Paul Taylor

Well founded coalgebras.

1995-6

These are slides for various conference talks and seminars that I gave in 1995-6 on such titles as "Towards a unified treatment of induction and recursion" and "The General Recursion Theorem". Unfortunately it's not possible to reconstruct the exact set of slides used for particular talks, so they have been merged into one sequence.

7 Aug 1995. "Category Theory" (joint with CTCS, Cambridge)
26 Aug 1996. "Gödel '96" (Brno, Czech Rep.)
18 Oct 1996 Seminar at Warwick Univ. (UK)

Towards a Unified Treatment of Induction

self calling programs | infinite descent
structural induction | Peano induction
termination | course of values
trees (loop-free graphs) | minimal elements
loop variants | König's lemma
well founded relations | $\in$-induction
term algebras - parsing, unification, wffs
initial T-algebras | $\Pi X.\,(T(X)\to X)\to X$
left adjoints (univ. props.) | limits, ends
closures | sets & ordinals
induction vs. recursion | Crole's FIX objects
rank of a functor | dilators
Scott induction | fixed points in SDT
techniques for proving strong normalisation
co-induction co-recursion
Freyd's algebraically compact categories

Structural Recursion for Free Theories

Example: $\mathbb{N}$ is the free algebra for the algebraic theory with one constant (nullary operation symbol) and one unary operation symbol, with no equations.

Let Ω be a set of operation-symols
 $ar(r)$ arity of $r \in \Omega$ (not necessarily finite)
An algebra for (Ω, ar) is a set A with a multiplication table $r_A : A^{ar(r)} \to A$ for each $r \in \Omega$.

More concisely,

$$\begin{array}{ccc} A^{ar(r)} & \hookrightarrow & \coprod_{r\in\Omega} A^{ar(r)} \stackrel{\text{def}}{=} TA \\ & {\scriptstyle r_A}\searrow \quad \swarrow{\scriptstyle ev_A} & \\ & A & \end{array}$$

Homomorphism

$$\begin{array}{ccc} \coprod_{r\in\Omega} A^{ar(r)} & \xrightarrow{\ \coprod f^{ar(r)}\ } & \coprod B^{ar(r)} \\ \parallel & & \parallel \\ TA & \xrightarrow{\ Tf\ } & TB \\ {\scriptstyle ev_A}\downarrow & & \downarrow{\scriptstyle ev_B} \\ A & \xrightarrow{\ f\ } & B \end{array}$$

Categorical definition of term algebra:
 the initial object of this category.

A Free Theory with no Initial Algebra

Let $T=P$ be the covariant powerset functor

$PX = \{U : U \subset X\}$ $\quad$ $Pf(U) = f_!(U) \equiv \{y : \exists x \in U.\, y = fx\}$

Any complete semilattice is an algebra for P with $ev_A = \bigwedge$ (conjunction).
(Not all algebras are of this form.

What is a P-coalgebra?

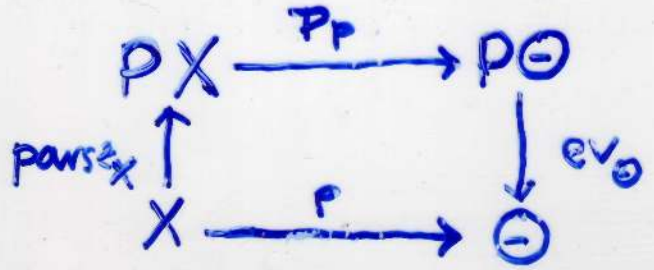
$$parse_X : X \longrightarrow PX$$

write $x < a$ if $x \in parse_X(a)$
i.e. $parse_X(a) = \{x : x < a\}$

If $X \hookrightarrow PX$ we say $<$ (or $parse_X$) is extensional

$$\frac{\{x : x < a\} = \{x : x < b\}}{a = b}$$

The Recursion Scheme (Osius)

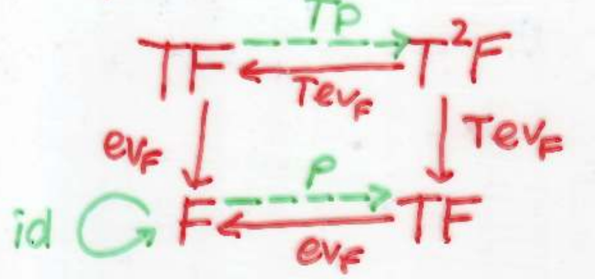


$$p(a) = ev_\Theta(\{p(x) : x < a\})$$

Parsing (Lambek) &
Induction (Lehmann-Smyth)

For any algebra $\begin{array}{c} TA \\ \downarrow{\scriptstyle ev_A} \\ A \end{array}$, $\begin{array}{c} T^2A \\ \downarrow{\scriptstyle Tev_A} \\ TA \end{array}$ is an algebra and $\begin{array}{ccc} T^2A & \xrightarrow{Tev_A} & TA \\ {\scriptstyle Tev_A}\downarrow & & \downarrow{\scriptstyle ev_A} \\ TA & \xrightarrow{ev_A} & A \end{array}$ is a homomorphism

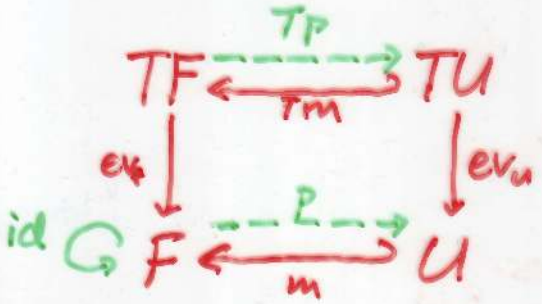
Let $ev_F : TF \to F$ be the initial algebra
Then ev_F has an inverse (ev_F^{-1} = $parse_F$).



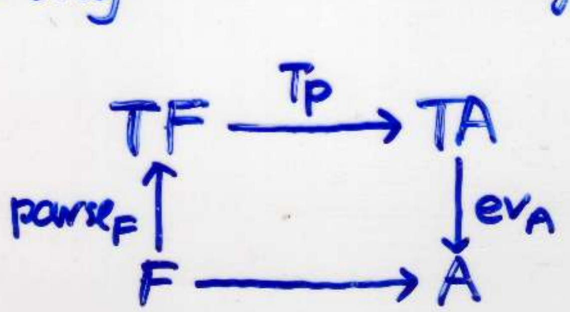
$p ; ev_F = id$

$Tp ; Tev_F = ev_F ; p$

Moreover if $U \hookrightarrow F$ is a subalgebra then $U \cong F$.



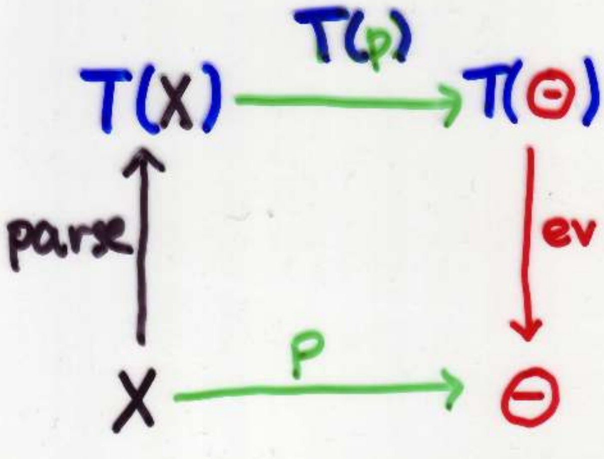
Often (usually) $parse_F$ is a better way of describing the term algebra F than ev_F is.



RECURSIVE PARADIGM

Recursive program for function $p(x)$

1. does case analysis on argument x to give zero or more sub-arguments $y_1, y_2, \ldots y_k$ (max k = arity)
2. calls itself on these sub-arguments in parallel to give sub-results $p(y_1), p(y_2), \ldots, p(y_k)$
3. calculates $p(x)$ from x and $p(y_i)$



Examples: $T(X)$ may be

- $X+N$ — while program
- X^2+X+A — one binary case, one unary and a base case
- $P(X)$ — covariant powerset – ideas from elementary set theory.

The Recursive Paradigm

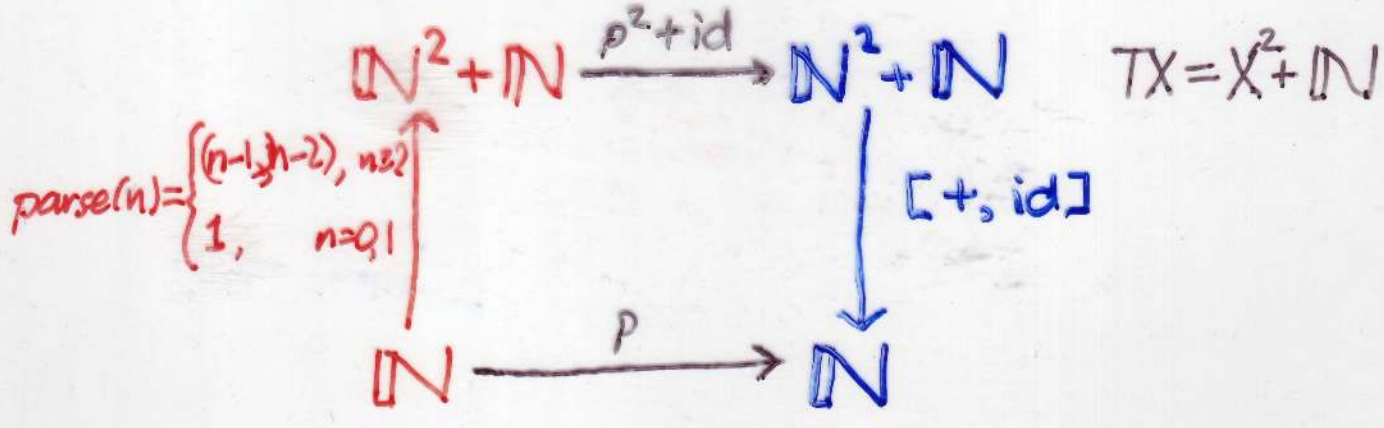
A recursive program which defines a (partial) function $p: X \to \Theta$ of one argument as follows:

① "parse" argument x into sub-arguments $\vec{x}$ (this always involves case analysis)

② call $p(\vec{x})$ in parallel

③ "evaluate" result $p(a)$ using $p(\vec{x})$ (and a)

$$p(n) = \begin{cases} \text{if } n \geq 2 & p(n-1) + p(n-2) \\ \text{if } n = 1 & 1 \\ \text{if } n = 0 & 1 \end{cases}$$



Although the function p which this program defines is unary, this is a binary recursion because there are (up to) two sub-arguments at each level.

Induction and Recursion for the Natural Numbers

$$0 \in \mathbb{N} \qquad\qquad 0 \neq sn$$

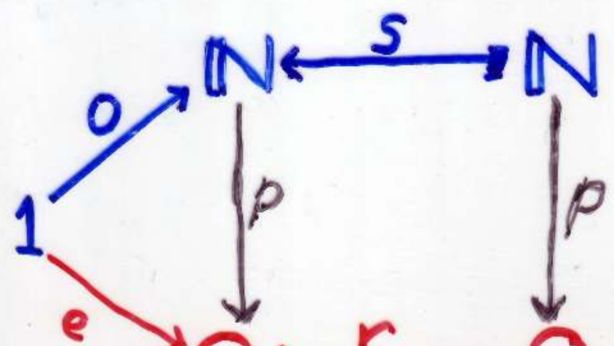
$$\frac{n \in \mathbb{N}}{sn \in \mathbb{N}} \qquad\qquad \frac{sn = sm}{n = m}$$

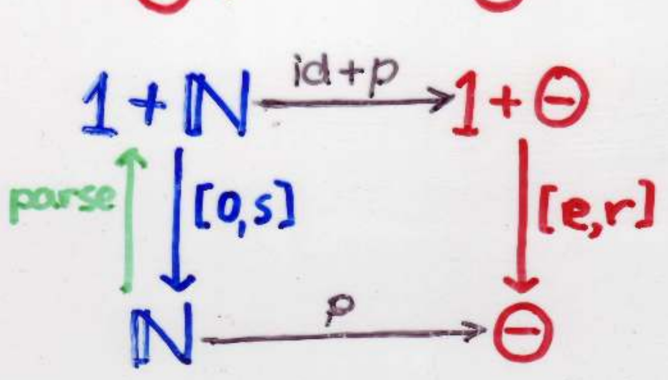
$$\forall \varphi. \frac{\forall n. (\forall m. m < n \Rightarrow \varphi[m]) \Rightarrow \varphi[n]}{\forall n. \varphi[n]}$$

$$e \in \Theta \qquad\qquad r : \Theta \to \Theta$$

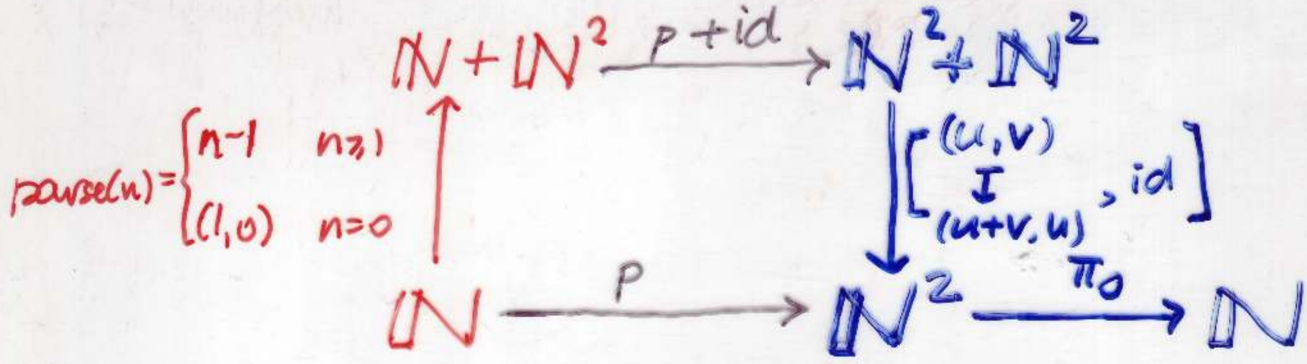
$$p(0) = e$$

$$p(sn) = r(p(n))$$

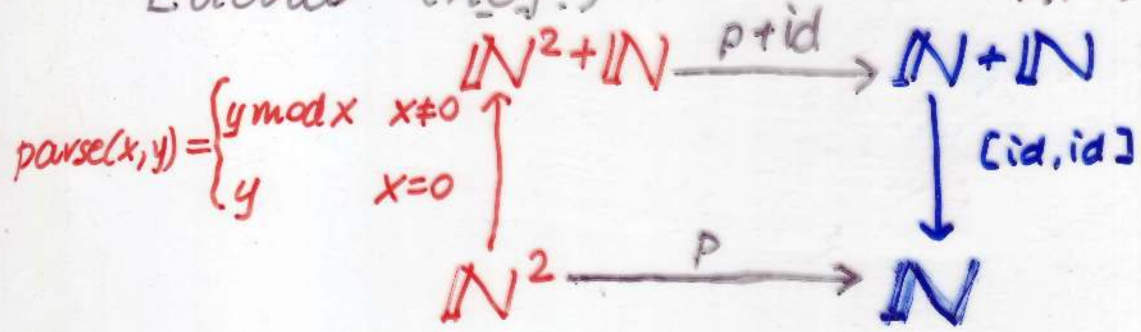




Quick Fibonacci $TX = X + \mathbb{N}^2$

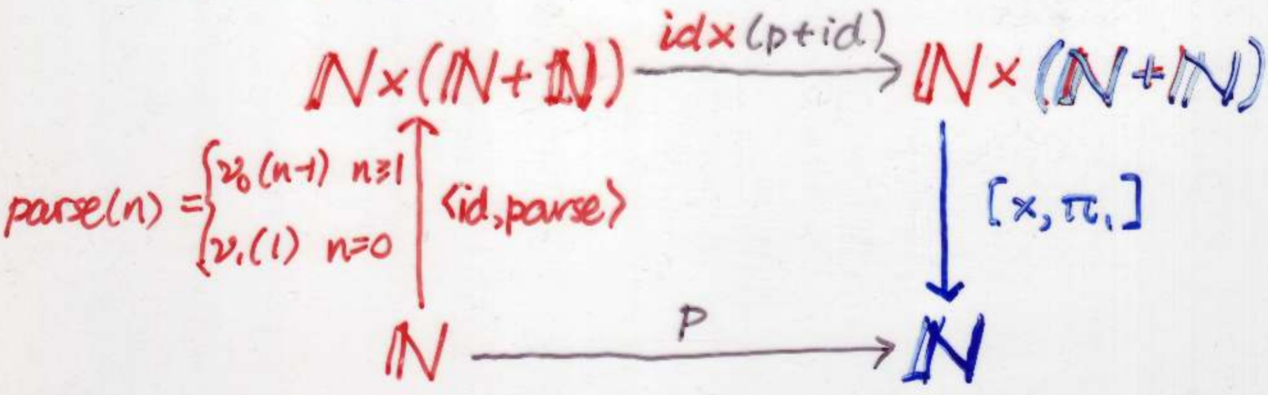


Euclid (h.c.f.) $TX = X + \mathbb{N}$



unary recursion defining a binary function

Factorial

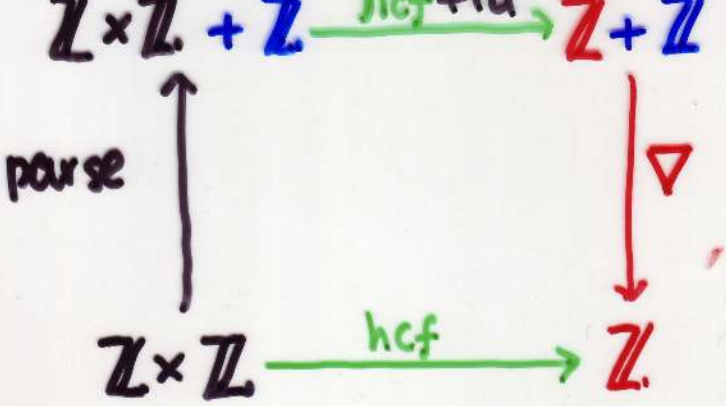


$TX = X + \mathbb{N}$ recursion with argument as parameter

EUCLIDEAN ALGORITHM

```
put x=a;
put y=b;                    ψ[x,y]
while x≠0 do                ψ[x,y] ∧ x≠0
    put z=y mod x;          |z|<|x| ∧ ∃n.z+nx=y
                              ∧ ψ[z,x]
    y:=x;                   ψ[z,y]
    x:=z;                   ψ[x,y]
    discard z;
od;                         x=0 ∧ ψ[x,y]
discard a,b,x
```

$$\psi[u,v] \equiv \forall m.\ m|a \wedge m|b \Leftrightarrow m|u \wedge m|v$$

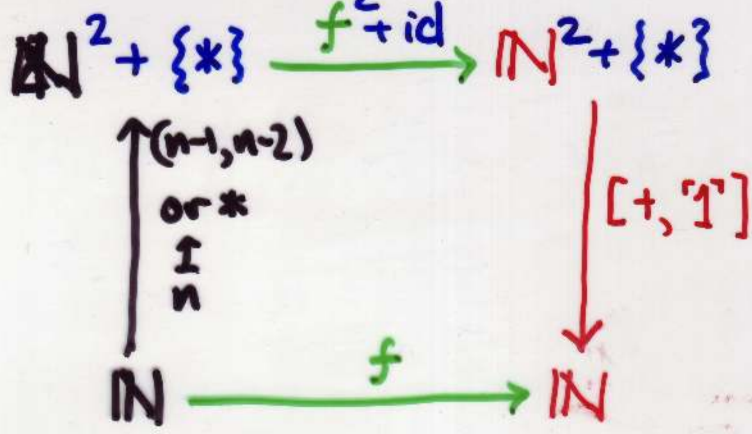


$$\text{parse}\,(x,y) = \begin{cases} (y \bmod x,\ x) & \text{if } x\neq 0 \\ y & \text{if } x=0 \end{cases}$$

FIBONACCI.

binary recursion
(exponential time)

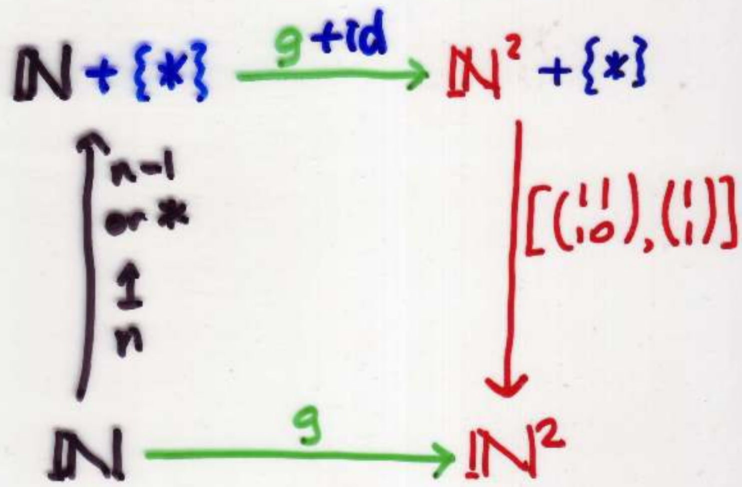
$$f(n) = \begin{cases} f(n-1) + f(n-2) & (n \geq 2) \\ 1 & (n = 0 \text{ or } 1) \end{cases}$$



Unary recursion
(linear time)

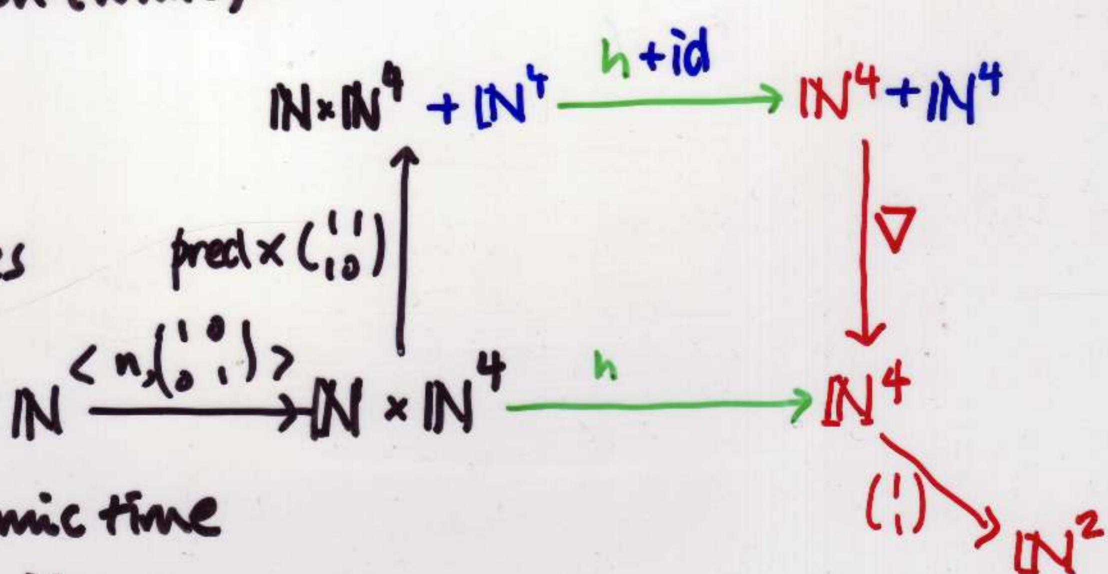
$$g(n) = \begin{pmatrix} f(n+1) \\ f(n) \end{pmatrix}$$

$$= \begin{cases} \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} g(n-1) & n \geq 1 \\ \begin{pmatrix} 1 \\ 1 \end{pmatrix} & n = 0 \end{cases}$$



tail recursion (while)

$M \cong \mathbb{N}^4$
(2×2) matrices

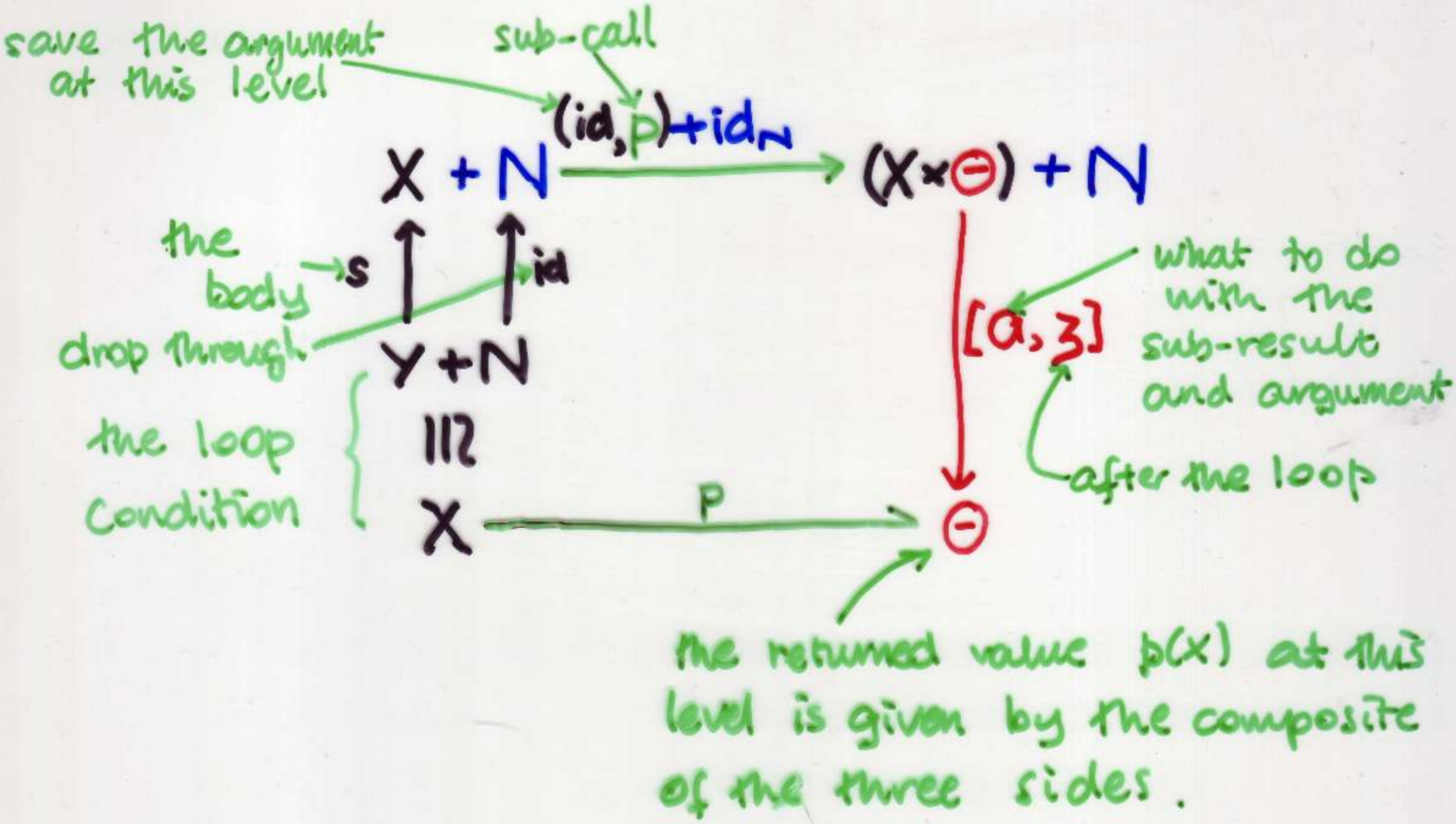


→ logarithmic time algorithm.

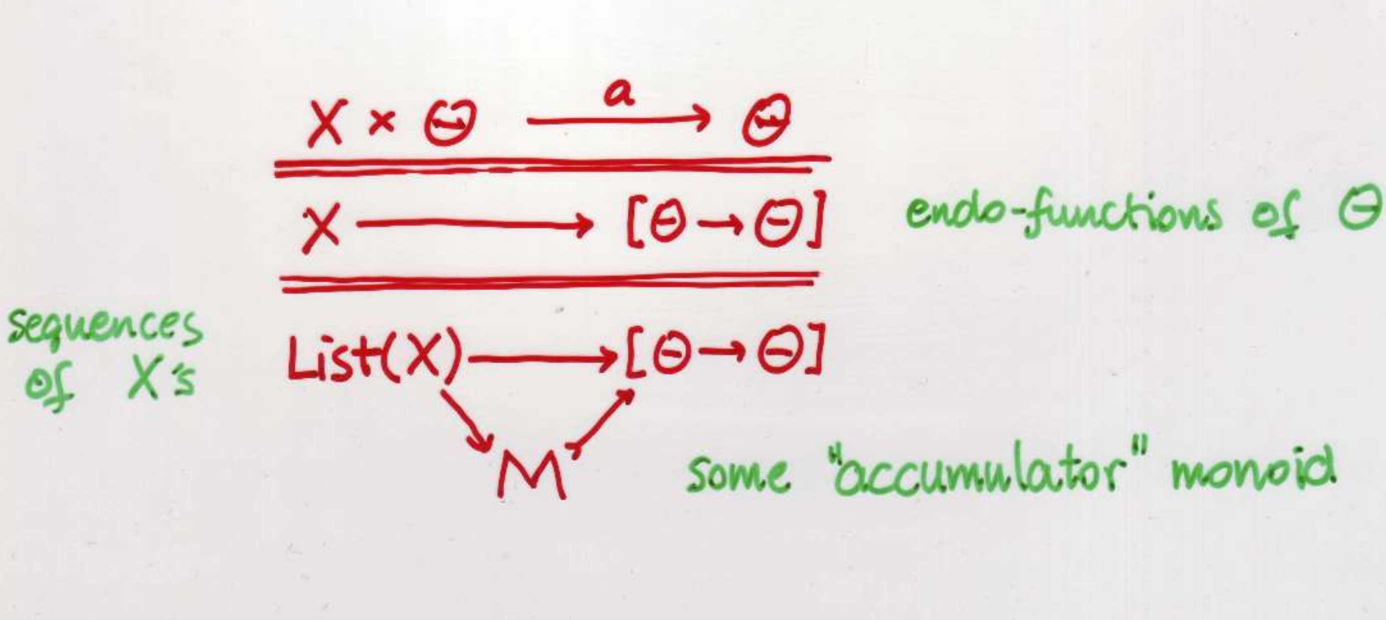
UNARY RECURSION

At most one sub-call at each level

so the functor is $(-)+N$

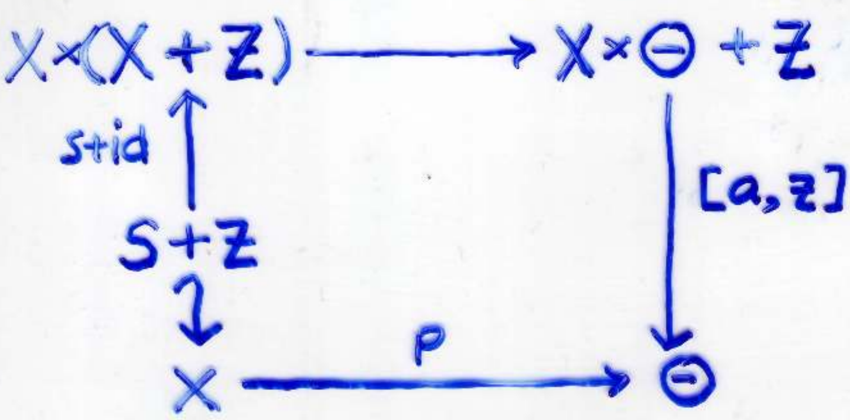


$$p(x) = \begin{cases} z(x) & \text{if } x \in N \\ a(x, s(x)) & \text{if } x \in Y \end{cases}$$



Tail Recursion (= while)

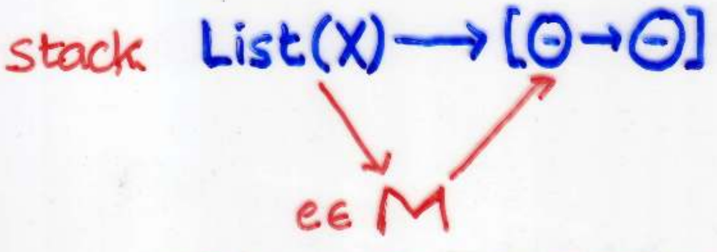
Any unary recursion problem may easily be put in the form



$S \xrightarrow{s} X$ beginning of loop body

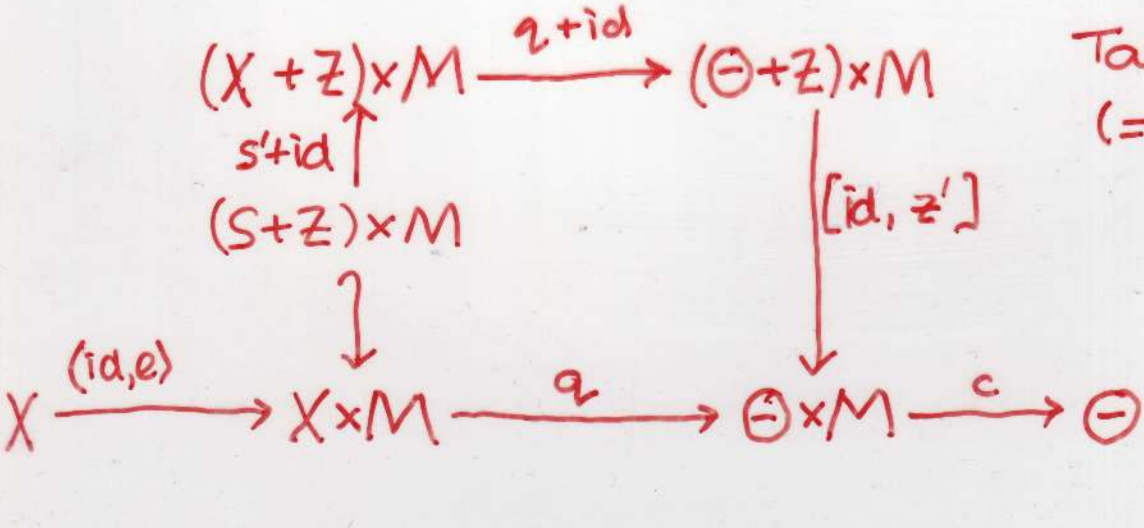
$Z \xrightarrow{z} \Theta$ continuation after the loop

$$\frac{\dfrac{X\times\Theta \xrightarrow{a} \Theta}{X \longrightarrow [\Theta\to\Theta]}}{}$$



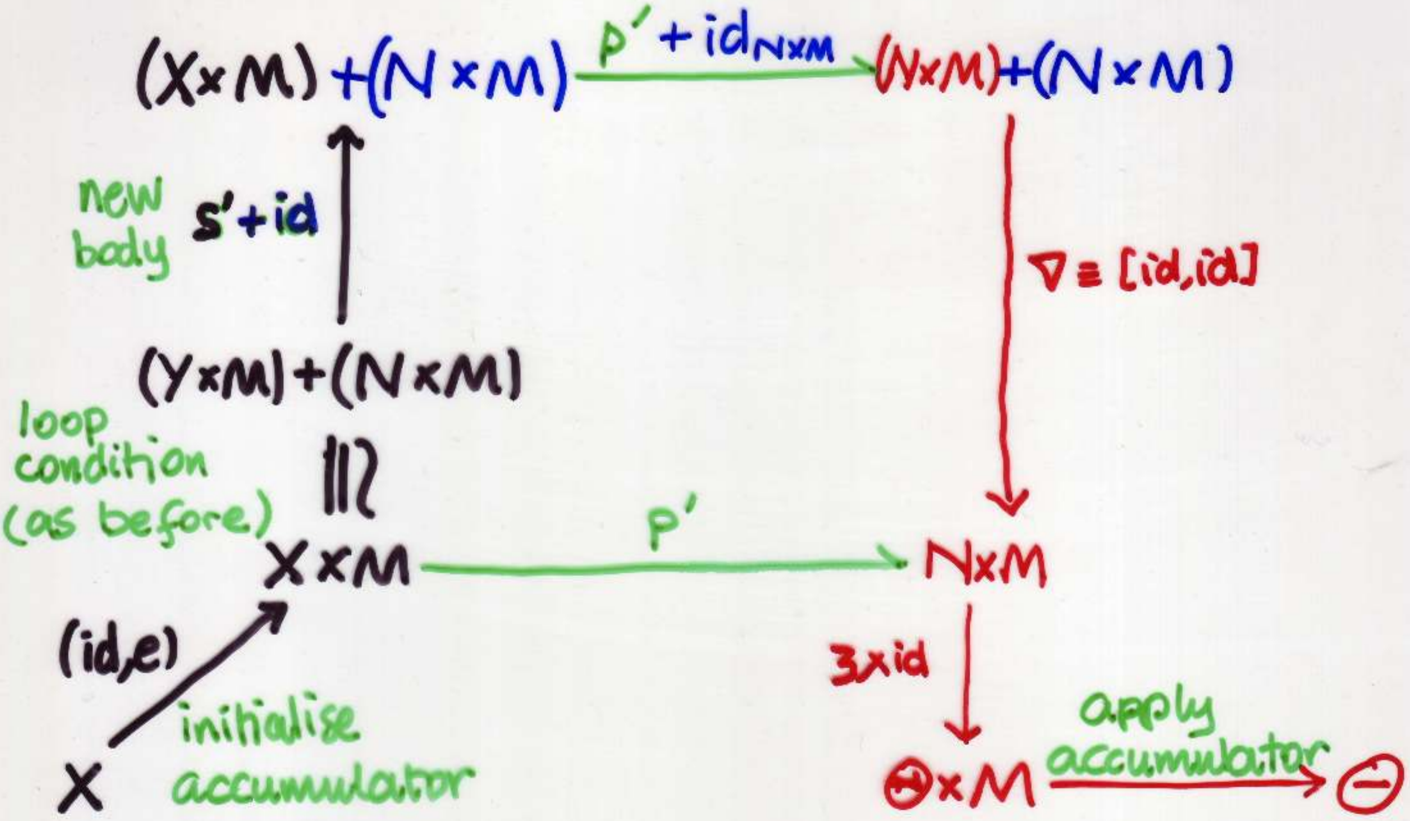
pending code to complete the loop

$$X \xrightarrow{b} M \qquad M\times\Theta \xrightarrow{c} \Theta$$



Tail recursive (= while)

UNARY RECURSION AS TAIL RECURSION WITH ACCUMULATOR



$$s'(x, m) = (s(x),\ s(x) *_M m)$$

original loop body — "multiplication" in M

APPLICATIONS:

(1) list operations (fold, map, reverse...)

(2) arithmetic (fibonacci, factorial...)

COMPILER OPTIMISATIONS

Arithmetic:

- recognise that result depends linearly on the subresult, maybe with constant term and arbitrary dependency on argument monoid of (n+1)×(n+1) matrices with matrix multiplication
- more complicated monoid of polynomials with substitution – not worth the trouble.

Lists

- pre- & ap-pending strings, maybe depending on the argument monoid of lists: essentially the same as using a stack, but more efficient by some (small) factor

Multiary recursion

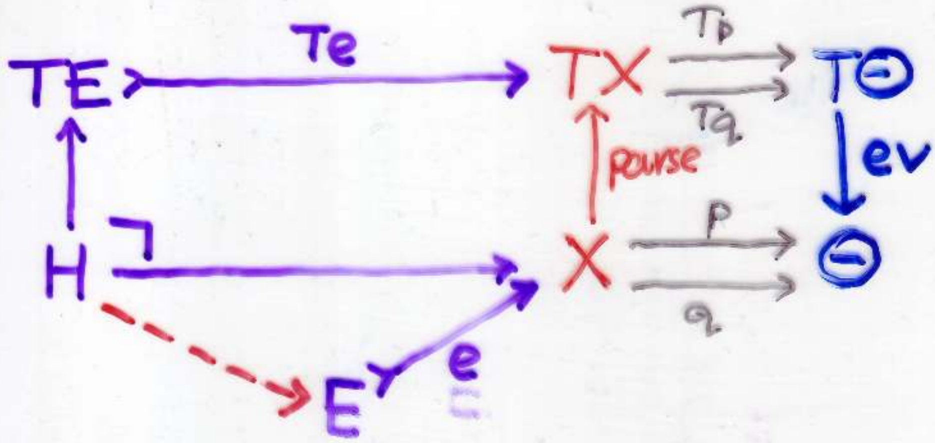
replace monoid by category, use a concrete representation of the Lawvere theory: unlikely to lead to any saving, but the theory exists.

Reducing arity (eg with Fibonacci)

don't know, but the categorical recursion scheme looks like a good notation.

The Induction Scheme

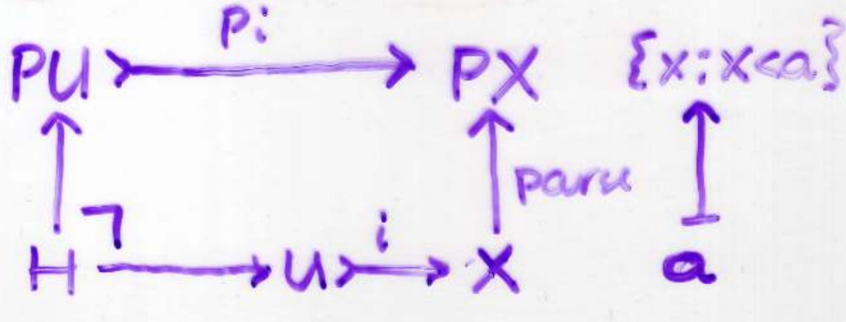
How do we know that the recursion scheme __has__ a __unique__ solution?



want to conclude that E = X.

Return to set theory (powerset)

$U = \{x : \varphi[x]\}$



$H = \{(V,a) : V \in PU,\ a \in X,\ \mathrm{parse}(a) = Pi(V)\}$

$V \subset U \subset X \quad \{x : x < a\} = V$

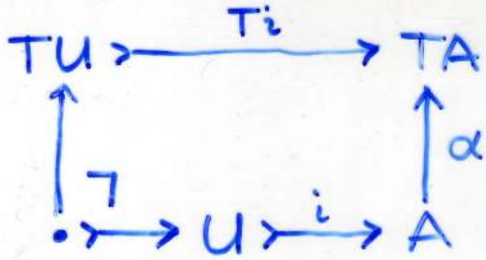
$H \cong \{a : \forall x.\, x < a \Rightarrow \varphi[x]\}$

$$\frac{H \subset U}{U = X} \qquad \frac{\forall a.\, (\forall x.\, x < a \Rightarrow \varphi[x]) \Rightarrow \varphi[a]}{\forall a.\, \varphi[a]}$$

Well foundedness w.r.t. a functor

A **coalgebra** $A \xrightarrow{\alpha} TA$ is **well founded**

if in any diagram

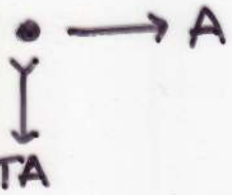


the inclusion $i: U \rightarrowtail A$ is an isomorphism.

With $T =$ covariant powerset this says

$$\frac{\forall x.\, [\forall y.\, y < x \Rightarrow \varphi(y)\,] \Rightarrow \varphi(x)}{\forall x.\, \varphi(x)}$$

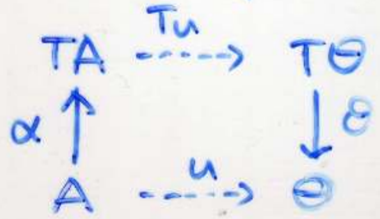
where $U = \{x : \varphi(x)\}$ and $\alpha(x) = \{y : y < x\}$.



is a minimal equationally free partial algebra (mefpa)

iff its $A \xrightarrow{\cong} A$ with $\alpha \downarrow TA$ a well founded.

For any algebra $T\Theta \xrightarrow{\vartheta} \Theta$ there's a unique homomorphism



A **recipe** is a cone over the diagram T-Alg → Set

≡ a generalised element of $\int^X (TX \to X) \to X$ (wedge)

$A \xrightarrow{\alpha} TA$ is a recipe morphism iff it is well founded

Requirements

category $\mathcal{C}$ (Set)

functor $T: \mathcal{C} \to \mathcal{C}$

(covariant powerset

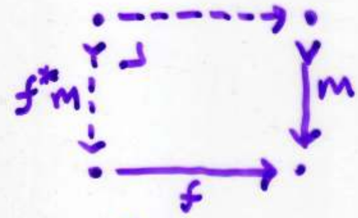
$\sum_{r \in R} (-)^{ar(r)} / G_r$

arity ↗ ↑ commutativity group)

class of "monos" $\mathcal{M}$

$\mathcal{M}$ predicates to test well foundedness

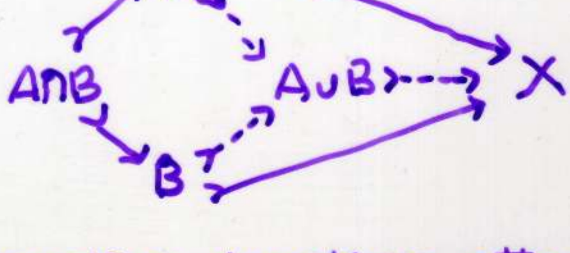
+ supports of partial maps



$f_! \equiv \exists_f \dashv f^* \dashv f_* \equiv \forall_f$

("Mostowski theorem") initial segments

T must preserve diagrams of this kind



pasting together attempts

"Transitive sets" for a functor

with T = covariant powerset

$A \xrightarrow{\alpha} TA$ binary relation

$$\begin{array}{ccccc} TU & & \xrightarrow{Ti} & & TA \\ \uparrow & & & & \uparrow \alpha \\ \bullet & \xrightarrow{?} & U & \xrightarrow{i} & A \end{array} \Rightarrow i \cong \quad \text{well founded}$$

$A \rightarrowtail^{\alpha} TA$ extensional

"transitive set"

ensemble (as I call them)

$$\begin{array}{ccc} TB & \xrightarrow{Ti} & TA \\ \beta \uparrow & & \uparrow \alpha \\ B & \xrightarrow{i} & A \end{array}$$

initial segment

$$\begin{array}{ccc} TB & \xrightarrow{Tf} & TA \\ \beta \uparrow & \subseteq & \uparrow \alpha \\ B & \xrightarrow{f} & A \end{array}$$

strictly monotone function

$b' < b \Rightarrow fb' < fb$

$$\begin{array}{ccc} TA & \xrightarrow{Tf} & T\Theta \\ \alpha \uparrow & & \downarrow r \\ A & \xrightarrow{f} & \Theta \end{array}$$

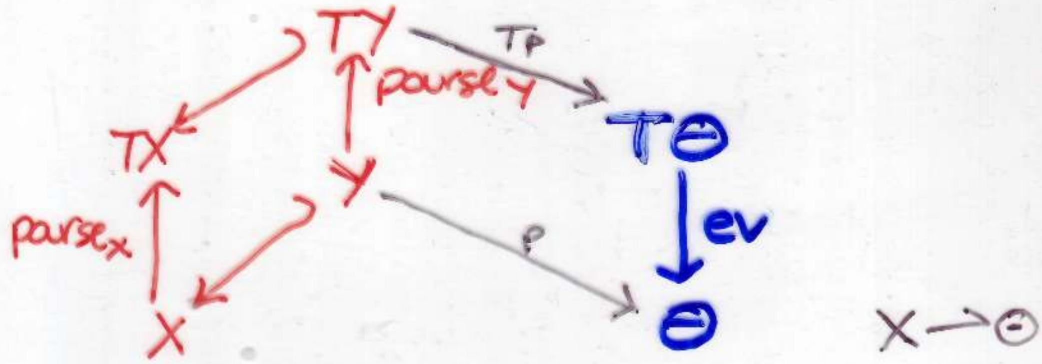
recursion

$f(a) = r(\{ f(a') : a' < a \})$

General Recursion Theorem

Induction ⇒ Recursion

Idea: attempt (partial solution)

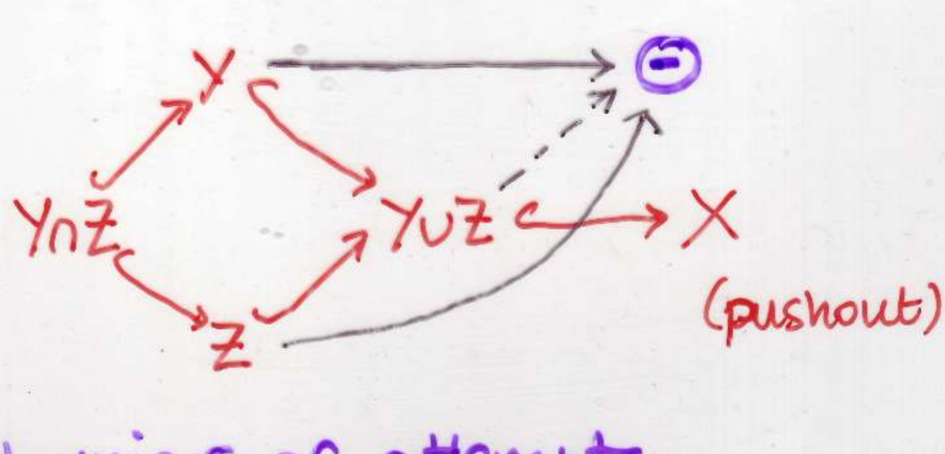


First attempt: $Y = \emptyset$ (strict initial object)

Apply T (preserves monos & inverse images)

Restrict to subcoalgebras (inverse image preserves unions)

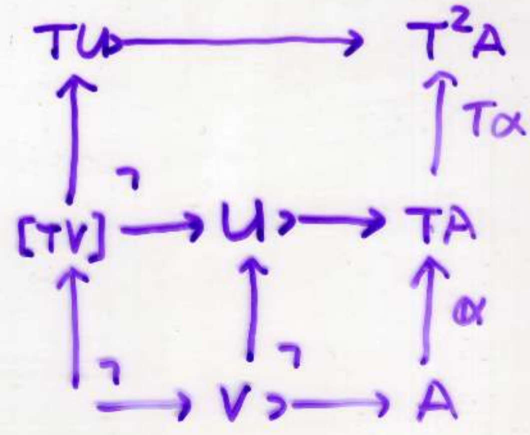
Pasting two attempts



Directed unions of attempts

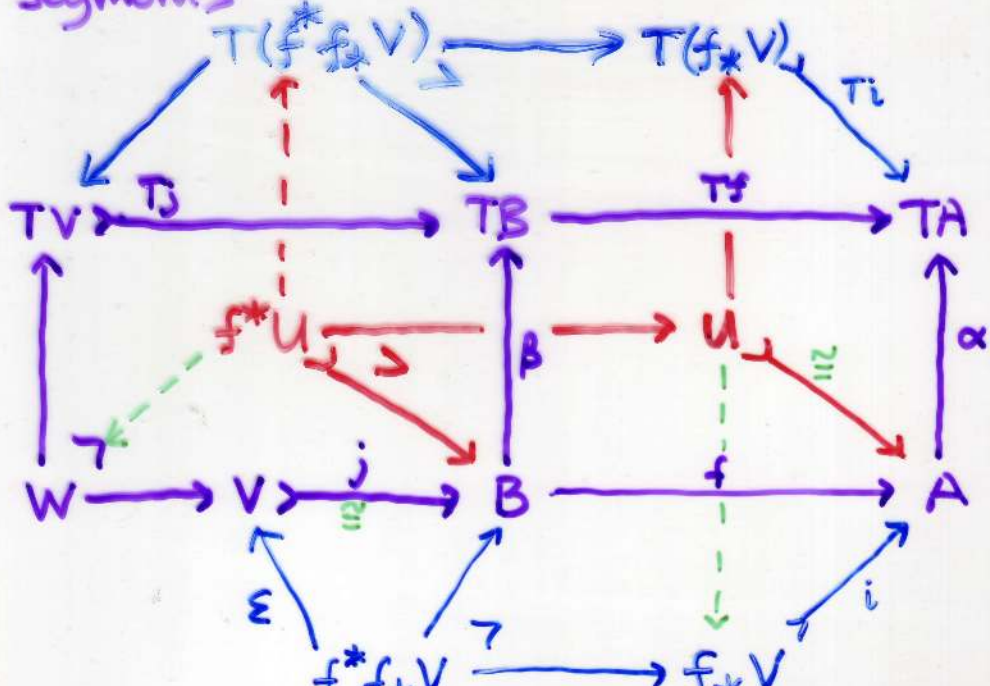
General Recursion Theorem
(constructions)

A wf $\Rightarrow$ TA wf

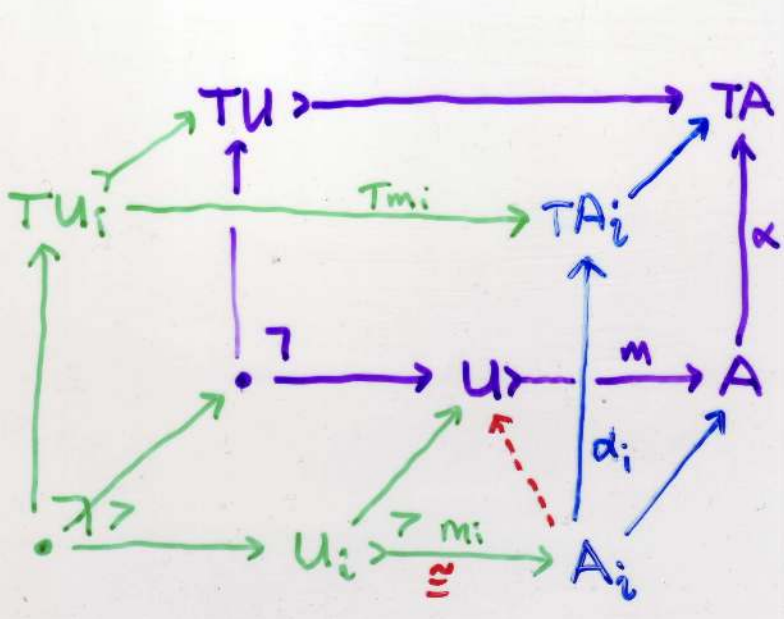


initial segments

A wf $\Rightarrow$ B wf

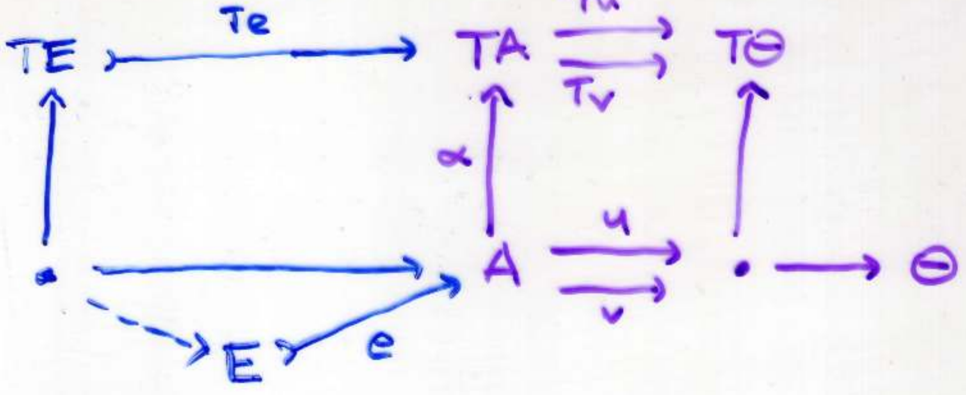


colimits



General Recursion Theorem

parallel attempts agree

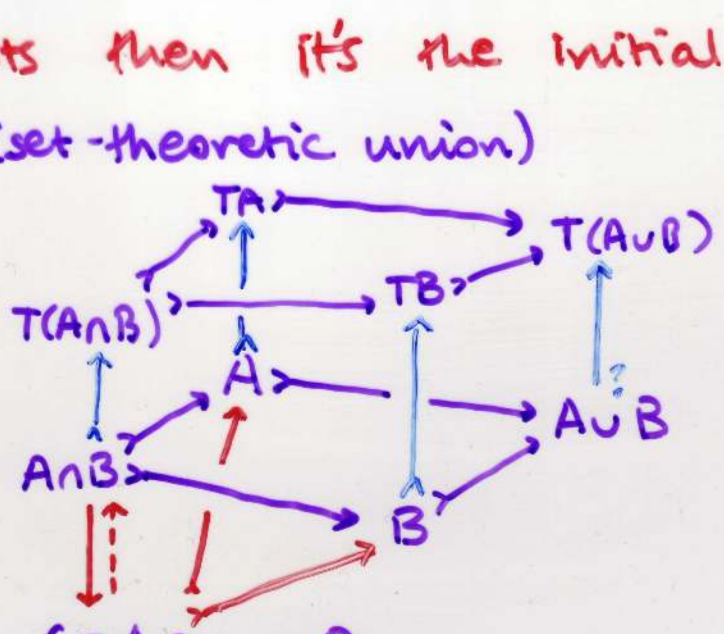


- support of an attempt is an initial segment
- the (partial) attempts $A \to \Theta$ form a directed set
  (uses pushouts of monos)
- maximal attempt is a homomorphism.

$\therefore$ $\exists!$ homomorphism $A \to \Theta$
(the general recursion theorem)

$\therefore$ if colimit of mefpas (terminal w.f. coalg) exists then it's the initial algebra.

- pushout (set-theoretic union)



The "von Neumann Hierarchy" of a functor

empty set $\qquad \emptyset \rightarrowtail T\emptyset$

powerset $\qquad A \xrightarrow{\alpha} TA \quad \Rightarrow \quad TA \xrightarrow{T\alpha} T^2A$

$$A \overset{\alpha}{\rightarrowtail} TA \quad \Rightarrow \quad TA \overset{T\alpha}{\rightarrowtail} T^2A$$

filtered colimit
(directed union)

$$(\operatorname{colim}^{\uparrow}_{i\in I} A_i) \to T(\operatorname{colim}^{\uparrow}_{i\in I} A_i)$$

$$\bigcup^{\uparrow}_{i\in I} A_i \rightarrowtail T(\bigcup^{\uparrow}_{i\in I} A_i)$$

initial segment

$$\begin{array}{ccc} TB & \xrightarrow{Tu} & TA \\ {\scriptstyle\beta}\uparrow & & \uparrow{\scriptstyle\alpha} \\ B & \xrightarrow{u} & A \\ & & \text{wf} \end{array} \quad \Rightarrow \quad B \text{ well founded}$$

$$\begin{array}{ccc} TB & \overset{Tu}{\rightarrowtail} & TA \\ {\scriptstyle\beta}\uparrow & & \uparrow{\scriptstyle\alpha} \\ B & \overset{u}{\rightarrowtail} & A \end{array}$$

intersection

$$\begin{array}{ccc} B & & A \\ & \nwarrow \quad \nearrow & \\ & A\cap B & \end{array}$$

as the greatest attempt $A \to B$ or vice versa.

binary union
"set theoretic"

$$\begin{array}{ccc} & A\cup B & \\ \nearrow & & \nwarrow \\ B & & A \\ \nwarrow & & \nearrow \\ & A\cap B & \end{array}$$

pushout

What if T is a monad?

If $\alpha : A \cong TA$ then

$m \equiv T\alpha ; \mu_A ; \alpha^{-1} : TA \to A$ is the structure of an algebra for any monad (T, η, μ)

Crole-Pitts FIX objects:

$\alpha : A \cong TA$ is free algebra

$$+ \quad \rightarrowtail A \underset{\eta_A}{\overset{\alpha}{\rightrightarrows}} TA$$ is a singleton (the fixed point)

call $\eta_A ; \alpha^{-1}$ the successor

In fact

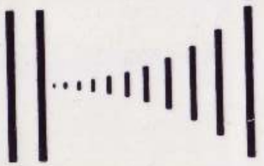
free T-algebra $\cong$ free $(T\eta\mu)$-algebra with successor.

$1 =$

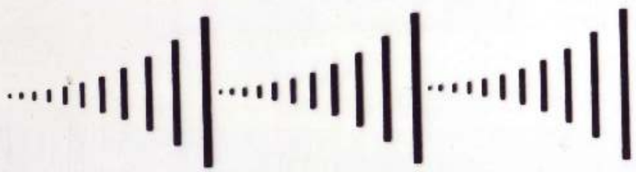
$2 =$ 

$3 =$ 

$\omega =$ 

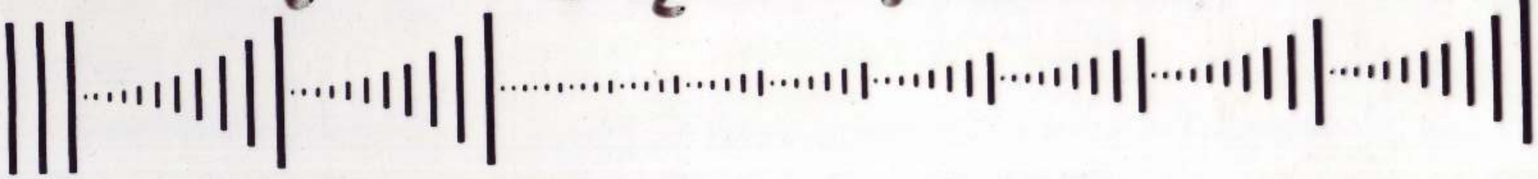
$\omega + 2 =$ 

$\omega 2 =$ 

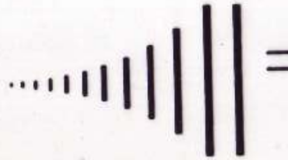
$\omega 3 =$ 

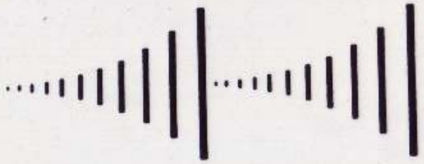
$\omega^2 =$ 

\ordinal{\omega^2 + \omega2 + 3}

$\omega^2 + \omega 2 + 3 =$ 

$\omega + 1 =$ 

$1 + \omega =$ 

$\omega 2 =$ 

INTUITIONISTIC ORDINALS & DIRECTEDNESS

$0 = \emptyset$

$1 = \{\emptyset\} = P(\emptyset)$

$2 = P(\{\emptyset\}) = \Omega$

new symbol: $\mathbf{2}$ two-cum-omega

$3 =$ lower subsets of $\mathbf{2}$

$$\alpha \in 3 \Leftrightarrow \begin{cases} \alpha \subset \mathbf{2} \text{ such that} \\ \forall p \in \mathbf{2}.\ p \in \alpha \to \bot \in \alpha \end{cases}$$

Such α need not be directed

so in $f^{\alpha}(\bot) \stackrel{\text{def}}{=} \bigvee \{ f(f^{\beta}(\bot)) : \beta \in \alpha \}$

the join is not directed

So cannot be formed in a cpo

FIRST DEFINITION OF ORDINALS DOESN'T WORK

How do we define (hereditarily) directed ordinals?

[B.T.W.: "directed" means in the binary sense

$\forall u, v\ \exists w.\ u, v \leq w$

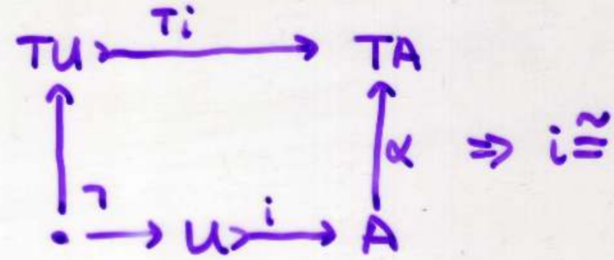
only, so $\emptyset$ is directed. Better: "semidirected"]

α^{+} = set of all lower directed subsets of α

[in SDT any $\alpha \to \Sigma$ automatically preserves $\wedge, \vee$]

The last word on Ordinals

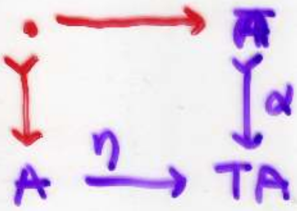
well founded



extensional

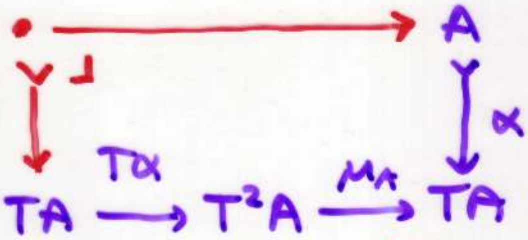
$$A \overset{\alpha}{\rightarrowtail} TA$$

"transitive"



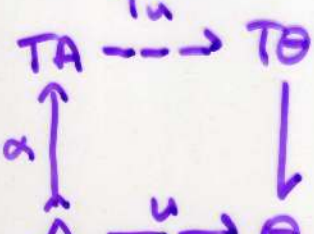
successor

T is a monad $T\eta\mu$

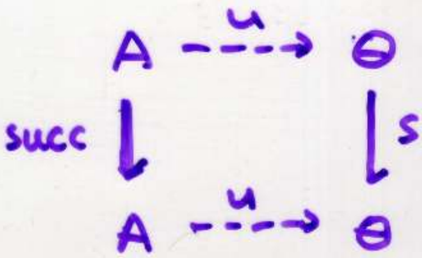


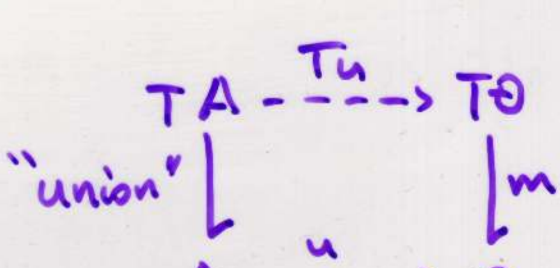
"union"

recursion for T



recursion for $T\eta\mu$





Examples

monad for algebraic theory $\mathcal{L}$

free T-algebra = free($\mathcal{L}$, s)-algebra

s = "brackets" (prevent computation)

covariant powerset on <u>Set</u>

transitive sets (ensembles)

with singleton $x \mapsto \{x\}$ and union

lower subsets of a poset

plump ordinals

Lower subsemilattices of a semilattice

plump directed ordinals

Scott-closed subsets of a dcpo

"ordinals" up to ω = fixed point

lift on dcpo same

powerset with restriction on size of subsets

sets & ordinals à la Joyal-Moerdijk

Ordinals

Try some other categories

posets TX = lower subsets of X

to get interesting notions of "ordinal", modify the notion of "mono"

ideally $X \hookrightarrow Y$ if it's the inclusion of a lower subset

binary semilattices

directed complete posets TX = Scott-closed sets

these ordinals admit fixed points (pace Burali-Forti!)

Other categories & functors?

Tarski's fixed point theorem?

Freyd's initial algebra – final coalgebra coincidence?

Elementary characterisation of transfinite iteration of functors (ax. replacement for toposes)