# A type theory with internal parametricity

Ambrus Kaposi
Eötvös Loránd Tudományegyetem
akaposi@inf.elte.hu

January 12, 2024

Parametricity is a way to express representation-independence. For example, in a language that satisfies parametricity, there is only one function of type $\Pi(A : \mathsf{Type}).A \to A$. This is the type of polymorphic functions which for every type $A$, compute an element of $A$ from an element of $A$. The idea is that representation-independent functions cannot inspect $A$, the only thing they can do is to return the element of $A$ that they take as input. Similarly, in a language with parametricity, there is no element of the type $\Pi(A : \mathsf{Type}).A$, and there are two elements of the type $\Pi(A : \mathsf{Type}).A \to A \to A$. The type $\Pi(A : \mathsf{Type}).A \to (A \to A) \to A$ is the type of abstract natural numbers, equivalent to the definition with Peano axioms. When working with such natural numbers, we do not need to choose between representations by Zermelo or von Neumann.

Parametricity was formalised by Reynolds [7] for the polymorphic lambda calculus as relation-preservation: polymorphic functions respect arbitrary relations. This implies the above example consequences, which can all be expressed for the polymorphic lambda calculus (System F). Bernardy et al. [3] extended parametricity to type theory, which is a full-scale language for the formalisation of mathematics, thus parametricity statements can be expressed in the language of type theory itself. However, parametricity is still *external*: inside type theory, we cannot prove that there is only one element of the type $\Pi(A : \mathsf{Type}).A \to A$, this is only a metatheorem. Internalising parametricity is difficult, because once we have a term witnessing parametricity in the language, it has to be parametric itself, and this induces higher dimensional structure: type theory has to be able to compute with arbitrary dimensional cubes. Type theories with internal parametricity have this higher dimensional structure explicitly built-in ([4, 5, 2, 6]).

In this talk I will introduce a new type theory with internal parametricity where the higher dimensional structure is emergent rather than built-in [1].

## References

[1] Thorsten Altenkirch, Yorgo Chamoun, Ambrus Kaposi, and Michael Shulman. Internal parametricity, without an interval. In *Proceedings of the 51st Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2024, London, UK, January 17 - 19, 2024*. ACM, 2024. Link to preprint.

[2] Jean-Philippe Bernardy, Thierry Coquand, and Guilhem Moulin. A presheaf model of parametric type theory. In Dan R. Ghica, editor, *The 31st Conference on the Mathematical Foundations of Programming Semantics, MFPS 2015, Nijmegen, The Netherlands, June 22-25, 2015*, volume 319 of *Electronic Notes in Theoretical Computer Science*, pages 67–82. Elsevier, 2015. `doi:10.1016/j.entcs.2015.12.006`.

[3] Jean-Philippe Bernardy, Patrik Jansson, and Ross Paterson. Parametricity and dependent types. In Paul Hudak and Stephanie Weirich, editors, *Proceeding of the 15th ACM SIGPLAN international conference on Functional programming, ICFP 2010, Baltimore, Maryland, USA, September 27-29, 2010*, pages 345–356. ACM, 2010. `doi:10.1145/1863543.1863592`.

[4] Jean-Philippe Bernardy and Guilhem Moulin. A computational interpretation of parametricity. In *Proceedings of the 27th Annual IEEE Symposium on Logic in Computer Science, LICS 2012, Dubrovnik, Croatia, June 25-28, 2012*, pages 135–144. IEEE Computer Society, 2012. `doi:10.1109/LICS.2012.25`.

[5] Jean-Philippe Bernardy and Guilhem Moulin. Type-theory in color. In Greg Morrisett and Tarmo Uustalu, editors, *ACM SIGPLAN International Conference on Functional Programming, ICFP'13, Boston, MA, USA - September 25 - 27, 2013*, pages 61–72. ACM, 2013. `doi:10.1145/2500365.2500577`.

[6] Evan Cavallo and Robert Harper. Internal parametricity for cubical type theory. *Log. Methods Comput. Sci.*, 17(4), 2021. `doi:10.46298/lmcs-17(4:5)2021`.

[7] John C. Reynolds. Types, abstraction and parametric polymorphism. In R. E. A. Mason, editor, *Information Processing 83, Proceedings of the IFIP 9th World Computer Congress, Paris, France, September 19-23, 1983*, pages 513–523. North-Holland/IFIP, 1983.