

Evaluation of Membership in MX Dependent Types

by Sven Nilsen, 2019

In this paper I represent evaluation of membership in MX grammar for dependent types.

The BNF^[1] of membership in MX grammar^[2] for dependent types^[3] is the following:

$$m ::= v : v$$

Where `v` is a variable/value.

Previously, I defined evaluation of variable in MX grammar for dependent types^[4]:

$$\text{eval_var} : \text{ctx} \rightarrow v \rightarrow (v, \text{opt}[v])$$

Let `m₀` be the membership to be evaluated and `ctx₀` the context to evaluate the membership in. The `eval_var` function is used on both left and right side to construct a new membership `m₁`:

$$\begin{aligned} (v_0 : v_1) &:= m_0 \\ m_1 &:= (\text{eval_var}(\text{ctx}_0)(v_0).0 : \text{eval_var}(\text{ctx}_1)(v_1).0) \end{aligned}$$

Here, the optional type information is ignored, but might be used by some versions to report an error if the types do not match. Type information can be ignored under evaluation assuming a correct program. Here, a correct program is assumed since the purpose is just to explain evaluation of membership.

The evaluated membership is found by looking for the definition of `m₁` in the context:

$$\text{if let some}(m_2) = \text{find_def}(\text{ctx}_0)(m_1) \text{ \{m}_2\} \text{ else \{m}_1\}$$
$$\text{find_def} : \text{ctx} \rightarrow m \rightarrow mx$$

Notice that this permits substituting e.g. `a : type₀` with a dependent type `b(c)`. This trick can be used for partial evaluation and theorem proving.

The complete code, assuming a correct program from the context:

$$\text{eval_mem} : \text{ctx} \rightarrow m \rightarrow mx$$
$$\begin{aligned} \text{eval_mem} &:= \lambda(\text{ctx}_0 : \text{ctx}) = \lambda(m_0 : m) = \{ \\ &\quad (v_0 : v_1) := m_0 \\ &\quad m_1 := (\text{eval_var}(\text{ctx}_0)(v_0).0 : \text{eval_var}(\text{ctx}_1)(v_1).0) \\ &\quad \text{if let some}(m_2) = \text{find_def}(\text{ctx}_0)(m_1) \text{ \{m}_2\} \text{ else \{m}_1\} \\ &\} \end{aligned}$$

References:

- [1] “Backus-Naur form”
Wikipedia
https://en.wikipedia.org/wiki/Backus%E2%80%93Naur_form
- [2] “MX Grammar for Dependent Types”
Sven Nilsen, 2019
https://github.com/advancedresearch/path_semantics/blob/master/papers-wip/mx-grammar-for-dependent-types.pdf
- [3] “Dependent type”
Wikipedia
https://en.wikipedia.org/wiki/Dependent_type
- [4] “Evaluation of Variable in MX Dependent Types”
Sven Nilsen, 2019
https://github.com/advancedresearch/path_semantics/blob/master/papers-wip/evaluation-of-variable-in-mx-dependent-types.pdf