Taking "algebraically" seriously in the definition of algebraically injective type

Martín Escardó

School of Computer Science,

University of Birmingham, UK

12th ASSUME seminar, 5th June 2025

Nottingham, UK

D injective:

$$\begin{tikzcd} X \arrow[r, hook, "\forall j"] \arrow[dr, "\forall f"'] & Y \arrow[d, dashed, "\exists f'"] \\ & D \end{tikzcd}$$

Algebraically injective:

Replace $\exists$ by Σ.

Abstract

Theorem. In a 1-topos, the following two categories are isomorphic, with an isomorphism that is the identity on objects:

1. Pullback-natural, associative, algebraically injective objects.
2. Algebras of the partial-map classifier (aka lifting) monad.

- Partial results towards the ∞-topos situation.
- We work in HoTT/UF.

I. Algebraic injectives (MSCS'2021)

Def. Algebraic injective structure on a type D consists of

1. An extension operation, for any types X and Y,

$$(-)|(-) : (X \to D) \times (X \hookrightarrow Y) \longrightarrow (Y \to D).$$

fibers are propositions.

2. For each map $f : X \to D$ and embedding $j : X \hookrightarrow Y$, a choice of an identification $(f|j) \circ j = f$, as illustrated by

$$\begin{tikzcd} X \arrow[rr, hook, "j"] \arrow[dr, "f"'] & & Y \arrow[dl, "f|j"] \\ & D & \end{tikzcd}$$

Some examples

MSCS' 2021

(They need univalence)

1. $D := \mathcal{U}$

(a)
$$\begin{tikzcd} X \arrow[rr, hook, "j"] \arrow[dr, "f"'] & & Y \arrow[dl] \\ & \mathcal{U} & \end{tikzcd}$$
$$(f \mid j)(y) := \prod_{(x,-) : \mathrm{fiber}\ j\ y} f(x).$$
(Right Kan extension.)

(b) Use Σ instead. (Left Kan extension.)

2. The type Ω of propositions. Use $\forall$ or $\exists$.

3. Universes of n-types.

4. Algebras of the lifting monad. We'll come back to this.

Homomorphisms of algebraic injectives

$$
\begin{tikzcd}
X \arrow[rr, hook, "j"] \arrow[dr, "f"'] \arrow[ddr, bend right] & & Y \arrow[dl, "f/j"] \arrow[ddl, bend left] \\
& D \arrow[d, "h"] & \\
& E &
\end{tikzcd}
$$

$$h \circ (f | j) = (h \circ f) | j$$

Pullback naturality

Previous examples are all pullback natural.

$$\begin{tikzcd}
A \arrow[r, hook, "k"] \arrow[d, "g"'] \arrow[dr, phantom, "\lrcorner", very near start] & B \arrow[d, "h"] \arrow[ddl, bend left=60, "(f \circ g) | k"] \\
X \arrow[r, hook, "j"] \arrow[dr, "f"'] & Y \arrow[d, "f/j"] \\
& D
\end{tikzcd}$$

$$\boxed{(f|j) \circ h = (f \circ g) | k}$$

It is essential that the square is a pullback.

Consider the non-pullback square

$$\begin{tikzcd}
0 \arrow[r, hook] \arrow[d] & 1 \arrow[d] \\
1 \arrow[r, hook] & 1
\end{tikzcd}$$

for a counter-example.

The counter-example in detail

$x : 1$

$$\begin{tikzcd}
0 \arrow[rr, hook, "!"] \arrow[d, "!"'] & & 1 \arrow[d, "!"] \arrow[ddl, bend left=60, "?"] \\
1 \arrow[rr, hook, "!"] \arrow[dr, "f"'] & & 1 \arrow[dl, "f"] \\
& D &
\end{tikzcd}$$

$$(f \circ !)|_! (*) = \bigsqcup_{x:0} (\ldots) =: \bot$$

To get a counter-example, just choose

$$f x = (\bot \to \emptyset)$$

So this counter-example works for the injective induced by any algebraic flabby structure.

Associativity

$$\begin{tikzcd} X \arrow[r, hook, "j"] \arrow[dr, "f"'] & Y \arrow[r, hook, "k"] \arrow[d, "f|j"] & Z \arrow[dl] \\ & D & \end{tikzcd}$$

$$f|(k \circ j) = (f|j)|k$$

Examples (MSCS'2021) $D := \mathcal{U}$ with extension given by Π or Σ.

II. Algebraic flabby structure

$$\begin{tikzcd} P \arrow[r, hook] \arrow[dr, "f"'] & 1 \arrow[d, "\sqcup f"] \\ & D \end{tikzcd}$$

$\boxed{\text{Every partial element of } D \text{ can be extended to a total element}}$

N.B. T.F.A.E.

1. The map $P \to 1$ is an embedding.
2. The type P is a proposition.

Trivial fact. Algebraic injective structure is, in particular, algebraic flabby structure.

Homomorphisms

$$\begin{tikzcd}
P \arrow[r, hook] \arrow[dr, "f"'] & 1 \arrow[d, "U_D f"] \arrow[dd, bend left, "U_E(h \circ f)"] \\
& D \arrow[d, "h"] \\
& E
\end{tikzcd}$$

$$\boxed{h \circ U_D f = U_E(h \circ f)}$$

construction

Algebraic flabby → algebraic injective

MSCS'2021

$$\begin{tikzcd} X \arrow[r, hook, "j"] \arrow[dr, "f"'] & Y \arrow[d, "f|j"] \\ & D \end{tikzcd}$$

$$(f|j)(y) := \bigsqcup_{\underbrace{(x,-) : \text{fiber } j\ y}_{P_y}} f\,x$$

Fact (new observation)

This algebraic injective structure is pullback natural. (Not only fiber-natural)

$$\begin{tikzcd} P_y \arrow[r] \arrow[d, "pr_1"'] & 1 \arrow[d, "\bigsqcup (f \circ pr_1)"] \\ X \arrow[r, "f"'] & D \end{tikzcd}$$

Fiberwise extension.

$$\begin{tikzcd} P_y \arrow[r, hook] \arrow[d, "pr_1"'] \arrow[dr, phantom, "\lrcorner"] & 1 \arrow[d, "y"] \arrow[dd, bend left, "\bigsqcup (f \circ pr_1)"] \\ X \arrow[r, hook, "j"] \arrow[dr, "f"'] & Y \arrow[d, "f|j"] \\ & D \end{tikzcd}$$

Get this by flabbiness.

III. The lifting monad

M.H.E & C. Knapp CSL'2017 & TypeTopology 2018

$$\mathcal{L}X := \sum_{P:\Omega} (P \xrightarrow{\varphi} X)$$

The type of partial elements of X.

$$\begin{array}{c} x \\ \downarrow \\ (\mathbb{1}, \lambda -. x) \end{array}$$

$$\begin{tikzcd}
X \arrow[r, "f"] \arrow[d, "\eta_X"'] & Y \arrow[d, "\eta_Y"] \\
\mathcal{L}X \arrow[r, "\mathcal{L}f"] & \mathcal{L}Y
\end{tikzcd}$$

$$(P, \varphi) \longmapsto (P, f \circ \varphi)$$

$$\text{is-defined} : \mathcal{L}X \to \Omega := pr_1$$

$$\text{value} : (l : \mathcal{L}X) \to \text{is-def}\ l \to X := pr_2$$

$$X \xrightarrow{g} \mathcal{L}Y$$

$$\mathcal{L}X \xrightarrow{g^\#} \mathcal{L}Y$$

$$(P, \varphi) \longmapsto \Big(\Big(\sum_{p:P} \text{is-def}(g(\varphi\, p))\Big), \lambda(p,d).\ \text{value}\,(g\,(\varphi\, p))\, d\Big)$$

is a prop, as required.

Monad algebras

1. Structure map

$$\bigsqcup : \mathcal{L} A \to A$$

Extend a partial element to a total element!

So $\mathcal{L}$-algebras give algebraic flabby structure. (MSCS'2021)

2. Unit law

$$\bigsqcup (\lambda(- : \mathbb{1}) . a) = a \quad \text{for every } a : A.$$

Extension "property", as for aflabby types. (extension <u>data</u>!)

3. Associativity law

$$\bigsqcup_{p : P} \bigsqcup_{q : Q\, p} f(p,q) = \bigsqcup_{r : \Sigma_{p : P} Q\, q} f\, r$$

$P : \Omega$

$Q : P \to \Omega$

$f : \Sigma_{p : P} Q\, p \to A$

Previously unaccounted for when discussing injectivity

IV. Putting I – III together

So $\mathcal{Z}$-algebra structure = associative algebraic flabby structure.

Lemma Let $\sqcup$ be the algebraic flabby structure induced by a given algebraic injective structure $|$ that is pullback natural.

Then $\sqcup$ is associative iff $|$ is associative

$\sqcup\sqcup = \sqcup$

$$\begin{tikzcd} \bullet \arrow[r, hook] \arrow[dr] & \bullet \arrow[r, hook] \arrow[d] & \bullet \arrow[dl] \\ & \bullet & \end{tikzcd}$$

Ongoing work. Replace "iff" by "$\simeq$".

(For sets we have this.)

Lemma Let $\mid$ be the algebraic injective structure induced by a given algebraic flabby structure $\sqcup$.

Then $\mid$ is always pullback natural. (We've already discussed this.)

Lemma The round trip $\sqcup \longmapsto \mid \longmapsto \sqcup'$ is always the identity on both extension operators and extension data.

Lemma The round trip $\mid \longmapsto \sqcup \longmapsto \mid'$ is the identity on extension operators iff $\mid$ is pullback natural.

But what about extension data?

Ongoing.

Theorem. Let D be any type.

1. Then

pullback-natural, associative injective structure on D

$\iff$ associative algebraic flabby structure on D

$=$ $\mathcal{X}$-algebra structure on D.

2. If D is a set, then "$\iff$" in (1) becomes an equivalence "$\simeq$".

- what is missing to always have a type equivalence?
    - check that the pullback-naturality data is unchanged by round trips.
    - check that the associativity data is unchanged by round trips.

(ongoing work, perhaps not difficult.)

- But there is still something else missing.

V. Is $\mathcal{L}$ really a monad?

Nobody knows what a monad on types is in HoTT/UF.

People do know what monads on ∞-toposes are, though.

But we don't know how to say that in the language of HoTT/UF.

The problem is how to specify coherence data for the monad laws.