Weak Type Theory is Rather Strong

Simon Boulier and Théo Winterhalter

Gallinette Project-Team, Inria Nantes France

Abstract

We often put computation at the core of our theories, showing Strong Normalisation and confluence of our calculi. Thanks to the Curry-Howard isomorphism we know that computation corresponds to cut-elimination, or simplification of proofs. In practice it is also useful because it allows to identify expressions such as $0+n$ and n. We claim however that computation does not bring more logical power to our type theories by means of a Coq formalisation of a translation from Extensional Type Theory to a Weak Intensional Type Theory with axioms K and functional extensionality. Our result extends to the case of 2-level type theories, making it relevant in the homotopy setting.

Computation. The basic computation rule is the β-reduction (where square brackets represent substitution):

$$(\lambda x.\ t)\ u \equiv t[x := u]$$

In dependent type theories, computation is not limited to β-reduction and includes computation rules for eliminators of inductive types (or pattern-matching). We can also extend the notion and think of an equational theory that includes reduction but also η-rules or even more recently in the case of Coq and Agda, definitional proof-irrelevance of propositions [3].

$$\begin{array}{rr} f \equiv \lambda x.\ f\ x & p \equiv (\pi_1\ p, \pi_2\ p) \\ \mathsf{S}\ n + m \equiv \mathsf{S}\ (n+m) & 0 + n \equiv n \\ (x : P : \mathsf{Prop}) \equiv (y : P : \mathsf{Prop}) & x \equiv \star : \mathsf{Unit} \end{array}$$

Weak type theories (WTT) are type theories without computation rules. Instead all of these rules are *weak*, i.e., assumed as equality axioms (for the type of propositional equality). For instance, β-reduction is represented by the following equality (where $x =_T y$ is the identity type representing propositional equality of x and y at type T).

$$\frac{\Gamma, x : A \vdash t : B \qquad \Gamma \vdash u : A}{\Gamma \vdash \beta(x.t, u) : (\lambda x.\ t)\ u =_{B[x:=u]} t[x := u]}$$

And the same goes for other equation rules. In particular we also make explicit the δ-reduction stating that a definition $id := t$ can be unfolded. We additionally require axioms K, functional extensionality and more surprising congruence rules for the other binders:

$$\frac{\Gamma \vdash p : t =_A t}{\Gamma \vdash \mathsf{K}(p) : p = \mathsf{refl}_A\ t} \qquad \frac{\Gamma, x : A \vdash p : f\ x = g\ x}{\Gamma \vdash \mathsf{funext}(x.p) : f = g}$$

$$\frac{\Gamma, x : A \vdash p : B_1 = B_2}{\Gamma \vdash \mathsf{cong}\Pi(x.p) : \Pi(x : A).B_1 = \Pi(x : A).B_2} \qquad \frac{\Gamma, x : A \vdash p : B_1 = B_2}{\Gamma \vdash \mathsf{cong}\Sigma(x.p) : \Sigma(x : A).B_1 = \Sigma(x : A).B_2}$$

Since our goal is to prove that extensional type theory (ETT) is conservative over WTT—every valid WTT type that is inhabited in ETT is inhabited in WTT—and since ETT proves these axioms, we have to assume them in WTT. The axioms $\mathsf{cong}\Pi$ and $\mathsf{cong}\Sigma$ do not seem to be derivable from funext.

Extensional Type Theory is distinguished from the usual Intensional Type Theory (ITT) by the reflection rule

$$\frac{\Gamma \vdash e : u =_A v}{\Gamma \vdash u \equiv v}$$

turning any provable equality into a conversion (i.e., computation rule in some sense). ITT trivially embeds into ETT so our result also extends to ITT to WTT.

Translating ETT to WTT. This work builds on previous work [5] of translating ETT to ITT: the idea is to take a typing derivation in ETT and produce a translation of the term (which was a decoration of the original term with rewriting of equalities) and a proof that it is typable in ITT.

This whole work has been formalised in Coq [2]. We proceed in two phases. For the first one, we take advantage of our previous formalisation by remarking that the way we deal with conversion in the target is mainly orthogonal to the translation itself. We thus remove conversion from the target and obtain one form of WTT with some extra axioms. For the second translation phase we remove these axioms by realising them using K, funext, and the necessary computation equalities, thus landing in WTT.

Homotopy. For those concerned by the use of axiom K to interpret equality, it is interesting to remark that the translation is done in a setting general enough that it can be instantiated to 2-level type theories [1]. By 2-level type theories we mean theories equipped with two equality types, one that is strict (with K and funext), and one that is the *usual* unconstrained equality meaning that in particular it can interpret the homotopy—or even univalent—equality (or path type).

Our translation can thus go from an *extensional* 2-level type theory (close in definition and usage to Voevodsky's Homotopy Type System [4]) to a *weak* 2-level type theory, that is a theory with a strict equality but no conversion.

Conclusion. We provide a translation from ETT to WTT that shows the former is conservative over the latter, thus proving that computation/conversion doesn't add logical power to type theory, including in a homotopy setting.

References

[1] Thorsten Altenkirch, Paolo Capriotti, and Nicolai Kraus. Extending homotopy type theory with strict equality. In *25th EACSL Annual Conference on Computer Science Logic, CSL 2016, Marseille, France*, 2016.

[2] S. Boulier, M. Sozeau, N. Tabareau, and T. Winterhalter. Formalisation of ETT to ITT (and to WTT), 2019. Available at https://github.com/TheoWinterhalter/ett-to-itt/tree/weak.

[3] Gaëtan Gilbert, Jesper Cockx, Matthieu Sozeau, and Nicolas Tabareau. Definitional Proof-Irrelevance without K. *Proceedings of the ACM on Programming Languages, POPL 2019, Cascais, Portugal*, 2019.

[4] Vladimir Voevodsky. A simple type system with two identity types. *Unpublished note*, 2013. https://www.math.ias.edu/vladimir/sites/math.ias.edu.vladimir/files/HTS.pdf.

[5] Théo Winterhalter, Matthieu Sozeau, and Nicolas Tabareau. Eliminating Reflection from Type Theory. In *8th ACM SIGPLAN International Conference on Certified Programs and Proofs, CPP 2019, Lisbonne, Portugal*, 2019.