# Effects, Substitution and Induction

## An Explosive *Ménage à Trois*

Pierre-Marie Pédrot[1] and Nicolas Tabareau[1]

Inria Rennes – Bretagne Atlantique, LS2N

Since the inception of dependent type theory, several people tried to apply the techniques coming from simply-typed settings to enrich it with new reasoning principles using effects, typically classical logic. The early attempts were mixed, if not outright failures. Most notably, Barthe and Uustalu showed that writing a typed CPS translation preserving dependent elimination was out of reach [1], and similarly Herbelin proved that CIC is inconsistent with computational classical logic [3].

Retrospectively, this should not have been that surprising. This incompatibility is the reflect of a very ancient issue: mixing classical logic with the axiom of choice, whose intuitionistic version is a consequence of dependent elimination, is a well-known source of foundational problems [5]. While in the literature much emphasis has been put on the particular case of classical logic, we argue here that this is an instance of a broader phenomenon, namely that side-effects are at odds with dependent type theory, in a pick two out of three conundrum. This mismatch is evocatively dubbed the *ménage à trois* and is embodied by the following theorem, which is a generalization of Herbelin's paradox.

**Theorem 1** (Explosive *ménage à trois*)**.** *A type theory that features observable effects and enjoys both arbitrary substitution and dependent elimination is logically inconsistent.*

We describe more in detail the premises of this theorem hereafter, where $\star$ stands for any proof term, not necessarily unique.

**Definition 1.** *Substitution* is the admissibility of the following rule.

$$\frac{\Gamma, x : A \vdash \star : B \qquad \Gamma \vdash u : A}{\Gamma \vdash \star : B\{x := u\}}$$

**Definition 2.** *Dependent elimination* on booleans is the admissibility of the following rule.

$$\frac{\Gamma, x : \mathbb{B} \vdash A : \square \qquad \Gamma \vdash \star : A\{x := \mathtt{true}\} \qquad \Gamma \vdash \star : A\{x := \mathtt{false}\}}{\Gamma, x : \mathbb{B} \vdash \star : A}$$

Finally, we need to express what it means for a type theory to be observably effectful. Intuitively, a type theory is pure when every term observationally behaves as a value. So a simple way to formalize what it means to be effectful is to say that there exists a boolean term which is not observationally equivalent to `true` nor `false`.

**Definition 3.** A type theory is *observably effectful* if there exists a closed term $\vdash t : \mathbb{B}$ that is not observationally equivalent to a value, that is, there exists a context $C$ such that $C[\mathtt{true}] \equiv \mathtt{true}$ and $C[\mathtt{false}] \equiv \mathtt{true}$, but $C[t] \equiv \mathtt{false}$, where $\equiv$ denotes definitional equality.

*Proof.* We define equality and empty type using the standard impredicative encoding, and we take $t$ and $C$ as provided by Definition 2. By dependent elimination, it holds that $x : \mathbb{B} \vdash C[x] = \mathtt{true}$. By substitution, $\vdash C[t] = \mathtt{true}$. By conversion and because $C[t] \equiv \mathtt{false}$, this implies $\vdash \mathtt{false} = \mathtt{true}$. But, by dependent elimination, we also have $\vdash \mathtt{false} = \mathtt{true} \to \bot$. $\square$

**Example 1.** It is possible to use `callcc` [2] to write a term `decide` : $\square \to \mathbb{B}$ that decides inhabitance of a type. Obviously, `decide` $A$ cannot evaluate to a value in general. Such terms are called *backtracking* or *non-standard*, and are the root of Herbelin's paradox.

Facing this impossibility theorem, we briefly list possible ways out and their trade-offs.

**No Effects** This is the good old CIC, featuring both substitution and dependent elimination.

**Call-by-value** Every function can expect its argument to be a value, which explains why dependent elimination is always valid: `true` and `false` are the only non-variable boolean values. Constrastingly, substitution is now by definition restricted to values. Generalizing it to arbitrary terms is not correct if there are effectful terms, as evidenced by the requirement of a *value restriction* in most systems. Albeit not strictly speaking dependent type theory, this is the path followed by PML [4].

**Call-by-name** In this setting, substitution always holds by construction. However, as already noticed in [6], dependent elimination is now lost in general. If there are effectful terms, knowing the behaviour of a predicate on boolean *values* is not enough to know the behaviour of the predicate in general, as there is a desynchronization between effects performed in the term and effects performed in the type during pattern-matching. This is what BTT [6] is all about.

**Boom** It is possible to satisfy the premises of the theorem, at the expense of consistency. The exceptional type theory [7] is such an instance. While seemingly concerning at first, one can argue that this is a paradigm shift from a dependent *type theory* to a dependently-typed *programming language*, where consistency is not relevant.

We will give more in-depth insights about these paradigms, and advocate for an encompassing theory called $\partial$CBPV [8]. This is a generalization of call-by-push-value to dependent types allowing for a uniform setting in which describe these effectful theories.

# References

[1] G. Barthe and T. Uustalu. Cps translating inductive and coinductive types. In *Proceedings of Partial Evaluation and Semantics-based Program Manipulation*, pages 131–142. ACM, 2002.

[2] T. Griffin. A formulae-as-types notion of control. In *Seventeenth Symposium on Principles of Programming Languages, San Francisco, California, USA, January 1990*, pages 47–58, 1990.

[3] H. Herbelin. On the degeneracy of sigma-types in presence of computational classical logic. In P. Urzyczyn, editor, *Seventh International Conference, TLCA '05, Nara, Japan. April 2005, Proceedings*, volume 3461 of *Lecture Notes in Computer Science*, pages 209–220. Springer, 2005.

[4] R. Lepigre. A classical realizability model for a semantical value restriction. In *25th European Symposium on Programming, ESOP 2016*, pages 476–502, 2016.

[5] P. Martin-Löf. 100 Years of Zermelo's Axiom of Choice: What was the problem with it? *Comput. J.*, 49(3):345–350, 2006.

[6] P.-M. Pédrot and N. Tabareau. An effectful way to eliminate addiction to dependence. In *32nd Symposium on Logic in Computer Science, LICS 2017, Reykjavik, Iceland, June 20-23, 2017*, pages 1–12. IEEE Computer Society, 2017.

[7] P.-M. Pédrot and N. Tabareau. Failure is Not an Option – An Exceptional Type Theory. In *27th European Symposium on Programming, ESOP 2018*, pages 245–271, 2018.

[8] P.-M. Pédrot and N. Tabareau. The Fire Triangle. Draft at https://www.pédrot.fr/articles/dcbpv.pdf, 2019.