Extending Cubical Agda with Internal Parametricity*

Antoine Van Muylder\textsuperscript{1}, Andrea Vezzosi, Andreas Nuyts\textsuperscript{1}, and Dominique Devriese\textsuperscript{1}

\textsuperscript{1}KU Leuven, Belgium

Abstract. Internally parametric type theories are type systems augmented with additional primitives and typing rules allowing the user to prove parametricity statements within the system, without resorting to axioms. We implement such a type system by extending the cubical type theory of Cubical Agda [17] with parametricity primitives proposed by Cavallo and Harper [9]. To assess the implementation, we formalize a general parametricity theorem within the system, which entails a large spectrum of free theorems including a Church encoding for the circle and a straightforward parametric model for System F\textsuperscript{1}.

A type-polymorphic function is \textit{parametric} if its type argument is merely used for typing, not for computing purposes. Such a parametric function necessarily applies the same algorithm irrespective of the type it is being used at. Reynolds’ relational parametricity [15, 13] is a semantic account of this property for, e.g., terms of System F (a.k.a. the second-order polymorphic lambda calculus). Useful information can systematically be extracted by only looking at the type of a parametric function. These facts commonly known as “free theorems” [18] provide, for instance, a formal explanation as to why there are only two functions with type $\forall \alpha. \alpha \to \alpha \to \alpha$.

Enforcing free theorems. Dependent type theory (DTT) has been proven to admit parametric models [16, 6, 12, 3]. Therefore, whenever a free theorem is needed, it can soundly be added as an axiom. In fact, evidence that a given closed term is parametric (w.r.t. a syntactic notion of relation) can even be obtained constructively: this is what parametricity translations [11, 1] achieve. However, parametricity is known to be logically independent from plain DTT [7] and this prevents the above meta-theoretical translations to be internalized. Hence, to obtain internal parametricity, novel principles must be added to plain DTT.

Internal parametricity for cubical type theory. Cavallo and Harper (CH) [9] extend cubical type theory [2] with parametricity primitives [5], in a style reminiscent of cubical type theory itself. Proofs of relatedness (versus equality) between $a_0, a_1 : A$ are built using functions $p : \text{Bl} \to A$ from an abstract \textit{bridge interval} $\text{Bl}$, satisfying $p(0) = a_0, p(1) = a_1$ definitionally. Such proofs are called \textit{bridges} and written $\lambda_{\text{Bl}} r.p(r) : \text{Bridge}_A a_0 a_1$ (versus paths $\lambda i.p(i) : \text{Path}_A a_0 a_1$). Contrary to path variables, the logic of bridge variables is sub-structural (affine): weakening and exchange hold, but not contraction. Concretely, one can eliminate a bridge $\Gamma_1, r : \text{Bl}, \Gamma_2 \vdash b : \text{Bridge}_A a_0 a_1$ at a bridge variable $r$ only if $r$ is \textit{fresh} for $b$, meaning that every free variable appearing in $b$ is in $\Gamma_1$ or is a bridge/path variable in $\Gamma_2$. This sub-structurality is crucial to formulate the inference rules of the \textit{extent} and \textit{Gel} primitives. The purpose of those primitives is to guarantee several \textit{bridge commutation principles}: theorems explaining how the $\text{Bridge}$ type former commutes with other type formers. The \textit{extent} primitive and its rules provide commutation with $\Pi$. For non-dependent functions this reads

*Van Muylder/Nuyts hold a PhD/Postdoctoral fellowship of the Research Foundation – Flanders (FWO).

\textsuperscript{1}Files, instructions, comparison with other systems: https://github.com/antoinevanmuylder/bridgy-lib.
(Π(α₀,α₁:A)(a:Bridge α₀ α₁),Bridge₀ₐ(f₀α₀)(f₁a₁)) ≃ Bridgeₐₐ → B f₀ f₁. The principle is analogous to
function extensionality and asserts that functions are related if they map related inputs to related outputs. The Gel primitive and its rules, together with univalence, prove commutation
with the universe: (A₀ → A₁ → Type) ≃ Bridgeₐₐ → Type A₀ A₁. This is analogous to univalence and
called relativity by CH. Commutation principles make bridges (and paths) behave as structured
relations (and isomorphisms, resp.). This is most blatant for types of algebraic structures. Con-

Contributions. We implement CH’s internally parametric type theory [9] on top of the
cubical type theory [8] underlying the Cubical Agda [17] proof assistant. As discussed above,
we must be able to generate freshness constraints for bridge variables during typechecking.
Our implementation hence reuses the existing affine variable infrastructure of Guarded Cubical
Agda [14]. Interestingly and unprecedented in Agda, the extent β-reduction, the rule has to operate on the normal form of M in the worst case scenario. Our

current implementation of extent β is sound but not complete because of this peculiar behaviour.
Implementation of Kan operations for Bridge,Gel is work in progress as well.

Our long term goals include assessing the precise expressivity of CH’s internal parametricity,
connecting it to existing alternate formulations (unary, Kripke, etc.) and evaluating its usefulness in practical applications. For now, we have already formalised a (binary) parametricity statement from which a wide range of free theorems ensue. A native reflexive graph is by
definition a type of vertices G equipped, for any g₀,g₁ : G with a type of edges G{g₀,g₁}
and an equivalence ηG : G{g₀,g₁} ≃ Bridge₀ₐ g₀g₁. The type of native reflexive graphs is of
course equivalent to Type, but ηG can contain non-trivial information. For instance, Type
equipped with relations A₀ → A₁ → Type as edges is native exactly thanks to relativity, and
formalizing the relativity theorem was non-trivial. Similarly, a native relator F between native
reflexive graphs G,H : Type acts both on vertices FVERT : G → H and on edges Fedge : G{g₀,g₁} → H{F₀g₀,F₁g₁} and the latter action must satisfy Path,(F₁bdg ♯ ηₐ)(ηᵣa ♯ Fedge)
where Fbdg = (λq,F₁bdg FVERT(gₓ)). Parametricity now reads as follows: for any native relator
F : G → Type and any function f : Πₓ:G Fₓ, inputs related by an edge e : G{g₀,g₁} result in a
proof (param : F.edge e (F₀g₀)(F₁g₁)). Bridge commutation principles ensure that native relators
abound. For instance the arrow relator Type × Type → Type : A,B → A → B is native, and
using the above param constant it is easy to show that (Πₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜ....
References