Internal and Observational Parametricity for Cubical Agda

Antoine Van Muylder, Andreas Nuyts, and Dominique Devriese

DistriNet, KU Leuven, Belgium

Introduction Relational parametricity [Rey83] of an object language $\mathscr{X}$ in a target language $\mathscr{Y}$ can be summarized as the following statement:

> **Parametricity:** For every ($\mathscr{S}$) type T in $\mathscr{X}$, we can define Reynolds' logical relation $[T]$ in $\mathscr{Y}$ and every ($\mathscr{P}$) program $t : T$ in $\mathscr{X}$ is self-related according to $[T]$ in $\mathscr{Y}$.

This statement contains two universal quantifications, which we have labeled with names $\mathscr{S}$ and $\mathscr{P}$ for the (meta)theories where these quantifications take place.

Two approaches exist to incorporate parametricity into proof assistants based on dependent type theory. On the one hand, parametricity translations from $\mathscr{X}$ to $\mathscr{Y}$ [BJP12, KL12, AM17] conveniently compute parametricity statements and their proofs solely based on *individual* well-typed polymorphic programs; individual in the sense that both quantifications take place in a metatheory. On the other hand, when $\mathscr{X} = \mathscr{Y}$, internally parametric type theories [NVD17, ND18, BCM15, Mou16, CH21] augment plain type theory with additional primitives that offer internal parametricity: the ability to write formal internal proofs that *any* polymorphic program of a *certain* type satisfies its parametricity statement. In other words, $\mathscr{P} = \mathscr{X} = \mathscr{Y}$ but $\mathscr{S}$ is still a metatheory. However these type theories lack mature proof assistant implementations and deriving parametricity in them involves low-level intractable proofs. We call a system observational if $\mathscr{S} = \mathscr{Y}$ and the assignment $T \mapsto [T]$ is available as a function in $\mathscr{Y}$.

We contribute Agda --bridges [VMND24]: the first practical internally parametric proof assistant. Agda --bridges is based on Cavallo and Harper's system (CH) for internally parametric HoTT [CH21] which is in turn based on Bernardy, Coquand and Moulin's system for internal parametricity [BCM15]. We provide the first mechanized proofs of crucial theorems for internal parametricity, like the relativity theorem (relational univalence). Moreover, Agda --bridges extends Agda --cubical [VMA21] and is capable of type-checking the cubical library [Agd].

We identify a high-level sufficient condition for proving internal parametricity which we call the structure relatedness principle (SRP) by analogy with the structure identity principle (SIP) of HoTT/UF. We state and prove a general parametricity theorem for types that satisfy the SRP. Our parametricity theorem lets us obtain one-liner proofs of standard internal free theorems. We observe that the SRP is harder to prove than the SIP and provide in Agda --bridges a shallowly embedded type theory to compose types that satisfy the SRP, which we call Relational Observational Type Theory (ROTT). Thus, while Agda --bridges is internally parametric in the above sense, for ROTT we have a situation where $\mathscr{X}$ = ROTT and $\mathscr{Y} = \mathscr{P} = \mathscr{S}$ = Agda --bridges. As such, ROTT is observationally but – unlike [ACKS24] – not internally parametric.

Cavallo and Harper's system (CH) Cavallo and Harper's system (CH) for internally parametric HoTT [CH21] is modelled in bicubical sets, i.e. presheaves over pairs of cubes. One of these cubes models a power of the *path interval* I from cubical HoTT [CCHM17, AFH18]. This is the 'walking' type with two elements i0 i1 : I that are equivalent. As such, equivalence of any two objects a0 a1 : A can be expressed as a *path* from a0 to a1, which is a function from I to A mapping i0 and i1 to a0 and a1 (resp.). Analogously, the other cube models a power of the *bridge interval* BI which is the 'walking' type with two elements bi0 bi1 : BI that are related. As such, relatedness of any two objects a0 a1 : A can be expressed as a *bridge* from a0 to a1, which is a function from BI to A mapping bi0 and bi1 to a0 and a1 (resp.).

The idea is that the type of bridges $\mathsf{Bridge}_{\mathsf{A}}\,\mathsf{a0}\,\mathsf{a1}$ will be equivalent to the logical relation defined by Reynolds [Rey83] by induction on the formation of A. For many concrete type formers, this can be proven internally, e.g. we can prove that a bridge in the Σ-type is a pair of bridges, the

second one dependent over the first one. However, additional primitives are needed for two specific type formers: the universe and the Π-type [BCM15, CH21]. The Gel primitive turns a relation $\mathsf{T0} \to \mathsf{T1} \to \mathsf{Type}$ into a bridge $\mathsf{Bridge}_{\mathsf{Type}}\,\mathsf{T0}\,\mathsf{T1}$, which is sufficient to prove the **relativity** theorem which states that bridges in the universe are equivalent to relations. The extent primitive produces a bridge $\mathsf{ff} : \mathsf{Bridge}_{(x:\mathsf{A})\to B\,x}\,\mathsf{f0}\,\mathsf{f1}$ from an operation fbrid that maps bridges $\mathsf{aa} : \mathsf{Bridge}_{\mathsf{A}}\,\mathsf{a0}\,\mathsf{a1}$ to dependent bridges $\mathsf{fbrid}\,\mathsf{aa} : \mathsf{BridgeP}_{i.\mathsf{B}\,i}\,(\mathsf{f0}\,\mathsf{a0})\,(\mathsf{f1}\,\mathsf{a1})$. The assignment $\mathsf{fbrid} \mapsto \mathsf{ff}$ is then also provably an equivalence, characterizing the logical relation in the function type.

Both primitives take essentially 3 arguments: two for the endpoints and one logical relation proof. Iterated use of Gel and extent allows the creation of bridge squares and cubes, and similarly takes 3^n arguments. This means that there is no opportunity for the user to define a bridge cube's behaviour on diagonals. For this reason, the bridge interval is *affine* [BCM15, CH21]: bridge variable contraction is disallowed, and the bridge cubes in the model have no diagonals.

Implementation in Agda --bridges Agda --bridges diverges from CH in two important respects: **(1)** The CH system uses cartesian path cubes [AFH18], but Agda --bridges extends Agda --cubical [VMA21] which uses De Morgan cubes [CCHM17]. As a consequence, the face constraint logic (a.k.a. cofibration logic) of Agda --bridges differs from CH. **(2)** Whereas CH prevents bridge variable contraction using an operation on contexts which removes the non-fresh parts, Agda --bridges uses a syntactic check that was already implemented as part of Agda --guarded [VV20].

Some important challenges faced during the implementation of Agda --bridges are: **(1)** The backwards-compatible adaptation of the Kan operations hcomp and transp to support face constraints mentioning not only path variables but also bridge variables. **(2)** The introduction of computation rules that perform intended *bridge variable capture*, namely the β-rule for extent and the η-rule for Gel. This capture is only safe and sound if the expression in which the bridge variable i is captured, is *semi-fresh* for i: it is allowed to mention i but not any variables susceptible to substitutions with expressions not fresh for i. Thus, a semi-freshness check had to be implemented as part of the reduction algorithm.

The Structure Relatedness Principle (SRP) Agda --bridges is internally parametric but not observationally so: all functions are known to preserve bridges, but it is not a priori clear that the bridge type is equivalent to the logical relation defined by Reynolds [Rey83], a property which we call the **structure relatedness principle (SRP)**. As indicated above, the SRP can be proven by *manual* induction on the formation of the type.[1] This is tedious in itself, but we argue that the SRP is moreover objectively harder to prove than the structure identity principle (SIP) in HoTT, because **(1)** a proof obligation is raised for the domain of the Π-type, **(2)** there is no J-rule for bridges, and **(3)** the bridge type at a mere proposition is again a mere proposition, *not* necessarily contractible and thus not automatically characterized up to equivalence.

To address this, we define the notion of a **relativistic reflexive graph (RRG)**. This is a type equipped with a relation that is equivalent to the bridge type (and therefore reflexive).[2] Of course, using univalence, it is immediately clear that every type is an RRG in exactly a single way up to equivalence. However, the idea is to take care to choose the correct relation up to definitional equality, namely to choose Reynolds' logical relation. Then the RRG structure proves the SRP for the underlying type. We build a raw[3] category with families (CwF) [Dyb95] of RRGs, which amounts to a shallow embedding of dependent type theory in Agda --bridges; we call this shallow syntax **Relational Observational Type Theory (ROTT)**. Users can then construct types satisfying the SRP by constructing them in ROTT. It follows internally to Agda --bridges that any Agda function between carriers of RRGs (ROTT types; including in particular System F types) respects the logical relations. We demonstrate our approach on Church encodings.

[1] Manual, because internal induction on types is not possible. It could perhaps be reconciled with univalence by treating Glue as a path constructor, but treating Gel as a bridge constructor may be problematic due to the affine nature of BI, and there's also the fact that Agda has an extensible universe with user-defined type constructors.

[2] This idea is analogous to the use of univalent categories/groupoids/graphs in HoTT [Sch20, Uni13].

[3] By 'raw', we mean we only implement the operations of a CwF and neglect to prove the axioms, which we expect to hold in a higher-dimensional way.

Acknowledgements We thank Andrea Vezzosi for continuously sharing with us his expertise and sound suggestions regarding Agda --cubical and Agda --bridges. We thank Rasmus Møgelberg and Andrea Vezzosi for welcoming the first author at the IT University of Copenhagen. Antoine Van Muylder holds a PhD fellowship (11H9921N) of the Research Foundation – Flanders (FWO). Andreas Nuyts holds a Postdoctoral fellowship (1247922N) of the Research Foundation – Flanders (FWO). This research is partially funded by the Research Fund KU Leuven and by the Research Foundation - Flanders (FWO; G030320N).

References

[ACKS24] Thorsten Altenkirch, Yorgo Chamoun, Ambrus Kaposi, and Michael Shulman. Internal parametricity, without an interval. In *To appear in: Proceedings of the 51st Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2024*. ACM, 2024.

[AFH18] Carlo Angiuli, Kuen-Bang Hou (Favonia), and Robert Harper. Cartesian cubical computational type theory: Constructive reasoning with paths and equalities. In Dan R. Ghica and Achim Jung, editors, *27th EACSL Annual Conference on Computer Science Logic, CSL 2018, September 4-7, 2018, Birmingham, UK*, volume 119 of *LIPIcs*, pages 6:1–6:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018. URL: https://doi.org/10.4230/LIPIcs.CSL.2018.6, doi:10.4230/LIPICS.CSL.2018.6.

[Agd] The Agda Community. A standard library for Cubical Agda. URL: https://github.com/agda/cubical.

[AM17] Abhishek Anand and Greg Morrisett. Revisiting parametricity: Inductives and uniformity of propositions. *CoRR*, abs/1705.01163, 2017. URL: http://arxiv.org/abs/1705.01163, arXiv:1705.01163.

[BCM15] 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. URL: https://doi.org/10.1016/j.entcs.2015.12.006, doi:10.1016/J.ENTCS.2015.12.006.

[BJP12] Jean-Philippe Bernardy, Patrik Jansson, and Ross Paterson. Proofs for free - parametricity for dependent types. *J. Funct. Program.*, 22(2):107–152, 2012. doi:10.1017/S0956796812000056.

[CCHM17] Cyril Cohen, Thierry Coquand, Simon Huber, and Anders Mörtberg. Cubical type theory: A constructive interpretation of the univalence axiom. *FLAP*, 4(10):3127–3170, 2017. URL: http://collegepublications.co.uk/ifcolog/?00019.

[CH21] Evan Cavallo and Robert Harper. Internal parametricity for cubical type theory. *Log. Methods Comput. Sci.*, 17(4), 2021. URL: https://doi.org/10.46298/lmcs-17(4:5)2021, doi:10.46298/LMCS-17(4:5)2021.

[Dyb95] Peter Dybjer. Internal type theory. In Stefano Berardi and Mario Coppo, editors, *Types for Proofs and Programs, International Workshop TYPES'95, Torino, Italy, June 5-8, 1995, Selected Papers*, volume 1158 of *Lecture Notes in Computer Science*, pages 120–134. Springer, 1995. doi:10.1007/3-540-61780-9_66.

[KL12] Chantal Keller and Marc Lasson. Parametricity in an impredicative sort. In Patrick Cégielski and Arnaud Durand, editors, *Computer Science Logic (CSL'12) - 26th International Workshop/21st Annual Conference of the EACSL, CSL 2012, September 3-6, 2012, Fontainebleau, France*, volume 16 of *LIPIcs*, pages 381–395. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2012. URL: https://doi.org/10.4230/LIPIcs.CSL.2012.381, doi:10.4230/LIPICS.CSL.2012.381.

[Mou16] Guilhem Moulin. *Internalizing Parametricity*. PhD thesis, Chalmers University of Technology, Gothenburg, Sweden, 2016. URL: http://publications.lib.chalmers.se/publication/235758-internalizing-parametricity.

[ND18] Andreas Nuyts and Dominique Devriese. Degrees of relatedness: A unified framework for parametricity, irrelevance, ad hoc polymorphism, intersections, unions and algebra in dependent type theory. In Anuj Dawar and Erich Grädel, editors, *Proceedings of the 33rd Annual*

ACM/IEEE Symposium on Logic in Computer Science, LICS 2018, Oxford, UK, July 09-12, 2018, pages 779–788. ACM, 2018. doi:10.1145/3209108.3209119.

[NVD17] Andreas Nuyts, Andrea Vezzosi, and Dominique Devriese. Parametric quantifiers for dependent type theory. *Proc. ACM Program. Lang.*, 1(ICFP):32:1–32:29, 2017. doi:10.1145/3110276.

[Rey83] 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.

[Sch20] Johannes Schipp von Branitz. Higher groups via displayed univalent reflexive graphs in cubical type theory. Master's thesis, Technische Universität Darmstadt, 10 2020.

[Uni13] The Univalent Foundations Program. *Homotopy Type Theory: Univalent Foundations of Mathematics*. Institute for Advanced Study, 2013. URL: https://homotopytypetheory.org/book/.

[VMA21] Andrea Vezzosi, Anders Mörtberg, and Andreas Abel. Cubical agda: A dependently typed programming language with univalence and higher inductive types. *J. Funct. Program.*, 31:e8, 2021. doi:10.1017/S0956796821000034.

[VMND24] Antoine Van Muylder, Andreas Nuyts, and Dominique Devriese. Internal and observational parametricity for cubical agda. *Proc. ACM Program. Lang.*, 8(POPL), jan 2024. doi:10.1145/3632850.

[VV20] Niccolò Veltri and Andrea Vezzosi. Formalizing π-calculus in guarded cubical agda. In Jasmin Blanchette and Catalin Hritcu, editors, *Proceedings of the 9th ACM SIGPLAN International Conference on Certified Programs and Proofs, CPP 2020, New Orleans, LA, USA, January 20-21, 2020*, pages 270–283. ACM, 2020. doi:10.1145/3372885.3373814.