The Dialectica Translation of Type Theory

Andrej Bauer[1] and Pierre-Marie Pédrot[2]

[1] University of Ljubljana
[2] INRIA

The Dialectica translation, introduced by Gödel in the eponymous journal [2] is a logical translation from intuitionistic higher-order arithmetic $\mathbf{HA}^{\omega}$ into System $\mathbf{T}$, a simply-typed λ-calculus with integers. In modern terms, one would call it a *realizability* interpretation, as all logical content is shifted into the meta-theory: proofs are erased into simply-typed realizers while logic is interpreted thanks to an orthogonality relation, allowing to discriminate pseudo-proofs from actual proofs. Thanks to this interpretation, one can realize two additional principles which are not provable in $\mathbf{HA}^{\omega}$, namely Markov's principle and the independence of premise.

Contrarily to Kreisel's modified realizability, which actually originated as a variant of Dialectica, realizers produced by the latter are not a mere computational erasure of the corresponding proof-term. Just as the Lafont-Reus-Streicher (LRS) CPS translation [3] can be synthetized back from Krivine's realizability [5], there is a novel program translation hidden beneath Dialectica's realizability. In a previous paper [6], the second author gave a dynamic explanation of an instance of this translation on the simply-typed λ-calculus by means of the Krivine machine, which showed that Dialectica gives a first-class account of two notions coming from the machine: 1. stacks, which are given a proper *counter* type $\mathbb{C}(A)$ for any source type A and 2. substitution, which is present in the machine in the form of closures. While stacks are already present in the LRS translation, first-class citizenship of substitution is not usually present in program translations, except maybe for call-by-need encodings.

In a nutshell, the simply-typed translation goes as follows. To any type A, it associates a witness type $\mathbb{W}(A)$, a counter type $\mathbb{C}(A)$ and an orthogonality relation $\perp_A$. To a term $\Gamma \vdash t : A$, it associates a term $\mathbb{W}(\Gamma) \vdash t^{\bullet} : \mathbb{W}(A)$ together with a family of terms $\mathbb{W}(\Gamma) \vdash t_x : \mathbb{C}(A) \rightarrow \mathfrak{M}\,\mathbb{C}(X)$ for each $(x : X) \in \Gamma$, where $\mathfrak{M}\,(-)$ stands for the finite multiset monad. This translation is computationally sound, i.e. assuming reasonable commutation rules on the multiset monad, it preserves conversion. Its direct-style interpretation can be thought of as a mix of an instrumentation of variable accesses together with a weak form of delimited continuations. Morally, t_x does the following: execute t in the machine, and each time x is dereferenced, add the current continuation (i.e. the stack of the machine) to a global multiset which is finally returned. This effect is commutative, in so far as stacks are returned without regard of the relative order of dereferencing, as we produce a multiset rather than a list. Such a desequentialization is vital for the preservation of conversion.

In the same paper, the second author showed that this translation readily adapts to the dependently-typed case, by providing a Dialectica translation of the whole calculus of construction with universes $\mathbf{CC}_{\omega}$ into $\mathbf{CC}_{\omega}$ plus Σ-types and a computational multiset monad. This shows that Dialectica is in a certain sense more intuitionistic than LRS, which is not known to exist in a dependent setting.

While Dialectica could also be independently adapted to handle simply-typed algebraic datatypes [7], the interpretation of dependent elimination remained difficult to tackle at first sight. Nonetheless, we show that by a simple tweak inspired by a call-by-push-value decomposition, one can recover the full dependent elimination principles, hence demonstrating that the Dialectica translation was indeed genuinely intuitionistic. The main theorem is the following.

Theorem 1. *Assuming reasonable commutation rules on the multiset monad, there exists a Dialectica translation from* **CIC** *into* **CIC** $+ \mathfrak{M}(-)$ *preserving typing and definitional equality.*

The main difference from the $\mathbf{CC}_\omega$ translation comes from the fact we have to make the counter type dependent on a witness to make the translation go through. It means that to any $A : \square$, we associate $\mathbb{W}(A) : \square$ and $\mathbb{C}(A) : \mathbb{W}(A) \to \square$ where $\mathbb{C}(A)$ is indexed. This triggers a handful of adaptations. A term $\Gamma \vdash M : A$ is now translated as:

1. A term $\mathbb{W}(\Gamma) \vdash M^\bullet : \mathbb{W}(A)$ (as before);
2. A family of terms $\mathbb{W}(\Gamma) \vdash M_x : \mathbb{C}(A)[M^\bullet] \to \mathfrak{M}\,\mathbb{C}(X)[x]$ for all $(x : X) \in \Gamma$.

Note how the type of counters produced by M_x depends on x, which is the crux of the trick. We give the translation of a few representative types to demonstrate how dependent the translation becomes.

	$\mathbb{W}(-) : \square$	$\mathbb{C}(-) : \mathbb{W}(-) \to \square$
$\square$	$\Sigma A^+ : \square . A^+ \to \square$	$\lambda \_ .\, 1$
$\Pi x : A.\, B$	$\Pi x : \mathbb{W}(A).\, \Sigma y : \mathbb{W}(B).\, \mathbb{C}(B)[y] \to \mathfrak{M}\,\mathbb{C}(A)[x]$	$\lambda f.\, \Sigma x : \mathbb{W}(A).\, \mathbb{C}(B)[\pi_1\,(f\,x)]$
$\Sigma x : A.\, B$	$\Sigma x : \mathbb{W}(A).\, \mathbb{W}(B)$	$\lambda (x, y).\, \mathbb{C}(A)[x] + \mathbb{C}(B)[y]$
$A + B$	$\mathbb{W}(A) + \mathbb{W}(B)$	$\lambda [\, x \Rightarrow \mathbb{C}(A)[x] \mid y \Rightarrow \mathbb{C}(B)[y]\,]$

This provides new hindsights on the De Paiva's linear decomposition of Dialectica [1]. Indeed, usually $\mathbb{C}(A + B)$ is defined as $\mathbb{C}(A) \times \mathbb{C}(B)$, but it turns out it is instead a pattern-matching over the corresponding witness. In a non-dependent setting, the two definitions agree, but here we need to be smarter.

The translation extends **CIC** with unexpected reasoning principles and can be seen as a way to expose intensional contents of the source theory. Most notably, it negates functional extensionality in a meaningful way: one can observe how a function uses its arguments, and in particular how many times, by looking at the multisets produced by its second component. This is very similar to the techniques developed in quantitative semantics [4], except that it gives access to the full expressive power of **CIC** and thus paves the way for the design of a type theory featuring an internal notion of complexity.

References

[1] V. de Paiva. A dialectica-like model of linear logic. In *Category Theory and Computer Science*, volume 389 of *Lecture Notes in Computer Science*, pages 341–356. Springer, 1989.

[2] K. Gödel. Über eine bisher noch nicht benützte Erweiterung des finiten Standpunktes. *Dialectica*, 12:280–287, 1958.

[3] Y. Lafont, B. Reus, and T. Streicher. Continuations semantics or expressing implication by negation. Technical Report 9321, Ludwig-Maximilians-Universitat, Munchen, 1993.

[4] J. Laird, G. Manzonetto, G. McCusker, and M. Pagani. Weighted relational models of typed lambda-calculi. In *LICS 2013*, pages 301–310, 2013.

[5] A. Miquel. Relating classical realizability and negative translation for existential witness extraction. In *TLCA 2009*, pages 188–202, 2009.

[6] P.-M. Pédrot. A functional functional interpretation. In *CSL-LICS 2014*, pages 77:1–77:10, New York, NY, USA, 2014. ACM.

[7] P.-M. Pédrot. *A Materialist Dialectica.* PhD thesis, Univ. Paris VII, 2015.