I missed lecture today to go to Dimitris Tsementzis's talk on Finite Inverse Categories.

fic	context
X	$(X : \text{Type})$
$X \to Y$	$(Y : \text{Type})(X : Y \to \text{Type})$
$I {\to_i} A {\rightrightarrows_{c,d}} O$ with $\mathcal{C}(I,O) = \{ci = di\}$	$(O : \text{Type})(A : O \to O \to \text{Type})\left(I : \prod_{x:O} A(x,x) \to \text{Type}\right)$
$I {\to_i} A {\rightrightarrows_{c,d}} O$ with $\mathcal{C}(I,O) = \{ci \neq di\}$	$(O : \text{Type})(A : O \to O \to \text{Type})\left(I : \prod_{x,y:O} A(x,y) \to \text{Type}\right)$
$I {\to_i} A {\rightrightarrows_{c,d}} O$ with $\mathcal{C}(I,O) = \{ci = di \neq e\}$	$(O : \text{Type})(A : O \to O \to \text{Type})\left(I : \prod_{x:O} A(x,x) \to O \to \text{Type}\right)$

Syntax: $\mathbb{T}_{\text{fic}}$

$$\frac{}{\bullet \vdash}, \frac{\Psi \vdash \Gamma}{\Psi, A : \Gamma \vdash}, \frac{\Psi \vdash}{\Psi \vdash \bullet}, \frac{\Psi, A : \Delta, \Psi' : L, \Psi, A : \Delta, \Psi' : \sigma : \Gamma \Rightarrow \Delta}{\Psi, A, \Psi' \vdash \Gamma, X : A\sigma}, \frac{\Psi \vdash \Gamma}{\Psi \vdash \varepsilon_r : \Gamma \Rightarrow \bullet, \Psi \vdash \text{id}_r : \Gamma \Rightarrow \Gamma, \text{etc}\ldots, \equiv}$$

such that $\text{id}_r \circ \sigma \equiv \sigma \equiv \sigma \circ \text{id}_r$, etc...

Theorem: There exists an interpretation of $\mathbb{T}_{\text{fic}}$ such that Soundness:

- $[\![\Psi \vdash]\!] \Leftrightarrow [\![\Psi]\!]$ is a fic
- $[\![\Psi \vdash \Gamma]\!] \Leftrightarrow [\![\Gamma]\!] : [\![\Psi]\!] \to \text{Set} \subset \text{Shape}_0$ (functor)
- $[\![\Psi \vdash \sigma : \Gamma \Rightarrow \Delta]\!] \Leftrightarrow [\![\sigma]\!] : [\![\Gamma]\!] \Rightarrow [\![\Delta]\!]$ (natural transformation)
- Analogous to Lindenbaum-Tarski algebras
- Where was the train going?

Theorem: Every [semantic] fic is derivable in this system: $\forall \text{fic } \mathcal{L}, \exists \Psi \vdash, \mathcal{L} \cong [\![\Psi]\!]$

Interpretation of $\mathbb{T}_{\text{fic}}$ in Coq

- $I(X) = \text{unit} : \text{Type}$
- $I(\Psi, A : \Gamma) = \sum_{X : I(\Psi)} I(\Psi, \Gamma)(X) \to \text{Type}$
- etc...
- $\text{FIC} \simeq \Phi \big| \mathbb{T}_{\text{fic}} \vDash \Phi = \text{Sig} \to_I$ Coq, Agda, CHTT, HoTT, etc... (but not Isabel)

Need

- Σ types
- non-dependent functors
- non-dependent types
- unit types
- associative constructors?

The metamathematics of mathematics formalized by type theory is the mathematics of fics

So, you can do type theory in any programming language! You can even actually store structured data like JSON files as elements, with a hierarchy naturally provided by nestedness. Elements of fics are dags, etc...