Toward a Theory of Contexts of Assumptions in Logical Frameworks

Amy Felty

University of Ottawa
Inria Sophia Antipolis Méditerranée

TYPES Meeting, May 12, 2014

Joint work with
Alberto Momigliano and Brigitte Pientka

Motivation: Comparing Systems

- We focus on logical frameworks that support the use of *higher-order abstract syntax*.
  - Commonalities:
    - ⋆ encode object-level binders with meta-level binders
    - ⋆ support for alpha-renaming and substitution
    - ⋆ encode axioms and inference rules using *hypothetical and parametric judgments*
  - Differences:
    - ⋆ how a system supports reasoning *about* hypothetical and parametric derivations, which requires support for *contexts* to keep track of hypotheses
    - ⋆ other features...

Comparing Systems (continued)

- Case studies we consider are in the domain of meta-theory of programming languages.
- Systems we have studied include:
  - based on type theory: Twelf, Beluga
  - based on proof theory: Abella, Hybrid
- We are also designing an Open challenge problem Repository for systems supporting reasoning with BInders (ORBI), for sharing HOAS benchmark problems. (Can be thought of as an intermediate language between *OTT* and *TPTP*.)
- We want to relate one framework to another with the aim of transferring theorems and proofs (some similar goals to the ProofCert project).

Outline

A First Example: Polymorphic λCalculus

Grammar: Types and Terms (does not enforce scope)

$$\begin{array}{lcl} \text{Types } A, B & ::= & \alpha \\ & & \mid \mathsf{arr}\ A\ B \\ & & \mid \mathsf{all}\ \alpha.A \end{array} \qquad \begin{array}{lcl} \text{Terms M} & ::= & x \\ & & \mid \mathsf{lam}\ x.M \mid \mathsf{app}\ M\ N \\ & & \mid \mathsf{tlam}\ \alpha.M \quad \mid \mathsf{tapp}\ M\ A \end{array}$$

Alternative: Well-Formed Terms Martin-Löf Style (enforces scope)

$$\frac{}{\mathsf{is\_tm}\ x}\ tm_v \qquad \frac{}{\mathsf{is\_tp}\ \alpha}\ tp_v$$

$$\frac{\begin{array}{c}\mathsf{is\_tm}\ M \\ \vdots \end{array}}{\mathsf{is\_tm}\ (\mathsf{lam}\, x.\, M)}\ tm_l^{x,tm_v} \qquad \frac{\begin{array}{c}\mathsf{is\_tm}\ M \\ \vdots \end{array}}{\mathsf{is\_tm}\ (\mathsf{tlam}\, \alpha.\, M)}\ tm_{tl}^{\alpha,tp_v}$$

$$\frac{\mathsf{is\_tm}\ M_1 \qquad \mathsf{is\_tm}\ M_2}{\mathsf{is\_tm}\ (\mathsf{app}\ M_1\ M_2)}\ tm_a \qquad \frac{\mathsf{is\_tm}\ M \qquad \mathsf{is\_tp}\ A}{\mathsf{is\_tm}\ (\mathsf{tapp}\ M\ A)}\ tm_{ta}$$

A Second Example (with Implicit Contexts)

Rules for "algorithmic" equality for the untyped λcalculus:

$$\frac{}{\text{is\_tm } x}\, x \qquad \frac{}{\text{aeq } x\ x}\, ae_v$$

$$\vdots$$

$$\frac{\text{aeq } M\ N}{\text{aeq } (\text{lam } x.\, M)\ (\text{lam } x.\, N)}\, ae_l^{x, ae_v} \qquad \frac{\text{aeq } M_1\ N_1 \quad \text{aeq } M_2\ N_2}{\text{aeq } (\text{app } M_1\ M_2)\ (\text{app } N_1\ N_2)}\, ae_a$$

+ Context-free representation scales from grammars to judgments
- Can we tell open vs. closed object?
- Meta-reasoning about such judgments requires explicit structured contexts.
- Explicit structural properties of assumptions are also important.

Explicit Contexts

Examples of contexts occurring in the above examples:

$$\begin{array}{lllll} \text{Type Context} & \Gamma & ::= & \cdot \mid \Gamma, \mathsf{is\_tp}\ \alpha \\ \text{Term/Type Context} & \Gamma & ::= & \cdot \mid \Gamma, \mathsf{is\_tp}\ \alpha \mid \Gamma, \mathsf{is\_tm}\ x \\ \text{Equality Context} & \Gamma & ::= & \cdot \mid \Gamma, \mathsf{is\_tm}\ x, \mathsf{aeq}\ x\ x \end{array}$$

In the latter, note that we are introducing the variable x *together* with the assumption aeq x x.

Issue: The use of ',' is ambiguous.

Our view: Contexts are structured sequences. We use ';' to separate atoms inside a "block."

$$\text{Equality Context} \quad \Gamma \quad ::= \quad \cdot \mid \Gamma, \mathsf{is\_tm}\ x; \mathsf{aeq}\ x\ x$$

Contexts as Structured Sequences

- A context is a sequence of declarations D where a declaration is a block of individual atomic assumptions separated by ';', which binds tighter than ','.

$$\begin{array}{lllll} \text{Atom} & A & & & \\ \text{Block of declaration} & D & ::= & A \mid D; A \\ \text{Context} & \Gamma & ::= & \cdot \mid \Gamma, D \\ \text{Schema} & S & ::= & D_s \mid D_s \mid S \end{array}$$

- A *schema* classify a context, where '|' describes alternatives.

$$\begin{array}{lll} S_{\alpha x} & ::= & \text{is\_tp } \alpha \mid \text{is\_tm } x \\ S_{xa} & ::= & \text{is\_tm } x; \text{aeq } x\ x \end{array}$$

- There are typing rules relating context and schemas (not shown here).
- **Convention:** $\Phi_{\alpha x}$ describes a context with schema $S_{\alpha x}$.

Polymorphic λCalculus Revisited (with Explicit Contexts)

$$\frac{\mathsf{is\_tm}\ x \in \Phi_{\alpha x}}{\Phi_{\alpha x} \vdash \mathsf{is\_tm}\ x}\ tm_v$$

$$\frac{\Phi_{\alpha x}, \mathsf{is\_tm}\ x \vdash \mathsf{is\_tm}\ M}{\Phi_{\alpha x} \vdash \mathsf{is\_tm}\ (\mathsf{lam}\, x.\, M)}\ tm_l \qquad \frac{\Phi_{\alpha x} \vdash \mathsf{is\_tm}\ M_1 \quad \Phi_{\alpha x} \vdash \mathsf{is\_tm}\ M_2}{\Phi_{\alpha x} \vdash \mathsf{is\_tm}\ (\mathsf{app}\ M_1\ M_2)}\ tm_a$$

$$\frac{\Phi_{\alpha x}, \mathsf{is\_tp}\ \alpha \vdash \mathsf{is\_tm}\ M}{\Phi_{\alpha x} \vdash \mathsf{is\_tm}\ (\mathsf{tlam}\, \alpha.\, M)}\ tm_{tl} \qquad \frac{\Phi_{\alpha x} \vdash \mathsf{is\_tm}\ M \quad \Phi_{\alpha x} \vdash \mathsf{is\_tp}\ A}{\Phi_{\alpha x} \vdash \mathsf{is\_tm}\ (\mathsf{tapp}\ M\ A)}\ tm_{ta}$$

Structural Rules

- More fine-grained view of structural rules, which can be applied inside a block or to a whole context.
- Slightly unusual presentation of rules based on

  operations on declarations:

  - Let $\mathrm{rm}_A : S \to S'$ be a total function taking $D \in S$ and returning $D' \in S'$ where D' is D with A removed, if $A \in D$; otherwise $D' = D$.
  - Let $\mathrm{perm}_\pi : S \to S'$ be a total function which permutes the elements of $D \in S$ according to π to obtain $D' \in S'$.

Example Operations on Declarations

$$
\begin{array}{lcl}
S_{\alpha x} & ::= & \mathsf{is\_tp}\ \alpha \mid \mathsf{is\_tm}\ x \\
S_{xa} & ::= & \mathsf{is\_tm}\ x; \mathsf{aeq}\ x\ x
\end{array}
$$

$$
\begin{array}{lcl}
S_{\alpha} & ::= & \mathsf{is\_tp}\ \alpha \\
S_{x} & ::= & \mathsf{is\_tm}\ x
\end{array}
$$

$$\mathsf{rm}_{\mathsf{aeq}\ x\ x} : S_{xa} \to S_x = \lambda d.\mathsf{case}\ d\ \mathsf{of}\ \mathsf{is\_tm}\ y; \mathsf{aeq}\ y\ y \mapsto \mathsf{is\_tm}\ y$$

$$
\begin{array}{lll}
\mathsf{rm}_{\mathsf{is\_tm}\ x} : S_{\alpha x} \to S_\alpha = \lambda d.\mathsf{case}\ d\ \mathsf{of} & \mid \mathsf{is\_tp}\ \alpha \mapsto \mathsf{is\_tp}\ \alpha \\
 & \mid \mathsf{is\_tm}\ y \mapsto \cdot
\end{array}
$$

(Note that the latter "removes" whole declarations.)

Structural Properties of Declarations

- Declaration Weakening:

$$\frac{\Gamma, \mathrm{rm}_A(D), \Gamma' \vdash J}{\Gamma, D, \Gamma' \vdash J}\ d\text{-}wk$$

- Declaration Strengthening:

$$\frac{\Gamma, D, \Gamma' \vdash J}{\Gamma, \mathrm{rm}_A(D), \Gamma' \vdash J}\ d\text{-}str(\dagger)$$

  with the proviso ($\dagger$) that A is irrelevant to J (e.g., *subordination*)

- Declaration Exchange

$$\frac{\Gamma, D, \Gamma' \vdash J}{\Gamma, \mathrm{perm}_\pi(D), \Gamma' \vdash J}\ d\text{-}exc$$

Structural Properties of Contexts

We extended those operations to act on contexts (rm^*_A, perm^*_π):

- Context Weakening

$$\frac{\mathrm{rm}^*_A(\Gamma) \vdash J}{\Gamma \vdash J}\ \textit{c-wk}$$

- Context Strengthening

$$\frac{\Gamma \vdash J}{\mathrm{rm}^*_A(\Gamma) \vdash J}\ \textit{c-str}(\dagger)$$

- Context Exchange

$$\frac{\Gamma \vdash J}{\mathrm{perm}^*_\pi(\Gamma) \vdash J}\ \textit{c-exc}$$

Example Revisited

- Recall:

$$\begin{array}{lcl} S_{\alpha x} & ::= & \mathsf{is\_tp}\ \alpha \mid \mathsf{is\_tm}\ x \\ S_{\alpha} & ::= & \mathsf{is\_tp}\ \alpha \\ \mathsf{rm}_{\mathsf{is\_tm}\ x} & : & S_{\alpha x} \to S_{\alpha} \end{array}$$

- For the rule for well formed type application, we wrote:

$$\frac{\Phi_{\alpha x} \vdash \mathsf{is\_tm}\ M \quad \Phi_{\alpha x} \vdash \mathsf{is\_tp}\ A}{\Phi_{\alpha x} \vdash \mathsf{is\_tm}\ (\mathsf{tapp}\ M\ A)}\ tm_{ta}$$

  Note that we also know $\mathsf{rm}^{*}_{\mathsf{is\_tm}\ x}(\Phi_{\alpha x}) \vdash \mathsf{is\_tp}\ A$

- Furthermore, we really have the following rule, where Γ is any context that can be strengthened to a context satisfying schema $S_{\alpha x}$ (and then further strengthened in the right premise).

$$\frac{\Gamma \vdash \mathsf{is\_tm}\ M \quad \Gamma \vdash \mathsf{is\_tp}\ A}{\Gamma \vdash \mathsf{is\_tm}\ (\mathsf{tapp}\ M\ A)}\ tm_{ta}$$

Reasoning and Contexts

Rules for "declarative" equality for the untyped λcalculus:

$$S_{xd} ::= \mathsf{is\_tm}\ x; \mathsf{deq}\ x\ x$$

$$\frac{\mathsf{deq}\ x\ x \in \Phi_{xd}}{\Phi_{xd} \vdash \mathsf{deq}\ x\ x}\ de_v \qquad \frac{\Phi_{xd}, \mathsf{is\_tm}\ x; \mathsf{deq}\ x\ x \vdash \mathsf{deq}\ M\ N}{\Phi_{xd} \vdash \mathsf{deq}\ (\mathsf{lam}\, x.\, M)\ (\mathsf{lam}\, x.\, N)}\ de_l$$

$$\frac{\Phi_{xd} \vdash \mathsf{deq}\ M_1\ N_1 \quad \Phi_{xd} \vdash \mathsf{deq}\ M_2\ N_2}{\Phi_{xd} \vdash \mathsf{deq}\ (\mathsf{app}\ M_1\ M_2)\ (\mathsf{app}\ N_1\ N_2)}\ de_a \qquad \frac{}{\Phi_{xd} \vdash \mathsf{deq}\ M\ M}\ de_r$$

$$\frac{\Phi_{xd} \vdash \mathsf{deq}\ N\ M}{\Phi_{xd} \vdash \mathsf{deq}\ M\ N}\ de_s \qquad \frac{\Phi_{xd} \vdash \mathsf{deq}\ M\ L \quad \Phi_{xd} \vdash \mathsf{deq}\ L\ N}{\Phi_{xd} \vdash \mathsf{deq}\ M\ N}\ de_t$$

Attempt (Completeness)

If $\Gamma_1 \vdash \mathsf{deq}\ M\ N$, *then* $\Gamma_2 \vdash \mathsf{aeq}\ M\ N$.

This statement does not contain enough information about how the two contexts Γ_1 and Γ_2 are related.

Two Approaches

Attempt (Completeness)

If $\Gamma_1 \vdash \mathsf{deq}\ M\ N$, *then* $\Gamma_2 \vdash \mathsf{aeq}\ M\ N$.

1. Context relations approach (R). Assume that Γ_1 and Γ_2 satisfy the appropriate schemas, and then specify how they are *related*.

$$S_{xd} ::= \mathsf{is\_tm}\ x; \mathsf{deq}\ x\ x \qquad S_{xa} ::= \mathsf{is\_tm}\ x; \mathsf{aeq}\ x\ x$$

(This approach is used by Abella and Hybrid.)

2. Generalized context approach (G). Use a single context in the theorem that contains all assumptions in the relevant judgments.

$$S_{da} ::= \mathsf{is\_tm}\ x; \mathsf{deq}\ x\ x; \mathsf{aeq}\ x\ x$$

(This approach is used by Twelf and Beluga.)

Generalized Contexts

$$S_{da} ::= \text{is\_tm } x; \text{deq } x\ x; \text{aeq } x\ x$$

Theorem (Completeness, G Version)

If $\Phi_{da} \vdash \text{deq } M\ N$, *then* $\Phi_{da} \vdash \text{aeq } M\ N$.

Proof of lambda case:

$$\frac{\Gamma, \text{is\_tm } x; \text{deq } x\ x \vdash \text{deq } M\ N}{\Gamma \vdash \text{deq } (\text{lam } x.\ M)\ (\text{lam } x.\ N)}\ de_l$$

$\Phi_{da} \vdash \text{deq } (\text{lam } x.\ M)\ (\text{lam } x.\ N)$	by assumption
$\Phi_{da}, \text{is\_tm } x; \text{deq } x\ x \vdash \text{deq } M\ N$	by de_l
$\Phi_{da}, \text{is\_tm } x; \text{deq } x\ x; \text{aeq } x\ x \vdash \text{deq } M\ N$	by *d-wk*
$\Phi_{da}, \text{is\_tm } x; \text{deq } x\ x; \text{aeq } x\ x \vdash \text{aeq } M\ N$	by i.h.
$\Phi_{da}, \text{is\_tm } x; \text{aeq } x\ x \vdash \text{aeq } M\ N$	by *d-str*
$\Phi_{da} \vdash \text{aeq } (\text{lam } x.\ M)\ (\text{lam } x.\ N)$	by ae_l

Context Relations

$$S_{xd} ::= \text{is\_tm } x; \text{deq } x\ x \qquad S_{xa} ::= \text{is\_tm } x; \text{aeq } x\ x$$

Theorem (Completeness, R Version)

Assume $\Phi_{xd} \sim \Phi_{xa}$. *If* $\Phi_{xd} \vdash \text{deq } M\ N$, *then* $\Phi_{xa} \vdash \text{aeq } M\ N$.

1. We can define $\Phi_{xd} \sim \Phi_{xa}$ using rm^*. Recall:

$$S_{da} ::= \text{is\_tm } x; \text{deq } x\ x; \text{aeq } x\ x$$

   $\Phi_{xd} \sim \Phi_{xa}$ iff there is a Φ_{da} satisfying S_{da} such that $\Phi_{xd} = \text{rm}^*_{\text{aeq } x\ x}(\Phi_{da})$ and $\Phi_{xa} = \text{rm}^*_{\text{deq } x\ x}(\Phi_{da})$.
2. Alternatively, we can define this relation inductively:

$$\frac{}{\cdot \sim \cdot} \qquad \frac{\Phi_{xd} \sim \Phi_{xa}}{\Phi_{xd}, \text{is\_tm } x; \text{deq } x\ x \sim \Phi_{xa}, \text{is\_tm } x; \text{aeq } x\ x}$$

Context Relations: Lambda Case Revisited

Theorem (Completeness, R Version)

Assume $\Phi_{xd} \sim \Phi_{xa}$. *If* $\Phi_{xd} \vdash \mathsf{deq}\ M\ N$, *then* $\Phi_{xa} \vdash \mathsf{aeq}\ M\ N$.

Proof of lambda case:

$$\frac{\Gamma, \mathsf{is\_tm}\ x; \mathsf{deq}\ x\ x \vdash \mathsf{deq}\ M\ N}{\Gamma \vdash \mathsf{deq}\ (\mathsf{lam}\, x.\, M)\ (\mathsf{lam}\, x.\, N)}\ de_l$$

$\Phi_{xd} \vdash \mathsf{deq}\ (\mathsf{lam}\, x.\, M)\ (\mathsf{lam}\, x.\, N)$	by assumption
$\Phi_{xd}, \mathsf{is\_tm}\ x; \mathsf{deq}\ x\ x \vdash \mathsf{deq}\ M\ N$	by de_l
$\Phi_{xd}, \mathsf{is\_tm}\ x; \mathsf{deq}\ x\ x \sim \Phi_{xa}, \mathsf{is\_tm}\ x; \mathsf{aeq}\ x\ x$	by def $\sim$
$\Phi_{xa}, \mathsf{is\_tm}\ x; \mathsf{aeq}\ x\ x \vdash \mathsf{aeq}\ M\ N$	by i.h.
$\Phi_{xa} \vdash \mathsf{aeq}\ (\mathsf{lam}\, x.\, M)\ (\mathsf{lam}\, x.\, N)$	by ae_l

Promotion

$$\begin{array}{lcl} S_{xa} & ::= & \text{is\_tm } x; \text{aeq } x\ x \\ S_{da} & ::= & \text{is\_tm } x; \text{deq } x\ x; \text{aeq } x\ x \end{array}$$

Proving completeness involves proving admissibility of reflexivity, symmetry, and transitivity. We consider the G version of reflexivity.

Lemma (Admissibility of Reflexivity, G Version)

If $\Phi_{xa} \vdash \text{is\_tm } M$ *then* $\Phi_{xa} \vdash \text{aeq } M\ M$.

Before using this lemma in the proof of completeness of algorithmic equality with respect to declarative equality, we must first "promote" it first to the larger context used in that theorem.

Lemma (Promotion, G Version)

If $\Phi_{da} \vdash \text{is\_tm } M$ *then* $\Phi_{da} \vdash \text{aeq } M\ M$.

Proving Promotion

$$\begin{array}{lcl} S_{xa} & ::= & \text{is\_tm } x; \text{aeq } x\ x \\ S_{da} & ::= & \text{is\_tm } x; \text{deq } x\ x; \text{aeq } x\ x \end{array}$$

Lemma (Promotion, G Version)

If $\Phi_{da} \vdash \text{is\_tm } M$ *then* $\Phi_{da} \vdash \text{aeq } M\ M$.

Proof:

$\Phi_{da} \vdash \text{is\_tm } M$	by assumption
$\Phi_{xa} \vdash \text{is\_tm } M$	by *c-str*
$\Phi_{xa} \vdash \text{aeq } M\ M$	by previous lemma
$\Phi_{da} \vdash \text{aeq } M\ M$	by *c-wk*

- In general, proofs of promotion for G versions of theorems require a combination of strengthening and weakening on contexts.
- R versions of promotion involve strengthening and/or weakening of one or both sides of a context relation.

Current and Future Work (1)

What started as work on comparing HOAS systems is bearing additional fruit.

- Translating theorems and proofs between systems
  - A possible unification of how contexts are mechanized in type theory and proof theory tools
  - Formally relating G and R versions will likely be an important step.
- Designing ORBI (Open challenge problem Repository for systems supporting reasoning with BInders)
  - A common intermediate language for benchmark sharing that uses a Beluga-like syntax enriched with *directives* so that the `ORBI2X` tools will compile it into legal Twelf/Beluga, Abella/Hybrid, etc.

Current and Future Work (2)

- Tool support
  - Many common lemmas such as structural rules and promotion lemmas. Wouldn't it be nice to have your logical framework support this?
  - A tool for parsing and translating ORBI syntax to the Coq version of Hybrid is under development. It is designed to be easily adapted to output Abella scripts. [Habli & Felty, PXTP 2013].

References

- Felty, Momigliano, & Pientka, The Next 700 Challenge Problems for Reasoning with Higher-Order Abstract Syntax Representations, 2014
  - Part 1—A Foundational View
    - ⋆ theory of contexts of assumptions (this talk)
    - ⋆ benchmark problems
    - ⋆ ORBI 0.1
  - Part 2—A Survey
    - ⋆ benchmark problems applied to Twelf, Beluga, Hybrid, and Abella
    - ⋆ comparison and discussion
- Open challenge problem Repository for systems reasoning with BInders: `https://github.com/pientka/ORBI/`