DEPENDENT TYPES SIMPLIFIED

TRISTAN BICE

Abstract

The standard approach to creating type systems is to start with a wishlist of inference rules and just hope that combining them does not break anything. Unsurprisingly, this often results in systems that are not even consistent, thus failing the most basic requirement of a logical system to be of any use or interest in mathematics. This is true not only of the early systems originally constructed by people such as Church, Curry and Martin-Löf (see [KR35], [Coq94] and [Hur95]) but right up to the more modern type systems underlying proof assistants such as Rocq and Lean (see [Car24]).

Even type systems created in this way that happen to be consistent often lack any simple semantics. Attempts to give semantics to such systems, particularly those based on dependent types, usually result in sophisticated interpretations that can not even be applied to terms in isolation. Rather, terms can only be interpreted as part of specific judgements with complicated rules changing the interpretation of the term based on the context of the judgement in question.

These issues are no doubt why type theory has had a limited impact upon mainstream mathematics and why classical systems like first order logic and ZFC in particular have remained the gold standard for over a century now, at least when it comes to systems that could realistically be viewed as a foundation for all mathematics. While type theorists tend to rail against this dominance of ZFC and insist that type theory is diametrically opposed to set theory, we aim to demonstrate in this talk that this need not be the case by outlining some simple dependent type systems with rigorous set theoretic semantics.

To do this we follow [Bic25] and approach dependent type systems in a more classical way, starting with the semantics on which we then base the syntactic system. Specifically, we first show how to interpret terms and the typing statements they form in an elementary but precise set theoretic way. A model for a collection of statements is then just an interpretation in which all those statements are valid. A semantic judgement $\Gamma \vDash X$ then indicates that every model for all the statements in Γ is a model for the statement X. Properties of $\vDash$ can then be taken as inference rules defining syntactic judgements $\Gamma \vdash X$, resulting in logical systems that are automatically sound and hence consistent, at least relative to ZFC.

References

[Bic25] Tristan Bice. Dependent types simplified, 2025. arXiv:2507.04071.

[Car24] Mario Carneiro. Lean4lean: Towards a verified typechecker for Lean, in Lean, 2024. arXiv:2403.14064.

[Coq94] Thierry Coquand. A new paradox in type theory. In *Logic, methodology and philosophy of science IX. Proceedings of the ninth international congress of logic, methodology and philosophy of science, Uppsala, Sweden, August 7–14, 1991*, pages 555–570. Amsterdam: North-Holland, 1994.

[Hur95] Antonius J. C. Hurkens. A simplification of Girard's paradox. In *Typed lambda calculi and applications. 2nd international conference, TLCA '95, Edinburgh, GB, April 10–12, 1995. Proceedings*, pages 266–278. Berlin: Springer-Verlag, 1995.

[KR35] S. C. Kleene and J. B. Rosser. The inconsistency of certain formal logics. *Ann. Math. (2)*, 36:630–636, 1935. doi:10.2307/1968646.