Subtyping in Dependently-Typed Higher-Order Logic

Colin Rothgang[1,2] Florian Rabe[3]

Imdea Software Institute, Madrid, Spain

Universidad Politécnica de Madrid, Madrid, Spain

Computer Science, FAU Erlangen-Nürnberg, Germany

September 30, 2025

FroCoS 2025

Motivation

Dependently-typed higher-order logic (DHOL) introduced 2023

- adds dependent types to higher-order logic (HOL)
- very close to HOL
- no propositions (only classical Booleans) and equality
- ATP support via translation to HOL and one native prover
- no propositions-as-types, no type universes

unlike dependent type theory

This talk: add refinement types $A|_p$ and quotient types $A/_r$

Prior work

- original paper (CADE 2023), journal version (ToCL 2025)
  Rothgang, Rabe, Benzmüller
- native DHOL prover Niederhauser, Brown, Kaliszyk, IJCAR 2024
- extension with choice operators
  Ranalter, Brown, Kaliszyk, LPAR 2024
- TPTP language for DHOL (presented this morning)
  Ranalter, Kaliszyk, Rabe, Sutcliffe, FroCoS 2025
- extension with polymorphism Ranalter, Rabe, Kaliszyk, under review
- extension with subtyping, refinement types and quotient types
  what I'm presenting

Recap: DHOL

$$
\begin{array}{lll}
T & ::= & \circ \mid T, c : A \mid T, a : (\Pi x{:}A.\ )^* \ \mathsf{tp} \mid T, \triangleright F \quad \text{theories} \\
\Gamma & ::= & . \mid \Gamma, x : A \mid \Gamma, \triangleright F \quad \text{contexts} \\
A, B & ::= & a\ t_1 \ldots t_n \mid \Pi x{:}A.\ B \mid \mathit{bool} \quad \text{types} \\
t, f, F & ::= & c \mid x \mid f\ t \mid \lambda x{:}A.\ t \mid t_1 =_A t_2 \quad \text{terms}
\end{array}
$$

Example (lists and vectors over *obj*)

$$
\begin{array}{l}
\mathrm{nat}\colon \mathsf{tp}, \quad \mathrm{zero}\colon \mathrm{nat}, \quad \mathrm{succ}\colon \mathrm{nat} \to \mathrm{nat}, \\
\mathit{obj}\colon \mathsf{tp}, \quad \mathrm{list}\colon \mathsf{tp}, \quad \mathrm{nil}\colon \mathrm{list}, \quad \mathrm{cons}\colon \mathit{obj} \to \mathrm{list} \to \mathrm{list}, \\
\mathrm{llist}\colon \mathrm{nat} \to \mathsf{tp}, \quad \mathrm{lnil}\colon \mathrm{llist}\ \mathrm{zero}, \\
\mathrm{lcons}\colon \Pi n{:}\mathrm{nat}.\ \mathit{obj} \to \mathrm{llist}\ n \to \mathrm{llist}\ (\mathrm{succ}\ n)
\end{array}
$$

Decidability and subtyping

DHOL typing undecidable

- given $f : \Pi x{:}A.\ B\ x$, formula $f(s) =_{B(s)} f(t)$ well-typed if $s =_A t$
- acceptable, as theorem proving undecidable anyways

subtyping

- needed for refinement types
- often avoided due to undecidability
- acceptable in DHOL, where type checking already undecidable

Subtyping

Definition (subtyping)

$\Gamma \vdash A \prec: B$ abbreviates $\Gamma,\ x : A \vdash x : B$.

$\prec:$ is reflexive and transitive, anti-symmetric by axiom

We can derive the usual variance rules for Π-types:

$$\frac{\Gamma \vdash A' \prec: A \quad \Gamma,\ x{:}A' \vdash B \prec: B'}{\Gamma \vdash \Pi x{:}A.\ B \prec: \Pi x{:}A'.\ B'} \qquad \frac{\Gamma \vdash A' \equiv A \quad \Gamma,\ x{:}A' \vdash B \equiv B'}{\Gamma \vdash \Pi x{:}A.\ B \equiv \Pi x{:}A'.\ B'}$$

Adding refinement types

Extend DHOL grammar with:

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

Note

There are no term-level changes: the canonical injection is a no-op

Example (Fixed-length lists)

$$\begin{aligned}
&\mathrm{length}\colon \mathrm{list} \to \mathrm{nat} \qquad \triangleright\ \mathrm{length}\ \mathrm{nil} =_{\mathrm{nat}} \mathrm{zero}\\
&\triangleright\ \forall\, x{:}\mathit{obj}.\ \forall\, l{:}\mathrm{list}.\ \mathrm{length}\ (\mathrm{cons}\ x\ l) =_{\mathrm{nat}} \mathrm{succ}\ (\mathrm{length}\ l)\\
&\mathrm{llist}\ n\ :=\ \mathrm{list}|_{\lambda l:\mathrm{list}.\ \mathrm{length}\ l =_{\mathrm{nat}} n}
\end{aligned}$$

Inference rules for refinement types

Formation:

$$\frac{\Gamma \vdash p : A \to \mathit{bool}}{\Gamma \vdash A|_p \ \mathsf{tp}}$$

Introduction:

$$\frac{\Gamma \vdash t : A \quad \Gamma \vdash p\ t}{\Gamma \vdash t : A|_p}$$

Elimination:

$$\frac{\Gamma \vdash t : A|_p}{\Gamma \vdash t : A} \qquad \frac{\Gamma \vdash t : A|_p}{\Gamma \vdash p\ t}$$

Refinement types: derived variance and congruence rules

Variance for refinement types:

$$\frac{\Gamma \vdash A \prec: A' \quad \Gamma,\ x{:}A,\ \triangleright\, p\ x \vdash p'\ x}{\Gamma \vdash A|_p \prec: A'|_{p'}}$$

Congruence:

$$\frac{\Gamma \vdash A \equiv A' \quad \Gamma \vdash p =_{A \to bool} p'}{\Gamma \vdash A|_p \equiv A'|_{p'}}$$

Improper subtype:

$$\frac{\Gamma \vdash A\ \mathsf{tp}}{\Gamma \vdash A \equiv A|_{\lambda x:A.\ \mathsf{true}}}$$

Subtyping:

$$\frac{\Gamma \vdash A|_p\ \mathsf{tp}}{\Gamma \vdash A|_p \prec: A}$$

The dual to refinements: quotients

refinement types	quotient types
canonical injection	canonical projection
condition in introduction rule	condition in elimination rule
subtypes	supertypes

Subtype hierarchy of refinements/quotients of A:

empty subtype

$A|_{\lambda x{:}A.\ \mathsf{false}} \prec: A|_p \prec: A|_{\lambda x{:}A.\ \mathsf{true}} \equiv A \equiv A/\lambda x, y{:}A.\ x =_A y \prec: A/r \prec: A/\lambda x, y{:}A.\ \mathsf{true}$

singleton supertype

$A \prec: B$ also means $s =_A t \Rightarrow s =_B t$

Quotient types

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

Formation:

$$\frac{\Gamma \vdash A \text{ tp} \qquad \Gamma \vdash r : A \to A \to \mathit{bool} \qquad \Gamma \vdash r \text{ is equivalence}}{\Gamma \vdash {}^{A}\!/_{r} \text{ tp}}$$

Introduction:

$$\frac{\Gamma \vdash t : A \qquad \Gamma \vdash {}^{A}\!/_{r} \text{ tp}}{\Gamma \vdash t : {}^{A}\!/_{r}}$$

Example (Sets as quotients of lists)

$$\mathrm{sameElements}\ s\ t := \forall z{:}\mathit{obj}.\ s\ \in\ z =_{\mathit{bool}} t\ \in\ z$$

$$\mathrm{set} \quad := \quad {}^{\mathrm{list}}\!/_{\mathrm{sameElements}},$$

for $\in$ suitably defined

Quotients: derived rules

Variance rule:

$$\frac{\Gamma \vdash A \prec: A' \quad \Gamma,\ x{:}A,\ y{:}A,\ \triangleright\, r\ x\ y \vdash r'\ x\ y}{\Gamma \vdash \quad A/r \prec: A'/r'}$$

Congruence:

$$\frac{\Gamma \vdash A \equiv A' \quad \Gamma \vdash r =_{A \to A \to bool} r'}{\Gamma \vdash A/r \equiv A'/r'}$$

Improper quotient:

$$\frac{\Gamma \vdash A\,\mathsf{tp}}{\Gamma \vdash A \equiv A/\lambda x, y{:}A.\ x =_A y}$$

Subtyping:

$$\frac{\Gamma \vdash A/r\,\mathsf{tp}}{\Gamma \vdash A \prec: A/r}$$

Normalizing refinement– and quotient types

Double refinements:

$$\vdash (A|_p)|_{p'} \equiv A|_{\lambda x{:}A.\ p\ x \wedge p'\ x}$$

Double quotient:

$$\vdash (A/_r)/_{r'} \equiv A/_{r'}$$

refinement of quotient:

$$\vdash (A/_r)|_p \equiv (A|_p)/_r$$

Refined codomain:

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

Quotiented domain:

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

Quotiented codomain ($\succ$: direction as new axiom):

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

Normalizing partial function types

Naive original expectation

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

so any type rewritable as quotient of refinement of plain DHOL type

Reality

- *only* $\succ$: *direction derivable*
- $\prec$: *direction false, since on the left* B *may use* $p\ x$
- *only an issue for dependent function types*

  no problem if B *doesn't require* $p\ x$

Normalizing refinement– and quotient types (continued)

Theorem (Normalizing Types)

All types can be rewritten into shape $(A|_p)/r$ *where*
$A, B ::= \mathit{bool} \mid a\ t^* \mid \Pi x{:}A|_p.\ B.$

- along with derived rules induces subtype checker
- doesn't require unfolding definition of subtyping
- partial functions make this less efficient
  proving completeness and meta-theoretical reasoning also harder

Ideas are welcome!

Soundness and completeness of the translation to HOL

- translation by dependency erasure
- erased information preserved in a PER for each DHOL type
- equality translated to PER

DHOL	HOL
type A	type $\overline{A}$ and PER $A^* : \overline{A} \to \overline{A} \to \mathit{bool}$
term $t : A$	term $\overline{t} : \overline{A}$ satisfying $A^*\ \overline{t}\ \overline{t}$
equality $s =_A t$	$A^*\ \overline{s}\ \overline{t}$

Extend translation with cases:

$$\overline{A|_p} := \overline{A} \qquad (A|_p)^*\ s\ t := A^*\ \overline{s}\ \overline{t} \wedge \overline{p}\ \overline{s} \wedge \overline{p}\ \overline{t}$$

$$\overline{A/_r} := \overline{A} \qquad (A/_r)^*\ s\ t := \overline{r}\ \overline{s}\ \overline{t} \wedge A^*\ \overline{s}\ \overline{s} \wedge A^*\ \overline{t}\ \overline{t}$$

Theorem

The translation is sound and complete (i.e. $\vdash \overline{s} : \mathit{bool}$ and $\vdash \overline{s}$ imply $\vdash s$).

Theorem proving: Associativity of function composition

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

$$\begin{aligned} \mathit{Functions}\ s\ t &:= (\mathrm{set} \to \mathrm{set})|_p\,/r, && \text{where} \\ p\ f &:= \forall x{:}\mathrm{set}.\ x \in s \Rightarrow (f\ x) \in t && \text{and} \\ r\ f\ g &:= \forall x{:}\mathrm{set}.\ x \in s \Rightarrow (f\ x) =_{\mathrm{set}} (g\ x) \end{aligned}$$

$$\begin{aligned} \triangleright\ & \forall s, t, u, v{:}\mathrm{set}.\ \forall f{:}\mathit{Functions}\ s\ t.\ \forall g{:}\mathit{Functions}\ t\ u. \\ & \forall h{:}\mathit{Functions}\ u\ v.\ h \circ (g \circ f) =_{\mathit{Functions}\ s\ v} (h \circ g) \circ f. \end{aligned}$$

The theorem's translation to HOL is easily proven by HOL ATPs.

Conclusion and future work

Conclusion

- extended DHOL with refinements $A|_p$ and quotients $A/_r$

  greatly increased expressivity
- type normalization results and subtype checker
- soundness and completeness theorems

Future work

- work on normalization for refined domains
- combine subtyping with polymorphism

  add subtype-bounded polymorphism

Conclusion and future work

Conclusion

- extended DHOL with refinements $A|_p$ and quotients A/r

  greatly increased expressivity
- type normalization results and subtype checker
- soundness and completeness theorems

Future work

- work on normalization for refined domains
- combine subtyping with polymorphism

  add subtype-bounded polymorphism

Thank you for your attention!

References

Niederhauser, J., Brown, C., and Kaliszyk, C.
Tableaux for automated reasoning in dependently-typed higher-order logic.
In *Automated Reasoning* (2024), C. Benzmüller, M. Heule, and R. Schmidt, Eds., Springer.

Ranalter, D., Brown, C., and Kaliszyk, C.
Experiments with choice in dependently-typed higher-order logic.
In *Proceedings of 25th Conference on Logic for Programming, Artificial Intelligence and Reasoning* (2024), N. Bj{\o}rner, M. Heule, and A. Voronkov, Eds., vol. 100 of *EPiC Series in Computing*, EasyChair, pp. 311–320.

Ranalter, D., Kaliszyk, C., Rabe, F., and Sutcliffe, G.
The Dependently Typed Higher-Order Form for the TPTP World.
In *Frontiers of Combining Systems* (2025), R. Thiemann and C. Weidenbach, Eds., Springer.

Ranalter, D., Rabe, F., and Kaliszyk, C.
Polymorphic Theorem Proving for DHOL, 2025.
under review.

Rothgang, C., Rabe, F., and Benzmüller, C.
Theorem Proving in Dependently Typed Higher-Order Logic.
In *Automated Deduction* (2023), B. Pientka and C. Tinelli, Eds., Springer, pp. 438–455.

Additional inference rules

$$\frac{\Gamma \vdash A \prec: B \quad \Gamma \vdash B \prec: A}{\Gamma \vdash A \equiv B} \qquad \frac{\Gamma \vdash p{:}A \to \mathit{bool}}{\Gamma \vdash A|_p \ \mathsf{tp}}$$

$$\frac{\Gamma \vdash t{:}A \quad \Gamma \vdash p\ t}{\Gamma \vdash t{:}A|_p} \qquad \frac{\Gamma \vdash t{:}A|_p}{\Gamma \vdash t{:}A} \qquad \frac{\Gamma \vdash t{:}A|_p}{\Gamma \vdash p\ t}$$

$$\frac{\Gamma \vdash s =_A t \quad \Gamma \vdash p\ s}{\Gamma \vdash s =_{A|_p} t}$$

$$\frac{\Gamma \vdash A\ \mathsf{tp} \quad \Gamma \vdash r{:}A \to A \to \mathit{bool} \quad \Gamma \vdash \mathrm{EqRel}(r)}{\Gamma \vdash A/r\ \mathsf{tp}}$$

Additional inference rules (continued)

$$\frac{\Gamma \vdash t : A \qquad \Gamma \vdash A/r \ \mathsf{tp}}{\Gamma \vdash t : A/r}$$

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

$$\frac{\Gamma \vdash s : A \qquad \Gamma \vdash t : A \qquad \Gamma \vdash r : A \to A \to \mathit{bool} \qquad \Gamma \vdash \mathrm{EqRel}(r)}{\Gamma \vdash (s =_{A/r} t) =_{\mathit{bool}} (r\ s\ t)}$$

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