-ary dependent type constructors $\mathtt{a}$, the translation generates an $n+2$-ary predicate $\mathtt{a}^*$ such that $\mathtt{a}^*\ \overline{t_1}\ \ldots\ \overline{t_n}$ is the PER for $\mathtt{a}\ t_1\ \ldots\ t_n$. For function types, the PER is defined using

Theories and contexts:

$$\frac{}{\vdash \circ\ \mathsf{Thy}} \qquad \frac{\vdash_{\mathsf{T}} x_1{:}A_1,\ \ldots,\ x_n{:}A_n\ \mathsf{Ctx}}{\vdash T,\ \mathsf{a}{:}\Pi x_1{:}A_1.\ \ldots \Pi x_n{:}A_n.\ \mathsf{tp}\ \mathsf{Thy}} \qquad \frac{\vdash_{\mathsf{T}} A\ \mathsf{tp}}{\vdash T,\ \mathsf{c}{:}A\ \mathsf{Thy}} \qquad \frac{\vdash_{\mathsf{T}} F{:}\mathsf{bool}}{\vdash T,\ \triangleright F\ \mathsf{Thy}}$$

$$\frac{\vdash T\ \mathsf{Thy}}{\vdash_{\mathsf{T}}.\ \mathsf{Ctx}} \qquad \frac{\Gamma \vdash_{\mathsf{T}} A\ \mathsf{tp}}{\vdash_{\mathsf{T}} \Gamma,\ x{:}A\ \mathsf{Ctx}} \qquad \frac{\Gamma \vdash_{\mathsf{T}} F{:}\mathsf{bool}}{\vdash_{\mathsf{T}} \Gamma,\ \triangleright F\ \mathsf{Ctx}}$$

Well-formedness and equality of types:

$$\frac{\mathsf{a}{:}\Pi x_1{:}A_1.\ \ldots \Pi x_n{:}A_n.\ \mathsf{tp}\ \text{in}\ T \qquad \Gamma \vdash_{\mathsf{T}} t_1{:}A_1 \quad \ldots \quad \Gamma \vdash_{\mathsf{T}} t_n{:}A_n[x_1/t_1]\ldots[x_{n-1}/t_{n-1}]}{\Gamma \vdash_T\ \mathsf{a}\ t_1\ \ldots\ t_n\ \mathsf{tp}} \qquad \frac{\vdash_{\mathsf{T}} \Gamma\ \mathsf{Ctx}}{\Gamma \vdash_{\mathsf{T}} \mathsf{bool}\ \mathsf{tp}} \qquad \frac{\Gamma \vdash_{\mathsf{T}} A\ \mathsf{tp} \quad \Gamma,\ x{:}A \vdash_{\mathsf{T}} B\ \mathsf{tp}}{\Gamma \vdash_{\mathsf{T}} \Pi x{:}A.\ B\ \mathsf{tp}}$$

$$\frac{\mathsf{a}{:}\Pi x_1{:}A_1.\ \ldots \Pi x_n{:}A_n.\ \mathsf{tp}\ \text{in}\ T \qquad \Gamma \vdash_{\mathsf{T}} s_1 =_{A_1} t_1\ \ldots\ \Gamma \vdash_{\mathsf{T}} s_n =_{A_n[x_1/t_1]\ldots[x_{n-1}/t_{n-1}]} t_n}{\Gamma \vdash_{\mathsf{T}} \mathsf{a}\ s_1\ \ldots\ s_n \equiv \mathsf{a}\ t_1\ \ldots t_n} \qquad \frac{\vdash_{\mathsf{T}} \Gamma\ \mathsf{Ctx}}{\Gamma \vdash_{\mathsf{T}} \mathsf{bool} \equiv \mathsf{bool}\ \mathsf{tp}} \qquad \frac{\Gamma \vdash_{\mathsf{T}} A \equiv A' \quad \Gamma, x{:}A \vdash_{\mathsf{T}} B \equiv B'}{\Gamma \vdash_{\mathsf{T}} \Pi x{:}A.\ B \equiv \Pi x{:}A'.\ B'}$$

Typing:

$$\frac{c{:}A'\ \text{in}\ T \qquad \Gamma \vdash_{\mathsf{T}} A' \equiv A}{\Gamma \vdash_{\mathsf{T}} \mathsf{c}{:}A} \qquad \frac{\Gamma,\ x{:}A \vdash_{\mathsf{T}} t{:}B \qquad A' \equiv A}{\Gamma \vdash_{\mathsf{T}} (\lambda x{:}A.\ t){:}\Pi x{:}A'.\ B} \qquad \frac{\Gamma \vdash_{\mathsf{T}} F{:}\mathsf{bool} \qquad \Gamma,\ \triangleright F \vdash_{\mathsf{T}} G{:}\mathsf{bool}}{\Gamma \vdash_{\mathsf{T}} F \Rightarrow G{:}\mathsf{bool}}$$

$$\frac{x{:}A'\ \text{in}\ \Gamma \qquad \Gamma \vdash_{\mathsf{T}} A' \equiv A}{\Gamma \vdash_{\mathsf{T}} x{:}A} \qquad \frac{\Gamma \vdash_{\mathsf{T}} f{:}\Pi x{:}A.\ B \qquad \Gamma \vdash_{\mathsf{T}} t{:}A}{\Gamma \vdash_{\mathsf{T}} f\ t{:}B[x/t]} \qquad \frac{\Gamma \vdash_{\mathsf{T}} s{:}A \qquad \Gamma \vdash_{\mathsf{T}} t{:}A}{\Gamma \vdash_{\mathsf{T}} s =_A t{:}\mathsf{bool}}$$

Equality: congruence, reflexivity, symmetry, β, η (derivable: transitivity, functional extensionality):

$$\frac{\Gamma \vdash_{\mathsf{T}} A \equiv A' \quad \Gamma,\ x{:}A \vdash_{\mathsf{T}} t =_B t'}{\Gamma \vdash_{\mathsf{T}} \lambda x{:}A.\ t =_{\Pi x:A.\ B} \lambda x{:}A'.\ t'} \qquad \frac{\Gamma \vdash_{\mathsf{T}} t =_A t' \quad \Gamma \vdash_{\mathsf{T}} f =_{\Pi x:A.\ B} f'}{\Gamma \vdash_{\mathsf{T}} f\ t =_B f'\ t'}$$

$$\frac{\Gamma \vdash_{\mathsf{T}} t{:}A}{\Gamma \vdash_{\mathsf{T}} t =_A t} \qquad \frac{\Gamma \vdash_{\mathsf{T}} t =_A s}{\Gamma \vdash_{\mathsf{T}} s =_A t} \qquad \frac{\Gamma \vdash_{\mathsf{T}} (\lambda x{:}A.\ s)\ t{:}B}{\Gamma \vdash_{\mathsf{T}} (\lambda x{:}A.\ s)\ t =_B s[x/t]} \qquad \frac{\Gamma \vdash_{\mathsf{T}} t{:}\Pi x{:}A.\ B}{\Gamma \vdash_{\mathsf{T}} t =_{\Pi x:A.\ B} \lambda x{:}A.\ t\ x}$$

Rules for validity: lookup, implication, Boolean equality and extensionality

$$\frac{\triangleright F\ \text{in}\ T \qquad \vdash_{\mathsf{T}} \Gamma\ \mathsf{Ctx}}{\Gamma \vdash_{\mathsf{T}} F} \qquad \frac{\Gamma,\ \triangleright F \vdash_{\mathsf{T}} G}{\Gamma \vdash_{\mathsf{T}} F \Rightarrow G} \qquad \frac{\Gamma \vdash_{\mathsf{T}} F =_{\mathsf{bool}} F' \qquad \Gamma \vdash_{\mathsf{T}} F'}{\Gamma \vdash_{\mathsf{T}} F}$$

$$\frac{\triangleright F\ \text{in}\ \Gamma \qquad \vdash_{\mathsf{T}} \Gamma\ \mathsf{Ctx}}{\Gamma \vdash_{\mathsf{T}} F} \qquad \frac{\Gamma \vdash_{\mathsf{T}} F \Rightarrow G \qquad \Gamma \vdash_{\mathsf{T}} F}{\Gamma \vdash_{\mathsf{T}} G} \qquad \frac{\Gamma \vdash_{\mathsf{T}} p\ \mathsf{true} \qquad \Gamma \vdash_{\mathsf{T}} p\ \mathsf{false}}{\Gamma, x{:}\mathsf{bool} \vdash_{\mathsf{T}} p\ x}$$

Figure 3 DHOL Rules

the usual condition for logical relations: functions are related if they map related inputs to related outputs.

3 Subtyping

Definition The treatment of quotients as subtypes and the use of different equality relations at different types are subtly difficult. Therefore, we first introduce a general definition that captures the essence of subtyping once and for all, and from which we will derive all concrete subtyping rules later on:

▶ **Definition 2** (Subtyping). $\Gamma \vdash_{\mathsf{T}} A \prec: B$ *abbreviates* $\Gamma, x{:}A \vdash_{\mathsf{T}} x{:}B$.

DHOL	HOL
type A	type $\overline{A}$ and PER $\mathtt{A}^*{:}\overline{A} \to \overline{A} \to \mathsf{bool}$
term $t{:}A$	term $\overline{t}{:}\overline{A}$ satisfying $\mathtt{A}^*\ \overline{t}\ \overline{t}$

Figure 4 Structure of DHOL→HOL translation

Theories and contexts, declaration-wise:

$$\overline{\circ} := \circ \qquad \overline{T,\ D} := \overline{T},\ \overline{D} \qquad \overline{\cdot} := \cdot \qquad \overline{T,\ D} := \overline{T},\ \overline{D}$$

$$\overline{\mathtt{a}{:}\Pi x_1{:}A_1.\ \ldots \Pi x_n{:}A_n.\ \mathsf{tp}} := \mathtt{a}{:}\mathsf{tp},\ \mathtt{a}^*{:}\overline{A_1} \to \ldots \to \overline{A_n} \to \mathtt{a} \to \mathtt{a} \to \mathsf{bool},$$
$$\triangleright\forall x_1{:}\overline{A_1}.\ \ldots \forall x_n{:}\overline{A_n}.\ \forall u, v{:}\mathtt{a}.\ \mathtt{a}^*\ x_1\ \ldots\ x_n\ u\ v \Rightarrow u =_a v$$

$$\overline{\mathtt{c}{:}A} := \mathtt{c}{:}\overline{A},\ \triangleright\mathtt{A}^*\ \mathtt{c}\ \mathtt{c} \qquad \overline{x{:}A} := x{:}\overline{A},\ \triangleright\mathtt{A}^*\ x\ x$$

$$\overline{\triangleright F} := \triangleright\overline{F} \qquad \triangleright\overline{F} := \triangleright\overline{F}$$

Types:

$$\overline{\mathtt{a}\ t_1\ \ldots\ t_n} := \mathtt{a} \qquad (\mathtt{a}\ \mathtt{t_1}\ \ldots\ \mathtt{t_n})^*\ s\ t := \mathtt{a}^*\ \overline{t_1}\ \ldots\ \overline{t_n}\ s\ t$$

$$\overline{\Pi x{:}A.\ B} := \overline{A} \to \overline{B} \qquad (\Pi\mathtt{x}{:}\mathtt{A}.\ \mathtt{B})^*\ f\ g := \forall x, y{:}\overline{A}.\ \mathtt{A}^*\ x\ y \Rightarrow \mathtt{B}^*\ (f\ x)\ (g\ y)$$

$$\overline{\mathsf{bool}} := \mathsf{bool} \qquad \mathsf{bool}^*\ s\ t := s =_{\mathsf{bool}} t$$

Terms:

$$\overline{\mathtt{c}} := \mathtt{c} \qquad \overline{x} := x \qquad \overline{\lambda x{:}A.\ t} := \lambda x{:}\overline{A}.\ \overline{t} \qquad \overline{f\ t} := \overline{f}\ \overline{t}$$

$$\overline{s =_A t} := \mathtt{A}^*\ \overline{s}\ \overline{t} \qquad \overline{F \Rightarrow G} := \overline{F} \Rightarrow \overline{G}$$

Figure 5 Definition of the Translation DHOL→HOL

Note that this definition is independent of any concrete type operators being part of the language. It is also very intuitive: users can immediately understand whether subtyping should hold. But it is not as general as one might think:

▶ **Lemma 3.** *In any extension of DHOL, $\Gamma \vdash_{\mathsf{T}} A \prec: B$ is equivalent to the derivability of*

$$\frac{\Gamma \vdash_{\mathsf{T}} t{:}A}{\Gamma \vdash_{\mathsf{T}} t{:}B}$$

Proof. Left-to-right: We construct the function $\lambda x{:}A.\ x{:}A \to B$ and derive the needed rule using the typing rule for function application.

Right-to-left: We start with $\Gamma, x{:}A \vdash_{\mathsf{T}} x{:}A$ and apply the derivable rule. ◀

Thus, our subtyping relation rules out *incidental* subtype instances, where the rule from Lem. 3 is only admissible but not derivable. For example, the empty type $A|_{\lambda x:A.\ \mathsf{false}}$ will be a subtype of any refinement of A, but not of all types. More generally, our definition precludes using induction on the terms of A to conclude $A \prec: B$. That restriction ensures that subtyping is preserved under, e.g., theory extensions, substitution, or language extensions. Importantly, subtyping preserves equality:

▶ **Lemma 4.** *In any extension of DHOL,* $\Gamma\vdash_{\mathsf{T}} A \prec: B$ *implies* $\Gamma, x{:}A, y{:}A, \triangleright x =_A y \vdash_{\mathsf{T}} x =_B y$ *(which is equivalent to* $\Gamma\vdash_{\mathsf{T}} \forall x{:}A.\ \forall y{:}A.\ x =_A y \Rightarrow x =_B y$*).*

If the rule $\frac{\Gamma\vdash_{\mathsf{T}} s =_A t}{\Gamma\vdash_{\mathsf{T}} s{:}A}(*)$ *is admissible, then the converse holds, too.*

Proof. Left-to-right: The subtyping assumption yields $\Gamma, x{:}A,\ y{:}A,\ \triangleright x =_A y \vdash_{\mathsf{T}} (\lambda x{:}A.\ x){:}A \to B$. Using the congruence of function application and reflexivity, we obtain $\Gamma, x{:}A,\ y{:}A,\ \triangleright x =_A y \vdash_{\mathsf{T}} (\lambda x{:}A.\ x)\ x =_B (\lambda x{:}A.\ x)\ y$, which yields $x =_B y$ by β-reduction.

Right-to-left: In context $\Gamma, x{:}A$ we can derive $x =_A x$ by reflexivity. The assumption now yields $\Gamma, x{:}A \vdash_{\mathsf{T}} x =_B x$, from which we get $\Gamma, x{:}A \vdash_{\mathsf{T}} x{:}B$ by $*$. ◀

Intuitively, the condition $*$ is necessary because establishing that $x =_B y$ is well-typed at all is already equivalent to showing that $x, y{:}B$. It is satisfied by DHOL and all extensions introduced in this paper but must be checked separately for each extension. If satisfied, we characterize subtyping through truth as $\Gamma\vdash_{\mathsf{T}} A \prec: B$ iff $\Gamma\vdash_{\mathsf{T}} \forall x, y{:}A.\ x =_A y \Rightarrow x =_B y$. But the practicality of this characterization depends on the design choices of the individual theorem provers, which may very well violate $*$ on purpose for efficiency or accidentally due to subtle implementation errors.

Subtype Ordering We want subtyping to be an order relation on types.

▶ **Lemma 5.** *In any extension of DHOL, subtyping is reflexive (in the sense that* $\Gamma\vdash_{\mathsf{T}} A \equiv B$ *implies* $\Gamma\vdash_{\mathsf{T}} A \prec: B$*) and transitive.*

Proof. Reflexivity: The assumption yields $\Gamma\vdash_{\mathsf{T}} \lambda x{:}A.\ x{:}A \to B$ and we also have $\Gamma\vdash_{\mathsf{T}} \lambda x{:}A.\ x{:}A \to A$. Applying both to a term t of type A and β-reducing yields the rule from from Lem. 3.
Transitivity: This follows immediately from Lem. 3. ◀

However, subtyping is not anti-symmetric with respect to $\equiv$, i.e., we might have $\Gamma\vdash_{\mathsf{T}} A \prec: B$ and $\Gamma\vdash_{\mathsf{T}} B \prec: A$ without being able to derive $\Gamma\vdash_{\mathsf{T}} A \equiv B$. We make it so by *adding* the following rule

$$\frac{\Gamma\vdash_{\mathsf{T}} A \prec: B \quad \Gamma\vdash_{\mathsf{T}} B \prec: A}{\Gamma\vdash_{\mathsf{T}} A \equiv B}\text{STantisym}$$

Note that this is the only *change* we are making to DHOL here — everything before has just been abbreviations. Our change is conservative in the following sense:

▶ **Theorem 6** (Conservativity for Plain DHOL)**.** *In DHOL as defined so far, we have that* $A \prec: B$ *iff* $A \equiv B$*.*

Proof. We show by induction on derivations that each term has a unique type up to type equality and that all term equality axioms satisfy the subject reduction property. ◀

In other words, DHOL (without the extension we are about to make) has no non-trivial subtyping at this point.

Derivable Rules As a first exercise of our definitions, we obtain the usual congruence and variance rule for function types:

▶ **Theorem 7** (Equality and Variance for Function Types). *The following rules are derivable*

$$\frac{\Gamma \vdash_{\mathsf{T}} A' \prec: A \quad \Gamma,\ x{:}A' \vdash_{\mathsf{T}} B \prec: B'}{\Gamma \vdash_{\mathsf{T}} \Pi x{:}A.\ B \prec: \Pi x{:}A'.\ B'} \qquad \frac{\Gamma \vdash_{\mathsf{T}} A' \equiv A \quad \Gamma,\ x{:}A' \vdash_{\mathsf{T}} B \equiv B'}{\Gamma \vdash_{\mathsf{T}} \Pi x{:}A.\ B \equiv \Pi x{:}A'.\ B'}$$

Note that the second rule in Lem. 7 is already part of DHOL (see Fig. 3). So derivability here means it is now derivable from the remaining rules and thus redundant.

Proof. The first rule is derived by expanding the definition of subtyping, using η-expansion of the function under consideration. The second rule is derived using (STantisym) and then establishing the two hypotheses using the variance rule and reflexivity of subtyping. ◀

4 Refinement types

Syntax To add refinement types, we add only one production to the grammar:

$$A ::= \quad A|_p \quad \text{type } A \text{ refined by predicate } p \text{ on } A$$

Note that we do not add productions for terms — refinement types only provide new typing properties for the existing terms.

Inference System The rules for, respectively, formation, introduction, elimination (two rules), and equality for refinement types are:

$$\frac{\Gamma \vdash_{\mathsf{T}} p{:}A \to \mathsf{bool}}{\Gamma \vdash_{\mathsf{T}} A|_p\ \mathsf{tp}} \qquad \frac{\Gamma \vdash_{\mathsf{T}} t{:}A \quad \Gamma \vdash_{\mathsf{T}} p\ t}{\Gamma \vdash_{\mathsf{T}} t{:}A|_p} \qquad \frac{\Gamma \vdash_{\mathsf{T}} t{:}A|_p}{\Gamma \vdash_{\mathsf{T}} t{:}A} \qquad \frac{\Gamma \vdash_{\mathsf{T}} t{:}A|_p}{\Gamma \vdash_{\mathsf{T}} p\ t} \qquad \frac{\Gamma \vdash_{\mathsf{T}} s =_A t \quad \Gamma \vdash_{\mathsf{T}} p\ s}{\Gamma \vdash_{\mathsf{T}} s =_{A|_p} t}$$

▶ **Example 8** (Refining Lists by Length). We extend Ex. 1 by obtaining fixed-length lists as a refinement of lists. First, we declare a predicate length on lists defined by two axioms:

$$\begin{aligned} &\mathsf{length}\colon \mathsf{list} \to \mathsf{nat} \\ &\triangleright\ \mathsf{length}\ \mathsf{nil} =_{\mathsf{nat}} \mathsf{zero} \\ &\triangleright\ \forall x{:}\mathtt{obj}.\ \forall l{:}\mathsf{list}.\ \mathsf{length}\ (\mathsf{cons}\ x\ l) =_{\mathsf{nat}} \mathsf{succ}\ (\mathsf{length}\ l) \end{aligned}$$

Now we can define $\mathsf{llist}\ n\ :=\ \mathsf{list}|_{\lambda l:\mathsf{list}.\ \mathsf{length}\ l =_{\mathsf{nat}} n}$. The constants for lnil and lcons are redundant, and we can instead derive the corresponding types for nil and cons:

$$\vdash \mathsf{nil}{:}\mathsf{llist}\ \mathsf{zero} \qquad n{:}\mathsf{nat} \vdash \mathsf{cons} : \Pi x{:}\mathtt{obj}.\ \Pi l{:}\mathsf{llist}\ n.\ \mathsf{llist}\ (\mathsf{succ}\ n)$$

Like for function types, we can *derive* the congruence and variance rules:

▶ **Theorem 9** (Congruence and Variance). *The following rules are derivable:*

$$\frac{\Gamma \vdash_{\mathsf{T}} A \prec: A' \quad \Gamma,\ x{:}A,\ \triangleright p\ x \vdash_{\mathsf{T}} p'\ x}{\Gamma \vdash_{\mathsf{T}} A|_p \prec: A'|_{p'}} \qquad \frac{\Gamma \vdash_{\mathsf{T}} A\ \mathsf{tp}}{\Gamma \vdash_{\mathsf{T}} A \equiv A|_{\lambda x:A.\ \mathsf{true}}}$$

$$\frac{\Gamma \vdash_{\mathsf{T}} A \equiv A' \quad \Gamma \vdash_{\mathsf{T}} p =_{A \to \mathsf{bool}} p'}{\Gamma \vdash_{\mathsf{T}} A|_p \equiv A'|_{p'}} \qquad \frac{\Gamma \vdash_{\mathsf{T}} A|_p\ \mathsf{tp}}{\Gamma \vdash_{\mathsf{T}} A|_p \prec: A}$$

Proof. To derive the first rule, we assume the hypotheses and $x{:}A|_p$. The elimination rules yield $x{:}A$ and $p\ x$, then the first hypothesis yields $x{:}A'$ and $p'\ x$, then the introduction rule yields $x{:}A|_p$.

To derive the second rule, we apply (STantisym) and use the introduction/elimination rules to show the two subtype relationships.

These then imply the other rules. ◀

5 Quotient types

Syntax To add quotient types we extend the grammar with only one production:

$$A ::= \ {}^{A}\!/_{r} \quad \text{quotient of } A \text{ by equivalence relation } r$$

Inference System The rules for, respectively, formation, introduction, elimination, and equality for quotient types are:

$$\frac{\Gamma\vdash_{\mathsf{T}} A\ \mathsf{tp} \quad \Gamma\vdash_{\mathsf{T}} r{:}A\to A\to\mathsf{bool} \quad \Gamma\vdash_{\mathsf{T}}\mathrm{EqRel}(r)}{\Gamma\vdash_{\mathsf{T}} {}^{A}\!/_{r}\ \mathsf{tp}} \qquad \frac{\Gamma\vdash_{\mathsf{T}} t{:}A \quad \Gamma\vdash_{\mathsf{T}} {}^{A}\!/_{r}\ \mathsf{tp}}{\Gamma\vdash_{\mathsf{T}} t{:}{}^{A}\!/_{r}}$$

$$\frac{\Gamma\vdash_{\mathsf{T}} s{:}{}^{A}\!/_{r} \quad \Gamma,\ x{:}A,\ \triangleright x =_{{}^{A}/_{r}} s\vdash_{\mathsf{T}} t{:}B \quad \Gamma,\ x{:}A,\ x'{:}A,\ \triangleright x =_{{}^{A}/_{r}} s,\ \triangleright x' =_{{}^{A}/_{r}} s\vdash_{\mathsf{T}} t =_B t[{}^{x}\!/_{x'}]}{\Gamma\vdash_{\mathsf{T}} t[{}^{x}\!/_{s}]{:}B[{}^{x}\!/_{s}]}$$

$$\frac{\Gamma\vdash_{\mathsf{T}} s{:}A \quad \Gamma\vdash_{\mathsf{T}} t{:}A \quad \Gamma\vdash_{\mathsf{T}} r{:}A\to A\to\mathsf{bool} \quad \mathrm{EqRel}(r)}{\Gamma\vdash_{\mathsf{T}} (s =_{{}^{A}/_{r}} t) =_{\mathsf{bool}} (r\ s\ t)}$$

where $\mathrm{EqRel}(r)$ abbreviates that r is an equivalence relation.

▶ **Example 10** (Sets). We extend Ex. 1 by obtaining sets as a quotient of lists. First, we define a contains-check for lists:

$$\mathsf{contains}\colon \mathsf{list}\to\mathtt{obj}\to\mathsf{bool}$$
$$\triangleright\ \forall x{:}\mathtt{obj}.\ \neg(\mathsf{contains}\ \mathsf{nil}\ x)$$
$$\triangleright\ \forall x{:}\mathtt{obj}.\ \forall y{:}\mathtt{obj}.\ \forall l{:}\mathsf{list}.\ (\mathsf{contains}\ (\mathsf{cons}\ y\ l)\ x) =_{\mathsf{bool}} (x =_{\mathtt{obj}} y \vee \mathsf{contains}\ l\ x)$$

Now we can define $\mathsf{set} := {}^{\mathsf{list}}\!/_{\lambda l:\mathsf{list}.\ \lambda m:\mathsf{list}.\ \forall x:\mathtt{obj}.\ \mathsf{contains}\ l\ x =_{\mathsf{bool}} \mathsf{contains}\ m\ x}$ as the type of lists containing the same elements. The equality at set immediately yields extensionality $\vdash \forall x,y{:}\mathsf{set}.\ x =_{\mathsf{set}} y \Leftrightarrow (\forall z{:}\mathtt{obj}.\ \mathsf{contains}\ x\ z =_{\mathsf{bool}} \mathsf{contains}\ y\ z)$.

Any $l{:}\mathsf{list}$ can be used as a representative of the respective equivalence class in set, and operations on sets can be defined via operations on lists, e.g., we can establish $\vdash \mathsf{conc} : \mathsf{set}\to\mathsf{set}\to\mathsf{set}$. To derive this, we assume $u{:}\mathsf{set}$ and apply the elimination rule twice. First we apply it with $B = \mathsf{list}\to\mathsf{set}$ and $t = \mathsf{conc}\ u$; we have to show $\mathsf{conc}\ x =_{\mathsf{list}\to\mathsf{set}} \mathsf{conc}\ x'$ under the assumption that x and y are equal as sets. That yields a term $\mathsf{conc}\ u : \mathsf{list}\to\mathsf{set}$. We assume $v{:}\mathsf{set}$ and apply the elimination rule again with $B = \mathsf{set}$ to obtain $\mathsf{conc}\ u\ v{:}\mathsf{set}$, and then conclude via λ-abstraction and η-reduction.

The elimination rule looks overly complex. It can be understood best by comparing it to the following, simpler and more intuitive rule

$$\frac{\Gamma,\ x{:}A\vdash_{\mathsf{T}} t{:}B \quad \Gamma,\ x{:}A,\ x'{:}A,\ \triangleright r\ x\ x'\vdash_{\mathsf{T}} t =_B t[{}^{x}\!/_{x'}]}{\Gamma,\ x{:}{}^{A}\!/_{r}\vdash_{\mathsf{T}} t{:}B}\ (*)$$

This rule captures the well-known condition that an operation t on A may be used to define an operation on ${}^{A}\!/_{r}$ if t maps equivalent representatives x, x' equally. Clearly, we can derive it from our elimination rule by putting $s = x$. But it is subtly weaker:

▶ **Example 11.** Continuing Ex. 10, assume a total order on $\mathtt{obj}$ and a function $g{:}\mathrm{list}|_{\mathrm{nonEmpty}} \to \mathtt{obj}$ picking the greatest from a non-empty list. We should be able to apply g to some $s{:}\mathrm{set}$ that we know to be non-empty. But if we try to apply $(*)$ to obtain $g\ s{:}\mathtt{obj}$, we find ourselves stuck trying to prove $g\ x =_{\mathtt{obj}} g\ x'$ for any x, x' that are representatives of an *arbitrary* equivalence class of lists. We are not allowed to use our additional knowledge that s is non-empty and thus only non-empty lists need to be considered. Thus, we cannot even derive that $g\ x$ is well-formed.

Our elimination rule remedies that: here we need to show $g\ x =_{\mathtt{obj}} g\ x'$ for any x, x' that are representatives of *the class of* s. Thus, we can use that x and x' are non-empty and that thus $g\ x$ is well-formed.

Like for function and refinement types, we can *derive* the congruence and variance rules:

▶ **Theorem 12** (Congruence and Variance). *The following rules are derivable:*

$$\frac{\Gamma \vdash_{\mathsf{T}} A \prec: A' \qquad \Gamma,\ x{:}A,\ y{:}A,\ \triangleright r\ x\ y \vdash_{\mathsf{T}} r'\ x\ y}{\Gamma \vdash_{\mathsf{T}} {}^{A}\!/_{r} \prec: {}^{A'}\!/_{r'}} \qquad \frac{\Gamma \vdash_{\mathsf{T}} A\ \mathsf{tp}}{\Gamma \vdash_{\mathsf{T}} A \equiv {}^{A}\!/_{\lambda x{:}A.\ \lambda y{:}A.\ x =_A y}}$$

$$\frac{\Gamma \vdash_{\mathsf{T}} A \equiv A' \qquad \Gamma \vdash_{\mathsf{T}} r =_{A \to A \to \mathsf{bool}} r'}{\Gamma \vdash_{\mathsf{T}} {}^{A}\!/_{r} \equiv {}^{A'}\!/_{r'}} \qquad \frac{\Gamma \vdash_{\mathsf{T}} {}^{A}\!/_{r}\ \mathsf{tp}}{\Gamma \vdash_{\mathsf{T}} A \prec: {}^{A}\!/_{r}}$$

Proof. To derive the first rule, we assume the hypotheses and $s{:}{}^{A}\!/_{r}$. We use the elimination rule with $B = {}^{A'}\!/_{r'}$ and $t = x$. We need to establish the second hypothesis of the elimination rule, which becomes $x{:}A,\ y{:}A,\ \triangleright x =_{{}^{A}/_{r}} s,\ \triangleright x' =_{{}^{A}/_{r}} s \vdash_{\mathsf{T}} x =_{{}^{A'}/_{r'}} x'$. We prove this by using the equality rule, which requires $x, x'{:}A'$ (which we show using $A \prec: A'$) and $r'\ x\ x'$, which follows from the second hypothesis.

To derive the second rule, we apply (STantisym) and use the introduction/elimination rules to show the two subtype relationships.

These then imply the other rules. ◀

6 Normalizing Types

Refinement and Quotient Types We can merge consecutive refinement and quotients:

▶ **Theorem 13** (Repeated Refinement/Quotient). *The following equalities are derivable whenever the LHS is well-formed*

$$\vdash (A|_{p})|_{p'} \equiv A|_{\lambda x{:}A.\ p\ x \wedge p'\ x} \qquad \vdash {}^{({}^{A}/_{r})}\!/_{r'} \equiv {}^{A}\!/_{\lambda x{:}A.\ \lambda y{:}A.\ r'\ x\ y} \qquad \vdash ({}^{A}\!/_{r})|_{p} \equiv {}^{(A|_{p})}\!/_{r}$$

Proof. For refinement-refinement, we first show that the RHS is well-formed: well-formedness of the LHS yields $p{:}A \to \mathsf{bool}$ and $p'{:}A|_{p} \to \mathsf{bool}$ and thus $p'\ x$ is well-formed because $\wedge$ is a *dependent* conjunction and $p\ x$ can be assumed while checking $p'\ x$. Verifying the equality is straightforward by showing subtyping in both directions.

For quotient-quotient, we first show that the RHS is well-formed: well-formedness of the LHS yields $r{:}A \to A \to \mathsf{bool}$ and $r'{:}{}^{A}\!/_{r} \to {}^{A}\!/_{r} \to \mathsf{bool}$, and $r'\ x\ y$ is well-formed because $A \prec: {}^{A}\!/_{r}$. The relation on the RHS is an equivalence relation because r' is. To verify the

type equality, we use Lem. 4 and show that both types induce the same equality on A. In particular, the type of r' already guarantees that it subsumes r.

For refinement-quotient, we first show that the RHS is well-formed: well-formedness of the LHS yields $r{:}A \to A \to \mathsf{bool}$ and $p{:}A/r \to \mathsf{bool}$. That implies $r{:}A|_p \to A|_p \to \mathsf{bool}$ and $p{:}A \to \mathsf{bool}$, which is needed for the well-formedness of the RHS. (Note the other direction does not hold in general.) To show the equality, we show both subtyping directions. For LHS $\prec{:}$ RHS, we assume $x{:}A/r$ and $p\ x$ and apply the elimination rule for quotients using $t = x$ and $B = {}^{(A|_p)}/_r$. (Critically, this step would not go through if we had only used the weaker rule $*$ in Sect. 5.) For RHS $\prec{:}$ LHS, we assume $x{:}{}^{(A|_p)}/_r$ and apply the elimination rule for quotients using $t = x$. ◀

Function Types and Subtyping We have 4 possible subtype situations for a function type: we can refine or quotient the domain or the codomain:

▶ **Theorem 14** (Refinement/Quotient in a Function Type). *The following judgments are derivable if either side is well-formed:*

$$\vdash \Pi x{:}A.\ (B|_p) \equiv (\Pi x{:}A.\ B)|_{\lambda f{:}\Pi x{:}A.\ B.\ \forall x{:}A.\ p\ (f\ x)}$$

$$\vdash \Pi x{:}A/r.\ B \equiv (\Pi x{:}A.\ B)|_{\lambda f{:}\Pi x{:}A.\ B.\ \forall x,y{:}A.\ r\ x\ y \Rightarrow (f\ x) =_B (f\ y)}$$

$$\vdash \Pi x{:}A.\ {}^B/_r \succ {}^{(\Pi x{:}A.\ B)}/_{\lambda f,g{:}\Pi x{:}A.\ B.\ \forall x{:}A.\ r\ (f\ x)\ (g\ x)}$$

The following one is derivable if the RHS is well-formed:

$$\vdash \Pi x{:}A|_p.\ B \succ {}^{(\Pi x{:}A.\ B)}/_{\lambda f,g{:}\Pi x{:}A.\ B.\ \forall x{:}A.\ p\ x \Rightarrow (f\ x) =_B (g\ x)}$$

Proof. Refined codomain: It is straightforward to prove both subtyping directions once we observe that terms on either side are given by $\lambda x{:}A.\ t$ where t has type B and satisfies p.

Quotiented domain: It is straightforward to prove both subtyping directions once we observe that both sides are subtypes of $\Pi x{:}A.\ B$ and that their elements must preserve r.

Quotiented codomain: We assume a term f of RHS-type and show $x{:}A \vdash f\ x{:}{}^B/_r$ using the quotient elimination rule.

Refined domain: We assume a term f of RHS-type and show $x{:}A|_p \vdash f\ x{:}B$ using the quotient elimination rule. Note that the well-formedness of the LHS does not imply the well-formedness of the RHS because the well-formedness of B might depend on the assumption $p\ x$ ◀

Maybe surprisingly, two of the subtyping laws in Thm. 14 are not equalities. The law for the refined domain must not be an equality:

▶ **Example 15** (Refined Domain). The issue here is that the assumption $p\ x$ makes more terms well-typed and thus there may be functions $\Pi x{:}A|_p.\ B$ that are not a restriction of a function $\Pi x{:}A.\ B$. Consider the theory $\mathsf{a}{:}\mathsf{bool} \to \mathsf{tp}$, $\mathsf{c}{:}\mathsf{a}\ \mathsf{true}$. Then $\mathsf{a}\ \mathsf{false}$ is empty and so are $\Pi x{:}\mathsf{bool}.\ \mathsf{a}\ x$ and its quotients. But with $p = \lambda x{:}\mathsf{bool}.\ x$, we have $\vdash \lambda x{:}\mathsf{bool}|_p.\ \mathsf{c} : \Pi x{:}\mathsf{bool}|_p.\ \mathsf{a}\ x$.

However, with the law for the quotiented codomain, we have some leeway that is related to which variant of the axiom of choice, if any, we want to adopt. Consider the following two statements

$$\vdash_{\mathsf{T}} \exists\, repr{:}{}^B/_r \to B.\ repr =_{{}^B/_r \to {}^B/_r} \lambda x{:}{}^B/_r.\ x \qquad f{:}\Pi x{:}A.\ {}^B/_r \vdash_{\mathsf{T}} \exists\, g{:}\Pi x{:}A.\ B.\ f =_{\Pi x{:}A.\ {}^B/_r} g$$

(Note that the first one is well-typed because repr also has type ${}^{B}\!/_{r} \to {}^{B}\!/_{r}$.) Both have a claim to be called the axiom of choice: The first one expresses that every equivalence relation has a system of representatives. The second generalizes this to a family of equivalence relations. The latter implies the former (put $A := {}^{B}\!/_{r}$ and $f := \lambda x{:}{}^{B}\!/_{r}.\ x$). In the simply-typed case the former also implies the latter (pick $\mathit{repr} \circ f$ for g); but in the dependently-typed case, where B and r may depend on x, the implication depends subtly on what other language features are around (e.g., Σ-types or choice).

Both statements construct a new term from existing term (repr behaves like the identity, and g like f) that has a different type but behaves the same up to quotienting. Adding the $\prec\!:$ direction to the law for the refined codomain would go a step further: it not only implies the existence of g from f but allows using f as a representative of the equivalence class of possible values for g. That is in keeping with our goal of avoiding changes of representation when transitioning between types:

▶ **Definition 16** (Quotiented Codomain). *We adopt as an additional axiom (whenever either side is well-formed):*

$$\vdash \Pi x{:}A.\ {}^{B}\!/_{r} \prec\!: {}^{(\Pi x:A.\ B)}\!/_{\lambda f, g:\Pi x:A.\ B.\ \forall x:A.\ r\ (f\ x)\ (g\ x)}$$

which is an equality in conjunction with Thm. 14.

Normalization Aggregating the above laws, we obtain a normalization algorithm for types:

▶ **Theorem 17** (Normalizing Types). *Every type is equal to a type of the form* ${}^{(A|_p)}\!/_{r}$ *where* $A ::= \mathsf{bool} \mid \mathsf{a}\ t^* \mid \Pi x{:}A|_p.\ B$.

Proof. Using Thm. 14 with the axiom from Def. 16, all refinements and quotients can be pushed out of all function types except for a single refinement of the domain; if there is no such refinement, we can use $p := \lambda x{:}A.\ \mathsf{true}$. And using Thm. 13, those can be collected into a single quotient+refinement. ◀

It is maybe surprising, and somewhat frustrating, that we need to allow for refined domains in the normal forms. Indeed, we initially expected being able to normalize those away as well, which would have allowed for a much more efficient algorithmic treatment. But we eventually found out, as discussed above, that is impossible.

7 Soundness and Completeness

We obtain a sound and complete theorem prover for DHOL via a translation to HOL. We build on the result in [14] and only describe the necessary extensions.

Translation We have added only two type operators to the grammar. We extend the translation from Fig. 5 that translates each DHOL type A to a HOL type $\overline{A}$ with a PER $\mathtt{A}^*$ on it:

$$\overline{A|_p} := \overline{A} \qquad (\mathtt{A}|_{\mathtt{p}})^*\ s\ t := \mathtt{A}^*\ s\ t \wedge \overline{p}\ s \wedge \overline{p}\ t$$
$$\overline{{}^{A}\!/_{r}} := \overline{A} \qquad ({}^{\mathtt{A}}\!/_{\mathtt{r}})^*\ s\ t := \overline{r}\ s\ t \wedge \mathtt{A}^*\ s\ s \wedge \mathtt{A}^*\ t\ t$$

Completeness HOL can prove the translations of all derivable DHOL judgments:

▶ **Theorem 18** (Completeness). *We have*

if in DHOL	*then in HOL*		
$\vdash T$ Thy	$\vdash \overline{T}$ Thy		
$\vdash_T \Gamma$ Ctx	$\vdash_{\overline{T}} \overline{\Gamma}$ Ctx		
$\Gamma \vdash_T A$ tp	$\overline{\Gamma} \vdash_{\overline{T}} \overline{A}$ tp	*and*	$\overline{\Gamma} \vdash_{\overline{T}} \mathtt{A}^* : \overline{A} \to \overline{A} \to \mathsf{bool}$ *and* $\mathtt{A}^*$ *is a PER*
$\Gamma \vdash_T A \equiv B$	$\overline{\Gamma} \vdash_{\overline{T}} \overline{A} \equiv \overline{B}$	*and*	$\overline{\Gamma}, x, y{:}\overline{A} \vdash_{\overline{T}} \mathtt{A}^*\ x\ y =_{\mathsf{bool}} \mathtt{B}^*\ x\ y$
$\Gamma \vdash_T A \prec: B$	$\overline{\Gamma} \vdash_{\overline{T}} \overline{A} \equiv \overline{B}$	*and*	$\overline{\Gamma}, x, y{:}\overline{B} \vdash_{\overline{T}} \mathtt{A}^*\ x\ y \Rightarrow \mathtt{B}^*\ x\ y$
$\Gamma \vdash_T t{:}A$	$\overline{\Gamma} \vdash_{\overline{T}} \overline{t}{:}\overline{A}$	*and*	$\overline{\Gamma} \vdash_{\overline{T}} \mathtt{A}^*\ \overline{t}\ \overline{t}$
$\Gamma \vdash_T F$	$\overline{\Gamma} \vdash_{\overline{T}} \overline{F}$		

Proof. Note that the subtyping claim is a slightly strengthened version of the claim obtained from the others by expanding the definition of $\prec:$. We adapt the proof from [14] with additional cases for all new productions and rules. The details are given in Appendix C. ◀

The case for subtyping in Thm. 18 gives us a criterion for which subtyping instances should hold. This allows to revisit our discussion following Thm. 14, which led us to adopt Def. 16:

▶ **Example 19** (PERs for a Quotiented Codomain). We calculate the PERs for both sides of the axiom of Def. 16:

$$(\Pi \mathtt{x{:}A}.\ \mathtt{B}/_{\mathtt{r}})^*\ f\ g = \forall x, y{:}\overline{A}.\ \mathtt{A}^*\ x\ y \Rightarrow (\overline{r}\ (f\ x)\ (g\ y) \wedge \mathtt{B}^*\ (f\ x)\ (f\ x) \wedge \mathtt{B}^*\ (g\ y)\ (g\ y))$$

which can be simplified to

$$\forall x{:}\overline{A}.\ \mathtt{A}^*\ x\ x \Rightarrow \overline{r}\ (f\ x)\ (g\ x) \wedge \mathtt{B}^*\ (f\ x)\ (f\ x) \wedge \mathtt{B}^*\ (g\ y)\ (g\ y)$$

which is exactly what we get when we unfold

$$\left((\Pi \mathtt{x{:}A}.\ \mathtt{B})/_{\lambda \mathtt{f}, \mathtt{g}:\Pi \mathtt{x{:}A}.\ \mathtt{B}.\ \forall \mathtt{x{:}A}.\ \mathtt{r}\ (\mathtt{f}\ \mathtt{x})\ (\mathtt{g}\ \mathtt{x})}\right)^*\ f\ g$$

This justifies adopting the axiom.

▶ **Example 20** (PERs for the a Refined Domain). We calculate the PERs for both sides of the subtyping law for a refined domain:

$$\begin{aligned}
&\left(\Pi \mathtt{x{:}A}|_{\mathtt{p}}.\ \mathtt{B}\right)^*\ f\ g = \forall x, y{:}\overline{A}.\ \mathtt{A}^*\ x\ y \wedge \overline{p}\ x \wedge \overline{p}\ y \Rightarrow \mathtt{B}^*\ (f\ x)\ (g\ y)\\
&\left((\Pi \mathtt{x{:}A}.\ \mathtt{B})/_{\lambda \mathtt{f}, \mathtt{g}:\Pi \mathtt{x{:}A}.\ \mathtt{B}.\ \forall \mathtt{x{:}A}.\ \mathtt{p}\ \mathtt{x} \Rightarrow (\mathtt{f}\ \mathtt{x}) =_{\mathtt{B}} (\mathtt{g}\ \mathtt{x})}\right)^*\ f\ g =\\
&\forall x{:}\overline{A}.\ \mathtt{A}^*\ x\ x \Rightarrow (\overline{p}\ x \Rightarrow \mathtt{B}^*\ (f\ x)\ (g\ x)) \wedge\\
&\left(\forall x, y{:}\overline{A}.\ \mathtt{A}^*\ x\ y \Rightarrow \mathtt{B}^*\ (f\ x)\ (f\ y)\right) \wedge\\
&\left(\forall x, y{:}\overline{A}.\ \mathtt{A}^*\ x\ y \Rightarrow \mathtt{B}^*\ (g\ x)\ (g\ y)\right)
\end{aligned}$$

These are indeed not equivalent in line with our observation from Ex. 15.

Soundness As discussed in [14], the converse theorem to completeness is much harder to state and prove. But it does carry over to DHOL with subtyping:

▶ **Theorem 21** (Soundness).

If $\Gamma \vdash^{\mathsf{DHOL}}_{\mathsf{T}} F : \mathsf{bool}$ *and* $\overline{\Gamma} \vdash^{HOL}_{\overline{T}} \overline{F}$, *then* $\Gamma \vdash^{\mathsf{DHOL}}_{\mathsf{T}} F$

In particular, if $\Gamma \vdash_{\mathsf{T}} s{:}A$ *and* $\Gamma \vdash_{\mathsf{T}} t : A$ *and* $\overline{\Gamma} \vdash_{\overline{T}} \mathtt{A}^* \ \overline{s} \ \overline{t}$, *then* $\Gamma \vdash s =_A t$.

Proof. The key idea is to transform a HOL-proof of $\overline{F}$ into one that is in the image of the translation, at which point we can read off a DHOL-proof of F. The full proof is given in Appendix D. ◀

Intuitively, the reverse directions of Thm. 18 holds if we have already established that all involved expressions are well-typed in DHOL. Like in [14], we can develop an intertwined type-checker and theorem prover that type-checks the conjecture generating a sequence of proof obligations, and then calls the HOL ATP on the proof obligations and the conjecture.

8 Application to Typed Set Theory

As a major case study, we sketch a formalization of typed set-theory. Throughout, we assume we can *define* identifiers rather than just declare them, and we use common infix notations where clear from the context.

We start with

$$\mathrm{set}{:}\mathsf{tp}, \quad \in {:}\mathrm{set} \to \mathrm{set} \to \mathsf{bool}, \quad \mathit{elem}\ s := \mathrm{set}|_{\lambda x:\mathrm{set}.\ x \in s}$$

where *elem* lifts every set to the type level. We have previously used this idea for typed set theory [9] in plain LF without any support for subtyping. There, we needed explicit reasoning about refinement, which massively complicated the development. In DHOL with subtyping, these formalizations are much more elegant.

We skip the routine formalization of the axioms, definitions, and theorems for untyped set theory. For example, for untyped pairing we get operations and theorems:

$$\times{:}\mathrm{set} \to \mathrm{set} \to \mathrm{set}, \quad \mathit{pair}{:}\mathrm{set} \to \mathrm{set} \to \mathrm{set}, \quad \triangleright \forall x, y, s, t{:}\mathrm{set}.\ x \in s \wedge y \in t \ \Rightarrow\ \mathit{pair}\ x\ y \in s \times t$$

(where we omit the definitions and proofs).

We can now use that to easily define a typed pairing operator:

$$\mathit{tpair} : \Pi s, t{:}\mathrm{set}.\ \mathit{elem}\ s \to \mathit{elem}\ t \to \mathit{elem}\ s \times t \quad := \quad \lambda s, t{:}\mathrm{set}.\ \mathit{pair}$$

Type-checking this declaration yields the proof obligation

$$x{:}\mathit{elem}\ s,\ y{:}\mathit{elem}\ t \vdash \mathit{pair}\ x\ y : \mathit{elem}\ s \times t$$

which is exactly the corresponding untyped theorem. Similarly, all constructions of untyped set theory can be lifted to their typed counterparts.

Moreover, we can represent the *set* of functions from s to t as the *type* $\mathit{Functions}\ s\ t := {}^{(s \to t)|_p}\!/_r$ where

$$p\ f = \forall x{:}\mathrm{set}.\ x \in s \Rightarrow (f\ x) \in t$$

$$r\ f\ g = \forall x{:}\mathrm{set}.\ x \in s \Rightarrow (f\ x) =_{\mathrm{set}} (g\ x)$$

We can then define function application and composition $\circ$ in the usual way, leading to the conjecture that the composition of functions from s to t and from t to u yields a function from s to u:

$$\triangleright \forall s, t, u{:}\mathrm{set}.\ \forall f{:}\mathit{Functions}\ s\ t.\ \forall g{:}\mathit{Functions}\ t\ u.\ \forall x{:}\mathrm{set}.\ x \in s \Rightarrow ((g\ \circ\ f)\ x) \in u$$

9 Conclusion and Future Work

DHOL combines higher-order logic with dependent types, obtaining an intuitive and expressive language, albeit with undecidable typing. We have doubled down on this design in two ways to obtain an extension of DHOL with two type constructors that practitioners often demand from language designers: refinement and quotient types.

Firstly, like dependent function types, refinement and quotient types require *dependent* types, i.e., terms occurring in types. Moreover, both are inherently undecidable and are therefore near-impossible to add as an afterthought to a type theory with decidable typing. But in DHOL, they can be added very elegantly. Secondly, the semantics of DHOL is defined via a translation to HOL. Critically, this translation maps every DHOL-type to a HOL-type with a partial equivalence relation (PER) on it. Because PERs are closed under refinements and quotients, it became feasible to adapt the existing translation as well as the soundness/completeness proof to obtain the corresponding results for our extended DHOL.

We used an extensional subtyping approach, where $A \prec: B$ holds iff all A-terms also have type B. That enabled us to prove all the expected variance and normalization laws — with one unexpected exception: we do not have a normalization algorithm that eliminates function types with refined domains (in other words: partial functions). That makes the normal forms of types and thus the task of deriving efficient subtype-checking algorithms more complex. Future work must investigate how to improve on the latter.

Extending DHOL with choice operators and sigma types also remains for future work.

We also want to use DHOL to guide future improvements to existing refinement type systems for programming languages like e.g. Quotient Haskell for the Haskell language. These systems extends the programming language with refinement types (and in case of Quotient Haskell also quotient types) in order to obtain a lightweight specification language. Like in our work on DHOL, they use a translation to obtain ATP support. But unlike those systems which typically prioritizes proof obligations that are efficiently checkable by SMTs, DHOL focuses on rigorously working out the general case. Furthermore, our soundness proof enables proof reconstruction and checking, whereas the trusted codebase of refinement type systems typically includes an entire SMT solver. A combination of these advantages is so far lacking.

References

1 P. Andrews. *An Introduction to Mathematical Logic and Type Theory: To Truth Through Proof*. Academic Press, 1986.

2 B. Hewer and G. Hutton. Quotient Haskell: Lightweight Quotient Types for All. *Proc. ACM Program. Lang.*, 8(POPL), 2024. doi:10.1145/3632869.

3 G. Bancerek, C. Byliński, A. Grabowski, A. Korniłowicz, A. Naumowicz R. Matuszewski, K. Pak, and J. Urban. Mizar: State-of-the-art and beyond. In M. Kerber, J. Carette, C. Kaliszyk, F. Rabe, and V. Sorge, editors, *Intelligent Computer Mathematics*, page 261–279, Cham, 2015. Springer International Publishing.

4 A. Church. A Formulation of the Simple Theory of Types. *Journal of Symbolic Logic*, 5(1):56–68, 1940.

5 R. Constable, S. Allen, H. Bromley, W. Cleaveland, J. Cremer, R. Harper, D. Howe, T. Knoblock, N. Mendler, P. Panangaden, J. Sasaki, and S. Smith. *Implementing Mathematics with the Nuprl Development System*. Prentice-Hall, 1986.

6 Coq Development Team. The Coq Proof Assistant: Reference Manual. Technical report, INRIA, 2015.

7 L. de Moura, S. Kong, J. Avigad, F. van Doorn, and J. von Raumer. The Lean Theorem Prover (System Description). In A. Felty and A. Middeldorp, editors, *Automated Deduction*, page 378–388, Cham, 2015. Springer International Publishing.

8 M. Gordon. HOL: A Proof Generating System for Higher-Order Logic. In G. Birtwistle and P. Subrahmanyam, editors, *VLSI Specification, Verification and Synthesis*, page 73–128. Kluwer-Academic Publishers, 1988.

9 M. Iancu and F. Rabe. Formalizing Foundations of Mathematics. *Mathematical Structures in Computer Science*, 21(4):883–911, 2011.

10 P. Martin-Löf. An Intuitionistic Theory of Types: Predicative Part. In *Proceedings of the '73 Logic Colloquium*, pages 73–118. North-Holland, 1974.

11 J. Niederhauser, C. Brown, and C. Kaliszyk. Tableaux for automated reasoning in dependently-typed higher-order logic, 2024. under review.

12 U. Norell. The Agda WiKi, 2005. `http://wiki.portal.chalmers.se/agda`.

13 S. Owre, J. Rushby, and N. Shankar. PVS: A Prototype Verification System. In D. Kapur, editor, *11th International Conference on Automated Deduction (CADE)*, pages 748–752. Springer, 1992.

14 C. Rothgang, F. Rabe, and C. Benzmüller. Theorem Proving in Dependently Typed Higher-Order Logic. In B. Pientka and C. Tinelli, editors, *Automated Dedution*, pages 438–455. Springer, 2023.

A Summary of logics and translations

In this section we collect the inference rules of the logics and the definition of the overall translation. We name the rules and enumerate the cases in the definition of the translation for reference in the proofs in the subsequent appendices.

A.1 HOL rules

Theories and contexts:

$$\frac{}{\vdash \circ\ \mathsf{Thy}}\text{thyEmpty} \qquad \frac{\vdash T\ \mathsf{Thy}}{\vdash T,\ A\ \mathsf{tp}\ \mathsf{Thy}}\text{thyType} \qquad \frac{\vdash_T A\ \mathsf{tp}}{\vdash T,\ \mathsf{c}{:}A\ \mathsf{Thy}}\text{thyConst} \qquad \frac{\vdash_T F{:}\mathsf{bool}}{\vdash T,\ \triangleright F\ \mathsf{Thy}}\text{thyAxiom}$$

$$\frac{\vdash T\ \mathsf{Thy}}{\vdash_T .\ \mathsf{Ctx}}\text{ctxEmpty} \qquad \frac{\Gamma\vdash_T A\ \mathsf{tp}}{\vdash_T \Gamma,\ x{:}A\ \mathsf{Ctx}}\text{ctxVar} \qquad \frac{\Gamma\vdash_T F{:}\mathsf{bool}}{\vdash_T \Gamma,\ \triangleright F\ \mathsf{Ctx}}\text{ctxAssume}$$

Lookup in theory and context:

$$\frac{A{:}\mathsf{tp}\ \text{in}\ T \qquad \vdash_T \Gamma\ \mathsf{Ctx}}{\Gamma\vdash_T A\ \mathsf{tp}}\text{type} \qquad \frac{\mathsf{c}{:}A'\ \text{in}\ T \qquad \Gamma\vdash_T A'\equiv A}{\Gamma\vdash_T \mathsf{c}{:}A}\text{const} \qquad \frac{\triangleright F\ \text{in}\ T \qquad \vdash_T \Gamma\ \mathsf{Ctx}}{\Gamma\vdash_T F}\text{axiom}$$

$$\frac{x{:}A'\ \text{in}\ \Gamma \qquad \Gamma\vdash_T A'\equiv A}{\Gamma\vdash_T x{:}A}\text{var} \qquad \frac{\triangleright F\ \text{in}\ \Gamma \qquad \vdash_T \Gamma\ \mathsf{Ctx}}{\Gamma\vdash_T F}\text{assume}$$

Well-formedness and equality of types:

$$\frac{\vdash_T \Gamma\ \mathsf{Ctx}}{\Gamma\vdash_T \mathsf{bool}\ \mathsf{tp}}\text{bool} \qquad \frac{\Gamma\vdash_T A\ \mathsf{tp} \qquad \Gamma\vdash_T B\ \mathsf{tp}}{\Gamma\vdash_T A\to B\ \mathsf{tp}}\text{arrow} \qquad \frac{\Gamma\vdash_T A\ \mathsf{tp}}{\Gamma\vdash_T A\equiv A}\text{congBase} \qquad \frac{\Gamma\vdash_T A\equiv A' \qquad \Gamma\vdash_T B\equiv B'}{\Gamma\vdash_T A\to B\equiv A'\to B'}\text{cong}{\to}$$

Typing:

$$\frac{\Gamma,\ x{:}A\vdash_T t{:}B}{\Gamma\vdash_T (\lambda x{:}A.\ t){:}A\to B}\text{lambda} \qquad \frac{\Gamma\vdash_T f{:}A\to B \qquad \Gamma\vdash_T t{:}A}{\Gamma\vdash_T f\ t{:}B}\text{appl} \qquad \frac{\Gamma\vdash_T s{:}A \qquad \Gamma\vdash_T t{:}A}{\Gamma\vdash_T s=_A t{:}\mathsf{bool}}{=}\text{type}$$

Term equality, congruence, reflexivity, symmetry, β, η:

$$\frac{\Gamma\vdash_T A\equiv A' \qquad \Gamma,\ x{:}A\vdash_T t=_B t'}{\Gamma\vdash_T \lambda x{:}A.\ t=_{A\to B}\lambda x{:}A'.\ t'}\text{cong}\lambda\ (\text{xi}) \qquad \frac{\Gamma\vdash_T t=_A t' \qquad \Gamma\vdash_T f=_{A\to B} f'}{\Gamma\vdash_T f\ t=_B f'\ t'}\text{congAppl}$$

$$\frac{\Gamma\vdash_T t{:}A}{\Gamma\vdash_T t=_A t}\text{refl} \qquad \frac{\Gamma\vdash_T t=_A s}{\Gamma\vdash_T s=_A t}\text{sym} \qquad \frac{\Gamma\vdash_T (\lambda x{:}A.\ s)\ t{:}B}{\Gamma\vdash_T (\lambda x{:}A.\ s)\ t=_B s[x/t]}\text{beta} \qquad \frac{\Gamma\vdash_T t{:}A\to B \qquad x\ \text{not in}\ \Gamma}{\Gamma\vdash_T t=_{A\to B}\lambda x{:}A.\ t\ x}\text{eta}$$

Rules for implication:

$$\frac{\Gamma\vdash_T F{:}\mathsf{bool} \qquad \Gamma\vdash_T G{:}\mathsf{bool}}{\Gamma\vdash_T F\Rightarrow G{:}\mathsf{bool}}{\Rightarrow}\text{type} \qquad \frac{\Gamma\vdash_T F{:}\mathsf{bool} \qquad \Gamma,\ \triangleright F\vdash_T G}{\Gamma\vdash_T F\Rightarrow G}{\Rightarrow}\text{I} \qquad \frac{\Gamma\vdash_T F\Rightarrow G \qquad \Gamma\vdash_T F}{\Gamma\vdash_T G}{\Rightarrow}\text{E}$$

Congruence for validity, Boolean extensionality, and non-emptiness of types:

$$\frac{\Gamma\vdash_T F=_{\mathsf{bool}} F' \qquad \Gamma\vdash_T F'}{\Gamma\vdash_T F}\text{cong}{\vdash} \qquad \frac{\Gamma\vdash_T p\ \mathsf{true} \qquad \Gamma\vdash_T p\ \mathsf{false}}{\Gamma,x{:}\mathsf{bool}\vdash_T p\ x}\text{boolExt} \qquad \frac{\Gamma\vdash_T F{:}\mathsf{bool} \qquad \Gamma,x{:}A\vdash_T F}{\Gamma\vdash_T F}\text{nonempty}$$

In the soundness proof, we will occasionally use the existence of a HOL term of given type A (whoose existence follows from rule (nonempty)), so we denote this term by w_A.

Figure 6 HOL Rules

A.2 Derived rules

Using the rules given in Figure 6 we can derive a number of additional useful rules.

A.3 Admissible rules for HOL

The following lemma collects a few routine meta-theorems that we make use of later on:

▶ **Lemma 22.** *Given the inference rules for HOL (cfg. Figure 6), the following rules are admissible:*

$$\frac{\vdash_{\mathsf{T}} \Gamma\ \mathsf{Ctx}}{\vdash T\ \mathsf{Thy}}\ ctxThy \qquad \frac{\Gamma \vdash_{\mathsf{T}} A\ \mathsf{tp}}{\vdash_{\mathsf{T}} \Gamma\ \mathsf{Ctx}}\ tpCtx \qquad \frac{\Gamma \vdash_{\mathsf{T}} t{:}A}{\Gamma \vdash_{\mathsf{T}} A\ \mathsf{tp}}\ typingTp \qquad \frac{\Gamma \vdash_{\mathsf{T}} F}{\Gamma \vdash_{\mathsf{T}} F{:}\mathsf{bool}}\ validTyping$$

$$\frac{\mathsf{c}{:}A\ in\ T}{\Gamma \vdash_{\mathsf{T}} \mathsf{c}{:}A}\ constS \qquad \frac{x{:}A\ in\ \Gamma}{\Gamma \vdash_{\mathsf{T}} x{:}A}\ varS$$

$$\frac{\Gamma \vdash_{\mathsf{T}} A\ \mathsf{tp}}{\Gamma \vdash_{\mathsf{T}} A \equiv A}\equiv refl \qquad \frac{\Gamma \vdash_{\mathsf{T}} A \equiv A'}{\Gamma \vdash_{\mathsf{T}} A' \equiv A}\equiv sym \qquad \frac{\Gamma \vdash_{\mathsf{T}} A \equiv A' \quad \Gamma \vdash_{\mathsf{T}} A' \equiv A''}{\Gamma \vdash_{\mathsf{T}} A \equiv A''}\equiv trans$$

$$\frac{\Gamma \vdash_{\mathsf{T}} s =_A t}{\Gamma \vdash_{\mathsf{T}} s{:}A}\ eqTyping \qquad \frac{\Gamma \vdash_{\mathsf{T}} F \Rightarrow G}{\Gamma \vdash_{\mathsf{T}} F{:}\mathsf{bool}}\ implTypingL \qquad \frac{\Gamma \vdash_{\mathsf{T}} F \Rightarrow G}{\Gamma \vdash_{\mathsf{T}} G{:}\mathsf{bool}}\ implTypingR$$

$$\frac{\Gamma \vdash_{\mathsf{T}} s{:}A \quad \Gamma \vdash_{\mathsf{T}} s{:}A'}{\Gamma \vdash_{\mathsf{T}} A \equiv A'}\ typesUnique \qquad \frac{\Gamma \vdash_{\mathsf{T}} f\ t{:}B \quad \Gamma \vdash_{\mathsf{T}} f{:}A \to B}{\Gamma \vdash_{\mathsf{T}} t{:}A}\ typingWf$$

$$\frac{\Gamma \vdash_{\mathsf{T}} t{:}A \quad \Gamma \vdash_{\mathsf{T}} f\ t{:}B}{\Gamma \vdash_{\mathsf{T}} f{:}A \to B}\ applType \qquad \frac{\Gamma,\ x{:}B \vdash_{\mathsf{T}} s{:}A \quad \Gamma \vdash_{\mathsf{T}} t{:}B}{\Gamma \vdash_{\mathsf{T}} s[x/t]{:}A}\ rewriteTyping$$

$$\frac{\Gamma \vdash_{\mathsf{T}} F{:}\mathsf{bool} \quad \Gamma \vdash_{\mathsf{T}} G}{\Gamma,\ \triangleright F \vdash_{\mathsf{T}} G}\ monotonic{\vdash} \qquad \frac{\Gamma \vdash_{\mathsf{T}} A\ \mathsf{tp} \quad \Gamma \vdash_{\mathsf{T}} J \qquad for\ any\ statement \vdash_{\mathsf{T}} J}{\Gamma,\ x{:}A \vdash_{\mathsf{T}} J}\ var{\vdash}$$

$$\frac{\Gamma,\ x{:}A \vdash_{\mathsf{T}} F{:}\mathsf{bool}}{\Gamma \vdash_{\mathsf{T}} \forall\, x{:}A.\ F{:}\mathsf{bool}}\forall type \qquad \frac{\Gamma,\ x{:}A \vdash_{\mathsf{T}} F}{\Gamma \vdash_{\mathsf{T}} \forall\, x{:}A.\ F}\forall I \qquad \frac{\Gamma \vdash_{\mathsf{T}} \forall\, x{:}A.\ F \quad \Gamma \vdash_{\mathsf{T}} t{:}A}{\Gamma \vdash_{\mathsf{T}} F[x/t]}\forall E$$

$$\frac{\Gamma\ \mathsf{Ctx} \quad F\ in\ \Gamma}{\Gamma \vdash_{\mathsf{T}} F{:}\mathsf{bool}}\ assTyping \qquad \frac{\Gamma \vdash_{\mathsf{T}} t =_A t' \quad \Gamma \vdash_{\mathsf{T}} A \equiv A' \quad \Gamma \vdash_{\mathsf{T}} t{:}A'}{\Gamma \vdash_{\mathsf{T}} t'{:}A'}\ cong{:}$$

$$\frac{\Gamma \vdash_{\mathsf{T}} F =_{\mathsf{bool}} \mathsf{true}}{\Gamma \vdash_{\mathsf{T}} F} = \mathsf{true} \qquad \frac{\Gamma \vdash_{\mathsf{T}} F}{\Gamma \vdash_{\mathsf{T}} F =_{\mathsf{bool}} \mathsf{true}}\mathsf{true} = \qquad \frac{\Gamma,\ \triangleright F \vdash_{\mathsf{T}} G \quad \Gamma,\ \triangleright G \vdash_{\mathsf{T}} F}{\Gamma \vdash_{\mathsf{T}} F =_{\mathsf{bool}} G}\ propExt$$

$$\frac{\Gamma,\ x{:}A \vdash_{\mathsf{T}} f\ x =_B f'\ x \qquad \Gamma \vdash_{\mathsf{T}} f{:}A \to B \qquad \Gamma \vdash_{\mathsf{T}} f'{:}A \to B}{\Gamma \vdash_{\mathsf{T}} f =_{A \to B} f'}\ extensionality$$

$$\frac{\Gamma \vdash_{\mathsf{T}} s =_A t \quad \Gamma \vdash_{\mathsf{T}} t =_A u}{\Gamma \vdash_{\mathsf{T}} s =_A u}\ trans \qquad \frac{\Gamma \vdash_{\mathsf{T}} s =_A s' \quad \Gamma \vdash_{\mathsf{T}} t =_A t'}{\Gamma \vdash_{\mathsf{T}} (s =_A t) =_{\mathsf{bool}} (s' =_A t')} = cong$$

$$\frac{\Gamma \vdash_{\mathsf{T}} A \equiv A' \quad \Gamma,\ x{:}A \vdash_{\mathsf{T}} F =_{\mathsf{bool}} F'}{\Gamma \vdash_{\mathsf{T}} \forall\, x{:}A.\ F =_{\mathsf{bool}} \forall\, x{:}A'.\ F'}\forall cong \qquad \frac{\Gamma \vdash_{\mathsf{T}} F =_{\mathsf{bool}} F' \quad \Gamma \vdash_{\mathsf{T}} G =_{\mathsf{bool}} G'}{\Gamma \vdash_{\mathsf{T}} F \Rightarrow G =_{\mathsf{bool}} F' \Rightarrow G'}\Rightarrow cong$$

$$\frac{\Gamma,\ x{:}A \vdash_{\mathsf{T}} F \Rightarrow G}{\Gamma \vdash_{\mathsf{T}} \forall\, x{:}A.\ F \Rightarrow \forall\, x{:}A.\ G}\forall\Rightarrow \qquad \frac{\Gamma \vdash_{\mathsf{T}} G \Rightarrow G' \quad \Gamma \vdash_{\mathsf{T}} F' \Rightarrow F}{\Gamma \vdash_{\mathsf{T}} (F \Rightarrow G) \Rightarrow (F' \Rightarrow G')}\Rightarrow Funct$$

$$\frac{\Gamma \vdash_{\mathsf{T}} F =_{\mathsf{bool}} F' \quad \Gamma \vdash_{\mathsf{T}} F}{\Gamma \vdash_{\mathsf{T}} F'}{\vdash}\ cong \qquad \frac{\Gamma \vdash_{\mathsf{T}} F[x/t] \quad \Gamma \vdash_{\mathsf{T}} t =_A t' \quad \Gamma,\ x{:}A \vdash_{\mathsf{T}} F{:}\mathsf{bool}}{\Gamma \vdash_{\mathsf{T}} F[x/t']}\ rewrite$$

This Lemma 22 is already proven for the version of HOL in the paper[14] that originally introduced DHOL.

Furthermore, using the definitions of the connectives and quantifiers we can prove the rules:

$$\frac{\Gamma \vdash_{\mathsf{T}} F{:}\mathsf{bool} \quad \Gamma \vdash_{\mathsf{T}} G{:}\mathsf{bool}}{\Gamma \vdash_{\mathsf{T}} F \wedge G{:}\mathsf{bool}}\wedge \qquad \frac{\Gamma \vdash_{\mathsf{T}} F =_{\mathsf{bool}} F' \quad \Gamma \vdash_{\mathsf{T}} F =_{\mathsf{bool}} F'}{\Gamma \vdash_{\mathsf{T}} (F \wedge G) =_{\mathsf{bool}} (F' \wedge G')}\wedge\mathrm{Cong}$$

$$\frac{\Gamma \vdash_{\mathsf{T}} F \quad \Gamma \vdash_{\mathsf{T}} G}{\Gamma \vdash_{\mathsf{T}} F \wedge G}\wedge\mathrm{I} \qquad \frac{\Gamma \vdash_{\mathsf{T}} F \wedge G}{\Gamma \vdash_{\mathsf{T}} F}\wedge\mathrm{El} \qquad \frac{\Gamma \vdash_{\mathsf{T}} F \wedge G}{\Gamma \vdash_{\mathsf{T}} G}\wedge\mathrm{Er}$$

and similar rules for the other boolean connectives.

▶ **Remark 1.** *Observe that many of the rules derived for HOL in Lemma 22 still hold in DHOL. In particular, the rules (ctxThy), (tpCtx), (typingTp) and (validTyping) can be proven by the same method. The rules (monotonic$\vdash$), (var$\vdash$), ($\forall$type), ($\forall$E), ($\forall$I), (assTyping), (= true), (true =), (propExt), (extensionality), ($\forall$cong), ($\forall\Rightarrow$), ($\Rightarrow$Funct), ($\vdash$ cong), (rewrite) and the introduction and elimination rules for the (dependent) conjunction can be derived in DHOL with the same proofs. Also the rules ($\equiv$ refl) and ($\equiv$ sym) can be proven easily in DHOL by induction on the type equality rules.*

A.4 DHOL rules

Theories and contexts:

$$\frac{}{\vdash \circ\ \mathsf{Thy}}\text{thyEmpty} \qquad \frac{\vdash_T x_1{:}A_1,\ \ldots,\ x_n{:}A_n\ \mathsf{Ctx}}{\vdash T,\ \mathsf{a}{:}\Pi x_1{:}A_1.\ \ldots \Pi x_n{:}A_n.\ \mathsf{tp}\ \mathsf{Thy}}\text{thyType'}$$

$$\frac{\vdash_T A\ \mathsf{tp}}{\vdash T,\ \mathsf{c}{:}A\ \mathsf{Thy}}\text{thyConst} \qquad \frac{\vdash_T F{:}\mathsf{bool}}{\vdash T,\ \triangleright F\ \mathsf{Thy}}\text{thyAxiom}$$

$$\frac{\vdash T\ \mathsf{Thy}}{\vdash_T.\ \mathsf{Ctx}}\text{ctxEmpty} \qquad \frac{\Gamma\vdash_T A\ \mathsf{tp}}{\vdash_T\Gamma,\ x{:}A\ \mathsf{Ctx}}\text{ctxVar} \qquad \frac{\Gamma\vdash_T F{:}\mathsf{bool}}{\vdash_T\Gamma,\ \triangleright F\ \mathsf{Ctx}}\text{ctxAssume}$$

Well-formedness and equality of types:

$$\frac{\mathsf{a}{:}\Pi x_1{:}A_1.\ \ldots \Pi x_n{:}A_n.\ \mathsf{tp}\ \text{in}\ T \qquad \Gamma\vdash_T t_1{:}A_1 \quad\ldots\quad \Gamma\vdash_T t_n{:}A_n[x_1/t_1]\ldots[x_{n-1}/t_{n-1}]}{\Gamma\vdash_T \mathsf{a}\ t_1\ \ldots\ t_n\ \mathsf{tp}}\text{type'}$$

$$\frac{\Gamma\vdash_T p{:}\Pi x{:}A.\ \mathsf{bool}}{\Gamma\vdash_T A|_p\ \mathsf{tp}}|_p\mathsf{tp} \qquad \frac{\vdash_T\Gamma\ \mathsf{Ctx}}{\Gamma\vdash_T\mathsf{bool}\ \mathsf{tp}}\text{bool} \qquad \frac{\Gamma\vdash_T A\ \mathsf{tp} \quad \Gamma,\ x{:}A\vdash_T B\ \mathsf{tp}}{\Gamma\vdash_T\Pi x{:}A.\ B\ \mathsf{tp}}\text{pi}$$

$$\frac{\Gamma\vdash_T A\ \mathsf{tp} \qquad \Gamma\vdash_T r{:}\Pi x_1{:}A.\ \Pi x_2{:}A.\ \mathsf{bool} \qquad \Gamma\vdash_T\mathrm{EqRel}(r)}{\Gamma\vdash_T A/r\ \mathsf{tp}}\text{Q}$$

Type equality:

$$\frac{\mathsf{a}{:}\Pi x_1{:}A_1.\ \ldots \Pi x_n{:}A_n.\ \mathsf{tp}\ \text{in}\ T \qquad \Gamma\vdash_T s_1 =_{A_1} t_1\ \ldots\ \Gamma\vdash_T s_n =_{A_n[x_1/t_1]\ldots[x_{n-1}/t_{n-1}]} t_n}{\Gamma\vdash_T\mathsf{a}\ s_1\ \ldots\ s_n\ \equiv\ \mathsf{a}\ t_1\ \ldots t_n}\text{congBase'} \qquad \frac{\Gamma\vdash A\ \prec{:}\ B \qquad \Gamma\vdash B\ \prec{:}\ A}{\Gamma\vdash_T A\ \equiv\ B}\text{STantisym}$$

$$\frac{\vdash_T\Gamma\ \mathsf{Ctx}}{\Gamma\vdash_T\mathsf{bool}\ \equiv\ \mathsf{bool}\ \mathsf{tp}}\equiv\mathsf{bool} \qquad \frac{\Gamma\vdash_T A\ \equiv\ A' \quad \Gamma,x{:}A\vdash_T B\ \equiv\ B'}{\Gamma\vdash_T\Pi x{:}A.\ B\ \equiv\ \Pi x{:}A'.\ B'}\text{congΠ}$$

Typing:

$$\frac{c{:}A'\ \text{in}\ T \qquad \Gamma\vdash_T A'\ \equiv\ A}{\Gamma\vdash_T\mathsf{c}{:}A}\text{const'} \qquad \frac{x{:}A'\ \text{in}\ \Gamma \qquad \Gamma\vdash_T A'\ \equiv\ A}{\Gamma\vdash_T x{:}A}\text{var'}$$

$$\frac{\Gamma,\ x{:}A\vdash_T t{:}B \qquad \Gamma\vdash_T A\ \equiv\ A'}{\Gamma\vdash_T(\lambda x{:}A.\ t){:}\Pi x{:}A'.\ B}\text{lambda'} \qquad \frac{\Gamma\vdash_T f{:}\Pi x{:}A.\ B \qquad \Gamma\vdash_T t{:}A}{\Gamma\vdash_T f\ t{:}B[x/t]}\text{appl'}$$

$$\frac{\Gamma\vdash_T F{:}\mathsf{bool} \qquad \Gamma,\ \triangleright F\vdash_T G{:}\mathsf{bool}}{\Gamma\vdash_T F\ \Rightarrow\ G{:}\mathsf{bool}}\Rightarrow\text{type'} \qquad \frac{\Gamma\vdash_T s{:}A \qquad \Gamma\vdash_T t{:}A}{\Gamma\vdash_T s =_A t{:}\mathsf{bool}}=\text{type}$$

$$\frac{\Gamma\vdash_T t{:}A \qquad \Gamma\vdash_T p\ t}{\Gamma\vdash_T t{:}A|_p}|_p\mathrm{I} \qquad \frac{\Gamma\vdash_T t{:}A|_p}{\Gamma\vdash_T t{:}A}|_p\mathrm{E1} \qquad \frac{\Gamma\vdash_T t{:}A \qquad \Gamma\vdash_T\mathrm{EqRel}(r)}{\Gamma\vdash_T t{:}A/r}\text{QI}$$

$$\frac{\Gamma\vdash_T s{:}A/r \qquad \Gamma,\ x{:}A,\ x =_{A/r} s\vdash_T t{:}B \qquad \Gamma,\ x{:}A,\ x'{:}A,\ x =_{A/r} s,\ x' =_{A/r} s\vdash_T t =_B t[x/x']}{\Gamma\vdash_T t[x/s]{:}B[x/s]}\text{quotE}$$

Term equality; congruence, reflexivity, symmetry, β, η:

$$\frac{\Gamma\vdash_\mathsf{T} A \equiv A' \quad \Gamma,\ x{:}A\vdash_\mathsf{T} t =_B t'}{\Gamma\vdash_\mathsf{T} \lambda x{:}A.\ t =_{\Pi x:A.\ B} \lambda x{:}A'.\ t'}\text{cong}\lambda' \qquad \frac{\Gamma\vdash_\mathsf{T} t =_A t' \quad \Gamma\vdash_\mathsf{T} f =_{\Pi x:A.\ B} f'}{\Gamma\vdash_\mathsf{T} f\ t =_B f'\ t'}\text{congApp}l'$$

$$\frac{\Gamma\vdash_\mathsf{T} t{:}A}{\Gamma\vdash_\mathsf{T} t =_A t}\text{refl} \qquad \frac{\Gamma\vdash_\mathsf{T} t =_A s}{\Gamma\vdash_\mathsf{T} s =_A t}\text{sym} \qquad \frac{\Gamma\vdash_\mathsf{T} (\lambda x{:}A.\ s)\ t{:}B}{\Gamma\vdash_\mathsf{T} (\lambda x{:}A.\ s)\ t =_B s[x/t]}\text{beta} \qquad \frac{\Gamma\vdash_\mathsf{T} t{:}\Pi x{:}A.\ B}{\Gamma\vdash_\mathsf{T} t =_{\Pi x:A.\ B} \lambda x{:}A.\ t\ x}\text{etaPi}$$

$$\frac{\Gamma\vdash_\mathsf{T} s =_A t \quad \Gamma\vdash_\mathsf{T} p\ s}{\Gamma\vdash_\mathsf{T} s =_{A|_p} t}|_p\,\text{Eq} \qquad \frac{\Gamma\vdash_\mathsf{T} s{:}A \quad \Gamma\vdash_\mathsf{T} t{:}A \quad \Gamma\vdash_\mathsf{T} r{:}A\to A\to\mathsf{bool} \quad \text{EqRel}(r)}{\Gamma\vdash_\mathsf{T} (s =_{A/r} t) =_\mathsf{bool} (r\ s\ t)}\text{Q}=$$

Rules for validity:

$$\frac{\triangleright F \text{ in } T \quad \vdash_\mathsf{T}\Gamma\ \mathsf{Ctx}}{\Gamma\vdash_\mathsf{T} F}\text{axiom} \qquad \frac{\triangleright F \text{ in } \Gamma \quad \vdash_\mathsf{T}\Gamma\ \mathsf{Ctx}}{\Gamma\vdash_\mathsf{T} F}\text{assume}$$

$$\frac{\Gamma\vdash_\mathsf{T} F{:}\mathsf{bool} \quad \Gamma,\ \triangleright F\vdash_\mathsf{T} G}{\Gamma\vdash_\mathsf{T} F\Rightarrow G}\Rightarrow\text{I} \qquad \frac{\Gamma\vdash_\mathsf{T} F\Rightarrow G \quad \Gamma\vdash_\mathsf{T} F}{\Gamma\vdash_\mathsf{T} G}\Rightarrow\text{E}$$

$$\frac{\Gamma\vdash_\mathsf{T} F =_\mathsf{bool} F' \quad \Gamma\vdash_\mathsf{T} F'}{\Gamma\vdash_\mathsf{T} F}\text{cong}\vdash \qquad \frac{\Gamma\vdash_\mathsf{T} p\ \mathsf{true} \quad \Gamma\vdash_\mathsf{T} p\ \mathsf{false}}{\Gamma,\ x{:}\mathsf{bool}\vdash_\mathsf{T} p\ x}\text{boolExt}$$

$$\frac{\Gamma\vdash_\mathsf{T} t{:}A|_p}{\Gamma\vdash_\mathsf{T} p\ t}|_p\,\text{E2}$$

We also have the axiom (16).

Finally, we modify the rule for the non-emptiness of types: we allow the existence of empty dependent types and only require that for each HOL type in the image of the translation there exists one non-empty DHOL type translated to it (rather than requiring all dependent types translated to it to be non-empty). Observe that either restricting to the fragment HOL of DHOL or translating to it then yields the non-emptyness assumptions for HOL types.

B The translation from DHOL into HOL

Before actually going into the soundness and completeness proofs, we repeat and enumerate the cases in the definition of the translation, so we can reference them in the following.

▶ **Definition 23** (Translation). *We define a translation from DHOL to HOL syntax by induction on the Grammar.*

We use the notation $\overrightarrow{x{:}A}, \overrightarrow{\Pi x{:}A.}, \overrightarrow{A}$ *and* $\overrightarrow{x}$ *to denote* $x{:}A_1, \ldots, x_n{:}A_n$, $\Pi x_1{:}A_1.\ \ldots \Pi x_n{:}A_n.$, $A_1\to\ldots\to A_n$ *and* $x_1\ \ldots\ x_n$ *respectively.*

The cases for theories and contexts are:

$$\overline{\circ} := \circ \qquad \text{(PT1)}$$

$$\overline{T,\ D} := \overline{T},\ \overline{D} \qquad \textit{where}$$

$$\begin{aligned}\overline{\mathsf{a}{:}\overrightarrow{\Pi x{:}A.}\ \mathsf{tp}} := \mathsf{a}{:}\mathsf{tp},\ & \mathsf{a}^*{:}\overrightarrow{A}\to\mathsf{a}\to\mathsf{a}\to\mathsf{bool},\\ & \triangleright\forall\overrightarrow{x{:}A}.\ \forall u,v,w{:}\mathsf{a}.\ (\mathsf{a}\ \overrightarrow{x})^*\ u\ v\Rightarrow\left((\mathsf{a}\ \overrightarrow{x})^*\ v\ w\Rightarrow(\mathsf{a}\ \overrightarrow{x})^*\ u\ w\right),\\ & \triangleright\forall\overrightarrow{x{:}A}.\ \forall u,v{:}\mathsf{a}.\ (\mathsf{a}\ \overrightarrow{x})^*\ u\ v\Rightarrow(\mathsf{a}\ \overrightarrow{x})^*\ v\ u,\\ & \triangleright\forall\overrightarrow{x{:}A}.\ \forall u,v{:}\mathsf{a}.\ (\mathsf{a}\ \overrightarrow{x})^*\ v\ v\Rightarrow(\mathsf{a}\ \overrightarrow{x})^*\ u\ v =_\mathsf{bool} u =_\mathsf{a} v\end{aligned} \qquad \text{(PT2)}$$

$$\overline{\mathtt{c}:A} := \mathtt{c} : \overline{A}, \quad \triangleright \mathtt{A}^*\ \mathtt{c}\ \mathtt{c} \tag{PT3}$$
$$\overline{\triangleright F} := \triangleright \overline{F} \tag{PT4}$$

$$\overline{\cdot} := \cdot \tag{PT5}$$
$$\overline{\Gamma, x:A} := \overline{\Gamma},\ x:\overline{A}, \triangleright \mathtt{A}^*\ x\ x \tag{PT6}$$
$$\overline{\Gamma, \triangleright F} := \overline{\Gamma},\ \triangleright \overline{F} \tag{PT7}$$

The case of $\overline{A}$ *and* $\mathtt{A}^*\ s\ t$ *for types* A *are:*

$$\overline{(\mathtt{a}\ t_1\ \ldots\ t_n)} := \mathtt{a} \tag{PT8}$$
$$(\mathtt{a}\ \mathtt{t_1}\ \ldots\ \mathtt{t_n})^*\ s\ t := \mathtt{a}^*\ \overline{t_1}\ \ldots\ \overline{t_n}\ s\ t \tag{PT9}$$
$$\overline{\Pi x{:}A.\ B} := \overline{A} \to \overline{B} \tag{PT10}$$
$$(\Pi \mathtt{x}{:}\mathtt{A}.\ \mathtt{B})^*\ f\ g := \forall x, y{:}\overline{A}.\ \mathtt{A}^*\ x\ y \Rightarrow \mathtt{B}^*\ (f\ x)\ (g\ y) \tag{PT11}$$
$$\overline{\mathsf{bool}} := \mathsf{bool} \tag{PT12}$$
$$\mathsf{bool}^*\ s\ t := s =_{\mathsf{bool}} t \tag{PT13}$$

$$\overline{A|_p} := \overline{A} \tag{PT14}$$
$$(\mathtt{A}|_{\mathtt{p}})^*\ s\ t := \mathtt{A}^*\ s\ t \wedge \overline{p}\ s \wedge \overline{p}\ t \tag{PT15}$$
$$\overline{A/r} := \overline{A} \tag{PT16}$$
$$(\mathtt{A}/\mathtt{r})^*\ s\ t := \overline{r}\ s\ t \wedge \mathtt{A}^*\ s\ s \wedge \mathtt{A}^*\ t\ t \tag{PT17}$$

The cases for terms are:

$$\overline{\mathtt{c}} := \mathtt{c} \tag{PT18}$$
$$\overline{x} := x \tag{PT19}$$
$$\overline{\lambda x{:}A.\ t} := \lambda x{:}\overline{A}.\ \overline{t} \tag{PT20}$$
$$\overline{f\ t} := \overline{f}\ \overline{t} \tag{PT21}$$
$$\overline{F \Rightarrow G} := \overline{F} \Rightarrow \overline{G} \tag{PT22}$$
$$\overline{s =_A t} := \mathtt{A}^*\ \overline{s}\ \overline{t} \tag{PT23}$$

C Completeness proof

To simplify the inductive arguments, we will actually prove the following slightly stronger version of the theorem:

▶ **Theorem 24** (Completeness). *We have*

$\vdash T$ Thy	*implies*	$\vdash\ \overline{T}$ Thy		(1)
$\vdash_{\mathsf{T}} \Gamma$ Ctx	*implies*	$\vdash_{\overline{T}} \overline{\Gamma}$ Ctx		(2)
$\Gamma \vdash_{\mathsf{T}} A$ tp	*implies*	$\overline{\Gamma} \vdash_{\overline{T}} \overline{A}$ tp	*and* $\overline{\Gamma} \vdash_{\overline{T}} \mathtt{A}^*{:}\overline{A} \to \overline{A} \to \mathsf{bool}$	(3)
$\Gamma \vdash_{\mathsf{T}} A \equiv B$	*implies*	$\overline{\Gamma} \vdash_{\overline{T}} \overline{A} \equiv \overline{B}$	*and* $\overline{\Gamma}, x{:}\overline{A} \vdash_{\overline{T}} \mathtt{A}^*\ x\ x =_{\mathsf{bool}} \mathtt{B}^*\ x\ x$	(4)
$\Gamma \vdash_{\mathsf{T}} t{:}A$	*implies*	$\overline{\Gamma} \vdash_{\overline{T}} \overline{t}{:}\overline{A}$	*and* $\overline{\Gamma} \vdash_{\overline{T}} \mathtt{A}^*\ \overline{t}\ \overline{t}$	(5)

In case of $\prec:$ *we strengthen the first claim of*

$$\overline{\Gamma}, x{:}, \triangleright \mathtt{A}^*\ x\ x \vdash_{\overline{T}} x{:}\overline{B}$$

to $\overline{\Gamma} \vdash_{\overline{T}} \overline{A} \equiv \overline{B}$ *yielding:*

$$\Gamma \vdash_{\mathsf{T}} A \prec: B \quad \textit{implies } \overline{\Gamma} \vdash_{\overline{T}} \overline{A} \equiv \overline{B} \quad \textit{and } \overline{\Gamma},\ x, y{:}\overline{B} \vdash_{\overline{T}} \mathtt{A}^*\ x\ y \Rightarrow \mathtt{B}^*\ x\ y \tag{6}$$

$$\Gamma \vdash_{\mathsf{T}} F \quad \textit{implies } \overline{\Gamma} \vdash_{\overline{T}} \overline{F} \tag{7}$$

In case of term equality, we strengthen the claim to:

$$\Gamma \vdash_{\mathsf{T}} t =_A t' \quad \textit{implies } \overline{\Gamma} \vdash_{\overline{T}} \mathtt{A}^*\ \bar{t}\ \overline{t'} \quad \textit{and } \overline{\Gamma} \vdash_{\overline{T}} \bar{t}{:}\overline{A} \quad \textit{and } \overline{\Gamma} \vdash_{\overline{T}} \overline{t'}{:}\overline{A} \tag{8}$$

Furthermore, the typing relations $\mathtt{A}^*$ *are symmetric and transitive on all well-formed types* A*:*

$$\Gamma \vdash_{\mathsf{T}} A\ \mathsf{tp} \quad \textit{implies} \quad \overline{\Gamma} \vdash_{\overline{T}} \forall x, y{:}\overline{A}.\ \mathtt{A}^*\ x\ y \Rightarrow \mathtt{A}^*\ y\ x \tag{9}$$

$$\Gamma \vdash_{\mathsf{T}} A\ \mathsf{tp} \quad \textit{implies} \quad \overline{\Gamma} \vdash_{\overline{T}} \forall x, y, z{:}\overline{A}.\ \mathtt{A}^*\ x\ y \Rightarrow (\mathtt{A}^*\ y\ z \Rightarrow \mathtt{A}^*\ x\ z) \tag{10}$$

Additionally the substitution lemma holds, i.e.,

$$\Gamma,\ x{:}A \vdash_{\mathsf{T}} t{:}B \textit{ and } \Gamma \vdash u{:}A \quad \textit{implies} \quad \overline{\Gamma} \vdash_{\overline{T}} \overline{t[x/u]} =_{\overline{B}} \bar{t}[x/\bar{u}] \tag{11}$$

$$\Gamma,\ x{:}A \vdash_{\mathsf{T}} B\ \mathsf{tp} \textit{ and } \Gamma \vdash_{\mathsf{T}} u{:}B \quad \textit{implies} \quad \overline{\Gamma} \vdash_{\overline{T}} \overline{B[x/u]} \equiv \overline{B}[x/\bar{u}] \tag{12}$$

In the following lines, we assume that if $t = \lambda y{:}C.\ s$ *for* s *of type* D*, then* $B = \Pi y{:}C.\ D$ *(this is enough in practice and we cannot easily show more).*

$$\Gamma,\ x{:}A \vdash_{\mathsf{T}} t{:}B \quad \textit{implies} \quad \overline{\Gamma},\ x, x'{:}\overline{A},\ \triangleright \mathtt{A}^*\ x\ x' \vdash_{\overline{T}} \mathtt{B}^*\ \bar{t}\ \bar{t}[x/x'] \tag{13}$$

Here Case 4 looks weaker than in the original statement, but is easily seen to be equivalent. The equivalence proof uses induction on the shape of the types (reducing the claim to base types), propositional extensionality and the PER axioms.

Proof of Theorem 24. Firstly, we will prove the substitution lemma by induction on the grammar, i.e. by induction on the shape of the terms and types.

Afterwards, we will prove completeness of the translation w.r.t. all DHOL judgements by induction on the derivations. This means that we consider the inference rules of DHOL and prove that if completeness holds for the assumptions of a DHOL inference rule, then it also holds for the conclusion of the rule. For the inductive steps for some typing rules, namely (=type), we also require the fact that for any (well-formed) type A in DHOL we have $\mathtt{A}^*{:}\overline{A} \to \overline{A} \to \mathsf{bool}$. This follows directly from how the $\mathtt{A}^*$ are generated/defined in the translation.

C.1 Substitution lemma and symmetry and transitivity of the typing relations

Since the translation of types commutes with the type productions of the grammar (12) is obvious.

We show (11) by induction on the grammar of DHOL. If x is not a free variable in t, then $\overline{t[x/u]} = \overline{t} = \overline{t}[x/\overline{u}]$ and the claim (11) follows by rule (refl). So assume that x is a free variable of t.

If t is a variable, then by assumption (that x is a free variable in t) it follows that $t = x$ and thus $\overline{t[x/u]} = \overline{u} = \overline{t}[x/\overline{u}]$ and the claim follows by rule (refl).

If t is a λ-term $\lambda y{:}A.\ s$, then by induction hypothesis we have $\overline{\Gamma},\ y{:}\overline{A} \vdash_{\overline{T}} \overline{s[x/u]} =_{\overline{A}} \overline{s}[x/\overline{u}]$, where A is the type of s. By rule (congλ), the claim of $\overline{\Gamma} \vdash_{\overline{T}} \overline{\lambda y{:}A.\ s[x/u]} =_{\overline{B}} \overline{\lambda y{:}A.\ s}[x/\overline{u}]$ follows.

If t is a function application $f\ s$, then by induction hypothesis we have $\overline{\Gamma} \vdash_{\overline{T}} \overline{s[x/u]} =_{\overline{A}} \overline{s}[x/\overline{u}]$ and $\overline{\Gamma} \vdash_{\overline{T}} \overline{f[x/u]} =_{\overline{A}\to\overline{B}} \overline{f}[x/\overline{u}]$, where A is the type of s. By rule (congAppl), the claim of $\overline{\Gamma} \vdash_{\overline{T}} \overline{(f\ s)\,[x/u]} =_{\overline{B}} \overline{f\ s}[x/\overline{u}]$ follows.

If t is an equality $s =_A s'$, then by induction hypothesis we have $\overline{\Gamma} \vdash_{\overline{T}} \overline{s[x/u]} =_{\overline{A}} \overline{s}[x/\overline{u}]$ and $\overline{\Gamma} \vdash_{\overline{T}} \overline{s'[x/u]} =_{\overline{A}} \overline{s'}[x/\overline{u}]$, where A is the type of s and s'. By rule ($=$cong), the claim of $\overline{\Gamma} \vdash_{\overline{T}} \overline{(s =_A s')\,[x/u]} =_{\mathsf{bool}} \overline{(s =_A s')}\,[x/\overline{u}]$ follows.

Before we can show (13), we first need to prove the symmetry and transitivity of the typing relations: We can prove both by induction on the type A. Denote $\Delta := \overline{\Gamma},\ x, y{:}\overline{A},\ \triangleright\mathtt{A}^*\ x\ y$ and $\Theta := \overline{\Gamma},\ x, y, z{:}\overline{A},\ \triangleright\mathtt{A}^*\ x\ y,\ \triangleright\mathtt{A}^*\ y\ z$ respectively. If we can show $\Delta \vdash_{\overline{T}} \mathtt{A}^*\ y\ x$ and $\Theta \vdash_{\overline{T}} \mathtt{A}^*\ x\ z$ respectively, then the claims (9) and (10) follows by the rules ($\Rightarrow$I), ($\forall$I), (varS) and (assume). Those are therefore the claims we are going to show.

Observe that for types declared in the theory T, the symmetry and transitivity of $\mathtt{A}^*$ follows from the axiom generated by the translation (in case (PT2)) of the type declaration declaring A. This follows from the symmetry and transitivity of equality and (11).

If A is bool, the typing relation is $=_{\mathsf{bool}}$ which is symmetric and transitive by the rules (sym) and (trans) respectively. In these cases the claims follows by the rule (assume) and rule (sym) resp. by rule (assume) and rule (trans).

If A is a Π-type $\Pi x{:}C.\ D$ we have $\mathtt{C}^*\ f\ g = \forall\, x, y{:}\overline{C}.\ \mathtt{C}^*\ x\ y \Rightarrow \mathtt{D}^*\ f\ x\ g\ y$. Then we have

$$\Delta = \overline{\Gamma},\ x, y{:}\overline{A},\ \triangleright\forall\, w{:}\overline{C}.\ \forall\, w'{:}\overline{C}.\ \mathtt{C}^*\ w\ w' \Rightarrow \mathtt{D}^*\ (x\ w)\ (y\ w')$$

and

$$\Theta = \overline{\Gamma},\ x, y, z{:}\overline{A},\ \triangleright\forall\, w{:}\overline{C}.\ \forall\, w'{:}\overline{C}.\ \mathtt{C}^*\ w\ w' \Rightarrow \mathtt{D}^*\ (x\ z)\ (y\ z'),\ \triangleright\forall\, w{:}\overline{C}.\ \forall\, w'{:}\overline{C}.\ \mathtt{C}^*\ w\ w' \Rightarrow \mathtt{D}^*\ (y\ w)\ (z\ w').$$

The claim is

$$\Delta \vdash_{\overline{T}} \forall\, w, w'{:}\overline{C}.\ \mathtt{C}^*\ w\ w' \Rightarrow \mathtt{D}^*\ (y\ w)\ (x\ w')$$

and

$$\Theta \vdash_{\overline{T}} \forall\, w, w'{:}\overline{C}.\ \mathtt{C}^*\ w\ w' \Rightarrow \mathtt{D}^*\ (x\ w)\ (z\ w')$$

respectively.

We can prove the claim for (9) by

$$\Delta,\ w, w'{:}\overline{C},\ \triangleright\mathtt{C}^*\ w\ w' \vdash_{\overline{T}}$$

$\mathtt{C}^*\ w\ w' \Rightarrow \mathtt{D}^*\ (x\ w)\ (y\ w')$ (∀E),(∀E),(assume) (14)

$\Delta,\ w, w'{:}\overline{C},\ \triangleright\mathtt{C}^*\ w\ w' \vdash_{\overline{T}}$
$\mathtt{D}^*\ (x\ w)\ (y\ w')$ (⇒E),(14),(assume) (15)

$\Delta,\ w, w'{:}\overline{C},\ \triangleright\mathtt{C}^*\ w\ w' \vdash_{\overline{T}}$
$\mathtt{D}^*\ (x\ w)\ (y\ w') \Rightarrow \mathtt{D}^*\ (y\ w)\ (x\ w')$ induction hypothesis (16)

$\Delta,\ w, w'{:}\overline{C},\ \triangleright\mathtt{C}^*\ w\ w' \vdash_{\overline{T}}$
$\mathtt{D}^*\ (x\ w)\ (y\ w')$ (⇒E),(16),(15) (17)

$\Delta,\ w, w'{:}\overline{C} \vdash_{\overline{T}} \mathtt{C}^*\ w\ w' \Rightarrow \mathtt{D}^*\ (y\ w)\ (x\ w')$ (⇒I),(17) (18)

$\Delta \vdash_{\overline{T}} \forall\, w, w'{:}\overline{C}.\ \mathtt{C}^*\ w\ w' \Rightarrow \mathtt{D}^*\ (y\ w)\ (x\ w')$ (∀I),(∀I),(18)

We can prove the claim for (10) similarly. For this denote $\Lambda := \Theta,\ w, w'{:}\overline{C},\ \triangleright\mathtt{C}^*\ w\ w'$.

$\Lambda \vdash_{\overline{T}} \mathtt{C}^*\ w\ w' \Rightarrow \mathtt{D}^*\ (x\ w)\ (y\ w')$ (∀E),(∀E),(assume) (19)

$\Lambda \vdash_{\overline{T}} \mathtt{C}^*\ w\ w' \Rightarrow \mathtt{D}^*\ (y\ w)\ (z\ w')$ (∀E),(∀E),(assume) (20)

$\Lambda \vdash_{\overline{T}} \mathtt{D}^*\ (x\ w)\ (y\ w')$ (⇒E),(19),(assume) (21)

$\Lambda \vdash_{\overline{T}} \mathtt{D}^*\ (y\ w)\ (z\ w')$ (⇒E),(20),(assume) (22)

$\Lambda \vdash_{\overline{T}} \mathtt{D}^*\ (x\ w)\ (y\ w') \Rightarrow \big(\mathtt{D}^*\ (y\ w)\ (z\ w')$
$\Rightarrow \mathtt{D}^*\ (x\ w)\ (z\ w')\big)$ induction hypothesis (23)

$\Lambda \vdash_{\overline{T}} \mathtt{D}^*\ (y\ w)\ (z\ w') \Rightarrow \mathtt{D}^*\ (x\ w)\ (z\ w')$ (⇒E),(23),(21) (24)

$\Lambda \vdash_{\overline{T}} \mathtt{D}^*\ (x\ w)\ (z\ w')$ (⇒E),(24),(22) (25)

$\Theta,\ w, w'{:}\overline{C} \vdash_{\overline{T}} \mathtt{C}^*\ w\ w' \Rightarrow \mathtt{D}^*\ (x\ w)\ (z\ w')$ (⇒E),(25),(assume) (26)

$\overline{\Gamma} \vdash_{\overline{T}} \forall\, w, w'{:}\overline{C}.\ \mathtt{C}^*\ w\ w' \Rightarrow \mathtt{D}^*\ (x\ w)\ (z\ w')$ (∀I),(∀I),(26)

If A is a quotient-type ${}^{B}\!/_{r}$ we have $\mathtt{A}^*\ s\ t = \overline{r}\ s\ t \wedge \mathtt{A}^*\ s\ s \wedge \mathtt{A}^*\ s\ s$ for all terms $s, t{:}\overline{A}$. Observe that the assumption $\mathrm{EqRel}(r) := \forall\, x, y{:}B.\ (x =_B y \Rightarrow r\ x\ y) \wedge (r\ x\ y \Rightarrow r\ y\ x) \wedge (\forall\, z{:}B.\ r\ x\ y \wedge r\ y\ z \Rightarrow r\ x\ z)$ for ${}^{B}\!/_{r}$ is translated to $\forall\, x{:}\overline{B}.\ \mathtt{B}^*\ x\ x \Rightarrow \forall\, y{:}\overline{B}.\ \mathtt{B}^*\ y\ y \Rightarrow (\mathtt{B}^*\ x\ y \Rightarrow \overline{r}\ x\ y) \wedge (\overline{r}\ x\ y \Rightarrow \overline{r}\ y\ x) \wedge (\forall\, z{:}\overline{B}.\ \mathtt{B}^*\ z\ z \Rightarrow \overline{r}\ x\ y \wedge \overline{r}\ y\ z \Rightarrow \overline{r}\ x\ z)$, which implies that $\overline{r}$ is an equivalence for terms x satisfying $\mathtt{B}^*\ x\ x$. Therefore, $\mathtt{A}^*$ is also an equivalence relation.

It remains to consider the case of $A = B|_p$. In this case, the claim is $\Delta \vdash_{\overline{T}} \mathtt{B}^*\ y\ x \wedge \overline{p}\ y \wedge \overline{p}\ x$ respectively $\Theta \vdash_{\overline{T}} \mathtt{B}^*\ x\ z \wedge \overline{p}\ x \wedge \overline{p}\ z$. Applying the induction hypothesis for type B yields $\Delta \vdash_{\overline{T}} \mathtt{B}^*\ y\ x$ respectively $\Theta \vdash_{\overline{T}} \mathtt{B}^*\ x\ z$. So it remains to show that $\Delta \vdash_{\overline{T}} \overline{p}\ y \wedge \overline{p}\ x$ and $\Theta \vdash_{\overline{T}} \overline{p}\ x \wedge \overline{p}\ z$ respectively hold. We can show them using rule (∧I) given $\Delta \vdash_{\overline{T}} \overline{p}\ y$ and $\Delta \vdash_{\overline{T}} \overline{p}\ x$ respectively $\Theta \vdash_{\overline{T}} \overline{p}\ x$ and $\Theta \vdash_{\overline{T}} \overline{p}\ x$. Those statements follow from rule (assume) and the elimination rules of ∧.

This concludes the proof of (9) and (10).

We show (13) by induction on the grammar: Without loss of generality we may assume that $B =: B'|_p$ for B' either a quotient–, a base– or a Π-type. This is due to the fact that quotinet–, base– and Π-types B' can be written as $B'|_{\lambda x:B'.\ \mathsf{true}}$ and types of the form $B''|_p|_q$ can be rewritten as $B''|_{\lambda x:B''.\ p\ x \wedge q\ x}$.

If t is a constant or variable then $\overline{t}[x/x'] = \overline{t}$ and by case (PT6) resp. by case (PT4) in the definition of the translation, we have $\mathtt{A}^*\ \overline{t}\ \overline{t}$. So the claim holds.

If t is a λ-term $\lambda y{:}C.\ s$ and $B' = \Pi z{:}C.\ D$, then by induction hypothesis we have

$$\overline{\Gamma},\ x, x'{:}\overline{A},\ \triangleright\mathtt{A}^*\ x\ x' \vdash_{\overline{T}} \mathtt{D}^*\ \overline{s}\ \overline{s}[x/x'].$$

By the rules ($\forall$I), ($\Rightarrow$I), we yield

$$\overline{\Gamma} \vdash_{\overline{T}} \forall\, x, y{:}\overline{A}.\ \mathtt{A}^*\ x\ y \Rightarrow \mathtt{D}^*\ \overline{s}\ \overline{s}[x/x'].$$

By definition (PT11) this is exactly

$$\overline{\Gamma},\ x, x'{:}\overline{A},\ \triangleright\mathtt{A}^*\ x\ x' \vdash_{\overline{T}} \mathtt{B}'^*\ \overline{t}\ \overline{t}[x/x'].$$

Since t is a λ-term, by assumption we have that $B \equiv B' = B|_{\lambda z:B.\ \mathsf{true}}$, so the claim follows trivially.

If t is a function application $f\ s$ with f of type $\Pi z{:}C.\ D$ and s of type C, then by assumption $B = D \equiv B' = B|_{\lambda z:B.\ \mathsf{true}}$, so it suffices to prove that

$$\overline{\Gamma},\ x, x'{:}\overline{A},\ \triangleright\mathtt{A}^*\ x\ x' \vdash_{\overline{T}} \mathtt{D}^*\ \overline{f\ s}\ \overline{f\ s}[x/x'].$$

By induction hypothesis and (11) we then have:

$$\overline{\Gamma},\ x, x'{:}\overline{A},\ \triangleright\mathtt{A}^*\ x\ x' \vdash_{\overline{T}} (\Pi \mathsf{z}{:}\mathtt{C}.\ \mathtt{D})^*\ \overline{f}\ \overline{f}[x/x']$$

and

$$\overline{\Gamma},\ x, x'{:}\overline{A},\ \triangleright\mathtt{A}^*\ x\ x' \vdash_{\overline{T}} \mathtt{C}^*\ \overline{s}\ \overline{s}[x/x']. \tag{27}$$

By definition (PT11), we can unpack the former to:

$$\overline{\Gamma},\ x, x'{:}\overline{A},\ \triangleright\mathtt{A}^*\ x\ x' \vdash_{\overline{T}} \forall\, z, z'{:}\overline{C}.\ \mathtt{C}^*\ z\ z' \Rightarrow (\Pi \mathsf{z}{:}\mathtt{C}.\ \mathtt{D})^*\ \overline{f}\ z\ \overline{f}[x/x']\ z'[x/x'] \tag{28}$$

Using the rules ($\forall$E) and ($\Rightarrow$E) (using (28)) to plug in $\overline{s}$ resp. $\overline{s}[x/x']$ for z, z' in (28), we yield:

$$\overline{\Gamma},\ x, x'{:}\overline{A},\ \triangleright\mathtt{A}^*\ x\ x' \vdash_{\overline{T}} (\Pi \mathsf{z}{:}\mathtt{C}.\ \mathtt{D})^*\ \overline{f}\ \overline{s}\ \overline{f}[x/x']\ \overline{s}[x/x']$$

which is exactly the desired result.

By definition (PT13), the typing relation for type bool is ordinary equality, so the cases of t being an implication or Boolean equality are in fact special cases of (11), which is already proven above. It remains to consider the case of t being an equality $s =_C s'$ for $C \not\equiv \mathsf{bool}$. In this case, the induction hypothesis implies that

$$\overline{\Gamma},\ x, x'{:}\overline{A},\ \triangleright\mathtt{A}^*\ x\ x' \vdash_{\overline{T}} \mathtt{C}^*\ \overline{s}\ \overline{s}[x/x'] \tag{29}$$

and

$$\overline{\Gamma},\ x, x'{:}\overline{A},\ \triangleright\mathtt{A}^*\ x\ x' \vdash_{\overline{T}} \mathtt{C}^*\ \overline{s'}\ \overline{s'}[x/x'] \tag{30}$$

We need to prove

$$\overline{\Gamma},\ x, x'{:}\overline{A},\ \triangleright\mathtt{A}^*\ x\ x' \vdash_{\overline{T}} \mathtt{C}^*\ \overline{s}\ \overline{s'} =_{\mathsf{bool}} \mathtt{C}^*\ \overline{s}[x/x']\ \overline{s'}[x/x'].$$

If we can show

$$\overline{\Gamma},\ x, x'{:}\overline{A},\ \triangleright\mathtt{A}^*\ x\ x',\ \triangleright\mathtt{C}^*\ \overline{s}\ \overline{s'} \vdash_{\overline{T}} \mathtt{C}^*\ \overline{s}[x/x']\ \overline{s'}[x/x']$$

and similarly also

$$\overline{\Gamma},\ x, x'{:}\overline{A},\ \triangleright\mathtt{A}^*\ x\ x',\ \triangleright\mathtt{C}^*\ \overline{s}[x/x']\ \overline{s'}[x/x'] \vdash_{\overline{T}} \mathtt{C}^*\ \overline{s}\ \overline{s'},$$

then the claim follows by rule (propExt).

Both follows from the transitivity (10) of the typing relation $\mathtt{C}^*$.

C.2 Proof of remaining soundness theorem by induction on DHOL derivations

C.2.1 Well-formedness of theories

Well-formedness of DHOL theories can be shown using the rules (thyEmpty), (thyType'), (thyConst) and (thyAxiom):

(thyEmpty):

$\vdash \circ$ Thy	(thyEmpty)	(31)
$\vdash^{\mathsf{H}} \overline{\circ}$ Thy	(thyEmpty)	

(thyType'):

$\vdash_{\mathsf{T}} x_1{:}A_1, \ldots, x_n{:}A_n$ Ctx	by assumption	(32)
$\vdash_{\overline{T}} x_1{:}\overline{A_1}, \mathtt{A_1}^*\ x_1\ x_1, \ldots, x_n{:}\overline{A_n}, \triangleright \mathtt{A_n}^*\ x_n\ x_n$ Ctx	induction hypothesis,(32)	(33)
$\vdash^{\mathsf{H}} \overline{T}$ Thy	(ctxThy),(33)	(34)
$\vdash^{\mathsf{H}} \overline{T}$, a:tp Thy	(thyType),(34)	(35)
$\vdash^{\mathsf{H}} \overline{T,\ \mathtt{a}{:}\Pi x_1{:}A_1.\ \ldots \Pi x_n{:}A_n.\ \mathsf{tp}}$ Thy	PT2,(35)	

(thyConst):

$\vdash_{\mathsf{T}} A$ tp	by assumption	(36)
$\vdash_{\overline{T}} \overline{A}$ tp	induction hypothesis,(36)	(37)
$\vdash^{\mathsf{H}} \overline{T}$, c:$\overline{A}$ Thy	(thyConst),(37)	(38)
$\vdash^{\mathsf{H}} \overline{T,\ c{:}A}$ Thy	PT3,(38)	(39)

(thyAxiom):

$\vdash_{\mathsf{T}} F$:bool	by assumption	(40)
$\vdash_{\overline{T}} \overline{F}$:bool	induction hypothesis,(40)	(41)
$\vdash^{\mathsf{H}} \overline{T}$, $\triangleright\overline{F}$ Thy	(thyAxiom),(41)	(42)
$\vdash^{\mathsf{H}} \overline{T,\ \triangleright F}$ Thy	PT4,(42)	(43)

C.2.2 Well-formedness of contexts

Well-formedness of contexts can be concluded using the rules (ctxEmpty), (ctxVar) and (ctxAssume):

(ctxEmpty):

$\vdash T$ Thy	by assumption	(44)
$\vdash^{\mathsf{H}} \overline{T}$ Thy	induction hypothesis,(44)	(45)
$\vdash_{\overline{T}}$. Ctx	(ctxEmpty),(45)	(46)
$\vdash_{\overline{T}}$- Ctx	PT5,(46)	(47)

(ctxVar):

$\Gamma \vdash_{\mathsf{T}} A$ tp	by assumption	(48)
$\overline{\Gamma} \vdash_{\overline{T}} \overline{A}$ tp	induction hypothesis,(48)	(49)
$\overline{\Gamma} \vdash_{\overline{T}} \mathtt{A}^*{:}\overline{A} \to \overline{A} \to$ bool	induction hypothesis,(48)	(50)
$\vdash_{\overline{T}} \overline{\Gamma}, x{:}\overline{A}$ Ctx	(ctxVar),(49)	(51)
$\overline{\Gamma}, x{:}\overline{A} \vdash_{\overline{T}} \mathtt{A}^*{:}\overline{A} \to \overline{A} \to$ bool	(var$\vdash$),(49),(50)	(52)
$\overline{\Gamma}, x{:}\overline{A} \vdash_{\overline{T}} \mathtt{A}^*\ x\ x$:bool	(appl),(52),(varS)	(53)
$\vdash_{\overline{T}} \overline{\Gamma}, x{:}\overline{A}, \mathtt{A}^*\ x\ x$ Ctx	(ctxAssume),(53)	(54)
$\vdash_{\overline{T}} \overline{\Gamma, x{:}A}$ Ctx	PT6,(54)	(55)

(ctxAssume):

$\Gamma \vdash_{\mathsf{T}} F$:bool	by assumption	(56)
$\overline{\Gamma} \vdash_{\overline{T}} \overline{F}$:bool	induction hypothesis,(56)	(57)
$\vdash_{\overline{T}} \overline{\Gamma}, \triangleright \overline{F}$ Ctx	(ctxAssume),(57)	(58)
$\vdash_{\overline{T}} \overline{\Gamma, \triangleright F}$ Ctx	PT7,(58)	(59)

C.2.3 Well-formedness of types

Well-formedness of types can be shown in DHOL using the rules (type'), (bool), (pi), (Q) and ($|_p$ tp):

(type'):

$\mathsf{a}{:}\Pi x_1{:}A_1. \ \ldots \Pi x_n{:}A_n.$ tp in T	by assumption	(60)
$\Gamma \vdash_{\mathsf{T}} t_1{:}A_1$	by assumption	(61)
$\vdots$		
$\Gamma \vdash_{\mathsf{T}} t_n{:}A_n[x_1/t_1] \ldots [x_n/t_n]$	by assumption	(62)
$\overline{\Gamma} \vdash_{\overline{T}} \overline{t_1}{:}\overline{A_1}$	induction hypothesis,(61)	(63)
$\vdots$		
$\overline{\Gamma} \vdash_{\overline{T}} \overline{t_n}{:}\overline{A_n}$	induction hypothesis,(62)	(64)
a:tp in $\overline{T}$	PT2,(60)	(65)
$\mathsf{a}^*{:}\overline{A_1} \to \ldots \overline{A_n} \to \mathsf{a} \to \mathsf{a} \to$ bool in $\overline{T}$	PT2,(60)	(66)
$\overline{\Gamma} \vdash_{\overline{T}} \mathsf{a}^*{:}\overline{A_1} \to \ldots \overline{A_n} \to \mathsf{a} \to \mathsf{a} \to$ bool	(constS),(66)	(67)
$\overline{\Gamma} \vdash_{\overline{T}} \mathsf{a}^*\ \overline{t_1}{:}\overline{A_2} \to \ldots \overline{A_n} \to \mathsf{a} \to \mathsf{a} \to$ bool	(appl),(67),(63)	(68)
$\vdots$		
$\overline{\Gamma} \vdash_{\overline{T}} \mathsf{a}^*\ \overline{t_1}\ \ldots\ \overline{t_n}{:}\mathsf{a} \to \mathsf{a} \to$ bool	(appl),previous line,(64)	(69)
$\vdash_{\overline{T}} \overline{\Gamma}$ Ctx	(tpCtx),(typingTp),(67)	(70)
$\overline{\Gamma} \vdash_{\overline{T}} \mathsf{a}$ tp	(type),(65),(70)	

$\overline{\Gamma} \vdash_{\overline{T}} (\mathtt{a}\ \mathtt{t}_1\ \ldots\ \mathtt{t}_n)^* : \overline{\mathtt{a}} \to \overline{\mathtt{a}} \to \mathsf{bool}$ PT21,(69)

(bool):

$\vdash_{\mathsf{T}} \Gamma\ \mathsf{Ctx}$	by assumption	(71)
$\vdash_{\overline{T}} \overline{\Gamma}\ \mathsf{Ctx}$	induction hypothesis,(71)	(72)
$\vdash_{\overline{T}} \mathsf{bool}\ \mathsf{tp}$	(bool),(72)	(73)
$\vdash_{\overline{T}} \overline{\mathsf{bool}}\ \mathsf{tp}$	PT12,(73)	

bool^* is just a notation of $=_{\mathsf{bool}}$ which is of type $\mathsf{bool} \to \mathsf{bool} \to \mathsf{bool}$ in HOL, as desired.

(pi):

$\Gamma \vdash_{\mathsf{T}} A\ \mathsf{tp}$	by assumption	(74)
$\Gamma,\ x{:}A \vdash_{\mathsf{T}} B\ \mathsf{tp}$	by assumption	(75)
$\overline{\Gamma} \vdash_{\overline{T}} \overline{A}\ \mathsf{tp}$	induction hypothesis,(74)	(76)
$\overline{\Gamma},\ x{:}\overline{A},\ \triangleright\mathtt{A}^*\ x\ x \vdash_{\overline{T}} \overline{B}\ \mathsf{tp}$	induction hypothesis,(75)	(77)
$\overline{\Gamma} \vdash_{\overline{T}} \overline{B}\ \mathsf{tp}$	HOL types context independent,(77)	(78)
$\overline{\Gamma} \vdash_{\overline{T}} \overline{A} \to \overline{B}\ \mathsf{tp}$	(arrow),(76),(78)	(79)
$\overline{\Gamma} \vdash_{\overline{T}} \mathtt{A}^* : \overline{A} \to \overline{A} \to \mathsf{bool}$	induction hypothesis,(74)	(80)
$\overline{\Gamma} \vdash_{\overline{T}} \mathtt{B}^* : \overline{B} \to \overline{B} \to \mathsf{bool}$	induction hypothesis,(75)	(81)
$\overline{\Gamma} \vdash_{\overline{T}} \overline{\Pi x{:}A.\ B}\ \mathsf{tp}$	PT10,(79)	
$\overline{\Gamma} \vdash_{\overline{T}} (\Pi \mathtt{x}{:}\mathtt{A}.\ \mathtt{B})^* : (\overline{\Pi x{:}A.\ B}) \to (\overline{\Pi x{:}A.\ B}) \to \mathsf{bool}$	PT11,(80),(81)	

(Q):

$\Gamma \vdash_{\mathsf{T}} A\ \mathsf{tp}$	by assumption	(82)
$\Gamma \vdash_{\mathsf{T}} r : \Pi x_1{:}A.\ \Pi x_2{:}A.\ \mathsf{bool}$	by assumption	(83)
$\overline{\Gamma} \vdash_{\overline{T}} \overline{A}\ \mathsf{tp}$	induction hypothesis,(82)	(84)
$\overline{\Gamma} \vdash_{\overline{T}} \mathtt{A}^* : \overline{A} \to \overline{A} \to \mathsf{bool}$	induction hypothesis,(82)	(85)
$\overline{\Gamma} \vdash_{\overline{T}} \overline{r} : \overline{A} \to \overline{A} \to \mathsf{bool}$	induction hypothesis,(83)	(86)
$\overline{\Gamma} \vdash_{\overline{T}} \overline{A/r}\ \mathsf{tp}$	PT16,(84)	
$\overline{\Gamma},\ x,y{:}\overline{A} \vdash_{\overline{T}} \mathtt{A}^*\ x\ x : \mathsf{bool}$	(appl),(appl),(85),(var),(var)	(87)
$\overline{\Gamma},\ x,y{:}\overline{A} \vdash_{\overline{T}} \mathtt{A}^*\ y\ y : \mathsf{bool}$	(appl),(appl),(85),(var),(var)	(88)
$\overline{\Gamma},\ x,y{:}\overline{A} \vdash_{\overline{T}} \overline{r}\ x\ x : \mathsf{bool}$	(appl),(appl),(86),(var),(var)	(89)
$\overline{\Gamma},\ x,y{:}\overline{A} \vdash_{\overline{T}} \overline{r}\ x\ y \wedge \mathtt{A}^*\ x\ x \wedge \mathtt{A}^*\ y\ y : \mathsf{bool}$	(∧),(89),(∧),(87),(88)	(90)
$\overline{\Gamma} \vdash_{\overline{T}} \lambda x,y{:}\overline{A}.\ \overline{r}\ x\ y \wedge \mathtt{A}^*\ x\ x \wedge \mathtt{A}^*\ y\ y : \overline{A} \to \overline{A} \to \mathsf{bool}$		
	(lambda),(lambda),(90)	(91)
$\overline{\Gamma} \vdash_{\overline{T}} (\mathtt{A}/\mathtt{r})^* : (\overline{A/r}) \to (\overline{A/r}) \to \mathsf{bool}$	PT16,PT17,(91)	

($|_p$ tp):

$\Gamma \vdash_{\mathsf{T}} p : \Pi x{:}A.\ \mathsf{bool}$	by assumption	(92)

$\overline{\Gamma} \vdash_{\overline{T}} \overline{p} : \overline{A} \to \mathsf{bool}$ induction hypothesis,(92) (93)

$\overline{\Gamma} \vdash_{\overline{T}} \overline{A} \to \mathsf{bool}\ \mathsf{tp}$ (typingTp),(93) (94)

Since statements of shape $\vdash B \to C\ \mathsf{tp}$ only provable using rule (arrow):

$\overline{\Gamma} \vdash_{\overline{T}} \overline{A}\ \mathsf{tp}$ see above,(94) (95)

$\overline{\Gamma} \vdash_{\overline{T}} \overline{A|_p}\ \mathsf{tp}$ PT14,(95)

$\overline{\Gamma} \vdash_{\overline{T}} \forall\, x, y{:}\overline{A}.\ \mathtt{A}^*\ x\ y \Rightarrow \overline{p}\ x =_{\mathsf{bool}} \overline{p}\ y$ induction hypothesis,PT11,(92) (96)

$\overline{\Gamma},\ x, y{:}\overline{A} \vdash_{\overline{T}} \mathtt{A}^*\ x\ y \Rightarrow \overline{p}\ x =_{\mathsf{bool}} \overline{p}\ y$ (monotonic$\vdash$),(∀E),(monotonic$\vdash$), (∀E),(monotonic$\vdash$),(96),(var),(var) (97)

$\overline{\Gamma},\ x, y{:}\overline{A} \vdash_{\overline{T}} \mathtt{A}^*\ x\ y{:}\mathsf{bool}$ (implTypingL),(97) (98)

$\overline{\Gamma},\ x, y{:}\overline{A} \vdash_{\overline{T}} \mathtt{A}^* : \overline{A} \to \overline{A} \to \mathsf{bool}$ (applType),(var),(applType),(var),(98) (99)

Since x, y don't occur in $\mathtt{A}^*$ and HOL types are context independent:

$\overline{\Gamma} \vdash_{\overline{T}} \mathtt{A}^* : \overline{A} \to \overline{A} \to \mathsf{bool}$ see above,(99) (100)

$\overline{\Gamma} \vdash_{\overline{T}} (\mathtt{A}|_{\mathtt{p}})^* : \overline{A} \to \overline{A} \to \mathsf{bool}$ PT15,(100)

C.2.4 Type-equality

Type-equality can be shown using the rules (congBase'),(STantisym), (congΠ) and ($\equiv$ bool): Observe that by the rules (var$\vdash$), (congAppl), (var), instead of proving $\overline{\Gamma},\ x, y{:}\overline{A} \vdash_{\overline{T}} \mathtt{A}^*\ x\ y =_{\mathsf{bool}} \mathtt{A}^*\ x\ y$ we may simply prove $\overline{\Gamma} \vdash_{\overline{T}} \mathtt{A}^* =_{\overline{A} \to \overline{A} \to \mathsf{bool}} \mathtt{A'}^*\ x\ y$.

(congBase'):

$\mathtt{a} : \Pi x_1{:}A_1.\ \ldots \Pi x_n{:}A_n.\ \mathsf{tp}$ in T by assumption (101)

$\Gamma \vdash_{\mathsf{T}} s_1 =_{A_1} t_1$ by assumption (102)

$\vdots$

$\Gamma \vdash_{\mathsf{T}} s_n =_{A_n[x_1/t_1]\ldots[x_{n-1}/t_{n-1}]} t_n$ by assumption (103)

$\mathtt{a} : \mathsf{tp}$ in $\overline{T}$ PT2,(101) (104)

$\mathtt{a}^* : \overline{A_1} \to \ldots \to \overline{A_n} \to \overline{\mathtt{a}} \to \overline{\mathtt{a}} \to \mathsf{bool}$ in $\overline{T}$ PT2,(101) (105)

$\overline{\Gamma} \vdash_{\overline{T}} \overline{s_1} =_{\overline{A_1}} \overline{t_1}$ induction hypothesis,(102) (106)

$\vdots$

$\overline{\Gamma} \vdash_{\overline{T}} \overline{s_n} =_{\overline{A_n}} \overline{t_n}$ induction hypothesis,(103) (107)

$\vdash_{\overline{T}} \overline{\Gamma}\ \mathsf{Ctx}$ (tpCtx),(typingTp),(eqTyping),(106) (108)

$\overline{\Gamma} \vdash_{\overline{T}} \mathtt{a} : \mathsf{tp}$ (type),(104),(108) (109)

$\overline{\Gamma} \vdash_{\overline{T}} \mathtt{a} \equiv \mathtt{a}$ (congBase),(109) (110)

$\overline{\Gamma} \vdash_{\overline{T}} \mathtt{a}^* =_{\overline{A_1} \to \ldots \to \overline{A_n} \to \overline{\mathtt{a}} \to \overline{\mathtt{a}} \to \mathsf{bool}} \mathtt{a}^*$ (refl),(constS),(105),(108) (111)

$\overline{\Gamma} \vdash_{\overline{T}} \mathtt{a}^*\ \overline{s_1} =_{\overline{A_2} \to \ldots \to \overline{A_n} \to \overline{\mathtt{a}} \to \overline{\mathtt{a}} \to \mathsf{bool}} \mathtt{a}^*\ \overline{t_1}$ (congAppl),(106),(111) (112)

$\vdots$

$\overline{\Gamma} \vdash_{\overline{T}} \mathtt{a}^*\ \overline{s_1}\ \ldots\ \overline{s_n} =_{\overline{\mathtt{a}} \to \overline{\mathtt{a}} \to \mathsf{bool}} \mathtt{a}^*\ \overline{t_1}\ \ldots\ \overline{t_n}$ (congAppl),(107),previous line (113)

$\overline{\Gamma} \vdash_{\overline{T}} \overline{\mathtt{a}\ s_1\ \ldots\ s_n} \equiv \overline{\mathtt{a}\ t_1\ \ldots\ t_n}$ PT8,(110)

$\overline{\Gamma} \vdash_{\overline{T}} \vdash_{\overline{T}} (\mathtt{a}\ \mathtt{s_1}\ \ldots\ \mathtt{s_n})^* =_{\overline{\mathtt{a}} \to \overline{\mathtt{a}} \to \mathsf{bool}} (\mathtt{a}\ \mathtt{t_1}\ \ldots\ \mathtt{t_n})^*$ PT9,(113)

(STantisym):

$\Gamma \vdash_{\mathsf{T}} A \prec: A'$	by assumption	(114)
$\Gamma \vdash_{\mathsf{T}} A' \prec: A$	by assumption	(115)
$\overline{\Gamma} \vdash_{\overline{T}} \overline{A} \equiv \overline{A'}$	induction hypothesis,(114)	(116)
$\Gamma,\ x,y{:}\overline{A} \vdash_{\overline{T}} \mathtt{A}^*\ x\ y \Rightarrow \mathtt{A'}^*\ x\ y$	induction hypothesis,(114)	(117)
$\overline{\Gamma},\ x{:}\overline{A} \vdash_{\overline{T}} \mathtt{A}^*\ x\ x \Rightarrow \mathtt{A'}^*\ x\ x$	(∀E),(∀I),(117),(var)	(118)
$\Gamma,\ x,y{:}\overline{A'} \vdash_{\overline{T}} \mathtt{A'}^*\ x\ y \Rightarrow \mathtt{A}^*\ x\ y$	induction hypothesis,(115)	(119)
$\overline{\Gamma},\ x{:}\overline{A'} \vdash_{\overline{T}} \mathtt{A'}^*\ x\ x \Rightarrow \mathtt{A}^*\ x\ x$	(∀E),(∀I),(119),(var)	(120)
$\overline{\Gamma} \vdash_{\overline{T}} \forall x{:}\overline{A}.\ \mathtt{A'}^*\ x\ x \Rightarrow \mathtt{A}^*\ x\ x$	(cong$\vdash$),(∀cong),(≡trans),(116),(refl),(∀I),(120)	(121)
$\overline{\Gamma},\ x{:}\overline{A} \vdash_{\overline{T}} \mathtt{A'}^*\ x\ x \Rightarrow \mathtt{A}^*\ x\ x$	(∀E),(var$\vdash$),(121),(var)	(122)
$\overline{\Gamma},\ x{:}\overline{A} \vdash_{\overline{T}} \mathtt{A}^*\ x\ x =_{\mathsf{bool}} \mathtt{A'}^*\ x\ x$	(propExt),(118),(122)	

(congΠ):

$\Gamma \vdash_{\mathsf{T}} A \equiv A'$	by assumption	(123)
$\Gamma,\ x{:}A \vdash_{\mathsf{T}} B \equiv B'$	by assumption	(124)
$\overline{\Gamma} \vdash_{\overline{T}} \overline{A} \equiv \overline{A'}$	induction hypothesis,(123)	(125)
$\overline{\Gamma} \vdash_{\overline{T}} \mathtt{A}^* =_{\overline{A}\to\overline{A}\to\mathsf{bool}} \mathtt{A'}^*$	induction hypothesis,(123)	(126)
$\overline{\Gamma},\ x{:}\overline{A},\ \triangleright\mathtt{A}^*\ x\ x \vdash_{\overline{T}} \overline{B} \equiv \overline{B'}$	induction hypothesis,(124)	(127)

Since $\equiv$ is context independent in HOL:

$\overline{\Gamma} \vdash_{\overline{T}} \overline{B} \equiv \overline{B'}$	explanation,(127)	(128)
$\overline{\Gamma} \vdash_{\overline{T}} \overline{A} \to \overline{B} \equiv \overline{A'} \to \overline{B'}$	(cong→),(125),(128)	(129)
$\overline{\Gamma} \vdash_{\overline{T}} \overline{\Pi x{:}A.\ B} \equiv \overline{\Pi x{:}A'.\ B'}$	PT10,(129)	
$\overline{\Gamma},\ x{:}\overline{A},\ \triangleright\mathtt{A}^*\ x\ x \vdash_{\overline{T}} \mathtt{B}^* =_{\overline{B}\to\overline{B}\to\mathsf{bool}} \mathtt{B'}^*$	induction hypothesis,(124)	(130)
$\overline{\Gamma},\ f{:}\overline{A}\to\overline{B},\ x{:}\overline{A} \vdash_{\overline{T}} \mathtt{A}^*\ x\ x \Rightarrow \mathtt{B}^*\ (f\ x)\ (f\ x)$		
$=_{\mathsf{bool}} \mathtt{A'}^*\ x\ x \Rightarrow \mathtt{B}^*\ (f\ x)\ (f\ x)$	(rewrite),(refl),(126)	(131)
$\overline{\Gamma},\ f{:}\overline{A}\to\overline{B},\ x{:}\overline{A} \vdash_{\overline{T}} \mathtt{A}^*\ x\ x \Rightarrow \mathtt{B}^*\ (f\ x)\ (f\ x)$		
$=_{\mathsf{bool}} \mathtt{A'}^*\ x\ x \Rightarrow \mathtt{B'}^*\ (f\ x)\ (f\ x)$	(rewrite),(131),(130)	(132)
$\overline{\Gamma},\ f{:}\overline{A}\to\overline{B} \vdash_{\overline{T}} \forall x{:}\overline{A}.\ \mathtt{A}^*\ x\ x \Rightarrow (\mathtt{B}^*\ (f\ x)\ (f\ x)) =_{\mathsf{bool}}$		
$\forall x{:}\overline{A'}.\ \mathtt{A'}^*\ x\ x \Rightarrow (\mathtt{B'}^*\ (f\ x)\ (f\ x))$	(∀cong),(125),(132)	(133)
$\overline{\Gamma} \vdash_{\overline{T}} (\Pi \mathtt{x}{:}\mathtt{A}.\ \mathtt{B})^* =_{\overline{A}\to\overline{A}\to\mathsf{bool}} \left(\Pi \mathtt{x}{:}\mathtt{A'}.\ \mathtt{B'}\right)^*$	PT20,(congλ),(133)	

($\equiv$ bool):

$\vdash_{\mathsf{T}} \Gamma\ \mathsf{Ctx}$	by assumption	(134)
$\vdash_{\overline{T}} \overline{\Gamma}\ \mathsf{Ctx}$	induction hypothesis,(134)	(135)
$\overline{\Gamma} \vdash_{\overline{T}} \mathsf{bool}\ \mathsf{tp}$	(congBase),(bool),(135)	

bool^* is a well-typed relation on bool by definition.

C.2.5 Subtyping

Subtyping can be shown using the axiom (16).

907 **(16):**

908 We need to check that the translation of axiom (16) holds in HOL. (16) states that whenever either
909 side is well-formed, we have:

910 $$\vdash \Pi x{:}A.\ {}^{B}\!/_{r} \prec: \overline{(\Pi x{:}A.\ B)/\lambda f,g{:}\Pi x{:}A.\ B.\ \forall x{:}A.\ r\ (f\ x)\ (g\ x)}$$

If either side is well-formed it follows that A, B are well-formed and r is equivalence relation on A. We then need to prove that $\Pi x{:}A.\ {}^{B}\!/_{r} \equiv \overline{(\Pi x{:}A.\ B)/\lambda f,g{:}\Pi x{:}A.\ B.\ \forall x{:}A.\ r\ (f\ x)\ (g\ x)}$ holds, which is immediate from the definition of the translation (both sides are just $\overline{A} \to \overline{B}$) and that in a context containing $x, y{:}\overline{B}$ we have

$$(\Pi \mathtt{x}{:}\mathtt{A}.\ {}^{\mathtt{B}}\!/_{\mathtt{r}})^{*}\ x\ y \Rightarrow (\overline{(\Pi \mathtt{x}{:}\mathtt{A}.\ \mathtt{B})/\lambda \mathtt{f},\mathtt{g}{:}\Pi \mathtt{x}{:}\mathtt{A}.\ \mathtt{B}.\ \forall \mathtt{x}{:}\mathtt{A}.\ \mathtt{r}\ (\mathtt{f}\ \mathtt{x})\ (\mathtt{g}\ \mathtt{x})})^{*}\ x\ y.$$

911 However, we have already shown in Example 19 that both PER applications reduce to the same
912 formula, so the implication must be valid in HOL.

913 C.2.6 Typing

914 Typing can be shown using the rules (const'), (var'), (quotE), (lambda'), (appl'), (⇒type'), (=type),
915 ($|_p$ I), ($|_p$ E1), (QI):

916 **(const'):**

917 $\mathtt{c}{:}A'$ in T	by assumption	(136)
918 $\Gamma \vdash_{\mathsf{T}} A' \equiv A$	by assumption	(137)
919 $\mathtt{c}{:}\overline{A'}$ in $\overline{T}$	PT3,(136)	(138)
920 $\triangleright \mathtt{A}'^{*}\ \mathtt{c}\ \mathtt{c}$ in $\overline{T}$	PT3,(136)	(139)
921 $\overline{\Gamma} \vdash_{\overline{T}} \overline{A'} \equiv \overline{A}$	induction hypothesis,(137)	(140)
922 $\overline{\Gamma}, x{:}\overline{A'} \vdash_{\overline{T}} \mathtt{A}'^{*}\ x\ x =_{\mathsf{bool}} \mathtt{A}^{*}\ x\ x$	induction hypothesis,(137)	(141)
923 $\overline{\Gamma} \vdash_{\overline{T}} \forall x{:}\overline{A}.\ \mathtt{A}'^{*}\ x\ x =_{\mathsf{bool}} \mathtt{A}^{*}\ x\ x$	(∀I),(∀I),(141)	(142)
924 $\overline{\Gamma} \vdash_{\overline{T}} \mathtt{c}{:}\overline{A}$	(const),(138),(140)	(143)
925 $\overline{\Gamma} \vdash_{\overline{T}} \overline{\mathtt{c}}{:}\overline{A}$	PT3,(143)	
926 $\overline{\Gamma} \vdash_{\overline{T}} \mathtt{A}'^{*}\ \overline{\mathtt{c}}\ \overline{\mathtt{c}}$	PT3,(axiom),(139)	(144)
927 $\overline{\Gamma} \vdash_{\overline{T}} \mathtt{A}^{*}\ \overline{\mathtt{c}}\ \overline{\mathtt{c}}$	(cong$\vdash$),(∀E),(142),(144)	

928 **(var'):**

929 $x{:}A'$ in Γ	by assumption	(145)
930 $\Gamma \vdash_{\mathsf{T}} A' \equiv A$	by assumption	(146)
931 $x{:}\overline{A'}$ in $\overline{\Gamma}$	PT3,(145)	(147)
932 $\triangleright \mathtt{A}'^{*}\ x\ x$ in $\overline{\Gamma}$	PT3,(145)	(148)
933 $\overline{\Gamma} \vdash_{\overline{T}} \overline{A'} \equiv \overline{A}$	induction hypothesis,(146)	(149)
934 $\overline{\Gamma}, x{:}\overline{A'} \vdash_{\overline{T}} \mathtt{A}'^{*}\ x\ x =_{\mathsf{bool}} \mathtt{A}^{*}\ x\ x$	induction hypothesis,(146)	(150)
935 $\overline{\Gamma} \vdash_{\overline{T}} \forall x{:}\overline{A}.\ \mathtt{A}'^{*}\ x\ x =_{\mathsf{bool}} \mathtt{A}^{*}\ x\ x$	(∀I),(150)	(151)
936 $\overline{\Gamma} \vdash_{\overline{T}} x{:}\overline{A}$	(var),(147),(149)	(152)
937 $\overline{\Gamma} \vdash_{\overline{T}} \overline{x}{:}\overline{A}$	PT3,(152)	
938 $\overline{\Gamma} \vdash_{\overline{T}} \mathtt{A}'^{*}\ \overline{x}\ \overline{x}$	PT3,(assume),(148)	(153)
939 $\overline{\Gamma} \vdash_{\overline{T}} \mathtt{A}^{*}\ \overline{x}\ \overline{x}$	(cong$\vdash$),(∀E),(151),(153)	

(quotE):

$$\Gamma \vdash_\mathsf{T} s{:}A/_r \quad \text{by assumption} \quad (154)$$

$$\Gamma,\ x{:}A,\ \triangleright x =_{A/_r} s \vdash_\mathsf{T} t{:}B \quad \text{by assumption} \quad (155)$$

$$\Gamma,\ x{:}A,\ x'{:}A,\ \triangleright x =_{A/_r} s,\ \triangleright x' =_{A/_r} s \vdash_\mathsf{T} t = _B\ t[x/x'] \quad \text{by assumption} \quad (156)$$

$$\overline{\Gamma} \vdash_{\overline{T}} \overline{s}{:}\overline{A} \quad \text{induction hypothesis,(154)} \quad (157)$$

$$\overline{\Gamma},\ x{:}\overline{A},\ \triangleright \mathtt{A}^*\ x\ x,\ \triangleright (\mathtt{A}/\mathtt{r})^*\ \overline{x}\ \overline{s} \vdash_{\overline{T}} \overline{t}{:}\overline{B} \quad \text{induction hypothesis,(155)} \quad (158)$$

$$\overline{\Gamma},\ x{:}A,\ \triangleright \mathtt{A}^*\ x\ x,\ x'{:}A,\ \triangleright \mathtt{A}^*\ x'\ x',\ \triangleright (\mathtt{A}/\mathtt{r})^*\ x\ \overline{s},\ \triangleright (\mathtt{A}/\mathtt{r})^*\ x'\ \overline{s} \vdash_{\overline{T}} \mathtt{B}^*\ \overline{t}\ \overline{t}[x/x'] \quad \text{induction hypothesis,(156)} \quad (159)$$

Since typing is context independent in HOL:

$$\overline{\Gamma},\ x{:}\overline{A} \vdash_{\overline{T}} \overline{t}{:}\overline{B} \quad \text{explanation,(158)} \quad (160)$$

$$\overline{\Gamma} \vdash_{\overline{T}} \overline{t}[x/\overline{s}]{:}\overline{B} \quad \text{(rewriteTyping),(160),(157)} \quad (161)$$

$$\overline{\Gamma} \vdash_{\overline{T}} \overline{t}[x/\overline{s}]{:}\overline{B[x/s]} \quad \text{HOL types are simple,(161)}$$

Since $\mathtt{B}^*$ is transitive we can simplify (159) to:

$$\overline{\Gamma},\ x{:}A,\ \triangleright \mathtt{A}^*\ x\ x,\ \triangleright (\mathtt{A}/\mathtt{r})^*\ x\ \overline{s} \vdash_{\overline{T}} \mathtt{B}^*\ \overline{t}\ \overline{t}[x/\overline{s}] \quad \text{explanation,(159)} \quad (162)$$

By symmetry and transitivity of $\mathtt{B}^*$, we yield also $\mathtt{B}^*\ \overline{t}[x/\overline{s}]\ \overline{t}[x/\overline{s}]$ in the same context. Since this formula no longer depends on x and an (known to be well-typed) equality assumption with an otherwise unused variable on one side is not useful for proving in HOL, the same must also be derivable in context $\overline{\Gamma}$.

$$\overline{\Gamma} \vdash_{\overline{T}} \mathtt{B}^*\ \overline{t}[x/\overline{s}]\ \overline{t}[x/\overline{s}] \quad \text{explanation,(162)}$$

(lambda'):

$$\Gamma,\ x{:}\overline{A} \vdash_\mathsf{T} t{:}B \quad \text{by assumption} \quad (163)$$

$$\Gamma \vdash_\mathsf{T} A \equiv A' \quad \text{by assumption} \quad (164)$$

$$\Gamma,\ x{:}\overline{A},\ \triangleright \mathtt{A}^*\ x\ x \vdash_{\overline{T}} \overline{t}{:}\overline{B} \quad \text{induction hypothesis,PT6,(163)} \quad (165)$$

$$\overline{\Gamma} \vdash_{\overline{T}} \overline{A} \equiv \overline{A'} \quad \text{induction hypothesis,(164)} \quad (166)$$

$$\overline{\Gamma} \vdash_{\overline{T}} \mathtt{A}^* =_{\overline{A}\to\overline{A}\to\mathsf{bool}} \mathtt{A'}^* \quad \text{induction hypothesis,(164)} \quad (167)$$

$$\Gamma,\ x{:}\overline{A},\ \triangleright \mathtt{A}^*\ x\ x \vdash_{\overline{T}} \mathtt{B}^*\ \overline{t}\ \overline{t} \quad \text{induction hypothesis,PT6,(163)} \quad (168)$$

$$\Gamma,\ x,y{:}\overline{A},\ \triangleright \mathtt{A}^*\ x\ y \vdash_{\overline{T}} \mathtt{B}^*\ \overline{t}\ \overline{t}[x/y] \quad (13),(168) \quad (169)$$

$$\Gamma \vdash_\mathsf{T} \forall x,y{:}\overline{A}.\ \mathtt{A}^*\ x\ y \Rightarrow \mathtt{B}^*\ \overline{t}\ \overline{t}[x/y] \quad (\forall\mathrm{I}),(\Rightarrow\mathrm{I}),(169) \quad (170)$$

$$\Gamma \vdash_\mathsf{T} \forall x,y{:}\overline{A}.\ \mathtt{A'}^*\ x\ y \Rightarrow \mathtt{B}^*\ \overline{t}\ \overline{t}[x/y] \quad \text{(rewrite),(170),(167)} \quad (171)$$

$$\Gamma,\ x{:}A \vdash_{\overline{T}} \overline{t}{:}\overline{B} \quad \text{typing independent of assumptions,(165)} \quad (172)$$

$$\overline{\Gamma} \vdash_{\overline{T}} (\lambda x{:}\overline{A}.\ \overline{t}){:}\overline{A} \to \overline{B} \quad \text{(lambda),(172)} \quad (173)$$

Since in HOL equal types are necessarily identical, it follows:

$$\overline{\Gamma} \vdash_{\overline{T}} (\lambda x{:}\overline{A}.\ \overline{t}){:}\overline{A'} \to \overline{B} \quad \text{explanation,(173),(166)} \quad (174)$$

$$\overline{\Gamma} \vdash_{\overline{T}} \overline{\lambda x{:}A.\ t}{:}\overline{\Pi x{:}A'.\ B} \quad \text{PT20,PT10,(174)}$$

$$\overline{\Gamma} \vdash_{\overline{T}} (\Pi \mathtt{x}{:}\mathtt{A'}.\ \mathtt{B})^* \quad \overline{\lambda x{:}A.\ t} \quad \text{PT11,(171)}$$

(appl'):

$\Gamma \vdash_\mathsf{T} f : \Pi x{:}A.\ B$	by assumption	(175)
$\Gamma \vdash_\mathsf{T} t : A$	by assumption	(176)
$\overline{\Gamma} \vdash_{\overline{T}} \overline{f} : \overline{A} \to \overline{B}$	induction hypothesis,PT10,(175)	(177)
$\overline{\Gamma} \vdash_{\overline{T}} (\Pi \mathtt{x{:}A.\ B})^*\ \overline{f}\ \overline{f}$	induction hypothesis,PT10,(175)	(178)
$\overline{\Gamma} \vdash_{\overline{T}} \forall x{:}\overline{A}.\ \forall y{:}\overline{A}.$		
$\quad \mathtt{A}^*\ x\ y \Rightarrow \mathtt{B}^*\ (\overline{f}\ x)\ (\overline{f}\ y)$	PT11,(178)	(179)
$\overline{\Gamma} \vdash_{\overline{T}} \overline{t} : \overline{A}$	induction hypothesis,(176)	(180)
$\overline{\Gamma} \vdash_{\overline{T}} \mathtt{A}^*\ \overline{t}\ \overline{t}$	induction hypothesis,(176)	(181)
$\overline{\Gamma} \vdash_{\overline{T}} \mathtt{A}^*\ \overline{t}\ \overline{t} \Rightarrow \mathtt{B}^*\ (\overline{f}\ \overline{t})\ (\overline{f}\ \overline{t})$	(∀E),(∀E),(179),(180),(180)	(182)
$\overline{\Gamma} \vdash_{\overline{T}} \mathtt{B}^*\ (\overline{f}\ \overline{t})\ (\overline{f}\ \overline{t})$	(⇒E),(182),(181)	(183)
$\overline{\Gamma} \vdash_{\overline{T}} \overline{f}\ \overline{t} : \overline{B}$	(appl),(177),(180)	(184)
$\overline{\Gamma} \vdash_{\overline{T}} \overline{f\ t} : \overline{B}$	PT21,(184)	
$\overline{\Gamma} \vdash_{\overline{T}} \mathtt{B}^*\ \overline{f\ t}\ \overline{f\ t}$	PT21,(182)	

(⇒type'):

$\Gamma \vdash_\mathsf{T} F : \mathsf{bool}$	by assumption	(185)
$\Gamma,\ \triangleright F \vdash_\mathsf{T} G : \mathsf{bool}$	by assumption	(186)
$\overline{\Gamma} \vdash_{\overline{T}} \overline{F} : \mathsf{bool}$	induction hypothesis,(185)	(187)
$\overline{\Gamma},\ \overline{F} \vdash_{\overline{T}} \overline{G} : \mathsf{bool}$	induction hypothesis,(186)	(188)
$\overline{\Gamma} \vdash_{\overline{T}} \overline{G} : \mathsf{bool}$	typing is independent of assumptions,(188)	(189)
$\overline{\Gamma} \vdash_{\overline{T}} \overline{F} \Rightarrow \overline{G} : \mathsf{bool}$	(⇒type),(187),(189)	(190)
$\overline{\Gamma} \vdash_{\overline{T}} \overline{F \Rightarrow G} : \mathsf{bool}$	PT22,(190)	(191)
$\overline{\Gamma} \vdash_{\overline{T}} \mathsf{bool}^*\ \overline{F \Rightarrow G}\ \overline{F \Rightarrow G}$	(PT13),(refl),(191)	

(=type):

$\Gamma \vdash_\mathsf{T} s : A$	by assumption	(192)
$\Gamma \vdash_\mathsf{T} t : A$	by assumption	(193)
$\overline{\Gamma} \vdash_{\overline{T}} \overline{s} : \overline{A}$	induction hypothesis,(192)	(194)
$\overline{\Gamma} \vdash_{\overline{T}} \overline{t} : \overline{A}$	induction hypothesis,(193)	(195)
$\overline{\Gamma} \vdash_{\overline{T}} \mathtt{A}^*\ \overline{s}\ \overline{s}$	induction hypothesis,(192)	(196)
$\overline{\Gamma} \vdash_{\overline{T}} \mathtt{A}^*\ \overline{s}\ \overline{s} : \mathsf{bool}$	(validTyping),(196)	(197)
$\overline{\Gamma} \vdash_{\overline{T}} \mathtt{A}^* : \overline{A} \to \overline{A} \to \mathsf{bool}$	(applType),(196),(applType),(196),(197)	(198)
$\overline{\Gamma} \vdash_{\overline{T}} \mathtt{A}^*\ \overline{s}\ \overline{t} : \mathsf{bool}$	(appl),(appl),(198),(194),(195)	(199)
$\overline{\Gamma} \vdash_{\overline{T}} \mathsf{bool}^*\ (\mathtt{A}^*\ \overline{s}\ \overline{t})\ (\mathtt{A}^*\ \overline{s}\ \overline{t})$	(PT13),(refl),(199)	

($|_p$ I):

$\Gamma \vdash_\mathsf{T} t : A$	by assumption	(200)
$\Gamma \vdash_\mathsf{T} p\ t$	by assumption	(201)

$\overline{\Gamma} \vdash_{\overline{T}} \overline{t}:\overline{A}$	induction hypothesis,(200)	(202)
$\overline{\Gamma} \vdash_{\overline{T}} \mathtt{A}^* \ \overline{t} \ \overline{t}$	induction hypothesis,(200)	(203)
$\overline{\Gamma} \vdash_{\overline{T}} \overline{p} \ \overline{t}$	induction hypothesis,(201)	(204)
$\overline{\Gamma} \vdash_{\overline{T}} (\mathtt{A}\vert_{\mathtt{p}})^* \ \overline{t} \ \overline{t}$	PT15,(∧I),(203),(∧I),(204),(204)	
$\overline{\Gamma} \vdash_{\overline{T}} \overline{t}:\overline{A\vert_p}$	PT14,(202)	

($\vert_p$ E1):

$\Gamma \vdash_{\mathsf{T}} t:A\vert_p$	by assumption	(205)
$\overline{\Gamma} \vdash_{\overline{T}} \overline{t}:\overline{A}$	induction hypothesis,(205)	
$\overline{\Gamma} \vdash_{\overline{T}} \mathtt{A}^* \ \overline{t} \ \overline{t} \wedge \overline{p} \ \overline{t}$	induction hypothesis,(205)	(206)
$\overline{\Gamma} \vdash_{\overline{T}} \mathtt{A}^* \ \overline{t} \ \overline{t}$	(∧El),(206)	

(QI):

$\Gamma \vdash_{\mathsf{T}} t:A$	by assumption	(207)
$\Gamma \vdash_{\mathsf{T}} \mathrm{EqRel}(r)$	by assumption	(208)
$\overline{\Gamma} \vdash_{\overline{T}} \overline{t}:\overline{A}$	induction hypothesis,(207)	(209)
$\overline{\Gamma} \vdash_{\overline{T}} \overline{t}:\overline{(A/r)}$	PT15,(209)	
$\overline{\Gamma} \vdash_{\overline{T}} \mathtt{A}^* \ \overline{t} \ \overline{t}$	induction hypothesis,(207)	(210)

As shown as Subsection C.1 (208) implies that $\overline{r}$ is an equivalence on terms x satisfying $\mathtt{A}^* \ x \ x$. It follows that $\overline{r} \ \overline{t} \ \overline{t}$ holds.

$\overline{\Gamma} \vdash_{\overline{T}} \overline{r} \ \overline{t} \ \overline{t}$	explanation,(210)	(211)
$\overline{\Gamma} \vdash_{\overline{T}} \overline{r} \ \overline{t} \ \overline{t} \wedge \mathtt{A}^* \ \overline{t} \ \overline{t} \wedge \mathtt{A}^* \ \overline{t} \ \overline{t}$	definition of ∧,(211),(210)	(212)
$\overline{\Gamma} \vdash_{\overline{T}} (\mathtt{A}/\mathtt{r})^* \ \overline{t} \ \overline{t}$	PT17,(212)	

C.2.7 Term equality

Fix a context. By rule (rewrite), if we can show for two DHOL terms $s,t:A$ that $\overline{s} =_{\overline{A}} \overline{t}$ and additionally that $\mathtt{A}^* \ \overline{s} \ \overline{s}$, then $\mathtt{A}^* \ \overline{t} \ \overline{t}$ and $\mathtt{A}^* \ \overline{s} \ \overline{t}$ follow. By rule (eqTyping) and rule (sym) we further yield $\overline{s}:\overline{A}$ and $\overline{t}:\overline{A}$. This reduces the completeness claim for a term-equality $s =_A t$ to showing $\overline{s} =_{\overline{A}} \overline{t}$ and $\mathtt{A}^* \ \overline{s} \ \overline{s}$.

Term equality can be shown using the rules (congλ'), (congAppl'), (refl), (sym), (beta), (etaPi) and (Q$=$) in DHOL.

(congλ')

This case will use (13).

$\Gamma \vdash_{\mathsf{T}} A \equiv A'$	by assumption	(213)
$\Gamma, \ x:A \vdash_{\mathsf{T}} t =_B t'$	by assumption	(214)
$\overline{\Gamma} \vdash_{\overline{T}} \overline{A} \equiv \overline{A'}$	induction hypothesis,(213)	(215)
$\overline{\Gamma}, \ x:\overline{A}, \ \triangleright\mathtt{A}^* \ x \ x \vdash_{\overline{T}} \mathtt{B}^* \ \overline{t} \ \overline{t'}$	induction hypothesis,(214)	(216)
$\overline{\Gamma}, \ z:\overline{A}, \ \triangleright\mathtt{A}^* \ z \ z \vdash_{\overline{T}} \mathtt{B}^* \ \overline{t}[x/z] \ \overline{t'}[x/z]$	α-renaming,(216)	(217)

$$\overline{\Gamma},\ z{:}\overline{A} \vdash_{\overline{T}} \mathtt{A}^*\ z\ z \Rightarrow \mathtt{B}^*\ \overline{t}[x/z]\ \overline{t'}[x/z] \quad (\Rightarrow\text{I}),(217) \quad (218)$$

$$\overline{\Gamma} \vdash_{\overline{T}} \forall z{:}\overline{A}.\ \mathtt{A}^*\ z\ z \Rightarrow \mathtt{B}^*\ \overline{t}[x/z]\ \overline{t'}[x/z] \quad (\forall\text{I}),(218) \quad (219)$$

$$\overline{\Gamma},\ x,y{:}\overline{A} \vdash_{\overline{T}} \forall z{:}\overline{A}.\ \mathtt{A}^*\ z\ z \Rightarrow \mathtt{B}^*\ \overline{t}[x/z]\ \overline{t'}[x/z] \quad (\text{var}\vdash),(\text{var}\vdash),(219) \quad (220)$$

$$\overline{\Gamma},\ x,y{:}\overline{A} \vdash_{\overline{T}} \mathtt{A}^*\ x\ x \Rightarrow \mathtt{B}^*\ \overline{t}\ \overline{t'} \quad (\text{var}\vdash),(\text{var}\vdash),(220) \quad (221)$$

$$\overline{\Gamma},\ x,y{:}\overline{A},\ \triangleright\mathtt{A}^*\ x\ y \vdash_{\overline{T}} \mathtt{A}^*\ x\ x \Rightarrow \mathtt{B}^*\ \overline{t}\ \overline{t'} \quad (\text{monotonic}\vdash),(221) \quad (222)$$

$$\overline{\Gamma},\ x,y{:}\overline{A},\ \triangleright\mathtt{A}^*\ x\ y \vdash_{\overline{T}} \mathtt{A}^*\ x\ x \quad (13),(\text{assume}),(\text{assume}) \quad (223)$$

$$\overline{\Gamma},\ x,y{:}\overline{A},\ \triangleright\mathtt{A}^*\ x\ y \vdash_{\overline{T}} \mathtt{B}^*\ \overline{t}\ \overline{t'} \quad (\Rightarrow\text{E}),(222),(223) \quad (224)$$

$$\overline{\Gamma},\ x,y{:}\overline{A},\ \triangleright\mathtt{A}^*\ x\ y \vdash_{\overline{T}} \mathtt{B}^*\ \overline{t}\ \overline{t'}[x/y] \quad (13),(224),(\text{assume}) \quad (225)$$

$$\overline{\Gamma},\ x,y{:}\overline{A} \vdash_{\overline{T}} \mathtt{A}^*\ x\ y \Rightarrow \mathtt{B}^*\ \overline{t}\ \overline{t'}[x/y] \quad (\Rightarrow\text{I}),(225) \quad (226)$$

$$\overline{\Gamma} \vdash_{\overline{T}} \forall x{:}\overline{A}.\ \forall y{:}\overline{A}.\ \mathtt{A}^*\ x\ y \Rightarrow \mathtt{B}^*\ \overline{t}\ \overline{t'}[x/y] \quad (\forall\text{I}),(\forall\text{I}),(226) \quad (227)$$

$$\overline{\Gamma},\ x{:}\overline{A},\ \triangleright\mathtt{A}^*\ x\ x \vdash_{\overline{T}} \overline{t}{:}\overline{B} \quad \text{induction hypothesis},(235) \quad (228)$$

$$\overline{\Gamma},\ x{:}\overline{A},\ \triangleright\mathtt{A}^*\ x\ x \vdash_{\overline{T}} \overline{t'}{:}\overline{B} \quad \text{induction hypothesis},(235) \quad (229)$$

Since in HOL typing is independent of context assumptions:

$$\overline{\Gamma},\ x{:}\overline{A} \vdash_{\overline{T}} \overline{t}{:}\overline{B} \quad \text{explanation},(228) \quad (230)$$

$$\overline{\Gamma},\ x{:}\overline{A} \vdash_{\overline{T}} \overline{t'}{:}\overline{B} \quad \text{explanation},(229) \quad (231)$$

$$\overline{\Gamma} \vdash_{\overline{T}} \lambda x{:}\overline{A}.\ \overline{t}{:}\overline{A \to B} \quad (\text{lambda}),(230) \quad (232)$$

$$\overline{\Gamma} \vdash_{\overline{T}} \lambda x{:}\overline{A}.\ \overline{t'}{:}\overline{A \to B} \quad (\text{lambda}),(231) \quad (233)$$

$$\overline{\Gamma} \vdash_{\overline{T}} \overline{\lambda x{:}A.\ t = _{\Pi x:A.\ B} \lambda x{:}A'.\ t'} \quad (\text{PT23}),(227)$$

$$\overline{\Gamma} \vdash_{\overline{T}} \overline{\lambda x{:}A.\ t}{:}\overline{\Pi x{:}A.\ B} \quad (\text{PT10}),(\text{PT20}),(232)$$

$$\overline{\Gamma} \vdash_{\overline{T}} \overline{\lambda x{:}A.\ t'}{:}\overline{\Pi x{:}A.\ B} \quad (\text{PT10}),(\text{PT20}),(233)$$

(congAppl'):

$$\Gamma \vdash_{\mathsf{T}} t = _A t' \quad \text{by assumption} \quad (234)$$

$$\Gamma \vdash_{\mathsf{T}} f = _{\Pi x:A.\ B} f' \quad \text{by assumption} \quad (235)$$

$$\overline{\Gamma} \vdash_{\overline{T}} \mathtt{A}^*\ \overline{t}\ \overline{t'} \quad \text{induction hypothesis},(234) \quad (236)$$

$$\overline{\Gamma} \vdash_{\overline{T}} \forall x{:}\overline{A}.\ \forall y{:}\overline{A}.\ \mathtt{A}^*\ x\ y \Rightarrow (\Pi \mathtt{z}{:}\mathtt{A}.\ \mathtt{B})^*\ \overline{f}\ x\ \overline{f'}\ y \quad \text{induction hypothesis},(235) \quad (237)$$

$$\overline{\Gamma} \vdash_{\overline{T}} \overline{t}{:}\overline{A} \quad \text{induction hypothesis},(234) \quad (238)$$

$$\overline{\Gamma} \vdash_{\overline{T}} \overline{t'}{:}\overline{A} \quad \text{induction hypothesis},(234) \quad (239)$$

$$\overline{\Gamma} \vdash_{\overline{T}} \mathtt{A}^*\ \overline{t}\ \overline{t'} \Rightarrow (\Pi \mathtt{z}{:}\mathtt{A}.\ \mathtt{B})^*\ \overline{f}\ \overline{t}\ \overline{f'}\ \overline{t'} \quad (\forall\text{E}),(\forall\text{E}),(237),(238),(239) \quad (240)$$

$$\overline{\Gamma} \vdash_{\overline{T}} \overline{f}{:}\overline{A \to B} \quad \text{induction hypothesis},(235) \quad (241)$$

$$\overline{\Gamma} \vdash_{\overline{T}} \overline{f'}{:}\overline{A \to B} \quad \text{induction hypothesis},(235) \quad (242)$$

$$\overline{\Gamma} \vdash_{\overline{T}} (\Pi \mathtt{z}{:}\mathtt{A}.\ \mathtt{B})^*\ \overline{f}\ \overline{t}\ \overline{f'}\ \overline{t'} \quad (\Rightarrow\text{E}),(240),(236)$$

$$\overline{\Gamma} \vdash_{\overline{T}} \overline{f}\ \overline{t}{:}\overline{B} \quad (\text{appl}),(241),(238)$$

$$\overline{\Gamma} \vdash_{\overline{T}} \overline{f'}\ \overline{t'}{:}\overline{B} \quad (\text{appl}),(241),(238)$$

(refl):

$\Gamma \vdash_{\mathsf{T}} t{:}A$	by assumption	(243)
$\overline{\Gamma} \vdash_{\overline{T}} \overline{t}{:}\overline{A}$	induction hypothesis,(243)	(244)
$\overline{\Gamma} \vdash_{\overline{T}} \overline{t} =_{\overline{A}} \overline{t}$	(refl),(244)	
$\overline{\Gamma} \vdash_{\overline{T}} \mathtt{A}^*\ \overline{t}\ \overline{t}$	induction hypothesis,(243)	

(sym):

$\Gamma \vdash_{\mathsf{T}} s =_A t$	by assumption	(245)
$\overline{\Gamma} \vdash_{\overline{T}} \mathtt{A}^*\ \overline{s}\ \overline{t}$	induction hypothesis,(245)	(246)
$\overline{\Gamma} \vdash_{\overline{T}} \overline{t}{:}\overline{A}$	induction hypothesis,(245)	(247)
$\overline{\Gamma} \vdash_{\overline{T}} \overline{s}{:}\overline{A}$	induction hypothesis,(245)	(248)
$\overline{\Gamma} \vdash_{\overline{T}} \mathtt{A}^*\ \overline{t}\ \overline{s}$	(∀E),(∀E),(⇒E),(9),(246),(247),(248)	

(beta):

$\Gamma \vdash_{\mathsf{T}} (\lambda x{:}A.\ s)\ t{:}B$	by assumption	(249)
$\overline{\Gamma} \vdash_{\overline{T}} (\lambda x{:}\overline{A}.\ \overline{s})\ \overline{t}{:}\overline{B}$	induction hypothesis,PT20,(249)	(250)
$\overline{\Gamma} \vdash_{\overline{T}} (\lambda x{:}\overline{A}.\ \overline{s})\ \overline{t} =_{\overline{B}} \overline{s}[x/t]$	(beta),(250)	(251)
$\overline{\Gamma} \vdash_{\overline{T}} (\lambda x{:}\overline{A}.\ \overline{s})\ \overline{t} =_{\overline{B}} \overline{s[x/t]}$	(11),(251)	(252)
$\overline{\Gamma} \vdash_{\overline{T}} \overline{(\lambda x{:}A.\ s)\ t} =_{\overline{B}} \overline{s[x/t]}$	PT20,PT21,(252)	
$\overline{\Gamma} \vdash_{\overline{T}} (\Pi \mathtt{x}{:}\mathtt{A}.\ \mathtt{B})^*\ ((\lambda x{:}A.\ s)\ t)\ ((\lambda x{:}A.\ s)\ t)$	induction hypothesis,(249)	

(etaPi):

$\Gamma \vdash_{\mathsf{T}} t{:}\Pi x{:}A.\ B$	by assumption	(253)
$\overline{\Gamma} \vdash_{\overline{T}} \overline{t}{:}\overline{A} \to \overline{B}$	PT10,induction hypothesis,(253)	(254)
$\overline{\Gamma} \vdash_{\overline{T}} \overline{t} =_{\overline{A}\to\overline{B}} \lambda x{:}\overline{A}.\ \overline{t}\ x$	(eta),(254)	(255)
$\overline{\Gamma} \vdash_{\overline{T}} \overline{t} =_{\overline{\Pi x:A.\ B}} \overline{\lambda x{:}A.\ t\ x}$	PT20,PT10,(255)	
$\overline{\Gamma} \vdash_{\overline{T}} (\Pi \mathtt{x}{:}\mathtt{A}.\ \mathtt{B})^*\ \overline{t}\ \overline{t}$	induction hypothesis,(254)	

($|_p$ Eq):

$\Gamma \vdash_{\mathsf{T}} s =_A t$	by assumption	(256)
$\Gamma \vdash_{\mathsf{T}} p\ s$	by assumption	(257)
$\overline{\Gamma} \vdash_{\overline{T}} \mathtt{A}^*\ \overline{s}\ \overline{t}$	induction hypothesis,(256)	(258)
$\overline{\Gamma} \vdash_{\overline{T}} \overline{p}\ \overline{s}$	induction hypothesis,(257)	(259)

By (27) it follows:

$\overline{\Gamma} \vdash_{\overline{T}} \overline{p}\ \overline{t}$	explanation,(257),(258)	(260)
$\overline{\Gamma} \vdash_{\overline{T}} \mathtt{A}^*\ \overline{s}\ \overline{t} \wedge \overline{p}\ \overline{s} \wedge \overline{p}\ \overline{t}$	definition of ∧,(258),(259),(260)	(261)
$\overline{\Gamma} \vdash_{\overline{T}} (\mathtt{A}|_{\mathtt{p}})^*\ s\ t$	PT15,(261)	
$\overline{\Gamma} \vdash_{\overline{T}} \overline{s}{:}\overline{(A/p)}$	(validTyping),(∧El),(261)	

(Q =):

$\Gamma \vdash_\tau s : A$	by assumption	(262)
$\Gamma \vdash_\tau t : A$	by assumption	(263)
$\Gamma \vdash_\tau r : A \to A \to \mathsf{bool}$	by assumption	(264)
$\overline{\Gamma} \vdash_{\overline{T}} \overline{s} : \overline{A}$	induction hypothesis,(262)	(265)
$\overline{\Gamma} \vdash_{\overline{T}} \mathtt{A}^* \ \overline{s} \ \overline{s}$	induction hypothesis,(262)	(266)
$\overline{\Gamma} \vdash_{\overline{T}} \overline{t} : \overline{A}$	induction hypothesis,(263)	(267)
$\overline{\Gamma} \vdash_{\overline{T}} \mathtt{A}^* \ \overline{t} \ \overline{t}$	induction hypothesis,(262)	(268)
$\overline{\Gamma} \vdash_{\overline{T}} \overline{r} : \overline{A} \to \overline{A} \to \mathsf{bool}$	induction hypothesis,(264)	(269)
$\overline{\Gamma} \vdash_{\overline{T}} \overline{r} \ \overline{s} \ \overline{t} : \mathsf{bool}$	(appl),(appl),(269),(265),(267)	(270)
$\overline{\Gamma} \vdash_{\overline{T}} \overline{r} \ \overline{s} \ \overline{t} =_{\mathsf{bool}} \overline{r} \ \overline{s} \ \overline{t}$	(refl),(270)	(271)
$\overline{\Gamma} \vdash_{\overline{T}} \left(\overline{r} \ \overline{s} \ \overline{t} \wedge \mathtt{A}^* \ \overline{s} \ \overline{s} \wedge \mathtt{A}^* \ \overline{t} \ \overline{t}\right) =_{\mathsf{bool}} \overline{r} \ \overline{s} \ \overline{t}$	definition of $\wedge$, (266),(268),(271)	(272)
$\overline{\Gamma} \vdash_{\overline{T}} \overline{s} = \overline{(A/r)} \ \overline{t}$	PT17,PT13,(272)	
$\overline{\Gamma} \vdash_{\overline{T}} \overline{s} : \overline{(A/r)}$	PT17,(267)	

C.2.8 Validity

Validity can be shown using the rules (axiom), (assume), ($\Rightarrow$I), ($\Rightarrow$E), (cong$\vdash$), (boolExt) and ($|_p$ E2).

(axiom)

$\triangleright F$ in T	by assumption	(273)
$\vdash_\tau \Gamma$ Ctx	by assumption	(274)
$\triangleright \overline{F}$ in $\overline{T}$	PT4,(273	(275)
$\vdash_{\overline{T}} \overline{\Gamma}$ Ctx	induction hypothesis,274	(276)
$\overline{\Gamma} \vdash_{\overline{T}} \overline{F}$	(axiom),(275),(276)	

(assume)

$\triangleright F$ in Γ	by assumption	(277)
$\vdash_\tau \Gamma$ Ctx	by assumption	(278)
$\triangleright \overline{F}$ in $\overline{\Gamma}$	PT7,(277	(279)
$\vdash_{\overline{T}} \overline{\Gamma}$ Ctx	induction hypothesis,278	(280)
$\overline{\Gamma} \vdash_{\overline{T}} \overline{F}$	(assume),(279),(280)	

($\Rightarrow$I)

$\Gamma \vdash_\tau F : \mathsf{bool}$	by assumption	(281)
$\Gamma, \triangleright F \vdash_\tau G$	by assumption	(282)
$\overline{\Gamma} \vdash_{\overline{T}} \overline{F} : \mathsf{bool}$	induction hypothesis,(281)	(283)
$\overline{\Gamma}, \overline{F} \vdash_{\overline{T}} \overline{G}$	induction hypothesis,PT7,(282)	(284)
$\overline{\Gamma} \vdash_{\overline{T}} \overline{F} \Rightarrow \overline{G}$	($\Rightarrow$I),(283),(284)	(285)
$\overline{\Gamma} \vdash_{\overline{T}} \overline{F \Rightarrow G}$	PT22,(285)	

(⇒E)

$\Gamma \vdash_{\mathsf{T}} F \Rightarrow G$	by assumption	(286)
$\Gamma \vdash_{\mathsf{T}} F$	by assumption	(287)
$\overline{\Gamma} \vdash_{\overline{T}} \overline{F} \Rightarrow \overline{G}$	induction hypothesis,PT22,(286)	(288)
$\overline{\Gamma} \vdash_{\overline{T}} \overline{F}$	induction hypothesis,(287)	(289)
$\overline{\Gamma} \vdash_{\overline{T}} \overline{G}$	(⇒E),(288),(289)	

(cong⊢)

$\Gamma \vdash_{\mathsf{T}} F =_{\mathsf{bool}} F'$	by assumption	(290)
$\Gamma \vdash_{\mathsf{T}} F'$	by assumption	(291)
$\overline{\Gamma} \vdash_{\overline{T}} \overline{F} =_{\mathsf{bool}} \overline{F'}$	(PT13),induction hypothesis,(290)	(292)
$\overline{\Gamma} \vdash_{\overline{T}} \overline{F'}$	induction hypothesis,(291)	(293)
$\overline{\Gamma} \vdash_{\overline{T}} \overline{F}$	(cong⊢),(292),(293)	

(boolExt)

$\Gamma \vdash_{\mathsf{T}} p\ \mathsf{true}$	by assumption	(294)
$\Gamma \vdash_{\mathsf{T}} p\ \mathsf{false}$	by assumption	(295)
$\overline{\Gamma} \vdash_{\overline{T}} \overline{p}\ \mathsf{true}$	induction hypothesis,PT21,(294)	(296)
$\overline{\Gamma} \vdash_{\overline{T}} \overline{p}\ \mathsf{false}$	induction hypothesis,PT21,(295)	(297)
$\overline{\Gamma} \vdash_{\overline{T}} \forall\, z{:}\mathsf{bool}.\ \overline{p}\ z$	(boolExt),(296),(297)	(298)
$\overline{\Gamma},\ x{:}\mathsf{bool} \vdash_{\overline{T}} \forall\, z{:}\mathsf{bool}.\ \overline{p}\ z$	(var⊢),(298)	(299)
$\overline{\Gamma},\ x{:}\mathsf{bool} \vdash_{\overline{T}} \overline{p}\ x$	(∀E),(299),(assume)	(300)
$\overline{\Gamma},\ x, y{:}\mathsf{bool},\ \triangleright x =_{\mathsf{bool}} y \vdash_{\mathsf{T}} \overline{p}\ x$	(monotonic⊢),(var⊢),(300)	(301)
$\overline{\Gamma},\ x, y{:}\mathsf{bool},\ \triangleright x =_{\mathsf{bool}} y \vdash_{\mathsf{T}} \overline{p}\ y$	(rewrite),(301),(assume)	(302)
$\overline{\Gamma},\ x, y{:}\mathsf{bool} \vdash_{\mathsf{T}} \mathsf{bool}^*\ x\ y \Rightarrow \overline{p}\ y$	(PT13),(⇒I),(302)	(303)
$\overline{\Gamma} \vdash_{\overline{T}} \forall\, x{:}\mathsf{bool}.\ \forall\, y{:}\mathsf{bool}.$ $\mathsf{bool}^*\ x\ y \Rightarrow \overline{p}\ y$	(∀I),(∀I),(303)	(304)
$\overline{\Gamma} \vdash_{\overline{T}} \overline{\forall\, x{:}\mathsf{bool}.\ p\ x}$	(PT23),(PT11),(304)	

($|_p$ E2)

$\Gamma \vdash_{\mathsf{T}} t{:}A|_p$	by assumption	(305)
$\overline{\Gamma} \vdash_{\overline{T}} (\mathtt{A}|_{\mathtt{p}})^*\ \overline{t}\ \overline{t}$	induction hypothesis,(305)	(306)
$\overline{\Gamma} \vdash_{\overline{T}} \overline{p}\ \overline{t}$	(∧Er),(∧Er),PT15,(306)	(307)
$\overline{\Gamma} \vdash_{\overline{T}} \overline{p\ t}$	PT21,(307)	

◀

D Soundness proof

The idea of the soundness proof is to transform HOL-proofs into DHOL-proofs. The proof is very involved, and we proceed in multiple steps:

1. prove that the translation is injective for terms of given DHOL type,
2. define quasi-preimages for terms not in image of translation,
3. given valid HOL derivation of translation of well-typed validity conjecture, choose DHOL types of quasi-preimages of terms in it,
4. modify derivation to make terms in it (almost) proper,
5. lift modified HOL derivation to DHOL derivation.

D.1 Type-wise injectivity of the translation

▶ **Definition 25.** *Let t be an ill-typed DHOL term with well-typed image $\overline{t}$ in HOL. In this case we will say that $\overline{t}$ is a* spurious *term w.rt. its preimage t. If the preimage is unique or clear from the context we will simply say that $\overline{t}$ is spurious. Similarly, a term $\overline{s}$ in HOL that is the image of a well-typed term s, will be called* proper *w.r.t its preimage s. A term tm in HOL that is not the image of any (well-typed or not) term is said to be* improper*.*

▶ **Lemma 26.** *Let Δ be a DHOL context and let Γ denote its translation. Given two DHOL terms s, t of type A and assuming s and t are not identical, it follows that $\overline{s}$ and $\overline{t}$ are not identical.*

Proof of Lemma 26. We prove this by induction on the shape of the types both equalities are over — in case both terms are equalities — and by subinduction on the shape of the two translated terms otherwise. We observe that terms created using a different top-level production are non-identical and will remain that way in the image. So we can go over the productions one by one and assuming type-wise injectivity for subterms show injectivity of applying them. Different constants are mapped to different constants and different variables to different variables, so in those cases there is nothing to prove. If two function applications or implications differ in DHOL then one of the two pairs of corresponding arguments must differ as well. By induction hypothesis so will the images of the terms in that pair. Since function application and implication both commute with the translation, it follows that the images of the function applications or implications also differ. Since the translations of the terms on both sides of an equality also show up in the translation, the same argument also works for two equalities over the same type. Similarly for lambda functions over the same type.

Consider now two equalities over different types that get identified by dependency-erasure.

In case of equalities over different base types, the typing relations that are applied in the images are different, so the images of the equalities differ. For equalities over different Π-types either the domain type or the codomain type must differ by rule (congΠ). If the domain types differ then the typing assumption after the two universal quantifiers of the translated equalities will differ. If the codomain types are different then the applications of the typing relations on the right of the $\Rightarrow$ of the translated equalities are the translations of the equalities yielded by applying the functions on both sides of the equalities to a freshly bound variable of the domain type. The translations of the equalities are only identical if those "inner equalities" are identical. Furthermore, the inner equalities are over types that are the codomain of the type the equalities are over. The claim then follows from the induction hypothesis.

Finally it remains to consider the case of equalities $s =_{A|_p} t$ and $s' =_{A'|_{p'}} t'$ over non-identical refinement types $A|_p$ and $A'|_{p'}$ where not both $A = A'$ and $p = p'$. If $p \neq p'$, then the translations have different subterms $\overline{p}\ s$ and $\overline{p'}\ s'$ and thus differ. If $A \neq A'$, then the first conjuncts in the translated equalities are the translations of equalities over the types A and A' respectively, which by the induction hypothesis have different translations. So in any case, the equalities have different images. The case of equalities over quotient types works analogously. ◀

D.2 Quasi-preimages for terms and validity statements in admissible HOL derivations

Firstly, we will consider the preimage of a typing relations $\mathtt{A}^*$ to be the equality symbol $\lambda x{:}A.\ \lambda y{:}A.\ x =_A y$ (if equality is treated as a (parametric) binary predicate rather than a production of the grammar this eta reduces to the symbol $=_A$).

Using this convention, we define the normalization of an improper HOL term, which is either a proper term or a spurious term. The normalization of an improper HOL term is defined by:

▶ **Definition 27.** *Let t be an improper HOL term. Then we define the normalization* $\mathsf{norm}\,[t]$ *of t by induction on the shape of t:*

$$\mathsf{norm}\,[\overline{t}] := t \tag{PT24}$$
$$\mathsf{norm}\,[\mathsf{norm}\,[s]] := \mathsf{norm}\,[s] \tag{PT25}$$
$$\mathsf{norm}\,[\mathtt{A}^*\ s] := \lambda y{:}\overline{A}.\ \mathtt{A}^*\ s\ y \tag{PT26}$$
$$\mathsf{norm}\,[\mathtt{A}^*] := \lambda x{:}\overline{A}.\ \lambda y{:}\overline{A}.\ \mathtt{A}^*\ x\ y \tag{PT27}$$
$$\mathsf{norm}\,[c] := \mathsf{c} \tag{PT28}$$
$$\mathsf{norm}\,[x] := x \tag{PT29}$$
$$\mathsf{norm}\,[f\ t] := \mathsf{norm}\,[f]\ \ \mathsf{norm}\,[t] \tag{PT30}$$
$$\mathsf{norm}\,[\lambda x{:}C.\ t] := \lambda x{:}C.\ \mathsf{norm}\,[t] \tag{PT31}$$

If F not of shape $\mathtt{A}^*\_\ \_ \Rightarrow \_$ or $\forall\, x'{:}\overline{A}.\ \mathtt{A}^*\ x\ x' \Rightarrow \_$:

$$\mathsf{norm}\,[\forall\, x{:}\overline{A}.\ F] := \mathsf{norm}\,[\forall\, x{:}\overline{A}.\ \mathtt{A}^*\ x\ xF] \tag{PT32}$$
$$\mathsf{norm}\,[\forall\, x{:}\overline{A}.\ \mathtt{A}^*\ x\ x \Rightarrow G] := \forall\, x, x'{:}\overline{A}.\ \mathtt{A}^*\ x\ x' \Rightarrow G \tag{PT33}$$
$$\mathsf{norm}\,[s =_{\overline{A}} t] := \mathtt{A}^*\ s\ t \tag{PT34}$$
$$\mathsf{norm}\,[s \Rightarrow t] := \mathsf{norm}\,[s] \Rightarrow \mathsf{norm}\,[t] \tag{PT35}$$

For terms t in the image of the translation, we define the normalization of t be be t itself.

▶ **Definition 28.** *Assume a well-formed DHOL theory T.*

We say that an HOL context Δ is proper *(relative to $\overline{T}$), iff there exists a well-formed HOL context Θ (relative to $\overline{T}$), s.t. there is a well-formed DHOL context Γ (relative to T) with $\overline{\Gamma} = \Theta$ and Θ can be obtained from Δ by adding well-typed typing assumptions. In this case, Γ is called a* quasi-preimage *of Δ. Inspecting the translation, it becomes clear that Γ is uniquely determined by the choices of the preimages of the types of variables without a typing assumption in Δ.*

Given a proper HOL context Δ *and a well-typed HOL formula* φ *over* Δ*, we say that* φ *is* quasi-proper *iff* $\mathsf{norm}\,[\varphi] = \overline{F}$ *for* $\Gamma \vdash_{\mathsf{T}} F : \mathsf{bool}$ *and* Γ *is a quasi-preimage of* Δ*. In that case, we call* F *a* quasi-preimage *of* φ*.*

Finally, we call a validity judgement $\Delta \vdash_{\overline{T}} \varphi$ *in HOL* proper *iff*

1. Δ *is proper,*
2. φ *is quasi-proper in context* Δ

In this case, we will call $\overline{\Gamma} \vdash_{\overline{T}} \overline{F}$ *a relativization of* $\Delta \vdash_{\overline{T}} \varphi$ *and* $\Gamma \vdash_{\mathsf{T}} F$ *a* quasi-preimage *of the statement* $\Delta \vdash_{\overline{T}} \varphi$*, where* Γ *is a quasi-preimage of* Δ *and* F *a quasi-preimage of* φ*. Additionally, for HOL terms with preimages we consider these preimages to be quasi-preimages of the HOL term as well.*

D.3 Transforming HOL derivations into admissible HOL derivations

It will be useful to distinguish between two different kinds of improper terms.

▶ **Definition 29.** *An improper term is called* almost proper *iff its normalization isn't spurious (w.r.t. a given quasi-preimage) and contains no spurious subterms, otherwise it is said to be* unnormalizably spurious*. This means that improper terms are almost proper iff their given quasi-preimage is well-typed. Since proper terms have well-typed preimages, they are almost proper (w.r.t. this preimage) as well.*

In order to lift a HOL derivation to DHOL, we first have to choose (quasi)-preimage types for all term occuring in it (at which point we use the notions of spurious and almost proper terms w.r.t. these DHOL types).

D.3.1 Choosing (quasi-)preimage types for a HOL derivation

▶ **Lemma 30** (Indexing lemma)**.** *Assume that* $\Gamma \vdash_{\mathsf{T}} F : \mathsf{bool}$ *holds in DHOL. Given a valid HOL derivation* D *of the statement* $\overline{\Gamma} \vdash_{\overline{T}} \overline{F}$*, we can choose a DHOL type* $T(t)$ *(called type index) for each occurence of a HOL term* t *in* D*, s.t. the following properties hold:*

1. $T(\overline{t}) = A'$ *with* $\overline{A} = \overline{A'}$ *for any DHOL term* t *satisfying* $\Gamma \vdash_{\mathsf{T}} t : A$*,*
2. $T(\mathsf{c}) = A$ *if* $\mathsf{c} : A$ *is a constant in* T*,*
3. $T(x) = A$ *if* $x : A$ *is variable declaration in* Γ*,*
4. $T(s) = T(t)$ *for* s, t *within an equality of the form* $s =_A t$ *for some HOL type* A*,*
5. $T(s) = T(t) = \mathsf{bool}$ *for* s, t *within an implication of the form* $s \Rightarrow t$*,*
6. $T(x) = A$ *for* x *in* $(\lambda x{:}\mathbb{B}.\ s)\ t$ *if* $T(t) = A$*,*
7. $T(s =_A t) = \mathsf{bool}$*,*
8. *for* x *in* $(\lambda x{:}\mathbb{B}.\ s)\ t$ *if* $T(t) = A$*,*
9. *when variables are moved from the context into a* λ*-binder or vice versa the index of said variable is preserved*
10. *whenever a term* t *occurs both in the assumptions and conclusions of a step* S *in* D*, the index of* t *is the same all those occurrences of* t *in* S*,*

11. *if the subterms* x, t *in a term* $\lambda x{:}\overline{B}.$ *in* D *satisfy* $T(x) = A$ *and* $T(t) = B$*, then it follows* $T(\lambda x{:}B.\ t) = \Pi x{:}A.\ B$.

Proof by induction of the shape of D**.** This lemma only holds for well-formed derivations of translations of well-typed conjectures over well-formed theories. It will not hold for arbitrary formulae (as can be seen by considering equalities between constants of equal HOL but different DHOL types). The proof of the lemma will therefore use that fact that the final statement in the derivation is the translation of a well-typed DHOL statement (which already determines the "correct" indices for the terms within that statement) and then show that for each step in a well-formed HOL proof concluding a statement that we can correctly index, the assumptions of that step can also be indexed correctly. We will thus proceed by "backwards induction" on the shape of the derivation D.

The assumptions of the theory and contexthood rules only contain terms already contained in the conclusions, so the associated cases in the proof are all trivial.

Similarly the assumptions of lookup rules and type well-formedness rules contain no additional terms, so those cases are also trivial.

The typing rules are about forming larger terms from subterms, so if those larger terms can be consistently (i.e. according to the claim of the lemma) indexed, then the same is necessarily also true for the subterms. Thus the typing rules also have only trivial cases. By the same arguments the cases for the congruence rules (congλ), (congAppl), the symmetry and transitivity rules for term equality and the rules (beta), (eta), ($\Rightarrow$type) and ($\Rightarrow$I) are also all trivial.

It remains to consider the cases for the rules ($\Rightarrow$E), (cong$\vdash$), (boolExt) and (nonempty).

D.3.1.1 Regarding (beta):

Here the assumptions of the rule contain the additional terms F and $F \Rightarrow G$. However as both terms are of type bool all their (quasi)-preimages have type bool as well and picking indices according to any of the quasi-preimages of $F \Rightarrow G$ will work.

D.3.1.2 Regarding (cong$\vdash$):

Here the assumptions of the rule contain the additional terms F' and $F =_{\mathsf{bool}} F'$. Both terms are of type bool, so by the same argument as in the previous case, we can index them consistently with the claim of this lemma.

D.3.1.3 Regarding (boolExt):

Here the assumptions of the rule contain the additional terms $p\ \mathsf{true}$ and $p\ \mathsf{false}$ and true and false. All these terms are of type bool, so by the same argument as in the previous two cases, we can index them consistently with the claim of this lemma.

D.3.1.4 Regarding (nonempty):

Here the second assumption contains an additional variable of type A. As this variable doesn't occur in any other terms, we can index it by the type of any of its (quasi)-preimages. ◀

▶ **Remark 2.** *In the following, we will use the term* type index *to refer to a choice of DHOL types for each HOL term in a derivation satisfying the properties of the previous lemma.*

D.3.2 Transforming unnormalizably spurious terms into almost proper terms in HOL derivations

▶ **Definition 31.** *A valid HOL derivation is called* admissible *iff we can choose quasi-preimages for all terms occurring in it s.t. all terms in the derivation are almost proper w.r.t. their chosen quasi-preimages.*

This definition is useful, since admissible derivations are precisely those HOL derivations that allow us to consistently lift the terms occurring in them to well-typed DHOL terms.

In the following, we describe a proof transformation which maps HOL derivations to admissible HOL derivations.

▶ **Definition 32.** *A* statement transformation *in a given logic is a map that maps statements in the logic to statements in the logic. Similarly an* indexed statement transformation *is a map that maps HOL statements with indexed terms to HOL statements.*

▶ **Definition 33.** *A* macro-step M for *an (indexed) statement transformation* T replacing *a step* S *in a derivation is a sequence of steps* $S_1, \dots, S_n$ *(called* micro-steps *of* M*) s.t. the assumptions of the* S_i *that are not concluded by* S_j *with* $j < i$ *are results of applying* T *to assumptions of step* S *and furthermore the conclusion of step* S_n *is the result of applying* T *to the conclusion of* S*. The assumptions of those* S_j *that are not concluded by previous micro-steps of* M *are called the* assumptions of macro-step M *and the conclusion of the last micro-step* S_n *of* M *is called the* conclusion of macro-step M*.*

Thus we we can replace each step in a derivation by a macro step replacing that step, we can transform that derivation to a derivation in which the given indexed statement transformation is applied to all statements. This is useful to simplify and normalize derivations to derivations with certain additional properties. In our case, we want to normalize a given HOL derivation into an admissible HOL derivation. Thus, we need to define an indexed statement transformation for which all terms in the image of the transformation are almost proper and the replace all steps in the derivation by macro steps for that statement transformation.

Since the notion of a quasi-proper term only makes sense once we fix a choice of type indices (in the sense of Lemma 30), the indexed statement transformation will actually depend on the choice of type indices.

▶ **Definition 34.** *A* normalizing statement transformation $\mathsf{sRed}\,(\cdot)$ *is defined to be an indexed statement transformation that replaces terms in statements as described below. The definition of the transformation of a term depends on its type index — a DHOL-type* A *(called* preimage type*) — for each term* t*. We will write those types as indices to the HOL terms, so for instance* t_A *indicates a HOL term* t *of type* $\overline{A}$ *and preimage type* A*.*

These preimage types are used to effectively associate to each term a type of a possible quasi-preimage (hence their name), which is useful as for λ*-functions there are quasi-preimages of potentially many different types. We require that for an indexed term* t_A*, term* t *has type* $\overline{A}$ *and that for almost proper terms* t_A *with unique quasi-preimage the quasi-preimage has type* A*.*

Since variables and lambda binders are the sole cause for HOL terms having multiple quasi-preimages, choosing indices for variables in HOL terms induces unique quasi-preimages (respecting those type indices). This uniqueness is a direct consequence of (the proof of) Lemma 26.

We will consider only those quasi-preimages that respect type indices, for the notions of unnormalizably spurious and almost proper terms.

With respect to these choices, the transformation will do the following two things (in this order) in order to "normalize" unnormalizably spurious terms to almost proper ones:

1. *apply beta and eta reductions and in case this doesn't yield almost proper terms*
2. *replace unnormalizably spurious function applications of type B by the "default terms" w_B of type B which is proper and whose existence is assumed for all HOL types.*

As we are assuming a valid HOL derivation indexed according to Lemma 30, we will only define this transformation on well-typed HOL terms with preimage types consistent with the indexing lemma. We can then define the transformation of t_A (denoted by $\mathsf{sRed}\,(t_A)$) by induction on the shape of t_A as follows:

$$\mathsf{sRed}\,(t_A) \quad :=t_A \quad \text{if } t \text{ has quasi-preimage of type } A \tag{SR1}$$

$$\mathsf{sRed}\,(f_{\prod x{:}A.\ B}\ t_A) \quad :=\mathsf{sRed}\,(\mathsf{sRed}\,(f_{\prod x{:}A.\ B})\ \mathsf{sRed}\,(t_A)) \quad \text{if } f_{\prod x{:}A.\ B}\ t_A \text{ not beta or eta reducible} \tag{SR2}$$

In the following cases, we assume that the term t_A in $\mathsf{sRed}\,(\cdot)$ on the left of $:=$ isn't almost proper with a quasi-preimage of type A:

$$\mathsf{sRed}\,(t_A) \quad :=\mathsf{sRed}\,\left(t_A^{\beta\eta}\right) \quad \text{if } t \text{ is beta or eta reducible} \tag{SR3}$$

$$\mathsf{sRed}\left(s_A =_{\overline{A}} t_{A'}\right) \quad :=\mathsf{sRed}\,(s_A) =_{\overline{A}} \mathsf{sRed}\,(t_A) \tag{SR4}$$

$$\mathsf{sRed}\,(F_{\mathsf{bool}} \Rightarrow G_{\mathsf{bool}}) \quad :=\mathsf{sRed}\,(F_{\mathsf{bool}}) \Rightarrow \mathsf{sRed}\,(G_{\mathsf{bool}}) \tag{SR5}$$

$$\mathsf{sRed}\,(\lambda x{:}A.\ s_B) \quad :=\lambda x{:}A.\ \mathsf{sRed}\,(s_B) \tag{SR6}$$

$$\mathsf{sRed}\left(\left(\mathsf{sRed}\,(f_{\prod x{:}A.\ B})_{\prod x{:}A.\ B}\ \mathsf{sRed}\,(t_{A'})_{A'}\right)_{B'}\right) \quad :=w_{\overline{B}} \quad \text{if } A \neq A' \text{ or } B \neq B' \tag{SR7}$$

▶ **Lemma 35.** *Assume a well-typed DHOL theory T and a conjecture $\Gamma \vdash_T \varphi$ with Γ well-formed and φ well-typed. Assume a valid HOL derivation D of $\overline{\Gamma} \vdash_{\overline{T}} \overline{\varphi}$. Choose type indices for the terms in D according to the properties of the indexing lemma (Lemma 30). Then, for any steps S in D we can construct a macro-step for the normalizing statement transformation replacing step S s.t. after replacing all steps by their macro-steps:*

- *the resulting derivation is valid,*
- *all terms occurring in the derivation are almost proper (wr.t. the quasi-preimages determined by the type indices).*

Proof of Lemma 35. We will show this by induction on the shape of D.

Firstly, we observe that there are no dependent types in HOL and the context and axioms contain no spurious subterms. Hence, well-formedness (of theories, contexts, types) and type-equality judgements are unaffected by the transformation. So there is nothing to prove for the well-formedness and type-equality rules (those steps can be replaced by a macro step containing exactly this single step).

D.3.2.1 Regarding the type indices:

We observe that the properties of the type indices provided by Lemma 30 ensure that the smallest unnormalizably spurious terms (i.e. without unnormalizably spurious proper subterms) are function applications in which function and argument are both almost proper. Furthermore, in such a case the function is not a λ-function.

It remains to consider the typing and validity rules and to construct macro steps for the steps in the derivation using them for the normalizing statement transformation.

Since terms indexed by a type A have type $\overline{A}$ it is easy to see from Definition 34 that the normalizing statement transformation replaces terms of type $\overline{A}$ by terms of type $\overline{A}$.

(const):

Since constants are proper terms, there is nothing to prove.

(var):

Since context variables are proper terms, there is nothing to prove.

(=type):

$$\Delta \vdash_{\overline{T}} \mathsf{sRed}\,(s)_A : \overline{A} \qquad \text{by assumption} \qquad (308)$$

$$\Delta \vdash_{\overline{T}} \mathsf{sRed}\,(t)_A : \overline{A} \qquad \text{by assumption} \qquad (309)$$

$$\Delta \vdash_{\overline{T}} \mathsf{sRed}\,(s)_A =_{\overline{A}} \mathsf{sRed}\,(t)_A : \mathsf{bool} \qquad (=\text{type}),(308),(309) \qquad (310)$$

$$\Delta \vdash_{\overline{T}} \mathsf{sRed}\,\left(s_A =_{\overline{A}} t_A\right)_{\mathsf{bool}} : \mathsf{bool} \qquad \text{SR4},(310)$$

(lambda):

$$\Delta,\ x_A : \overline{A} \vdash_{\overline{T}} \mathsf{sRed}\,(t_B)_B : \overline{B} \qquad \text{by assumption} \qquad (311)$$

$$\Delta \vdash_{\overline{T}} \left(\lambda x_A : \overline{A}.\ \mathsf{sRed}\,(t_B)_B\right) : \overline{A} \to \overline{B} \qquad (\text{lambda}),(311) \qquad (312)$$

If $\mathsf{sRed}\,(t)_B$ isn't an unnormalizably spurious function application $\mathsf{sRed}\,(f_{\Pi y:A'.\ B})\ x_A$ for which x doesn't appear in f:

$$\Delta \vdash_{\overline{T}} \mathsf{sRed}\,\left(\lambda x_A : \overline{A}.\ \mathsf{sRed}\,(t_B)_B\right) : \overline{A} \to \overline{B} \qquad \text{SR6},(312)$$

Else by (SR3) we have $\mathsf{sRed}\,\left(\lambda x_A : \overline{A}.\ \mathsf{sRed}\,(t_B)_B\right) = \mathsf{sRed}\,(f_{\Pi y:A.\ B})$. By the remark about the type of $\mathsf{sRed}\,(\cdot)$ it follows that $\mathsf{sRed}\,(f_{\Pi y:A.\ B})$ has type $\overline{\Pi y:A.\ B} = \overline{A} \to \overline{B}$.

$$\Delta \vdash_{\overline{T}} \mathsf{sRed}\,(f_{\Pi y:A'.\ B}) : \overline{A} \to \overline{B} \qquad \text{see above} \qquad (313)$$

$$\Delta \vdash_{\overline{T}} \mathsf{sRed}\,\left(\lambda x_A : \overline{A}.\ \mathsf{sRed}\,(t_B)_B\right) : \overline{A} \to \overline{B} \qquad (\text{SR3}),(313)$$

(appl):

$$\Delta \vdash_{\overline{T}} \mathsf{sRed}\,(f_{\Pi x:A.\ B}) : \overline{A} \to \overline{B} \qquad \text{by assumption} \qquad (314)$$

$$\Delta \vdash_{\overline{T}} \mathsf{sRed}\,(t_{A'}) : \overline{A} \qquad \text{by assumption} \qquad (315)$$

If $\mathsf{sRed}\,(f_{\Pi x{:}A.\ B})\ \mathsf{sRed}\,(t_{A'})_{A'}$ satisfies $A \equiv A'$:

$\Delta \vdash_{\overline{T}} \mathsf{sRed}\,(f_{\Pi x{:}A.\ B}\ t_{A'}) : \overline{B}$ SR1,(lambda),(314),(315)

If $\mathsf{sRed}\,(f_{\Pi x{:}A.\ B})\ \mathsf{sRed}\,(t_{A'})$ doesn't satisfy $A \equiv A'$ then the 6. property in Lemma 30 implies that f is not a lambda function and thus $\mathsf{sRed}\,(f_{\Pi x{:}A.\ B})\ \mathsf{sRed}\,(t_{A'})$ is not beta reducible. Thus by (SR2) and (SR7) we have

$$\mathsf{sRed}\,(f_{\Pi x{:}A.\ B}\ t_{A'}) = \mathsf{sRed}\,(\mathsf{sRed}\,(f_{\Pi x{:}A.\ B})_{\Pi x{:}A.\ B}\ \mathsf{sRed}\,(\mathsf{sRed}\,(t_{A'})_{A'})) = w_{\overline{B}}.$$

By the axiom schema asserting the existence of $w_{\overline{B}}$ we have $w_{\overline{B}} : \overline{B}$:

$\Delta \vdash_{\overline{T}} w_{\overline{B}} : \overline{B}$ axiom scheme (316)

$\Delta \vdash_{\overline{T}} \mathsf{sRed}\,(f_{\Pi x{:}A.\ B}\ t_{A'}) : \overline{B}$ (SR2),(SR7),(316)

(⇒type):

$\Delta \vdash_{\overline{T}} \mathsf{sRed}\,(F)_{\mathsf{bool}} : \mathsf{bool}$ by assumption (317)

$\Delta \vdash_{\overline{T}} \mathsf{sRed}\,(G)_{\mathsf{bool}} : \mathsf{bool}$ by assumption (318)

$\Delta \vdash_{\overline{T}} \mathsf{sRed}\,(F)_{\mathsf{bool}} \Rightarrow \mathsf{sRed}\,(G)_{\mathsf{bool}} : \mathsf{bool}$ (⇒type),(317),(318) (319)

$\Delta \vdash_{\overline{T}} \mathsf{sRed}\,(F_{\mathsf{bool}} \Rightarrow F_{\mathsf{bool}}) : \mathsf{bool}$ SR5,(319)

(axiom):

Since translations of axioms to HOL are always proper terms and the additionally generated axioms are almost proper, there is nothing to prove here.

(assume):

If the axiom is a typing axiom generated by the translation, it follows that it is almost proper. Similarly, if it is an axiom for a base type. Otherwise:

$\triangleright \mathsf{sRed}\,(F_{\mathsf{bool}})$ in Δ by assumption (320)

$\Delta \vdash_{\overline{T}} \mathsf{sRed}\,(F_{\mathsf{bool}})$ (assume),(320)

By assumption $\mathsf{sRed}\,(F)$ almost proper (with a quasi-preimage of type bool), so the conclusion of the rule is almost proper and there is nothing ot prove here.

(congλ):

$\Delta \vdash_{\overline{T}} A \equiv A'$ by assumption (321)

$\Delta,\ x_A{:}\overline{A} \vdash_{\overline{T}} \mathsf{sRed}\,(t_B =_{\overline{B}} t'_B)_{\mathsf{bool}}$ by assumption (322)

$\Delta,\ x_A{:}\overline{A} \vdash_{\overline{T}} \mathsf{sRed}\,(t_B)_B =_{\overline{B}} \mathsf{sRed}\,(t'_B)_B$ SR4,(322) (323)

$\Delta \vdash_{\overline{T}} \lambda x_A{:}\overline{A}.\ \mathsf{sRed}\,(t_B)_B =_{\overline{A}\to\overline{B}}$

$\lambda x_A{:}\overline{A}.\ \mathsf{sRed}\,(t'_B)_B$ (congλ),(321),(323) (324)

By assumption $\mathsf{sRed}\,(t)_B =_{\overline{B}} \mathsf{sRed}\,(t')_B$ almost proper with quasi-preimage consistent with type indices and $A \equiv A'$, thus also $\lambda x_A{:}\overline{A}.\ \mathsf{sRed}\,(t)_B =_{\overline{A}\to\overline{B}} \lambda x_A{:}\overline{A}.\ \mathsf{sRed}\,(t')_B$ almost proper with quasi-preimage consistent with type indices.

$\Delta \vdash_{\overline{T}} \mathsf{sRed}\,(\lambda x_A{:}\overline{A}.\ \mathsf{sRed}\,(t_B)_B =_{\overline{A}\to\overline{B}} \lambda x_A{:}\overline{A}.\ \mathsf{sRed}\,(t'_B)_B)$ SR6,SR4,(324)

(congAppl):

$\Delta \vdash_{\overline{T}} \mathsf{sRed}\,(t_A =_{\overline{A}} t'_A)$ by assumption (325)

$\Delta \vdash_{\overline{T}} \mathsf{sRed}\,(t_A)_A =_{\overline{A}} \mathsf{sRed}\,(t'_A)_A$ SR4,(325) (326)

$\Delta \vdash_{\overline{T}} \mathsf{sRed}\,(f\,\Pi x{:}A'.\ B =_{\overline{A}\to\overline{B}} f'\,\Pi x{:}A'.\ B)$ by assumption (327)

$\Delta \vdash_{\overline{T}} \mathsf{sRed}\,(f)_{\Pi x:A'.\ B} =_{\overline{A}\to\overline{B}} \mathsf{sRed}\,(f')_{\Pi x:A'.\ B}$ SR4,(327) (328)

Assume that $A \not\equiv A'$. By property 6. in Lemma 30 $\mathsf{sRed}\,(f)$ and $\mathsf{sRed}\,(f')$ are not λ-functions. Consequently, the applications $\mathsf{sRed}\,(f)\ \mathsf{sRed}\,(s)$ and $\mathsf{sRed}\,(f')\ \mathsf{sRed}\,(s')$ are not beta or eta reducible. Thus,

$$\mathsf{sRed}\,(\mathsf{sRed}\,(f)_{\Pi x:A.\ B}\ \mathsf{sRed}\,(t_A)_{A'}) = w_{\overline{B}}$$

and

$$\mathsf{sRed}\big(\mathsf{sRed}\,(f')_{\Pi x:A.\ B}\,\mathsf{sRed}\,(t'_A)_{A'}\big) = w_{\overline{B}}$$

and we yield:

$$\Delta \vdash_{\overline{T}} \mathsf{sRed}\,(\mathsf{sRed}\,(f)_{\Pi x:A'.\ B}\ \mathsf{sRed}\,(t)_A) =_{\overline{B}} \mathsf{sRed}\,(\mathsf{sRed}\,(f')_{\Pi x:A'.\ B}\ \mathsf{sRed}\,(t')_A) \quad \text{(refl)} \quad (329)$$

Otherwise the terms $\mathsf{sRed}\,(f)_{\Pi x:A.\ B}\ \mathsf{sRed}\,(t_A)_A$ and $\mathsf{sRed}\,(f')_{\Pi x:A.\ B}\ \mathsf{sRed}\,(t'_A)_A$ are almost proper with quasi-preimages consistent with type indices. It follows:

$$\mathsf{sRed}\,(\mathsf{sRed}\,(f)_{\Pi x:A.\ B}\ \mathsf{sRed}\,(t_A)_A) = \mathsf{sRed}\,(f)_{\Pi x:A.\ B}\ \mathsf{sRed}\,(t_A)_A$$

and

$$\mathsf{sRed}\,(\mathsf{sRed}\,(f')_{\Pi x:A.\ B}\ \mathsf{sRed}\,(t'_A)_A) = \mathsf{sRed}\,(f')_{\Pi x:A.\ B}\ \mathsf{sRed}\,(t'_A)_A$$

and thus:

$$\Delta \vdash_{\overline{T}} \mathsf{sRed}\,(\mathsf{sRed}\,(f)_{\Pi x:A.\ B}\ \mathsf{sRed}\,(t)_A) =_{\overline{B}} \mathsf{sRed}\,(\mathsf{sRed}\,(f')_{\Pi x:A.\ B}\ \mathsf{sRed}\,(t')_A) \quad \text{(congAppl),(326),(328)} \quad (330)$$

In either case, we concluded

$$\mathsf{sRed}\,(\mathsf{sRed}\,(f)_{\Pi x:A.\ B}\ \mathsf{sRed}\,(t)_A) =_{\overline{B}} \mathsf{sRed}\,(\mathsf{sRed}\,(f')_{\Pi x:A.\ B}\ \mathsf{sRed}\,(t')_A)\,.$$

By SR4 this is already the desired conclusion of:

$$\mathsf{sRed}\,\big(\mathsf{sRed}\,(f)_{\Pi x:A.\ B}\ \mathsf{sRed}\,(t)_A =_{\overline{B}} \mathsf{sRed}\,(f')_{\Pi x:A.\ B}\ \mathsf{sRed}\,(t')_A\big)\,.$$

(refl):

$\Delta \vdash_{\overline{T}} \mathsf{sRed}\,(t_A)_A : \overline{A}$ by assumption (331)

$\Delta \vdash_{\overline{T}} \mathsf{sRed}\,(t_A)_A =_{\overline{A}} \mathsf{sRed}\,(t_A)_A$ (refl),(331) (332)

$\Delta \vdash_{\overline{T}} \mathsf{sRed}\,(\mathsf{sRed}\,(t_A)_A =_{\overline{A}} \mathsf{sRed}\,(t_A)_A)$ SR4,(332)

(sym):

$\Delta \vdash_{\overline{T}} \mathsf{sRed}\,(t_A =_{\overline{A}} s_A)$	by assumption		(333)
$\Delta \vdash_{\overline{T}} \mathsf{sRed}\,(t_A)_A =_{\overline{A}} \mathsf{sRed}\,(s_A)_A$	SR4,(333)		(334)
$\Delta \vdash_{\overline{T}} \mathsf{sRed}\,(s_A)_A =_{\overline{A}} \mathsf{sRed}\,(t_A)_A$	(sym),(334)		(335)
$\Delta \vdash_{\overline{T}} \mathsf{sRed}\,(s_A =_{\overline{A}} t_A)$	SR4,(335)		

(beta):

$\Delta \vdash_{\overline{T}} \mathsf{sRed}\,((\lambda x_A{:}\overline{A}.\ s_B)\ t_{A'})_{B'} : \overline{B}$	by assumption		(336)

By property 6. in Lemma 30, it follows that $A = A'$. If $\mathsf{sRed}\,((\lambda x_A{:}\overline{A}.\ s_B)\ t_A)_{B'}$ is almost proper with quasi-preimage of type $B \equiv B'$, then $\mathsf{sRed}\,((\lambda x_A{:}\overline{A}.\ s_B)\ t_A)_{B'} = (\lambda x_A{:}\overline{A}.\ s_B)\ t_A$ and thus:

$\Delta \vdash_{\overline{T}} (\lambda x_A{:}\overline{A}.\ s_B)\ t_A : \overline{B}$	SR1,(336)		(337)
$\Delta \vdash_{\overline{T}} (\lambda x_A{:}\overline{A}.\ s_B)\ t_A =_{\overline{B}} s_B[x_A/t_A]$	(beta),(337)		(338)
$\Delta \vdash_{\overline{T}} \mathsf{sRed}\,((\lambda x_A{:}\overline{A}.\ s_B)\ t_A =_{\overline{B}} s_B[x_A/t_A])$	SR4,SR1,(338)		

Otherwise by SR2,

$$\mathsf{sRed}\,((\lambda x_A{:}\overline{A}.\ s_B)\ t_{A'}) = \mathsf{sRed}\,(s_B[x_A/t_{A'}])$$

and we yield:

$\Delta \vdash_{\overline{T}} \mathsf{sRed}\,((\lambda x_A{:}\overline{A}.\ s_B)\ t_{A'}) =_{\overline{B}} \mathsf{sRed}\,(s_B[x_A/t_{A'}])$	(refl),above observation		(339)
$\Delta \vdash_{\overline{T}} \mathsf{sRed}\,((\lambda x_A{:}\overline{A}.\ s_B)\ t_{A'} =_{\overline{B}} s_B[x_A/t_{A'}])$	SR4,(339)		

(eta):

$\Delta \vdash_{\overline{T}} \mathsf{sRed}\,(t_{\Pi x:A.\ B}) : \overline{A} \to \overline{B}$	by assumption		(340)
x not in Δ	by assumption		(341)

Since $\mathsf{sRed}\,(t_{\Pi x:A.\ B})$ is almost proper with quasi-preimage of type $\Pi x{:}A.\ B$, it follows that $\lambda x_A{:}\overline{A}.\ \mathsf{sRed}\,(t_{\Pi x:A.\ B})\ x_A$ is also almost proper with quasi-preimage of type $\Pi x{:}A.\ B$. It follows:

$\Delta \vdash_{\overline{T}} \mathsf{sRed}\,(t_{\Pi x:A.\ B}) =_{\overline{A}\to\overline{B}} \lambda x_A{:}\overline{A}.\ \mathsf{sRed}\,(t_{\Pi x:A.\ B})\ x_A$	(eta),(340),(341)		(342)
$\Delta \vdash_{\overline{T}} \mathsf{sRed}\,(t_{\Pi x:A.\ B} =_{\overline{A}\to\overline{B}} \lambda x_A{:}\overline{A}.\ t_{\Pi x:A.\ B}\ x_A)$	SR2,SR6,SR4,(342)		

(cong$\vdash$):

$\Delta \vdash_{\overline{T}} \mathsf{sRed}\,(F_{\mathsf{bool}} =_{\mathsf{bool}} {F'}_{\mathsf{bool}})$	by assumption		(343)
$\Delta \vdash_{\overline{T}} \mathsf{sRed}\,({F'}_{\mathsf{bool}})$	by assumption		(344)
$\Delta \vdash_{\overline{T}} \mathsf{sRed}\,(F_{\mathsf{bool}}) =_{\mathsf{bool}} \mathsf{sRed}\,({F'}_{\mathsf{bool}})$	SR4,(343)		(345)
$\Delta \vdash_{\overline{T}} \mathsf{sRed}\,(F_{\mathsf{bool}})$	(cong$\vdash$),(345),(344)		

(⇒I):

	$\Delta \vdash_{\overline{T}} \mathsf{sRed}\ (F_{\mathsf{bool}}) : \mathsf{bool}$	by assumption	(346)
$\Delta,\ \triangleright \mathsf{sRed}\ (F_{\mathsf{bool}})$	$\vdash_{\overline{T}} \mathsf{sRed}\ (G_{\mathsf{bool}})$	by assumption	(347)
	$\Delta \vdash_{\overline{T}} \mathsf{sRed}\ (F_{\mathsf{bool}}) \Rightarrow \mathsf{sRed}\ (G_{\mathsf{bool}})$	(⇒I),(346),(347)	(348)
	$\Delta \vdash_{\overline{T}} \mathsf{sRed}\ (F_{\mathsf{bool}} \Rightarrow G_{\mathsf{bool}})$	SR5,(348)	

(⇒E):

$\Delta \vdash_{\overline{T}} \mathsf{sRed}\ (F_{\mathsf{bool}} \Rightarrow G_{\mathsf{bool}})$		by assumption	(349)
$\Delta \vdash_{\overline{T}} \mathsf{sRed}\ (F_{\mathsf{bool}})$		by assumption	(350)
$\Delta \vdash_{\overline{T}} \mathsf{sRed}\ (F_{\mathsf{bool}}) \Rightarrow \mathsf{sRed}\ (G_{\mathsf{bool}})$		SR5,(349)	(351)
$\Delta \vdash_{\overline{T}} \mathsf{sRed}\ (G_{\mathsf{bool}})$		(⇒E),(351),(350)	

(boolExt):

$\Delta \vdash_{\overline{T}} \mathsf{sRed}\ (p_{\mathsf{bool}\to\mathsf{bool}}\ \mathsf{true}_{\mathsf{bool}})$		by assumption	(352)
$\Delta \vdash_{\overline{T}} \mathsf{sRed}\ (p_{\mathsf{bool}\to\mathsf{bool}}\ F_{\mathsf{bool}})$		by assumption	(353)
$\Delta \vdash_{\overline{T}} \mathsf{sRed}\ (p_{\mathsf{bool}\to\mathsf{bool}})\ \mathsf{true}$		(SR2),(SR1),(352)	(354)
$\Delta \vdash_{\overline{T}} \mathsf{sRed}\ (p_{\mathsf{bool}\to\mathsf{bool}})\ \mathsf{false}$		(SR2),(SR1),(353)	(355)
$\Delta \vdash_{\overline{T}} \forall\, x{:}\mathsf{bool}.\ \mathsf{sRed}\ (p_{\mathsf{bool}\to\mathsf{bool}})\ x$		(boolExt),(354),(355)	(356)
$\Delta \vdash_{\overline{T}} \mathsf{sRed}\ (\forall\, x{:}\mathsf{bool}.\ p_{\mathsf{bool}\to\mathsf{bool}}\ x)$		SR4,SR6,SR2,SR1,(356)	

◀

D.4 Lifting admissible HOL derivations of validity statements to DHOL

We finally have all required results to prove the soundness of the translation from DHOL to HOL.

Proof of Theorem 21. As shown in Lemma 35, we may assume that the proof of $\overline{\Gamma} \vdash_{\overline{T}} \overline{F}$ is admissible, so it only contains almost-proper terms. Consequently, whenever an equality $s =_{\overline{A}} t$ is derivable in HOL and s', t' are the quasi-preimages of s, t respectively, it follows that it's quasi-preimage $s' =_A t$ is well-typed in DHOL and thus $s'{:}A$ and $t'{:}A$. Without loss of generality (adding extra assumptions throughout the proof) we may assume that the context of the (final) conclusion is the translation of a DHOL context. By Lemma 26 the translation is term-wise injective.

Therefore, the translated conjecture is a proper validity statement with unique (quasi)-preimage in DHOL. If we can lift a derivation of the translated conjecture to a valid DHOL derivation of its quasi-preimage, the resulting derivation is a valid derivation of the original conjecture. This means, that it suffices to prove that we can lift admissible derivations of a proper validity statement S in HOL to a derivation of a quasi-preimage of S.

We prove this claim by induction on the validity rules of HOL as follows:

Given a validity rule R with assumptions $A_1, \ldots, A_n$, validity assumptions (assumptions that are validity statements) $V_1, \ldots, V_m$, non-judgement assumptions (meaning assumptions that something occurs in a context or theory) $N_1, \ldots, N_p$ and conclusion C we will show the following:

▷ Claim 36. Assuming that the A_i and the N_j hold.

1. Assume that the conclusion C is proper with quasi-preimage C^{-1}. Then the contexts C_i of the V_i are proper and the quasi-preimages of the V_i are well-formed.
2. Assume that whenever an V_i is proper its quasi-preimage (where we choose the same preimages for identical terms and types with several possible preimages) holds in DHOL and that the conclusion C is proper with quasi-preimage C^{-1}. Then, C^{-1} holds in DHOL.

Consider the first part of this claim, namely that if C is proper then the V_i are proper. Since all formulae appearing in the derivation are almost proper, this implies that the V_i themselves are proper and by construction (choice of quasi-preimage) the contexts of their quasi-preimages fit together with the context of C^{-1}.

The translation clearly implies that if an N_j holds in HOL, the corresponding non-judgement assumption ${N_j}^{-1}$ holds in DHOL (e.g. if $\overline{F}$ is an axiom in $\overline{T}$, then F must be an axiom in T).

Since the validity judgement being derived is proper, it follows from this first part of the claim that the validity assumptions of all validity rules in the derivation are proper.

By induction on the validity rules, if given an arbitrary validity rule R whose assumptions hold and whose validity assumptions all satisfy a property P we can show that P holds on the conclusion of R, then all derivable validity judgments have property P. Since all the validity assumptions and conclusions of validity rules in the derivation are proper, the property of having a derivable quasi-preimage is such a property. By this induction principle, it suffices to prove the claim for the validity rules in HOL.

We will therefore consider the validity rules one by one. For each rule we first prove the first part of the claim. Sometimes we also need that the quasi-preimages of some non-validity (typically typing) assumptions hold, so we will prove that this also follows from the conclusion being proper. Then the assumption of the second part, combined with the first part implies that the quasi-preimages of the V_i hold in DHOL and it is easy to prove that also C^{-1} holds in DHOL.

Throughout this proof we will use the notation $\tilde{t}$ to denote that t is some quasi-preimage of $\tilde{t}$. Since the translation is surjective on type-level we will only need this notation on term-level.

Validity can be shown using the rules (congλ), (eta), (congAppl), (cong$\vdash$), (beta), (refl), (sym), (assume), (axiom), ($\Rightarrow$I), ($\Rightarrow$E) and (boolExt).

(congλ):

Since the conclusion is proper, it follows that the preimage

$$\Gamma \vdash_{\mathsf{T}} \lambda x{:}A.\ t = _{\Pi x:A.\ B} \lambda x{:}A.\ t'$$

of the normalization

$$\overline{\Gamma} \vdash_{\overline{T}} \forall x{:}\overline{A}.\ \forall y{:}\overline{A}.\ \mathtt{A}^*\ x\ y \Rightarrow \mathtt{B}^*\ \tilde{t}\ \ x\ \widetilde{t'}\ \ y$$

of the conclusion is well-formed. By rule (eqTyping) and rule (sym) we obtain $\Gamma \vdash_{\mathsf{T}} \lambda x{:}A.\ t{:}\Pi x{:}A.\ B$ and $\Gamma \vdash_{\mathsf{T}} \lambda x{:}A.\ t'{:}\Pi x{:}A.\ B$ in DHOL.

$\Gamma \vdash_{\mathsf{T}} \lambda x{:}A.\ t{:}\Pi x{:}A.\ B$	see above	(357)
$\Gamma \vdash_{\mathsf{T}} \lambda x{:}A.\ t'{:}\Pi x{:}A.\ B$	see above	(358)
$\Gamma,\ y{:}A \vdash_{\mathsf{T}} (\lambda x{:}A.\ t)\ \ y{:}B$	(appl),(var$\vdash$),(357),(assume)	(359)
$\Gamma,\ y{:}A \vdash_{\mathsf{T}} (\lambda x{:}A.\ t')\ \ y{:}B$	(appl),(var$\vdash$),(358),(assume)	(360)
$\Gamma,\ y{:}A \vdash_{\mathsf{T}} (\lambda x{:}A.\ t)\ \ y = _B t[x/y]$	(beta),(359)	(361)
$\Gamma,\ y{:}A \vdash_{\mathsf{T}} (\lambda x{:}A.\ t')\ \ y = _B t'[x/y]$	(beta),(360)	(362)
$\Gamma,\ x{:}A \vdash_{\mathsf{T}} t{:}B$	α-renaming,(cong:),(361)	(363)
$\Gamma,\ x{:}A \vdash_{\mathsf{T}} t'{:}B$	α-renaming,(cong:),(362)	(364)
$\Gamma,\ x{:}A \vdash_{\mathsf{T}} t = _B t$	(=type),(363),(364)	

Clearly, $\Gamma,\ x{:}A \vdash_{\mathsf{T}} t = _B t'$ is a quasi-preimage of the validity assumption, so this proves the first part of the claim.

Regarding the second part:

$\Gamma,\ x{:}A \vdash_{\mathsf{T}} t = _B t'$	by assumption	(365)
$\Gamma \vdash_{\mathsf{T}} A\ \equiv\ A$	($\equiv$refl),(typingTp),(365)	(366)
$\Gamma \vdash_{\mathsf{T}} \lambda x{:}A.\ t = _{\Pi x:A.\ B} \lambda x{:}A.\ t'$	(congλ'),(366)	(367)

(eta):

Since the rule has no validity assumption, the first part of the claim holds.

For the second part, we still need the quasi-preimage of the assumption to hold, so we will show that it follows from the conclusion being proper.

Since the conclusion is proper, it follows that the preimage

$$\Gamma \vdash_{\mathsf{T}} t = _{\Pi x:A.\ B} \lambda x{:}A.\ t\ x$$

of the normalization

$$\overline{\Gamma} \vdash_{\overline{T}} \forall x{:}\overline{A}.\ \forall y{:}\overline{A}.\ \mathtt{A}^*\ x\ y \Rightarrow \mathtt{B}^*\ \tilde{t}\ \ x\ \left(\lambda x{:}\overline{A}.\ \tilde{t}\ \ x\right)\ \ y$$

of the conclusion is well-formed. By rule (eqTyping) and rule (sym) we obtain $\Gamma \vdash_{\mathsf{T}} t{:}\Pi x{:}A.\ B$ and $\Gamma \vdash_{\mathsf{T}} \lambda x{:}A.\ t\ \ x{:}\Pi x{:}A.\ B$ in DHOL. Clearly, $\Gamma \vdash_{\mathsf{T}} t{:}\Pi x{:}A.\ B$ is a quasi-preimage of the validity assumption, so this proves the quasi-preimage of the assumption of the rule.

Regarding the second part:

$\Gamma \vdash_{\mathsf{T}} t{:}\Pi x{:}A.\ B$	see above	(368)
$\Gamma \vdash_{\mathsf{T}} t = _{\Pi x:A.\ B} \lambda x{:}A.\ t\ x$	(etaPi),(368)	

(congAppl):

Since the conclusion is proper, it follows that the preimage

$$\Gamma \vdash_{\mathsf{T}} f\ t =_B f'\ t'$$

of the normalization

$$\mathtt{B}^*\ \widetilde{f}\ \widetilde{t}\ \widetilde{f'}\ \widetilde{t'}$$

of the conclusion is well-formed. By rule (eqTyping) and rule (sym) we obtain $\Gamma \vdash_{\mathsf{T}} f\ t : B$ and $\Gamma \vdash_{\mathsf{T}} f'\ t' : B$ in DHOL. Obviously, $\Gamma \vdash_{\mathsf{T}} t =_A t'$ and $\Gamma \vdash_{\mathsf{T}} f =_{\Pi x:A.\ B} f'$ are quasi-preimages of the validity assumptions.

Since the validity assumptions use the same context as the conclusion, it follows that they are both proper with uniquely determined context. As observed in the beginning of the proof if a proper assumption of a rule is an equality over a type $\overline{A}$, the induction hypothesis implies that the quasi-preimage of that assumption in which the equality is over type A must be well-formed. Hence both $\Gamma \vdash_{\mathsf{T}} t =_A t'$ and $\Gamma \vdash_{\mathsf{T}} f =_{\Pi x:A.\ B} f'$ are well-formed in DHOL, so we have proven the first part of the claim.

Regarding the second part of the claim:

$\Gamma \vdash_{\mathsf{T}} t =_A t'$	by assumption	(369)
$\Gamma \vdash_{\mathsf{T}} f =_{\Pi x:A.\ B} f'$	by assumption	(370)
$\Gamma \vdash_{\mathsf{T}} f\ t =_B f'\ t'$	(congAppl),(369),(370)	(371)

This is what we had to show.

(cong$\vdash$):

Since the conclusion is proper, it follows that the preimage

$$\Gamma \vdash_{\mathsf{T}} F$$

of the normalization

$$\overline{\Gamma} \vdash_{\overline{T}} \widetilde{F}$$

of the conclusion is well-formed. Thus we have $\Gamma \vdash_{\mathsf{T}} F : \mathsf{bool}$. Since the validity assumptions use the same context as the conclusion, it follows that they are both proper with uniquely determined. As observed in the beginning of the proof if a proper assumption of a rule is an equality over a type $\overline{A}$ (here $A = \overline{A} = \mathsf{bool}$), the induction hypothesis implies that the quasi-preimage of that assumption in which the equality is over type bool must be well-formed. Clearly, $\Gamma \vdash_{\mathsf{T}} F' =_{\mathsf{bool}} F$ and $\Gamma \vdash_{\mathsf{T}} F$ are the quasi-preimages of the two validity assumptions. Since the former is a validity statement about the quasi-preimage of an equality, it follows that $\Gamma \vdash_{\mathsf{T}} F' =_{\mathsf{bool}} F$ is well-formed. We have already seen that $\Gamma \vdash_{\mathsf{T}} F$ is well-typed. This shows the first part of the claim.

Regarding the second part:

$\Gamma \vdash_{\mathsf{T}} F' =_{\mathsf{bool}} F$	by assumption	(372)
$\Gamma \vdash_{\mathsf{T}} F'$	by assumption	(373)
$\Gamma \vdash_{\mathsf{T}} F$	(cong$\vdash$),(372),(373)	

(beta):

Since the rule has no validity assumptions, the first part of the claim trivially holds.

Since the conclusion is proper, it follows that the preimage

$$\Gamma\vdash_{\mathsf{T}}(\lambda x{:}A.\ s)\ t = \Pi_{x:A.\ B}\ s[x/t]$$

of the normalization

$$\overline{\Gamma}\vdash_{\overline{T}} \mathtt{B}^*\ (\lambda x{:}\overline{A}.\ \widetilde{s})\ \widetilde{t}\ \ \widetilde{s}\,[x/\widetilde{t}\,]$$

of the conclusion is well-formed. By rule (eqTyping), we obtain $\Gamma\vdash_{\mathsf{T}}(\lambda x{:}A.\ s)\ t{:}B$ in DHOL. Clearly, $\Gamma\vdash_{\mathsf{T}}(\lambda x{:}A.\ s)\ t{:}B$ is a quasi-preimage of the assumption of the rule, so we have proven that the quasi-preimage of the assumption of the rule holds in DHOL.

Regarding the second part:

$\Gamma\vdash_{\mathsf{T}}(\lambda x{:}A.\ s)\ t{:}B$	see above	(374)
$\Gamma\vdash_{\mathsf{T}}(\lambda x{:}A.\ s)\ t = \Pi_{x:A.\ B}\ s[x/t]$	(beta),(374)	

(refl):

Once again the rule has no validity assumptions, so the first part of the claim trivially holds.

Since the conclusion is proper, it follows that the preimage

$$\Gamma\vdash_{\mathsf{T}} t =_A t'$$

of the normalization

$$\overline{\Gamma}\vdash_{\overline{T}} \mathtt{A}^*\ \widetilde{t}\ \ \widetilde{t}$$

of the conclusion is well-formed. By Lemma 26 it follows that t and t' are identical so the quasi-preimage is $\Gamma\vdash_{\mathsf{T}} =_A t$. By rule (eqTyping), we obtain $\Gamma\vdash_{\mathsf{T}} t{:}A$ in DHOL, the quasi-preimage of the assumption of the rule.

Regarding the second part of the claim:

$\Gamma\vdash_{\mathsf{T}} t{:}A$	see above	(375)
$\Gamma\vdash_{\mathsf{T}} t =_A t$	(refl),(375)	

(sym):

Since the conclusion is proper, it follows that the preimage

$$\Gamma\vdash_{\mathsf{T}} t =_A s$$

of the normalization

$$\overline{\Gamma}\vdash_{\overline{T}} \mathtt{A}^*\ \widetilde{t}\ \ \widetilde{s}$$

of the conclusion is well-formed. By the rules (eqTyping) and (sym) both $\Gamma\vdash_{\mathsf{T}} t{:}A$ and $\Gamma\vdash_{\mathsf{T}} s{:}A$ follow. By rule (=type) it follows that $\Gamma\vdash_{\mathsf{T}} s =_A t$ is well-formed. Clearly, $\Gamma\vdash_{\mathsf{T}} s =_A t$ is the quasi-preimages of the validity assumption, so we have proven the first part of the claim.

Regarding the second part:

$\Gamma\vdash_{\mathsf{T}} t =_A s$	by assumption	(376)
$\Gamma\vdash_{\mathsf{T}} s =_A t$	(sym),(376)	(377)

(assume):

Once again, there are no validity assumption, so the first part of the claim is trivial.

Since the conclusion is proper, it follows that the preimage

$$\Gamma \vdash_{\mathsf{T}} F$$

of the normalization

$$\overline{\Gamma} \vdash_{\overline{T}} \widetilde{F}$$

of the conclusion is well-formed and thus $\Gamma \vdash_{\mathsf{T}} F : \mathsf{bool}$.

$\Gamma \vdash_{\mathsf{T}} F : \mathsf{bool}$	see above	(378)
$\Gamma \vdash_{\mathsf{T}} \mathsf{bool}\ \mathsf{tp}$	(typingTp),(378)	(379)
$\vdash_{\mathsf{T}} \Gamma\ \mathsf{Ctx}$	(tpCtx),(379)	(380)

The context assumption may be the translation of a context assumption in DHOL or a typing assumption added by the translation. In the latter case, F is of the form $F = \mathtt{A}^*\ x\ x$ for $x{:}A$ in Γ. In that case, the second part of the claim $\Gamma \vdash_{\mathsf{T}} F$ can be concluded as follows:

$\Gamma \vdash_{\mathsf{T}} x : A$	(var')	(381)
$\Gamma \vdash_{\mathsf{T}} x =_A t$	(refl),(381)	(382)
$\Gamma \vdash_{\mathsf{T}} F$	by assumption $F = \mathtt{A}^*\ x\ x$,(382)	

Otherwise:

$\triangleright F$ in Γ	by assumption	(383)
$\Gamma \vdash_{\mathsf{T}} F$	(assume),(383),(380)	

(axiom):

Once again, there are no validity assumption, so the first part of the claim is trivial.

Since the conclusion is proper, it follows that the preimage

$$\Gamma \vdash_{\mathsf{T}} F$$

of the normalization

$$\overline{\Gamma} \vdash_{\overline{T}} \widetilde{F}$$

of the conclusion is well-formed and thus $\Gamma \vdash_{\mathsf{T}} F : \mathsf{bool}$.

$\Gamma \vdash_{\mathsf{T}} F : \mathsf{bool}$	see above	(384)
$\Gamma \vdash_{\mathsf{T}} \mathsf{bool}\ \mathsf{tp}$	(typingTp),(384)	(385)
$\vdash_{\mathsf{T}} \Gamma\ \mathsf{Ctx}$	(tpCtx),(385)	(386)

The axiom may be the translation of an axiom in T, a typing axiom added by the translation or an axiom added for some base type A. In the first case, the second part of the claim follows by:

$\triangleright F$ in T	by assumption	(387)

$\Gamma \vdash_{\mathsf{T}} F$	(axiom),(387),(386)	

If the axiom is a typing axiom then its preimage states that some constant c of type A satisfies $\mathsf{c} =_A t$ which follows by rule (refl).

If the axiom is the PER axiom generated for some A type declared in T, then it's quasi-preimage states that equality on A implies itself which is obviously true.

(⇒I):

Since the conclusion is proper, it follows that the preimage

$$\Gamma \vdash_{\mathsf{T}} F \Rightarrow G$$

of the normalization

$$\overline{\Gamma} \vdash_{\overline{T}} \widetilde{F} \Rightarrow \widetilde{G}$$

of the conclusion is well-formed and thus $\Gamma \vdash_{\mathsf{T}} F \Rightarrow G$:bool.

$\Gamma \vdash_{\mathsf{T}} F \Rightarrow G$:bool	see above	(388)
$\Gamma \vdash_{\mathsf{T}} F$:bool	(implTypingL),(388)	(389)
$\Gamma \vdash_{\mathsf{T}} G$:bool	(implTypingR),(388)	(390)
$\Gamma, \triangleright F \vdash_{\mathsf{T}} G$:bool	(monotonic$\vdash$),(390)	

Obviously $\Gamma, \triangleright F \vdash_{\mathsf{T}} G$ is a quasi-preimage of the validity assumption of the rule, so the first part of the claim is proven.

Regarding the second part:

$\Gamma, \triangleright F \vdash_{\mathsf{T}} G$	by assumption	(391)
$\Gamma \vdash_{\mathsf{T}} F \Rightarrow G$	(⇒I),(389),(391)	

(⇒E):

Since the conclusion is proper, it follows that the preimage

$$\Gamma \vdash_{\mathsf{T}} G$$

of the normalization

$$\overline{\Gamma} \vdash_{\overline{T}} \widetilde{G}$$

of the conclusion is well-formed and thus $\Gamma \vdash_{\mathsf{T}} G$:bool.

Since the validity assumptions use the same context as the conclusion, it follows that they are both proper and uniquely determined.

Since the formula $\widetilde{F}$ (where $\overline{\Gamma} \vdash_{\overline{T}} \widetilde{F}$ is the second validity assumption) must be almost proper, it follows that its preimage F is well-typed i.e. $\Gamma \vdash_{\mathsf{T}} F$:bool.

$\Gamma \vdash_{\mathsf{T}} F$:bool	$\widetilde{F}$ almost proper	(392)
$\Gamma \vdash_{\mathsf{T}} G$:bool	see above	(393)
$\Gamma \vdash_{\mathsf{T}} F \Rightarrow G$:bool	(⇒type'),(392),(393)	

Clearly, $\Gamma \vdash_{\mathsf{T}} F \Rightarrow G$ and $\Gamma \vdash_{\mathsf{T}} F$ are quasi-preimages of the two validity assumptions of the rule, so we have proven the first part of the claim.

Regarding the second part:

$\Gamma \vdash_{\mathsf{T}} F \Rightarrow G$	by assumption	(394)
$\Gamma \vdash_{\mathsf{T}} F$	by assumption	(395)
$\Gamma \vdash_{\mathsf{T}} G$	($\Rightarrow$E),(394),(395)	

(boolExt):

Since the conclusion is proper, it follows that the preimage

$$\Gamma \vdash_{\mathsf{T}} \forall\, x{:}\mathsf{bool}.\ p\ x$$

of the normalization

$$\overline{\Gamma} \vdash_{\overline{T}} \forall\, x{:}\mathsf{bool}.\ \forall\, y{:}\mathsf{bool}.\ \mathsf{bool}^*\ x\ y \Rightarrow \mathsf{bool}^*\ (\lambda x{:}\mathsf{bool}.\ \mathsf{true})\ x\ \widetilde{\mathrm{p}}\ y$$

of the conclusion is well-formed and thus $\Gamma \vdash_{\mathsf{T}} \forall\, x{:}\mathsf{bool}.\ p\ x{:}\mathsf{bool}$. Expanding the definition of $\forall$ yields:

$\Gamma \vdash_{\mathsf{T}} \lambda x{:}\mathsf{bool}.\ \mathsf{true} =_{\Pi x:\mathsf{bool}.\ \mathsf{bool}} \lambda x{:}\mathsf{bool}.\ p\ x{:}\mathsf{bool}$	see above	(396)
$\Gamma \vdash_{\mathsf{T}} \lambda x{:}\mathsf{bool}.\ p\ x{:}\Pi x{:}\mathsf{bool}.\ \mathsf{bool}$	(eqTyping),(sym),(396)	(397)
$\Gamma \vdash_{\mathsf{T}} (\lambda x{:}\mathsf{bool}.\ p\ x)\ \ \mathsf{true}{:}\mathsf{bool}$	(appl),(397)	(398)
$\Gamma \vdash_{\mathsf{T}} (\lambda x{:}\mathsf{bool}.\ p\ x)\ \ F{:}\mathsf{bool}$	(appl),(397)	(399)
$\Gamma \vdash_{\mathsf{T}} p\ \mathsf{true} =_{\mathsf{bool}} (\lambda x{:}\mathsf{bool}.\ p\ x)\ \ \mathsf{true}$	(sym),(beta),(398)	(400)
$\Gamma \vdash_{\mathsf{T}} p\ \mathsf{false} =_{\mathsf{bool}} (\lambda x{:}\mathsf{bool}.\ p\ x)\ \ \mathsf{false}$	(sym),(beta),(399)	(401)
$\Gamma \vdash_{\mathsf{T}} p\ \mathsf{true}{:}\mathsf{bool}$	(eqTyping),(400)	
$\Gamma \vdash_{\mathsf{T}} p\ F{:}\mathsf{bool}$	(eqTyping),(401)	

Since $\Gamma \vdash_{\mathsf{T}} p\ \mathsf{true}$ and $\Gamma \vdash_{\mathsf{T}} p\ \mathsf{false}$ are clearly quasi-preimages of the two validity assumptions of the rule, we have proven the first part of the claim.

Regarding the second part:

$\Gamma \vdash_{\mathsf{T}} p\ \mathsf{true}$	by assumption	(402)
$\Gamma \vdash_{\mathsf{T}} p\ \mathsf{false}$	by assumption	(403)
$\Gamma \vdash_{\mathsf{T}} \forall\, x{:}\mathsf{bool}.\ p\ x$	(boolExt),(402),(403)	(404)

◀