The Joker Type

by Sven Nilsen, 2021

In this paper I introduce a Joker type, which is a kind of subjective/biased path.

An involution[1] `inv` is a map such that:

inv · inv <=> id

For the Joker Type, there exists an involution as an abstract type[2] constructor, such that:

inv(T) = U inv(U) = T

inv : type → type

With other words, the involution maps one type `T` into its mirror type `U`.
Notice that only types are mapped by `inv`, while members of types are not mapped by default.

A Joker Type `joker(T)` is defined as the following (inspired by Joker Calculus[3]):

a : T	=>	me(a) : joker(T)
b : U	=>	other(b) : joker(T)

Since `T` can be mapped into `U` using an involution, one also gets a mirror Joker Type `joker(U)`:

a : T	=>	other(a) : joker(U)
b : U	=>	me(b) : joker(U)

This means that a Joker Type has two members, one “me” and one “other”.
However, the mirror Joker Type also has two members,
where the “me” becomes the “other” and the “other” becomes the “me”.

In the paper “The Subjective Sense of IO”[4], I showed that subjectivity emerges from the sense of input and output and forms a Catușkoți[5]. By thinking about the Joker Type as an arrow from `a` to `b`, one can construct an Intuitionistic Propositional[6] representation of the Catușkoți:

me(a) : joker(T)	“me” is the subject
other(b) : joker(T)	“other” is the subject
eq_t_u : (T → U, U → T)	members of `T` and `U` can be mapped
inv(T) ¬= U ∨ inv(U) ¬= T	there is no involution between `T` and `U`

This means that the Joker Type is in some sense “subjective”. It is like a biased path[7].
The subjective property is important because one can introduce the “unbiased position” externally:

none(), some(me(a)), some(other(b)) : opt[joker(T)]

The type `opt[joker(T)]` is contractible[8], but in a “false/nothing” sense, which was predicted by Avatar Graphs[9]. This bridges the gap between Avatar Extensions[10] and constructive logic[6]. The sum type `joker(T) + ⊥` is a 1-avatar of `joker(T)`. The empty type `⊥` is the unit element of `+`. When `+` is lifted to multiplication, `⊥` becomes `1` and this gives meaning to Avatar Algebra[11].

References:

[1] "Involution (mathematics)"
Wikipedia
https://en.wikipedia.org/wiki/Involution_(mathematics)

[2] "Type theory"
Wikipedia
https://en.wikipedia.org/wiki/Type_theory

[3] "Joker Calculus"
Daniel Fischer, William Alexander Morris, Sven Nilsen, 2021
https://github.com/advancedresearch/path_semantics/blob/master/papers-wip2/joker-calculus.pdf

[4] "The Subjective Sense of IO"
Sven Nilsen, 2021
https://github.com/advancedresearch/path_semantics/blob/master/papers-wip2/the-subjective-sense-of-io.pdf

[5] "Catuṣkoṭi"
Wikipedia
https://en.wikipedia.org/wiki/Catu%E1%B9%A3ko%E1%B9%ADi

[6] "Intuitionistic logic"
Wikipedia
https://en.wikipedia.org/wiki/Intuitionistic_logic

[7] "Homotopy"
Wikipedia
https://en.wikipedia.org/wiki/Homotopy

[8] "homotopy level"
nLab
https://ncatlab.org/nlab/show/homotopy+level

[9] "Avatar Graphs"
Sven Nilsen, 2020-2021
https://github.com/advancedresearch/path_semantics/blob/master/papers-wip/avatar-graphs.pdf

[10] "Avatar Extensions"
AdvancedResearch – Summary page on Avatar Extensions
https://advancedresearch.github.io/avatar-extensions/summary.html

[11] "Avatar Algebra"
AdvancedResearch – Summary page on Avatar Extensions
https://advancedresearch.github.io/avatar-extensions/summary.html#avatar-algebra