Into the Wild: Heterogeneous Quivers and Level-Polymorphic (Co)limits in Cubical Agda

Pavel Turyansky, Alexander Gryzlov

15.05.2026

Motivation In dependently typed programming, one frequently encounters operators such as $\_ \times \_$ (cartesian product) and the covariant map `_<$>_` : $(A \to B) \to F\,A \to F\,B$, used in both mathematical and programming contexts. The cartesian product is the canonical example of a universal construction, while `_<$>_` is a recurring idiom for a functorial action on morphisms; both pose challenges for existing formalizations in two independent ways.

First, existing formalizations of category theory fix the universe level of objects and morphisms upfront. Conforming to this rigid discipline without (structured) type universe cumulativity makes it unsuitable for programming: consider the cartesian product $\mathbb{N} \times \mathsf{Type}_0$, where $\mathbb{N} : \mathsf{Type}_0$ and $\mathsf{Type}_0 : \mathsf{Type}_1$ inhabit different universes, so the standard categorical product, which requires a single level for all objects, demands an explicit lift. The problem can also be encountered in HoTT lemmas: facts about equivalences (composition, two-out-of-three, pre/postcomposition) are cleanest to prove as categorical statements in the *large* category of types; however, a small-category formalization forces a specialized restatement. One could maintain a parallel hierarchy of level-polymorphic constructions alongside the standard small ones, but this duplicates definitions and defeats the purpose of a unified notational framework.

Second, for `_<$>_` to make sense it must be binary, which fixes the target category to be the category of types and functions. The proper generalization is then a copresheaf action, leaving the source category arbitrary while valuing the action in types. Types, however, do not form a category in HoTT, motivating the wild framework below.

Level Polymorphism Following Allais [All19], we parameterize object types by a *vector* of universe levels rather than a single level. The key idea is that a large structure need not take a single universe level as parameter; instead it takes a natural number n and a function $\mathsf{Level}^n \to \mathsf{Level}$ that computes the level from a vector of level arguments:

Definition 1. *A* universe-polymorphic wild category *consists of* $n : \mathbb{N}$, $\ell_{\mathsf{Ob}} : \mathsf{Level}^n \to \mathsf{Level}$, $\mathsf{Ob} : (ls : \mathsf{Level}^n) \to \mathsf{Type}_{\ell_{\mathsf{Ob}}(ls)}$, *and* $\mathsf{Hom} : \{lxs\ lys : \mathsf{Level}^n\}\{\ell_{hom} : \mathsf{Level}^n \to \mathsf{Level}^n \to \mathsf{Level}\} \to Ob\ lxs \to Ob\ lys \to Type(\ell_{hom}\ lxs\ lys)$.

Setting $n = 0$ recovers small (fixed-level) categories; $n = 1$ with $\ell_{\mathsf{Ob}}(\ell) = \ell + 1$ gives the category of all types at finite levels, in which $\mathbb{N}$ and Type_0 coexist as objects without lifting. The vector length n must be chosen explicitly, but within that choice Agda's unifier is then able to infer all levels.

This approach to level polymorphism is independently useful and can be applied to any categorical structure. Note that the resulting structures are *large* by construction, so they are unable to express things like functor categories. But this is fine: every small (fixed-level) category can be cast into the large, universe-polymorphic framework, so constructions such as (co)limits need only be developed once at the large level and then specialized, rather than being duplicated for each small incarnation [Ste24b].

Reflexive Graphs The second problem arises because categories require Homs to be sets (h-level 2), which types in HoTT need not satisfy. Dropping this constraint yields *wild* categories. We can go further: removing additional structure (associativity, unitality, then composition itself) reveals that (co)limits depend only on a *reflexive graph* equipped with push/pull lenses [Ste24a]. Concretely, an oplax covariant push lens equips a family of reflexive graphs $B(x)$ over a base A with forward transport $\mathsf{push}^p_B : |B(x)| \to |B(y)|$ along each edge $p : x \approx_A y$ of the base, plus a unitor identifying transport along the reflexive edge with the identity; the dually-oriented pull lens describes backward transport ([Ste24a, Def. 54]). In particular, `_<$>_` is simply a push.

Manually writing down the definition of $\_ \times \_$ for reflexive graphs shows that we still have to somehow compose morphisms. Luckily, morphism composition itself can be expressed either as a *pull* or as a *push* in a specific family of reflexive graphs (Hom with one endpoint fixed). Pull corresponds to limit-like constructions, push to colimit-like ones; full composition is not needed.

For binary products, the universal property uses only pull, which we write infix as $\triangleleft$ (pulling a morphism back along another one on its target side): given projections $\pi_1 : \mathsf{Hom}(P, X)$, $\pi_2 : \mathsf{Hom}(P, Y)$ and morphisms $f : \mathsf{Hom}(Q, X)$, $g : \mathsf{Hom}(Q, Y)$, the pairing morphism $\langle f, g\rangle : \mathsf{Hom}(Q, P)$ satisfies $\langle f, g\rangle \triangleleft \pi_1 = f$ and $\langle f, g\rangle \triangleleft \pi_2 = g$. In the formalization:

```
record Product (X : Ob) (Y : Ob) (P : Ob) : Typeω where
  field π₁   : Hom P X
        π₂   : Hom P Y
        ⟨_,_⟩ : Hom Q X → Hom Q Y → Hom Q P
        ⟨⟩◁π₁ : ⟨ f , g ⟩ ◁ π₁ = f
        ⟨⟩◁π₂ : ⟨ f , g ⟩ ◁ π₂ = g
```

(For readability, universe-level arguments are elided throughout: each occurrence of `Ob` abbreviates `Ob lxs` for an implicit level vector `lxs : Level^n`, with separate vectors per object, and similarly for `Hom` and `Het`.) Dually, binary coproducts use only push.

Heterogeneity, polarization and quivers A standard method for splitting a structure into covariant and contravariant parts is to *polarize* its carrier [Ste24a, Neu23]. Sterling shows that one can simultaneously define algebraic structure and its homomorphisms (Section 1.4 of [Ste24a], magma example); the question is whether it's possible to do the same for wild categories. Applied to graphs, polarization replaces $\mathsf{Hom} : \mathsf{Ob} \to \mathsf{Ob} \to \mathscr{U}$ with a family between two distinct types:

Definition 2. *A* heterogeneous quiver *(*hetquiver*) on types* Ob^- *and* Ob^+ *is a family* $\mathsf{Het} : \mathsf{Ob}^- \to \mathsf{Ob}^+ \to \mathscr{U}$.

The resulting concept of heteromorphism, a morphism between objects of different sorts, appears sparingly in the literature [Péc12, Sat23, Ell07]. Lee's profunctorial (co)limits [Lee23] are closest to our universal constructions: he develops (co)limits via indexed profunctors over 2-categories, which we simplify to the quiver level.

This definition admits the same universe-polymorphic refinement as Definition 1:

```
record Quiver-onω m Ob⁻ n Ob⁺ ℓ-het : Typeω where
  field Het : Ob⁻ lxs → Ob⁺ lys → Type (ℓ-het lxs lys)
```

Hetquivers subsume all prior structures: 1) they can be specialized to the usual graphs by setting $\mathsf{Ob}^- = \mathsf{Ob}^+$, so homomorphisms are just a particular instance of heteromorphisms; 2) they can be equipped with further structure to yield profunctors.

Cubical Agda's Path^P is naturally a hetquiver. Given a line of types $A : \mathbb{I} \to \mathsf{Type}_\ell$, the dependent path type $\mathsf{Path}^P\, A$ relates elements of $A\,\mathsf{i0}$ to elements of $A\,\mathsf{i1}$.

Writing (co)limit definitions becomes *type-directed*: because arrows flow only from Ob^- to Ob^+, one cannot accidentally flip a morphism. Colimits inhabit Ob^+, limits inhabit Ob^-, and every variance error is caught by the type checker. For example, the general limit:

```
record Limit (d : Ob⁺) : Typeω where
  field apex     : Ob⁻
        ψ        : Het apex d
        lim-univ : is-universalω⁻ ψ
```

The apex lives in Ob^-, the universal morphism ψ is a heteromorphism, and universality states that any $u : \mathsf{Het}(x, d)$ factors uniquely through ψ via a homomorphism in $\mathsf{Hom}(x, \mathit{apex})$, all enforced by polarity. Being universal is a property after universe level instantiation. Exponentials and other weighted (co)limits are also expressible in this framework.

Formalization. The development is part of the `cubical-mini` library[1] in Cubical Agda. Approximately half of Sterling's paper [Ste24a] has been ported to the heterogeneous setting.

References

[All19] G. Allais. Generic level polymorphic n-ary functions. *TyDe '19*, pp. 14–26, 2019.

[Ell07] D Ellerman. Adjoint functors and heteromorphisms. *arXiv:0704.2207*, 2007.

[Lee23] S. Lee. Indexed profunctors over 2-categories. *arXiv:2302.06515*, 2023.

[Péc12] B. Pécsi. On Morita contexts in bicategories. *Appl. Categ. Struct.*, 20(4):415–432, 2012.

[Neu23] J. Neumann. Paranatural category theory. *arXiv:2307.09289*, 2023.

[Sat23] C. Sattler. Gluing along a profunctor. Unpublished note, 2023.

[Ste24a] J. Sterling. Reflexive graph lenses in univalent foundations. *arXiv:2404.07854*, 2024.

[Ste24b] J. Sterling. Large and small in univalent foundations. Blog post, https://www.jonmsterling.com/01I1/, 2024.

[Ste25] J. Sterling. Univalent foundations II. Lecture notes, 2025.

[1] https://github.com/cubical-mini/core, modules `Foundations.Base` and `Foundations.Lens`.