Recursive types in System F ?

$$\text{cata} : \forall \alpha.\ (F\alpha \to \alpha) \to \mu F \to \alpha$$
$$\simeq \mu F \to (\forall \alpha.\ (F\alpha \to \alpha) \to \alpha)$$

(Ex: prove the $\simeq$)

Th: In the parametric models of System F, for any functorial $F(\alpha)$
$(\forall \alpha.\ (F\alpha \to \alpha) \to \alpha)$
is the type forming the initial F-algebra
hence (informally)
$\widetilde{\text{cata}} : \mu F \to (\forall \alpha.\ (F\alpha \to \alpha) \to \alpha)$
is an isomorphism

Possible project: see the proof "Recursive types for free" (Wadler)

We write the iso in Haskell
$$F(\underline{\mu} F) \simeq \underline{\mu} F$$
where $\underline{\mu} F \equiv (\forall \alpha.\ (F\alpha \to \alpha) \to \alpha)$

Examples:
$\mathbb{N} = \mu F$ where $FX = 1 + X$
hence we recover the standard encoding
$$\mathbb{N} = \forall \alpha.\ (F\alpha \to \alpha) \to \alpha$$
$$= \forall \alpha.\ ((1 + \alpha) \to \alpha) \to \alpha$$
$$\simeq \forall \alpha.\ ((1 \to \alpha) \times (\alpha \to \alpha)) \to \alpha$$
$$\simeq \forall \alpha.\ \alpha \to (\alpha \to \alpha) \to \alpha$$

$\mathbb{N} = \forall \alpha.\ \alpha \to (\alpha \to \alpha) \to \alpha$

$\ulcorner 0 \urcorner = \Lambda \alpha.\ \lambda z : \alpha.\ \lambda s : \alpha \to \alpha.\ z$

$\ulcorner 1 \urcorner = \quad " \quad " \quad " \quad . \ s\ z$

$\ulcorner 2 \urcorner = \quad " \quad " \quad " \quad . \ s\ (s\ z)$

$\ldots$

$succ : \mathbb{N} \to \mathbb{N}$

$succ = \lambda n : \mathbb{N}.\ \Lambda \alpha.\ \lambda z : \alpha.\ \lambda s : \alpha \to \alpha.$

$\qquad s\ (n\ s\ z)$

$iter : \mathbb{N} \to \forall \alpha.\ \alpha \to (\alpha \to \alpha) \to \alpha$

$iter\ n = n$

$List\ \mathbb{N} = \mu F$ where $F = 1 + \mathbb{N} \times X$

$List\ \mathbb{N} = \forall \alpha.\ (F\alpha \to \alpha) \to \alpha$

$= \forall \alpha.\ ((1 + \mathbb{N} \times \alpha) \to \alpha) \to \alpha$

$\simeq \forall \alpha.\ \alpha \to ((\mathbb{N} \times \alpha) \to \alpha) \to \alpha$

$\simeq \forall \alpha.\ \alpha \to (\mathbb{N} \to \alpha \to \alpha) \to \alpha$

is the encoding for lists

$nil : List\ \mathbb{N}$

$nil = \Lambda \alpha.\ \lambda n : \alpha.\ \lambda c : \mathbb{N} \to \alpha \to \alpha.\ n$

$cons : \mathbb{N} \to List\ \mathbb{N} \to List\ \mathbb{N}$

$cons = \lambda n : \mathbb{N}.\ \lambda l : List\ \mathbb{N}.$

$\qquad \Lambda \alpha.\ \lambda n : \alpha.\ \lambda c : \mathbb{N} \to \alpha \to \alpha.$

$\qquad\quad c\ n\ (l\ \alpha\ n\ c)$

We experiment with this encoding in Haskell

Note that, if we allow any recursion at the type level, not only we get an inconsistent logic ($\tau \simeq \neg\tau$)

↑ (contravariant recursion!)

but we also get a fixed point combinator!

$fix : \forall \alpha.\ (\alpha \to \alpha) \to \alpha$

This can be constructed from the (contravariantly) recursive type

$R\,\alpha \simeq (R\,\alpha \to \alpha)$

We construct it in Haskell

Note: $R\,\alpha$ would be μF

where $F\,X = X \to \alpha$

is a contravariant functor,

i.e. if $g : A \to B$

then $F\,g : F\,B \to F\,A$

$h \longmapsto g ; h$

$F : \mathcal{C}^{op} \to \mathcal{C}$

Crafting models for recursive types requires a category with "enough" initial algebras

Note that for Haskell, we would also need an initial algebra for

$$FX = A \to X$$
$$FX = X \to A$$
$$FX = X \to X$$

which is problematic! Some of these are, intuitively, not even functors!!

In domain-based categories, the usual solution is to restrict the category so to ensure that everything is functorial (to the category of embedding-projections)

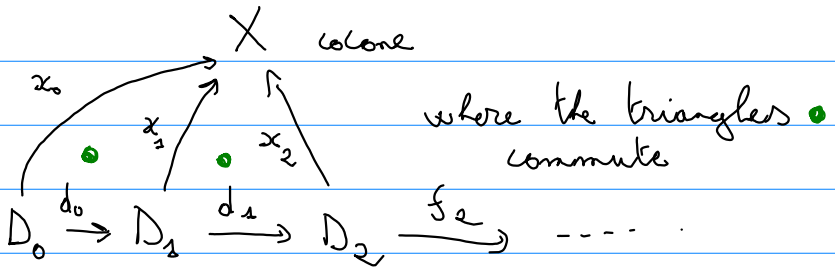
One manages to lift all the usual Kleene fixed point machinery to the category.

ω-CPO	$\rightsquigarrow$	$\mathcal{C}$ ω-cocomplete
$\perp$	$\rightsquigarrow$	0 initial
f continuous	$\rightsquigarrow$	F ω-cocontinuous functor
upper bound	$\rightsquigarrow$	cocone
sup	$\rightsquigarrow$	colimit

Very roughly

$\bot \sqsubseteq f(\bot) \sqsubseteq f^2(\bot) \cdots$

$0 \xrightarrow{!} F0 \xrightarrow{F!} F^2 0 \xrightarrow{F^2!} \cdots$



where the triangles commute

Colimit = cocone initial

One can then prove that the colimit

$X \simeq (\varinjlim_{n \geq 0} F^n 0)$

is the initial F-algebra

(hence $FX \simeq X$)