Staged Compilation and Generativity[*]

András Kovács

Eötvös Loránd University, Budapest, Hungary
kovacsandras@inf.elte.hu

The purpose of staged compilation is to write code-generating programs in a safe and ergonomic way. Although it is always possible to write metaprograms by simply manipulating strings or deeply embedded syntax trees, this is often error-prone and tedious. Staging is a way to have more guarantees about the safety and well-typing of metaprograms, and also a way to integrate object-level and meta-level syntaxes more organically.

Two-level type theory (2LTT) [1] was originally developed for the purpose of doing synthetic homotopy theory, by adding a metaprogramming layer on top of homotopy type theory [3]. However, it turns out that 2LTT is also a great framework for metaprogramming and staging in general; it is applicable to a wide range of object theories.

There is a natural semantics to 2LTT which justifies the metaprogramming view: this is the *presheaf model* of 2LTT. Here, meta-level types are presheaves over the underlying category of the object theory. Hence, every meta-level construction must be stable under the object-level morphisms.

Generativity Generativity means that we can only generate code, but not look inside object-level syntax and make decisions based on that. Generativity simplifies staging, and it is often enforced in practical implementations [2]. However, non-generative staging provides additional power and flexibility. A simple example for a non-generative feature is conversion checking. Assume that given an object-level type A, $\mathsf{Code}\,A$ is the type of meta-level programs which compute A-expressions. Conversion checking may be postulated as $\mathsf{conversion}_A : (t\,u : \mathsf{Code}\,A) \to (t = u) + (t \neq u)$. We aim to investigate generativity in the presheaf model. We observe that it depend on the choice of morphisms in the object theory.

If base morphisms are *substitutions*, then $\mathsf{conversion}$ does not hold in the model, because definitional inequality is not stable under substitution. For example, inequal variables may become equal after substitution.

If base morphisms are *weakenings*, then $\mathsf{conversion}$ holds. However, in this case the object theory can only support features which can be specified with weakenings. For instance, this allows simple types at at the object level, without any β-rules, but it does not allow polymorphism or dependent types. This is still sufficient for many staging applications.

Yoneda lemma It is worth to note that we get generativity statements in the presheaf model from the Yoneda lemma. For example, the natural transformation $\mathsf{y}\,\mathsf{Bool}_0 \Rightarrow \mathsf{Bool}_1$ must be a constant map in the model. How is it possible then to have non-generativity? The answer is that the Yoneda lemma can only meaningfully apply to *types* (as opposed to contexts) in the presheaf model, if context extension is preserved by Yoneda embedding. If base morphisms are weakenings, the base category does not have all binary products, and context extension is not preserved by y. Hence, the singleton context $\bullet, \mathsf{y}\,\mathsf{Bool}_0$ is not representable, and the Yoneda lemma does not say anything about dependency on $\mathsf{y}\,\mathsf{Bool}_0$.

[*]The author was supported by the European Union, co-financed by the European Social Fund (EFOP-3.6.3-VEKOP-16-2017-00002).

References

[1] Danil Annenkov, Paolo Capriotti, Nicolai Kraus, and Christian Sattler. Two-level type theory and applications. *ArXiv e-prints*, may 2019.

[2] Oleg Kiselyov. The design and implementation of BER metaocaml - system description. In Michael Codish and Eijiro Sumii, editors, *Functional and Logic Programming - 12th International Symposium, FLOPS 2014, Kanazawa, Japan, June 4-6, 2014. Proceedings*, volume 8475 of *Lecture Notes in Computer Science*, pages 86–102. Springer, 2014.

[3] The Univalent Foundations Program. *Homotopy Type Theory: Univalent Foundations of Mathematics*. https://homotopytypetheory.org/book, Institute for Advanced Study, 2013.