Logical framework-like encoding of inference rules[1]

Vladimir Voevodsky[2]

March, 2017

For an inference rule with premises $\Gamma \rhd T_1\, type$ and $\Gamma, x_1 : T_1 \rhd T_2\, type$ and product of the form $\Gamma \rhd Sum(T_1, x_1.T_2)\, type$ the format of the LF-style encoding is

$$Sum : \prod_{T_1 : Type} \prod_{T_2 : T_1 \to Type} Type$$

Therefore, types of the structure

$$(A_1 \to Type) \to \ldots \to (A_n \to Type) \to Type$$

where A_i are small type expressions, should be allowed.

Let us check something else. Suppose the premises are $\Gamma \rhd T_1\, type$, $\Gamma, x_1 : Sum(T_1, T_1) \rhd T_2\, type$ and the product is $\Gamma \rhd Sum_2(T_1, x_1.T_2)\, type$. Then

$$Sum_2 : \prod_{T_1 : Type} \prod_{Sum(T_1, T_1) \to Type} Type$$

The fact that types of structure

$$(Type \to Type) \to Type$$

are not necessary, somehow corresponds with the fact that in the syntax of type theory, variables are always elements of types and never types. If I could have as the premise $\Gamma, X : Type \rhd T : Type$ and as the product $\Gamma \rhd S(X.T)\, type$ then for the LF-style encoding I would have

$$S : \prod_{T : Type \to Type} Type$$

which is the same as

$$S : (Type \to Type) \to Type$$

If we had it, it would be the following - $T(X)$ would be any construction that from a type expression A makes another type expression $T(A)$ while S will be a construction that from any such construction makes a type. For example, S could be evaluating T on *nat*. If we add $\Gamma \rhd T_0\, type$ as another premise then we can have $S(T_0, X.T) = T(T(T_0/X)/X)$.

What is interesting however is that the structure $(\mathbf{RR}, \mathbf{LM})$ is directly related to the fact that the premises of the inference rules generating types or elements of types in the Agda-LF-style encoding can only have the form $Type$, $(\vec{x} : \vec{A}) \to Type$, B or $(\vec{x} : \vec{A}) \to B$, but never $Type \to Type$.

---

[1] *2000 Mathematical Subject Classification: 18D99, 18C50*

[2] School of Mathematics, Institute for Advanced Study, Princeton NJ, USA. e-mail: vladimir@ias.edu

To follow further with the LF-style encodings one needs to provide encodings of the inference rules for the type and element equalities. Suppose you have a universe U with the Ty constructor that produces a type from an element of the universe. One needs the rule with the premise $\Gamma \rhd a \equiv b : U$ and the product $\Gamma \rhd Ty(a) \equiv Ty(b)$.

One can encode a type equality rule as a pair of the problem is in how to use such equality rules in the subsequent inference rules