Unique Universal Binary Relations

by Sven Nilsen, 2020

In this paper I introduce unique universal binary relations, which simplifies abstract generalizations.

A binary relation is an ordered pair:

    (a, b)

This can also be modeled as a predicate `p`:

    p(a, b) : bool

A *universal* binary relation can infer `p` from `b`:

    role_of(b) = p

One says that `b` is assigned the “role” `p`. For any `q <¬=> p`, `q(a, b) = false`.
A *unique* binary relation has the property that `b` can be inferred from `p(a)`:

    p(a) = b

Unique universal binary relations also permit multiple predicates, e.g. `p(a) = b` and `q(a) = c`.

Members of types are unique universal binary relations:

false : bool	type_of(false, bool)	type_of(false) = bool
true : bool	type_of(true, bool)	type_of(true) = bool

The role of `bool` is `type_of`. Every relation to `bool` is a type judgement.
The values `false` and `true` can also be assigned a role `value_of` to perform computations:

not(false) = true	value_of(not(false), true)	value_of(not(false)) = true
not(true) = false	value_of(not(true), false)	value_of(not(true)) = false

Now, look at the following law in Type Theory for Cartesian products:

∀ x : X, y : Y { **(x, y) : (X, Y)** }

This law can be generalized over all Cartesian products, without quantification over all predicates:

∀ X, Y { role_of(X) == role_of(Y) => role_of(**(X, Y)**) == role_of(X) }	lift role
∀ x, y, X, Y { (x, X) ∧ (y, Y) => **((x, y)**, **(X, Y))** }	lift relation

value_of((not(false), not(true))) = (true, false)
type_of((not(false), not(true))) = (type_of(not(false)), type_of(not(true)))