Toward a Theory of Contexts of Assumptions in Logical Frameworks

Amy Felty[1], Alberto Momigliano[2], and Brigitte Pientka[3]

[1] School of Electrical Engineering and Computer Science, University of Ottawa, Canada, afelty@eecs.uottawa.ca
[2] Dipartimento di Informatica, Università degli Studi di Milano, Italy, momigliano@di.unimi.it
[3] School of Computer Science, McGill University, Montreal, Canada, bpientka@cs.mcgill.ca

In the beginning Gentzen created natural deduction, but then He switched to the sequent calculus in order to sort out the meta-theory. Something similar happened to logical frameworks supporting higher-order abstract syntax (HOAS): first Edinburgh LF adopted Martin-Löf's parametric-hypothetical judgments to encode object logics in such a way that contexts were left *implicit*. Later on, Twelf [5] had to provide some characterization of contexts (regular worlds) to verify the meta-theory of those very object logics. The same applies to λProlog vs. Abella [3] and Hybrid [2] and, in a more principled way, to Beluga [4].

One may argue that, prior to Girard, proof-theory had been somewhat oblivious to what contexts look like. Even sub-structural logics view a context of assumptions as a *flat* collection of formulas $A_1, A_2, \ldots, A_n$ listing its elements separated by commas. However, this turns out to be inadequate once we mechanize this matter, as it ignores that assumptions come in *blocks*. Consider as an object logic the typing rules for the polymorphic lambda-calculus:

$$
\begin{array}{c}
\overline{x\ \mathsf{term}}\ tm_x \qquad \overline{x : T_1}\ of_v \\
\vdots \\
\dfrac{M : T_2}{(\mathsf{lam}\, x.\, M) : (\mathsf{arr}\, T_1 T_2)}\ of_l^{tm_x, of_v}
\end{array}
\qquad\qquad
\begin{array}{c}
\overline{\alpha\ \mathsf{tp}}\ tp_v \\
\vdots \\
\dfrac{M : T_1}{(\mathsf{tlam}\, \alpha.\, M) : (\mathsf{all}\, \alpha.\, T)}\ of_{tl}^{tp_v}
\end{array}
$$

$$
\dfrac{M_1 : (\mathsf{arr}\, T_1 T_2) \quad M_2 : T_1}{(\mathsf{app}\ M_1\ M_2) : T_2}\ of_a
\qquad\qquad
\dfrac{M : (\mathsf{all}\, \alpha.\, T_1) \quad T_2\ \mathsf{tp}}{(\mathsf{tapp}\ M\ T_1) : [T_2/\alpha]T_1}\ of_{ta}
$$

We have proposed in [1] to view contexts as *structured sequences* of declarations D, where a declaration is a block of *unique* (atomic) assumptions separated by ';'.

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

A schema classifies contexts and consists of declarations D_s, possibly more general than those occurring in a concrete context having schema S. This yields for the above example

$$
\begin{array}{lcl}
\Gamma & ::= & \cdot \mid \Gamma, (x\ \mathsf{term}; x : T) \mid \Gamma, \alpha\ \mathsf{tp} \\
S & ::= & \alpha\ \mathsf{tp} + (x\ \mathsf{term}; x : T)
\end{array}
$$

where, e.g., the context $\alpha_1\ \mathsf{tp},\ (x_1\ \mathsf{term}; x_1 : (\mathsf{arr}\ \ \alpha_1\ \alpha_1)),\ (x_2\ \mathsf{term}; x_2 : \alpha_1)$ has schema S.

Since contexts are structured sequences, they admit structural *properties* on the level of sequences (for example by adding a new declaration) as well as inside a block of declarations (for example by adding an element to an existing declaration). We distinguish also between structural properties of a *concrete* context and structural properties of *all* contexts of a given schema.

We give a unified treatment of all such weakening/strengthening/exchange re-arrangements via total operations rm and perm that *remove* an element of a declaration, and *permute* elements within a declaration. For example, declaration weakening can be seen as:

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

Suppose now that we want to prove in a logical framework some meta-theorem involving different contexts, say "if $\Gamma_1 \vdash J_1$ then $\Gamma_2 \vdash J_2$", for Γ_i of schema S_i. HOAS-based logical frameworks have so far pursued two apparently different options:

(G) We reinterpret the statement in a *generalized context* containing all the relevant assumptions—we call this the *generalized context* approach, as taken in Twelf and Beluga—and prove "if $\Gamma_1 \cup \Gamma_2 \vdash J_1$ then $\Gamma_1 \cup \Gamma_2 \vdash J_2$", where "$\cup$" denotes the *join* of the two contexts.

(R) We state how two (or more) contexts are *related*—we call this the *context relations* approach. The statement becomes therefore "if $\Gamma_1 \sim \Gamma_2$ and $\Gamma_1 \vdash J_1$ then $\Gamma_2 \vdash J_2$", with an explicit and typically inductive definition of this relation. This approach is taken in Abella and Hybrid.

If we had a common grounding of both approaches, this would pave the way toward moving proofs from one system to another, in particular breaking the type/proof theory barrier. It turns out, roughly, that a context relation can be seen as the graph of one or more appropriate rm operation on a generalized context. Further, if we take the above join metaphor seriously, we can organize declarations and contexts in a *semi-lattice*, where $x \preceq y$ holds iff x can be reached from y by some rm operation on y. A generalized context will indeed be the (lattice-theoretic) join of two contexts and context relations can be identified by navigating the lattice starting from the join of the to-be-related contexts. Our ongoing effort is to use the lattice structure to give a declarative account *promotion/demotion* of theorems (known in the Twelf lingo as "context subsumption"), where a statement proven in a certain context can be used in a "related" one. We may formulate subsumption rules akin to upward and downward casting over the lattice order.

This work also has a practical outcome in our ongoing work designing *ORBI* (Open challenge problem Repository for systems supporting reasoning with BInders), a repository for sharing benchmark problems and their solutions for HOAS-based systems, in the spirit of TPTP [6].

References

[1] A. Felty, A. Momigliano, and B. Pientka. The next 700 challenge problems for reasoning with higher-order abstract syntax representations: Part 1—a foundational view. Submitted, 2014.

[2] A. P. Felty and A. Momigliano. Hybrid: A definitional two-level approach to reasoning with higher-order abstract syntax. *Journal of Automated Reasoning*, 48(1):43–105, 2012.

[3] A. Gacek. The Abella interactive theorem prover (system description). In *IJCAR 2008*, volume 5195 of *LNCS*, pages 154–161. Springer, 2008.

[4] B. Pientka and J. Dunfield. Beluga:a framework for programming and reasoning with deductive systems (system description). In *IJCAR 2010*, volume 6173 of *LNCS*, pages 15–21. Springer, 2010.

[5] C. Schürmann. The Twelf proof assistant. In *22nd International Conference on Theorem Proving in Higher Order Logics*, volume 567 of *LNCS*, pages 79–83. Springer, 2009.

[6] G. Sutcliffe. The TPTP problem library and associated infrastructure. *Journal of Automated Reasoning*, 43(4):337–362, 2009.