Appendix to "Reasoning with Higher-Order Abstract Syntax and Contexts: A Comparison" [1]

Amy Felty[1] and Brigitte Pientka[2]

[1] SITE, University of Ottawa, Ottawa, Canada
afelty@site.uottawa.ca
[2] School of Computer Science, McGill University, Montreal, Canada
bpientka@cs.mcgill.ca

A Appendix

This appendix includes and expands Section 2 of the paper [1] in order to make a more complete version of the sample problems available electronically. The expanded material includes definitions that were omitted from the paper as well as some additional theorems.

We intend to also make available implementations in various systems, as well as additional problems and their implementations. Some implementations are already available at `http://complogic.cs.mcgill.ca/beluga/benchmarks`.

A.1 Equality reasoning for lambda-terms

We begin by defining the syntax of the (untyped) lambda-calculus together with a declarative definition of equality which includes reflexivity and transitivity in addition to the structural rules, and then the algorithmic version of equality, which concentrates only on the structural rules. We model the declarative definition of equality by the judgment $\Psi \vdash \mathsf{equal}\ M\ N$ and the algorithmic one by the judgment $\Phi \vdash \mathsf{eq}\ M\ N$ and carefully define the context Ψ and Φ. The goal is to prove these two versions of equality to be equivalent.

Term $M ::= y \mid \mathsf{lam}\, x.\, M \mid \mathsf{app}\ M_1\ M_2$

Context $\Phi ::= \cdot \mid \Phi, \mathsf{equal}\ x\ x$
Context $\Psi ::= \cdot \mid \Psi, \mathsf{eq}\ x\ x$

Algorithmic Equality

$$\frac{\mathsf{eq}\ x\ x \in \Psi}{\Psi \vdash \mathsf{eq}\ x\ x} \qquad \frac{\Psi, \mathsf{eq}\ x\ x \vdash \mathsf{eq}\ M\ N}{\Psi \vdash \mathsf{eq}\ (\mathsf{lam}\, x.\, M)\ (\mathsf{lam}\, x.\, N)} \qquad \frac{\Psi \vdash \mathsf{eq}\ M_1\ N_1 \quad \Psi \vdash \mathsf{eq}\ M_2\ N_2}{\Psi \vdash \mathsf{eq}\ (\mathsf{app}\ M_1\ M_2)\ (\mathsf{app}\ N_1\ N_2)}$$

Declarative Equality

$$\frac{\mathsf{equal}\ x\ x \in \Phi}{\Phi \vdash \mathsf{equal}\ x\ x} \qquad \frac{\Phi, \mathsf{equal}\ x\ x \vdash \mathsf{equal}\ M\ N}{\Phi \vdash \mathsf{equal}\ (\mathsf{lam}\, x.\, M)\ (\mathsf{lam}\, x.\, N)} \qquad \frac{}{\Phi \vdash \mathsf{equal}\ M\ M}$$

$$\frac{\Phi \vdash \mathsf{equal}\ M_1\ N_1 \quad \Phi \vdash \mathsf{equal}\ M_2\ N_2}{\Phi \vdash \mathsf{equal}\ (\mathsf{app}\ M_1\ M_2)\ (\mathsf{app}\ N_1\ N_2)} \qquad \frac{\Phi \vdash \mathsf{equal}\ M\ L \quad \Phi \vdash \mathsf{equal}\ L\ N}{\Phi \vdash \mathsf{equal}\ M\ N}$$

It may be slightly unusual to keep the fact that a variable is equal to itself as a declaration in the context in both formulations. It is only strictly necessary in the first. There are two main reasons. 1) Explicitly introducing the appropriate assumption about each variable is a general methodology which scales to more expressive assumptions. For example, when we specify typing rules, we must introduce a typing context that keeps track of the fact that a given variable has a certain type. 2) Choosing this formulation will also make our proofs more elegant and compact, while at the same time highlight the issues which arise when working with two formal systems each using different assumptions.

We begin by proving that reflexivity and transitivity are indeed admissible from the algorithmic definition of equality.

Theorem 1 (Admissibility of Reflexivity and Transitivity).

1. *If Ψ contains premises for all the free variables in M, then $\Psi \vdash \mathsf{eq}\ M\ M$.*
2. *If $\Psi \vdash \mathsf{eq}\ M\ L$ and $\Psi \vdash \mathsf{eq}\ L\ N$ then $\Psi \vdash \mathsf{eq}\ M\ N$.*

The first theorem can be proven by induction on M. The second can be proven by induction on the first derivation. We now state that when we have a proof for $\mathsf{equal}\ M\ N$ then we also have a proof using algorithmic equality.

Attempt 1 (Completeness). *If $\Phi \vdash \mathsf{equal}\ M\ N$ then $\Psi \vdash \mathsf{eq}\ M\ N$.*

However, we note that this statement does not contain enough information about how the two contexts Φ and Ψ are related. In the base case, where we have that $\Phi \vdash \mathsf{equal}\ x\ x$, we must know that for every variable x in Φ there exists a corresponding assumption such that $\mathsf{eq}\ x\ x$ in Ψ. There are two solutions to this problem. 1) We state how two contexts are related and then assume that if this relation holds the theorem holds. 2) We generalize the context used in the theorem such that it contains both assumptions as follows

$$\text{Generalized context } \Gamma ::= \cdot \mid \Gamma, \mathsf{eq}\ x\ x, \mathsf{equal}\ x\ x$$

where we deliberately state that the assumption $\mathsf{eq}\ x\ x$ always occurs together with the assumption $\mathsf{equal}\ x\ x$, and then apply weakening and strengthening as needed to apply the equality inference rules. Both approaches can be mechanized and we discuss some of the trade-offs later. For now we will concentrate on the latter approach and state the revised generalized theorem.

Theorem 2 (Completeness). *If $\Gamma \vdash \mathsf{equal}\ M\ N$ then $\Gamma \vdash \mathsf{eq}\ M\ N$.*

Proof. Proof by induction on the first derivation. We show three cases which highlight the use of weakening and strengthening.

Case 1: Assumption from context
We know $\Gamma \vdash \mathsf{equal}\ x\ x$ where $\mathsf{equal}\ x\ x \in \Gamma$ by assumption. Because of the definition of Γ, we know that whenever we have an assumption $\mathsf{equal}\ x\ x$, we also must have an assumption $\mathsf{eq}\ x\ x$.

Case 2: Reflexivity rule
If the last step applied in the proof was the reflexivity rule $\Gamma \vdash \mathsf{equal}\ M\ M$, then we must show that $\Gamma \vdash \mathsf{eq}\ M\ M$. By the reflexivity lemma, we know that $\Psi \vdash \mathsf{eq}\ M\ M$. By weakening the context Ψ, we obtain the proof for $\Gamma \vdash \mathsf{eq}\ M\ M$.

Case 3: Equality rule for lambda-abstractions

$\Gamma \vdash \mathsf{equal}\ (\mathsf{lam}\, x.\, M)\ (\mathsf{lam}\, x.\, N)$	by assumption
$\Gamma, \mathsf{equal}\ x\ x \vdash \mathsf{equal}\ M\ N$	by decl. equality rule for lambda-abstraction
$\Gamma, \mathsf{eq}\ x\ x, \mathsf{equal}\ x\ x \vdash \mathsf{equal}\ M\ N$	by weakening
$\Gamma, \mathsf{eq}\ x\ x, \mathsf{equal}\ x\ x \vdash \mathsf{eq}\ M\ N$	by i.h.
$\Gamma, \mathsf{eq}\ x\ x \vdash \mathsf{eq}\ M\ N$	by strengthening
$\Gamma \vdash \mathsf{eq}\ (\mathsf{lam}\, x.\, M)\ (\mathsf{lam}\, x.\, N)$	by alg. equality rule for lambda-abstraction

This proof demonstrates many issues related to the treatment of bound variables and the treatment of contexts. First, we need to be able to apply a lemma which was proven in a context Ψ in a different context Γ. Second, we need to apply weakening and strengthening in the proof. Third, we need to be able to know the structure of the context and we need to be able to take advantage of it. We focus here on these structural properties of contexts, but of course many proofs also need the substitution lemma.

A.2 Reasoning about variable occurrences

In this example, we reason about the shape of terms instead of equality of terms. The idea is to compare terms up to variables. For example $\mathsf{lam}\, x.\, \mathsf{lam}\, y.\, \mathsf{app}\ x\ y$ would have the same shape as $\mathsf{lam}\, x.\, \mathsf{lam}\, y.\, \mathsf{app}\ y\ x$ but these two terms are obviously not equal. We use the judgment $\Phi \vdash \mathsf{shape}\ M_1\ M_2$ to describe that the term M_1 and the term M_2 have the same shape or structure. Thinking of the lambda-terms being described by a syntax tree, comparing the shape of two terms corresponds to comparing two syntax trees where we do not care about specific variable names which are at the leaves of it.

To define whether two lambda-terms have the same shape, we use two different judgments:

$\Psi \vdash \mathsf{shape}\ M\ N$	Terms M and N have the same shape
$\Psi \vdash \mathsf{varT}\ x$	x is a term variable

The context Ψ will have the following structure:

$$\text{Context}\ \ \Psi ::= \cdot \mid \Psi, \mathsf{varT}\ x$$

Next, we define when two lambda-terms have the same shape as follows:

$$\frac{\mathsf{varT}\ x \in \Psi \quad \mathsf{varT}\ y \in \Psi}{\Psi \vdash \mathsf{shape}\ x\ y} \qquad \frac{\Psi \vdash \mathsf{shape}\ M_1\ N_1 \quad \Psi \vdash \mathsf{shape}\ M_2\ N_2}{\Psi \vdash \mathsf{shape}\ (\mathsf{app}\ M_1\ M_2)\ (\mathsf{app}\ N_1\ N_2)}$$

$$\frac{\Psi, \mathsf{varT}\ x \vdash \mathsf{shape}\ M\ E}{\Psi \vdash \mathsf{shape}\ (\mathsf{lam}\, x.\, M)\ (\mathsf{lam}\, x.\, E)}$$

Finally, we define a judgment which counts how often variables occur in a lambda-term as follows:

$$\Psi \vdash \mathsf{varT{-}occ}\ M\ K \qquad \text{There are } K \text{ variables in the term } M$$

$$\frac{\mathsf{varT}\ x \in \Psi}{\Psi \vdash \mathsf{varT{-}occ}\ x\ 1} \qquad \frac{\Psi, \mathsf{varT}\ x \vdash \mathsf{varT{-}occ}\ T\ N}{\Psi \vdash \mathsf{varT{-}occ}\ (\mathsf{lam}\, x.\, T)\ N}$$

$$\frac{\Psi \vdash \mathsf{varT{-}occ}\ T_1\ N_1 \quad \Psi \vdash \mathsf{varT{-}occ}\ T_2\ N_2 \quad N = N_1 + N_2}{\Psi \vdash \mathsf{varT{-}occ}\ (\mathsf{app}\ T_1\ T_2)\ N}$$

First, we state that if two terms are equal they must have the same shape.

Theorem 3. *If* $\Psi \vdash \mathsf{eq}\ M_1\ M_2$ *then* $\Phi \vdash \mathsf{shape}\ M_1\ M_2$.

The proof of this theorem is a simpler version of the completeness proof we have given in the previous section. As in that proof, we need to either establish a context invariant which states the relationship between these two contexts or create a generalized context which contains both assumptions from Ψ and Φ.

We now prove that if M_1 and M_2 have the same shape, then they must have the same number of variables using the judgment $\Phi \vdash \mathsf{var{-}occ}\ M\ I$ where I describes the total number of variable occurrences in the term M. So for example, the total number of variable occurrences in the term $\mathsf{lam}\, x.\, \mathsf{lam}\, y.\, \mathsf{app}\ (\mathsf{app}\ y\ x)\ x$ is 3. If we think of the lambda-term as a syntax tree, then I describes the number of leaves in the syntax tree described by the term M. We give three different variations, intended to show differences among systems.

Theorem 4.

1. *If* $\Phi \vdash \mathsf{shape}\ M_1\ M_2$
   then there exists an I *such that* $\Phi \vdash \mathsf{var{-}occ}\ M_1\ I$ *and* $\Phi \vdash \mathsf{var{-}occ}\ M_2\ I$. *Furthermore* I *is unique.*
2. *If* $\Phi \vdash \mathsf{shape}\ M_1\ M_2$
   then for all I. $\Phi \vdash \mathsf{var{-}occ}\ M_1\ I$ *implies* $\Phi \vdash \mathsf{var{-}occ}\ M_2\ I$.
3. *If* $\Phi \vdash \mathsf{shape}\ M_1\ M_2$ *and* $\Phi \vdash \mathsf{var{-}occ}\ M_1\ I$ *then* $\Phi \vdash \mathsf{var{-}occ}\ M_2\ I$.

A.3 Reasoning about subterms in lambda-terms

For the next example, we define when a given lambda-term M is a subterm of another lambda-term N and hence we consider M to be structurally smaller than (or equal to) N using the following judgment:

$$\Psi \vdash \mathsf{lt}\ M\ N \qquad \text{Term } M \text{ is strictly smaller than } N$$
$$\Psi \vdash \mathsf{le}\ M\ N \qquad \text{Term } M \text{ is smaller than or equal to } N$$

Next, we define these judgments.

Term M is strictly smaller than N

$$\frac{\Psi, \mathsf{eq}\ x\ x \vdash \mathsf{le}\ N\ M}{\Psi \vdash \mathsf{lt}\ N\ (\mathsf{lam}\, x.\, M)} \qquad \frac{\Psi \vdash \mathsf{le}\ N\ M_1}{\Psi \vdash \mathsf{lt}\ N\ (\mathsf{app}\ M_1\ M_2)} \qquad \frac{\Psi \vdash \mathsf{le}\ N\ M_2}{\Psi \vdash \mathsf{lt}\ N\ (\mathsf{app}\ M_1\ M_2)}$$

Term M is smaller than or equal to N

$$\frac{\Psi \vdash \mathsf{eq}\ M\ N}{\Psi \vdash \mathsf{le}\ M\ N} \qquad \frac{\Psi \vdash \mathsf{lt}\ M\ N}{\Psi \vdash \mathsf{le}\ M\ N}$$

We concentrate here on stating a very simple intuitive theorem that says that if for all terms N, if N is smaller than K implies that N is also smaller than L, then clearly K is smaller than L.

Theorem 5. *If for all N. $\Psi \vdash \mathsf{le}\ N\ K$ implies $\Psi \vdash \mathsf{le}\ N\ L$ then $\Psi \vdash \mathsf{le}\ K\ L$.*

This theorem is interesting because in order to state it, we nest quantification and implications placing them outside the fragment of propositions directly expressible in systems such as Twelf.

References

1. Amy Felty and Brigitte Pientka. Reasoning with higher-order abstract syntax and contexts: A comparison. In Matt Kaufmann and Lawrence Paulson, editors, *International Conference on Interactive Theorem Proving (ITP)*, Lecture Notes in Computer Science. Spinger, 2010.