A Higher-Order Approach to Explicit Contexts in LF

JONATHAN M. STERLING

There are two standard techniques for representing dependently typed calculi in the LF. The first is to re-use the LF contexts for the object contexts, which can lead to difficulties when the meaning of hypothetico-general judgement in the object language is stronger than in the LF; this is, for instance, the case in MLTT 1979, where non-trivial functionality obligations are incurred in the sequent judgement.

Another technique advanced by Krary is to represent contexts explicitly, and then define a sequent judgement over them; this can be used to resolve the problem described above (and several others), but it comes at the cost of verbosity, and cannot be to encode *telescopes*, which are contexts where each successive term has the previous bound variables free in it. In this paper, I demonstrate a higher-order encoding of contexts and telescopes which can be used to faithfully encode the sequent judgement for MLTT 1979.

Categories and Subject Descriptors: []:

General Terms: Logical Frameworks

Additional Key Words and Phrases: Contexts, telescopes

1. INTRODUCTION

To motivate this work, let us consider an encoding of MLTT 1979 in the logical framework. The syntax contains two sorts, val (for canonical forms) and exp (for non-canonical forms), and is defined in the following LF signature: [1]

$$\frac{}{\mathsf{val} : \mathbf{type}} \qquad \frac{}{\mathsf{exp} : \mathbf{type}}$$

$$\frac{}{\mathsf{void} : \mathsf{val}} \qquad \frac{}{\mathsf{unit} : \mathsf{val}} \qquad \frac{A : \mathsf{exp} \quad B : \mathsf{exp} \to \mathsf{exp}}{\mathsf{pi}(A; [x]B\,x) : \mathsf{val}} \qquad \frac{A : \mathsf{exp} \quad B : \mathsf{exp} \to \mathsf{exp}}{\mathsf{sg}(A; [x]B\,x) : \mathsf{val}}$$

$$\frac{}{\mathsf{ax} : \mathsf{val}} \qquad \frac{E : \mathsf{exp} \to \mathsf{exp}}{\lambda[x]E\,x : \mathsf{val}} \qquad \frac{M : \mathsf{exp} \quad N : \mathsf{exp}}{\mathsf{pair}(M; N) : \mathsf{val}}$$

$$\frac{M : \mathsf{val}}{\uparrow M : \mathsf{exp}} \qquad \frac{M : \mathsf{exp} \quad N : \mathsf{exp}}{\mathsf{ap}(M; N) : \mathsf{exp}} \qquad \frac{M : \mathsf{exp}}{\mathsf{fst}(M) : \mathsf{exp}} \qquad \frac{M : \mathsf{exp}}{\mathsf{snd}(M) : \mathsf{exp}}$$

Then, the language is accorded a simple big-step operational semantics via the $\boxed{M \Rightarrow N}$ judgement:

[1] LF signatures are presented in the style of inference rules for clarity. Object judgement forms have a *mode* which expresses which arguments represent inputs, and which represent outputs; in our presentation, we color the inputs *MidnightBlue*, and the outputs *Maroon*. Proper assignment of modes may be used to construe a judgement as having algorithmic content.

$$\frac{M : \mathsf{exp} \quad N : \mathsf{val}}{M \Rightarrow N : \mathbf{type}}$$

$$\frac{}{\uparrow M \Rightarrow M} \Rightarrow_{\uparrow} \qquad \frac{E\, M \Rightarrow N}{\mathsf{ap}(\uparrow \lambda[x]E(x); M) \Rightarrow N} \Rightarrow_{\mathsf{ap}}$$

$$\frac{M \Rightarrow M'}{\mathsf{fst}(\uparrow \mathsf{pair}(M; N)) \Rightarrow M'} \Rightarrow_{\mathsf{fst}} \qquad \frac{N \Rightarrow N'}{\mathsf{fst}(\uparrow \mathsf{pair}(M; N)) \Rightarrow N'} \Rightarrow_{\mathsf{snd}}$$

2. THE NAÏVE HIGHER-ORDER APPROACH

Informally, there are four basic categorical judgements in MLTT 1979:

Form	Pronunciation
A *set*	A is a set
$A = B$ *set*	A and B are equal sets
$M \in A$	M is a member of A
$M = N \in A$	M and N are equal members of A

The subjects of these judgements are terms of sort exp. However, implicit in Martin-Löf's presentation are four prior judgements which operate only on canonical forms (i.e. terms of sort val). We will therefore introduce the following LF signature to account for this:

$$\frac{A : \mathsf{val}}{A\ \mathit{set}_{\mathsf{v}} : \mathbf{type}} \quad \frac{A : \mathsf{val} \quad B : \mathsf{val}}{A = B\ \mathit{set}_{\mathsf{v}} : \mathbf{type}} \quad \frac{M : \mathsf{val} \quad A : \mathsf{val}}{M \in_{\mathsf{v}} A : \mathbf{type}} \quad \frac{M : \mathsf{val} \quad N : \mathsf{val} \quad A : \mathsf{val}}{M = N \in_{\mathsf{v}} A : \mathbf{type}}$$

$$\frac{A : \mathsf{val}}{A\ \mathit{set} : \mathbf{type}} \quad \frac{A : \mathsf{exp} \quad B : \mathsf{exp}}{A = B\ \mathit{set} : \mathbf{type}} \quad \frac{M : \mathsf{exp} \quad A : \mathsf{exp}}{M \in A : \mathbf{type}} \quad \frac{M : \mathsf{exp} \quad N : \mathsf{exp} \quad A : \mathsf{exp}}{M = N \in A : \mathbf{type}}$$

Then, we will give the meanings of the judgements which deal with non-canonical forms first, anticipating the explanations of the canonical forms judgements.

$$\frac{A \Rightarrow A' \quad A'\ \mathit{set}_{\mathsf{v}}}{A\ \mathit{set}} \qquad \frac{\begin{matrix} A \Rightarrow A' \\ B \Rightarrow B' \end{matrix} \quad A' = B'\ \mathit{set}_{\mathsf{v}}}{A = B\ \mathit{set}}$$

$$\frac{\begin{matrix} M \Rightarrow M' \\ A \Rightarrow A' \end{matrix} \quad M' \in_{\mathsf{v}} A'}{M \in A} \qquad \frac{\begin{matrix} M \Rightarrow M' \\ N \Rightarrow N' \\ A \Rightarrow A' \end{matrix} \quad M' = N' \in_{\mathsf{v}} A'}{M = N \in A}$$

Following Martin-Löf, we give an extensional equality for all types (i.e. two types are equal when they have the same membership and equivalence relations); we could choose an intensional/structural equality for types, but for our purposes here, it is not necessary.

$$\frac{\begin{array}{l}\forall\{M\}.\ M \in_{\mathsf{v}} A \to M \in_{\mathsf{v}} B \\ \forall\{M\}.\ M \in_{\mathsf{v}} B \to M \in_{\mathsf{v}} A \\ \forall\{M,N\}.\ M = N \in_{\mathsf{v}} A \to M = N \in_{\mathsf{v}} B \\ \forall\{M,N\}.\ M = N \in_{\mathsf{v}} B \to M = N \in_{\mathsf{v}} A\end{array}}{A = B\ \mathit{set}_{\mathsf{v}}}$$

Then, all we have to do is define the remaining canonical forms judgements with respect to our syntax. For the base types, this is trivial:

$$\frac{}{\mathsf{void}\ \mathit{set}_{\mathsf{v}}} \quad \frac{}{\mathsf{unit}\ \mathit{set}_{\mathsf{v}}} \quad \frac{}{\mathsf{ax} \in_{\mathsf{v}} \mathsf{unit}} \quad \frac{}{\mathsf{ax} = \mathsf{ax} \in_{\mathsf{v}} \mathsf{unit}}$$

But for terms which involve binding, the rules will become more complicated. Informally, Martin-Löf gives the following formation rule for Π using hypothetico-general judgement:

$$\frac{A\ \mathit{set} \quad B(x)\ \mathit{set}\ (x \in A)}{\Pi(x \in A)B(x)\ \mathit{set}_{\mathsf{v}}} *$$

So we might be tempted to do something similar in the LF, as follows:

$$\frac{A\ \mathit{set} \quad \forall\{x\}.\ x \in A \to B(x)\ \mathit{set}}{\mathsf{pi}(A; [x]B\,x)\ \mathit{set}_{\mathsf{v}}} *'$$

This encoding, however, is not adequate, since the meaning of hypothetico-general judgement in Martin-Löf's informal metalangauge is not captured by the above attempt. Recall that hypothetico-general judgement in type theory implicitly requires *functionality* of type and term families; therefore, in order for our definition to be correct, we would have to adjust the rule to include functionality:

$$\frac{A\ \mathit{set} \quad \begin{array}{l}\forall\{x\}.\ x \in A \to B(x)\ \mathit{set} \\ \forall\{x,y\}.\ x = y \in A \to B(x) = B(y)\ \mathit{set}\end{array}}{\mathsf{pi}(A; [x]B\,x)\ \mathit{set}_{\mathsf{v}}}$$

We have at least got closer, but even this is not really satisfactory; we were unable to factor out the functionality constraints, and must be careful to place them in all the right places in each and every rule. Contrast this with Martin-Löf's definition of hypothetico-general judgement, which includes functionality from the start, allowing the other judgements to use this "off the shelf".

3. EXPLICIT CONTEXTS À LA KRARY

In order to factor out functionality constraints, it will be necessary to consider the use of contexts in the encoding, and then recast the judgements above as being sequents, i.e. judgements with respect to a context. Krary's first-order encoding of explicit contexts in the LF is essentially the following:

$$\frac{}{\mathsf{ctx} : \mathbf{type}} \qquad \frac{}{\cdot : \mathsf{ctx}} \qquad \frac{\Gamma : \mathsf{ctx} \quad X : \mathsf{exp} \quad A : \mathsf{exp}}{(\Gamma, X : A) : \mathsf{ctx}}$$

$$\frac{X : \mathsf{exp} \quad I : \mathbb{N}}{\mathsf{isvar}(X; I) : \mathbf{type}} \qquad \frac{X : \mathsf{exp} \quad Y : \mathsf{exp}}{\mathsf{precedes}(X; Y) : \mathbf{type}} \qquad \frac{\mathsf{isvar}(X; I) \quad \mathsf{isvar}(Y; J) \quad I < J}{\mathsf{precedes}(X; Y)}$$

$$\frac{\Gamma : \mathsf{ctx} \quad X : \mathsf{exp}}{\mathsf{bounded}(\Gamma; X) : \mathbf{type}} \qquad \frac{\mathsf{isvar}(X; I)}{\mathsf{bounded}(\cdot; X)} \qquad \frac{\mathsf{precedes}(Y; X) \quad \mathsf{bounded}(G; Y)}{\mathsf{bounded}(\Gamma, Y : A; X)}$$

$$\frac{\Gamma : \mathsf{ctx}}{\mathsf{ordered}(\Gamma) : \mathbf{type}} \qquad \frac{}{\mathsf{ordered}(\cdot)} \qquad \frac{\mathsf{bounded}(\Gamma; X)}{\mathsf{ordered}(\Gamma, X : A)}$$

Note that the well-formedness constraints above do not in practice protrude into the definitions of judgements which *use* explicit contexts; they are only used in metatheorems. This encoding is very practical for certain calculi, but it is concerning in our case for a few reasons:

(1) The correctness of this encoding depends very strongly on its *use*: this only truly encodes contexts if in every case a LF-variable is used for X. Moreover, the various well-formedness constraints are a bit inelegant, as is the concept of mapping LF variables to indices.

(2) It is not possible to construct such a context on its own as a syntactic object of (LF-)type ctx, where variables bound earlier in the context may appear free later in the context. This can be done otherwise by constructing an object of type ctx in an extended LF-context; the well-formedness of such a context is still an extrinsic property, though.

(3) A straightforward higher-order encoding is preferable (if possible) when working in the LF.

4. ENCODING TELESCOPES IN HIGHER-ORDER ABSTRACT SYNTAX

More precisely, it seems that what we really need is an encoding of *telescopes*, in the sense of De Bruijn. A telescope is essentially a sequence of type assignments, where variables bound earlier may appear free later, for instance:

$$\cdot, x_1 : A_1, x_2 : A_2(x_1), x_3 : A_3(x_1, x_2), \ldots, x_n : A_n(x_1, \ldots, x_{n-1})$$

To encode this in a straightforward, higher-order manner, we shall first stipulate that each variable in the telescope will be an LF-variable. Then, each type A_i in the telescope will be a term with $i - n$ bindings (i.e. it will be an i-ary λ-abstraction). We'll define by mutual induction a sort for telescopes tele, and a sort for open terms with respect to a telescope $[\Psi]$exp:

$$\frac{}{\mathsf{tele} : \mathbf{type}} \qquad \frac{\Psi : \mathsf{tele}}{[\Psi]\mathsf{exp} : \mathbf{type}}$$

$$\frac{}{\cdot : \mathsf{tele}} \qquad \frac{\Psi : \mathsf{tele} \quad A : [\Psi]\mathsf{exp}}{\Psi, A : \mathsf{tele}}$$

$$\frac{}{!M : [\cdot]\mathsf{exp}} \qquad \frac{M : \mathsf{exp} \to [\Psi]\mathsf{exp}}{@[x]M(x) : [\Psi, A]\mathsf{exp}}$$

Note that we do not put any well-typedness constraints on the arguments of the bound term constructors; we wish to separate object syntax from object judgements; this is particularly important for an object theory which does not have decidable type checking.