

**Universal construction** is about picking an object in a category *uniquely* by its relationship with other objects without making reference to its internal structure in this way

1. define a pattern
2. match it in the category (like in a query)
3. rank the matches by defining as comparison criteria the number of morphisms between matches
4. pick the *best* one: the one that has a unique morphism with any other

Universal construction defines the absolute best optimal fit, the perfect embodiment, the **ideal**. The unique morphism specify the perfect model what must be included in any candidate. the unique morphism *distills* the candidate to the essential, the perfect, the ideal.

The simplest possible pattern is a *single object* this brings to the definition of initial & terminal objects.

A more complex pattern is

$$X \xleftarrow{p_1} P \xrightarrow{p_2} Y$$

imposing a condition on projections and taking the pattern that is *final* we get to the definition of product.

This pattern-matching thing is formalized by functors.

See [1] Ch.9.1.

---

[1] B. Milewski, *Category Theory for Programmers* (2019).

---