510996

TYPES

Types for Proofs and Programs

Coordination Action
FP6-2002-IST-C

Deliverable: Short Course

Monads and more

Intensive course by Tarmo Uustalu, Institute of Cybernetics, Tallinn, Estonia. Intended audience: Postgraduates and researchers in Theoretical Computer Science.

Slides:

- Monday (upupdated)
- Tuesday (update)
- Wednesday
- Friday
  Slides about tree transducers

Course contents:

1. Monads and why they matter for a working programming language person
2. Combining monads: monad transformers, distributive laws, the coproduct of monads
3. Finer and coarser: Lawvere theories and arrows
4. Comonads and context-dependent computation
5. Notions of computation on trees

Time and place

Monday 14 May - Wednesday + Friday, 9:00-11:00 in C60 (may change), CS & IT.

Contact

Thorsten Altenkirch

Last modified: Thu May 24 10:06:18 BST 2007

Monads and More: Part 1

Tarmo Uustalu, Institute of Cybernetics, Tallinn

University of Nottingham, 14–18 May 2007
University of Udine, 2–6 July 2007

Outline

- Monads and why they matter for a working functional programmer: monads, Kleisli categories, monadic computation, strong and commutative monads, monadic semantics
- Combining monads: monads from adjunctions, distributive laws, the coproduct of monads
- Finer and coarser: Lawvere theories, arrows and Freyd categories
- Comonadic notions of computation: comonads and coKleisli categories, comonadic computation, in particular dataflow computation, lax/strong symmetric monoidal comonads, comonadic semantics
- Notions of computation on trees

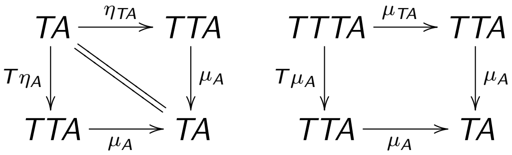
Prerequisites

- Basics of functional programming and typed lambda calculi
- From category theory:
  - functors, natural transformations
  - adjunctions
  - symmetric monoidal (closed) categories
  - Cartesian (closed) categories, coproducts
  - initial algebra, final coalgebra of a functor

Monads

- A *monad* on a category $\mathcal{C}$ is given by a
    - a functor $T : \mathcal{C} \to \mathcal{C}$ (the *underlying functor*),
    - a natural transformation $\eta : \mathrm{Id}_{\mathcal{C}} \stackrel{.}{\to} T$ (the *unit*),
    - a natural transformation $\mu : TT \stackrel{.}{\to} T$ (the *multiplication*)

  satisfying these conditions:



- This definition says that (T, η, μ) is a monoid in the endofunctor category $[\mathcal{C}, \mathcal{C}]$.

An alternative formulation: Kleisli triples

- A more combinatory formulation is the following.
- A *monad* (*Kleisli triple*) is given by
  - an object mapping $T : |\mathcal{C}| \to |\mathcal{C}|$,
  - for any object A, a map $\eta_A : A \to TA$,
  - for any map $k : A \to TB$, a map $k^\star : TA \to TB$ (the *Kleisli extension* operation)

  satisfying these conditions:
  - if $k : A \to TB$, then $k^\star \circ \eta_A = k$,
  - $\eta_A^\star = \mathrm{id}_{TA}$,
  - if $k : A \to TB$, $\ell : B \to TC$, then $(\ell^\star \circ k)^\star = \ell^\star \circ k^\star$.
- (Notice there are no explicit functoriality and naturality conditions.)

Monads vs. Kleisli triples

- There is a bijection between monads and Kleisli triples.
- Given T, η, μ, one defines
  - if $k : A \to TB$, then $k^\star =_{\mathrm{df}} \; TA \xrightarrow{Tk} TTB \xrightarrow{\mu_B} TB$.
- Given T (on objects only), η and $-^\star$, one defines
  - if $f : A \to B$, then
    $Tf =_{\mathrm{df}} (\; A \xrightarrow{f} B \xrightarrow{\eta_B} TB \;)^\star : TA \to TB$,
  - $\mu_A =_{\mathrm{df}} (\; TA \xrightarrow{\mathrm{id}_{TA}} TA \;)^\star : TTA \to TA$.

Kleisli category of a monad

- A monad T on a category $\mathcal{C}$ induces a category $\mathbf{Kl}(T)$ called the *Kleisli category* of T defined by
  - an object is an object of $\mathcal{C}$,
  - a map of from A to B is a map of $\mathcal{C}$ from A to TB,
  - $\mathrm{id}^T_A =_{\mathrm{df}} A \xrightarrow{\eta_A} TA$,
  - if $k : A \to^T B$, $\ell : B \to^T C$, then $\ell \circ^T k =_{\mathrm{df}} A \xrightarrow{k} TB \xrightarrow{T\ell} TTC \xrightarrow{\mu_C} TC$
- From $\mathcal{C}$ there is an identity-on-objects *inclusion functor* J to $\mathbf{Kl}(T)$, defined on maps by
  - if $f : A \to B$, then $Jf =_{\mathrm{df}} A \xrightarrow{f} B \xrightarrow{\eta_B} TB = A \xrightarrow{\eta_A} TA \xrightarrow{Tf} TB$.

Computational interpretation

- Think of $\mathcal{C}$ as the category of pure functions and of TA as the type of effectful computations of values of a type A.
- $\mathbf{Kl}(T)$ is then the category of effectful functions.
- $\eta_A : A \to TA$ is the identity function on A viewed as trivially effectful.
- $Jf : A \to TB$ is a general pure function $f : A \to B$ viewed as trivially effectful.
- $\mu_A : TTA \to TA$ flattens an effectful computation of an effectful computation.
- $k^\star : TA \to TB$ is an effectful function $k : A \to TB$ extended into one that can input an effectful computation.

Examples

- Exceptions monad:
  - $TA =_{\mathrm{df}} A + E$ where E is some object (of exceptions),
  - $\eta_A =_{\mathrm{df}} A \xrightarrow{\mathsf{inl}} A + E$,
  - $\mu_A =_{\mathrm{df}} (A + E) + E \xrightarrow{[\mathsf{id},\mathsf{inr}]} A + E$,
  - if $k : A \to B + E$, then $k^\star =_{\mathrm{df}} A + E \xrightarrow{[k,\mathsf{inr}]} B + E$.
- Output monad:
  - $TA =_{\mathrm{df}} A \times E$ where (E, e, m) is some monoid (of output traces), e.g., the type of lists of a fixed element type with nil and append,
  - $\eta_A =_{\mathrm{df}} A \xrightarrow{\mathsf{ur}} A \times 1 \xrightarrow{\mathsf{id} \times e} A \times E$,
  - $\mu_A =_{\mathrm{df}} (A \times E) \times E \xrightarrow{\mathsf{a}} A \times (E \times E) \xrightarrow{\mathsf{id} \times m} A \times E$,
  - if $k : A \to B \times E$, then
    $k^\star =_{\mathrm{df}} A \times E \xrightarrow{k \times \mathsf{id}} (B \times E) \times E \xrightarrow{\mathsf{a}} B \times (E \times E) \xrightarrow{\mathsf{id} \times m} B \times E$.

- Reader monad:
  - $TA =_{\mathrm{df}} E \Rightarrow A$ where E is some object (of environments),
  - $\eta_A =_{\mathrm{df}} \Lambda(A \times E \xrightarrow{\mathsf{fst}} A)$,
  - $\mu_A =_{\mathrm{df}} \Lambda((E \Rightarrow (E \Rightarrow A)) \times E \xrightarrow{\langle \mathsf{ev}, \mathsf{snd} \rangle} (E \Rightarrow A) \times E \xrightarrow{\mathsf{ev}} A)$,
  - if $k : A \to E \Rightarrow B$, then $k^\star =_{\mathrm{df}} \Lambda((E \Rightarrow A) \times E \xrightarrow{\langle \mathsf{ev}, \mathsf{snd} \rangle} A \times E \xrightarrow{k \times \mathsf{id}} (E \Rightarrow B) \times E \xrightarrow{\mathsf{ev}} B)$.

- Side-effect monad:
  - $TA =_{\mathrm{df}} S \Rightarrow A \times S$ where S is some object (of states),
  - $\eta_A =_{\mathrm{df}} \Lambda(A \times S \xrightarrow{id} A \times S)$,
  - $\mu_A =_{\mathrm{df}} \Lambda(S \Rightarrow ((S \Rightarrow A \times S) \times S) \times S$
    $\xrightarrow{\mathrm{ev}} (S \Rightarrow A \times S) \times S \xrightarrow{\mathrm{ev}} A \times S)$,
  - if $k : A \to S \Rightarrow B \times S$, then $k^\star =_{\mathrm{df}} \Lambda((S \Rightarrow A \times S) \times S$
    $\xrightarrow{\mathrm{ev}} A \times S \xrightarrow{k \times \mathrm{id}} (S \Rightarrow B \times S) \times S \xrightarrow{\mathrm{ev}} B \times S)$.
- Continuations monad:
  - $TA =_{\mathrm{df}} (A \Rightarrow R) \Rightarrow R$ where R is some object (of answers),
  - $\eta_A =_{\mathrm{df}} \Lambda(A \times (A \Rightarrow R) \xrightarrow{\mathrm{c}} (A \Rightarrow R) \times R \xrightarrow{ev} R)$,
  - if $k : A \to (B \Rightarrow R) \Rightarrow R$, then
    $k^\star =_{\mathrm{df}} \Lambda(((A \Rightarrow R) \Rightarrow R) \times (B \Rightarrow R)$
    $\xrightarrow{\mathrm{id} \times \Lambda(\Lambda^{-1}(k) \circ \mathrm{c})} ((A \Rightarrow R) \Rightarrow R) \times (A \Rightarrow R) \xrightarrow{\mathrm{ev}} R)$.

Strong functors

- A *strong functor* on a category $(\mathcal{C}, I, \otimes)$ is given by
  - an endofunctor F on $\mathcal{C}$,
  - together with a natural transformation $\mathsf{sl}_{A,B} : A \otimes FB \to F(A \otimes B)$ (the *(tensorial) strength*)

  satisfying

$$
\begin{array}{ccc}
I \otimes FA & \xrightarrow{\mathsf{sl}_{I,A}} & F(I \otimes A) \\
{\scriptstyle \mathsf{ul}_{FA}} \downarrow & & \downarrow {\scriptstyle F\mathsf{ul}_{A}} \\
FA & = & FA
\end{array}
$$

$$
\begin{array}{ccccc}
(A \otimes B) \otimes FC & & \xrightarrow{\mathsf{sl}_{A \otimes B, C}} & & F((A \otimes B) \otimes C) \\
{\scriptstyle \mathsf{a}_{A,B,FC}} \downarrow & & & & \downarrow {\scriptstyle F\mathsf{a}_{A,B,C}} \\
A \otimes (B \otimes FC) & \xrightarrow[\mathsf{id}_A \otimes \mathsf{sl}_{B,C}]{} & A \otimes F(B \otimes C) & \xrightarrow[\mathsf{sl}_{A,B \otimes C}]{} & F(A \otimes (B \otimes C))
\end{array}
$$

- A strong natural transformation between two strong functors (F, sl), (G, sl') is a natural transformation $\tau : F \dot{\to} G$ satisfying

$$\begin{array}{ccc} A \otimes FB & \xrightarrow{\mathsf{sl}_{A,B}} & F(A \otimes B) \\ {\scriptstyle \mathrm{id}_A \otimes \tau_B}\downarrow & & \downarrow{\scriptstyle \tau_{A \otimes B}} \\ A \otimes GB & \xrightarrow[\mathsf{sl}'_{A,B}]{} & G(A \otimes B) \end{array}$$

Strong monads

- A *strong monad* on a monoidal category $(\mathcal{C}, I, \otimes)$ is a monad (T, η, μ) together with a strength sl for T for which η and μ are strong, i.e., satisfy

$$\begin{array}{ccc} A\otimes B & \xlongequal{\quad} & A\otimes B \\ {\scriptstyle \mathrm{id}_A\otimes\eta_B}\downarrow & & \downarrow{\scriptstyle \eta_{A\otimes B}} \\ A\otimes TB & \xrightarrow[\mathrm{sl}_{A,B}]{} & T(A\otimes B) \end{array}$$

$$\begin{array}{ccccc} A\otimes TTB & \xrightarrow{\mathrm{sl}_{A,TB}} & T(A\otimes TB) & \xrightarrow{T\mathrm{sl}_{A,B}} & TT(A\otimes B) \\ {\scriptstyle \mathrm{id}_A\otimes\mu_B}\downarrow & & & & \downarrow{\scriptstyle \mu_{A\otimes B}} \\ A\otimes TB & & \xrightarrow[\mathrm{sl}_{A,B}]{} & & T(A\otimes B) \end{array}$$

(Note that Id is always strong and, if F, G are strong, then GF is strong.)

Commutative monads

- If $(\mathcal{C}, I, \otimes)$ is symmetric monoidal, then a strong functor (F, sl) is actually bistrong: it has a *costrength* $\mathsf{sr}_{A,B} : FA \otimes B \to F(A \otimes B)$ with properties symmetric to those of a strength defined by

$$\mathsf{sr}_{A,B} =_{\mathrm{df}} FA \otimes B \xrightarrow{\mathsf{c}_{FA,B}} B \otimes FA \xrightarrow{\mathsf{sl}_{B,A}} F(B \otimes A) \xrightarrow{F\mathsf{c}_{B,A}} F(A \otimes B)$$

- A bistrong monad $(T, \mathsf{sl}, \mathsf{sr})$ is called *commutative*, if it satisfies

$$\begin{array}{ccccc}
TA \otimes TB & \xrightarrow{\mathsf{sl}_{TA,B}} & T(TA \otimes B) & \xrightarrow{T\mathsf{sr}_{A,B}} & TT(A \otimes B) \\
\downarrow{\scriptstyle \mathsf{sr}_{A,TB}} & & & & \\
T(A \otimes TB) & & & & \downarrow{\scriptstyle \mu_{A \otimes B}} \\
\downarrow{\scriptstyle T\mathsf{sl}_{A,B}} & & & & \\
TT(A \otimes B) & & \xrightarrow[\mu_{A \otimes B}]{} & & T(A \otimes B)
\end{array}$$

Examples

- Exceptions monad:
  - $TA =_{\mathrm{df}} A + E$ where E is an object,
  - $\mathsf{sl}_{A,B} =_{\mathrm{df}} A \times (B + E) \xrightarrow{\mathsf{dr}} A \times B + A \times E \xrightarrow{\mathsf{id}+\mathsf{snd}} A \times B + E$.
- Output monad:
  - $TA =_{\mathrm{df}} A \times E$ where (E, e, m) is a monoid,
  - $\mathsf{sl}_{A,B} =_{\mathrm{df}} A \times (B \times E) \xrightarrow{\mathsf{a}^{-1}} (A \times B) \times E$.
- Reader monad:
  - $TA =_{\mathrm{df}} E \Rightarrow A$ where E is an object,
  - $\mathsf{sl}_{A,B} =_{\mathrm{df}} \Lambda((A \times (E \Rightarrow B)) \times E$
    $\xrightarrow{\mathsf{a}} A \times ((E \Rightarrow B) \times E) \xrightarrow{\mathsf{id}\times\mathsf{ev}} A \times B)$.

Tensorial vs. functorial strength

- A *functorially strong functor* on a monoidal closed category $(\mathcal{C}, I, \otimes, \multimap)$ is an endofunctor F on $\mathcal{C}$ with a natural transformation $\mathsf{fs}_{A,B} : A \multimap B \to FA \multimap FB$ internalizing the functorial action of F.
- There is a bijective correspondence between tensorially and functorially strong endofunctors, in fact an equivalence between their categories.
- Given fs, one defines sl by

$$\mathsf{sl}_{A,B} =_{\mathrm{df}} A \otimes FB \xrightarrow{\mathsf{coev} \otimes \mathsf{id}} (B \multimap A \otimes B) \otimes FB \xrightarrow{\Lambda^{-1}(\mathsf{fs})} F(A \otimes B)$$

- Given sl, one defines fs by

$$\mathsf{fs}_{A,B} =_{\mathrm{df}} \Lambda((A \multimap B) \otimes FA \xrightarrow{\mathsf{sl}} F((A \multimap B) \otimes A) \xrightarrow{F\mathsf{ev}} FB)$$

On **Set**, every monad is $(1, \times)$ strong

- Any endofunctor on **Set** has a unique functorial strength and any natural transformation between endofuctors on **Set** is functorially strong.
- Hence any monad on **Set** is both functorially and tensorially strong.

Effects

- Of course we want the Kleisli category of a monad to contain more maps than the base category.
- To describe those, we must single out some proper sources of effectfulness. How to choose those is a topic on its own.
- E.g., for the exceptions monad, an important map is $\mathsf{raise} =_{\mathrm{df}} E \xrightarrow{\mathsf{inr}} A + E$.

Semantics of pure typed lambda calculus

- Pure typed lambda calculus can be interpreted into any Cartesian closed category $\mathcal{C}$, e.g., **Set**.
- The interpretation is this:

$$
\begin{array}{rcl}
[\![K]\!] & =_{\mathrm{df}} & \text{an object of } \mathcal{C} \\
[\![A \times B]\!] & =_{\mathrm{df}} & [\![A]\!] \times [\![B]\!] \\
[\![A \Rightarrow B]\!] & =_{\mathrm{df}} & [\![A]\!] \Rightarrow [\![B]\!] \\
\\
[\![\underline{C}]\!] & =_{\mathrm{df}} & [\![C_0]\!] \times \ldots \times [\![C_{n-1}]\!] \\
\\
[\![(\underline{x})\, x_i]\!] & =_{\mathrm{df}} & \pi_i \\
[\![(\underline{x})\, let\ x \leftarrow t\ in\ u]\!] & =_{\mathrm{df}} & [\![(\underline{x}, x)\, u]\!] \circ \langle \mathrm{id}, [\![(x)\, t]\!] \rangle \\
[\![(\underline{x})\, fst(t)]\!] & =_{\mathrm{df}} & \mathrm{fst} \circ [\![(\underline{x})\, t]\!] \\
[\![(\underline{x})\, snd(t)]\!] & =_{\mathrm{df}} & \mathrm{snd} \circ [\![(\underline{x})\, t]\!] \\
[\![(\underline{x})\, (t_0, t_1)]\!] & =_{\mathrm{df}} & \langle [\![(\underline{x})\, t_0]\!], [\![(\underline{x})\, t_1]\!] \rangle \\
[\![(\underline{x})\, \lambda x t]\!] & =_{\mathrm{df}} & \Lambda([\![(\underline{x}, x)\, t]\!]) \\
[\![(\underline{x})\, t\ u]\!] & =_{\mathrm{df}} & \mathrm{ev} \circ \langle [\![(\underline{x})\, t]\!], [\![(\underline{x})\, u]\!] \rangle
\end{array}
$$

- This interpretation is sound: derivable typing judgements of the pure typed lambda calculus are valid, i.e.,

$$\underline{x} : \underline{C} \vdash t : A \text{ implies } [\![(\underline{x})\, t]\!] : [\![\underline{C}]\!] \to [\![A]\!]$$

  and the same holds true about all derivable equalities.
- This interpretation is also complete.

Pre-[Cartesian closed] structure of the Kleisli category of a strong monad

- Given a Cartesian (closed) category $\mathcal{C}$ and a $(1, \times)$ strong monad T on it, how much of that structure carries over to $\mathbf{Kl}(T)$?
- We can manufacture "pre-products" in $\mathbf{Kl}(T)$ using the products of $\mathcal{C}$ and the strength sl like this:

$$
\begin{array}{rcl}
A_0 \times^T A_1 & =_{\mathrm{df}} & A_0 \times A_1 \\
\mathsf{fst}^T & =_{\mathrm{df}} & \eta \circ \mathsf{fst} \\
\mathsf{snd}^T & =_{\mathrm{df}} & \eta \circ \mathsf{snd} \\
\langle k_0, k_1 \rangle^T & =_{\mathrm{df}} & \mathsf{sl}^\star \circ \mathsf{sr} \circ \langle k_0, k_1 \rangle
\end{array}
$$

$$\frac{k : C \to TA \quad \ell : C \times A \to TB}{\begin{array}{l}\ell \bullet^T k =_{\mathrm{df}} \\ \quad C \xrightarrow{\langle \mathsf{id}_C, k\rangle} C \times TA \xrightarrow{\mathsf{sl}_{C,A}} T(C \times A) \xrightarrow{\ell^\star} TB\end{array}}$$

$$\mathsf{fst}^T =_{\mathrm{df}} A_0 \times A_1 \xrightarrow{\mathsf{fst}} A_0 \xrightarrow{\eta} TA_0$$

$$\mathsf{snd}^T =_{\mathrm{df}} A_0 \times A_1 \xrightarrow{\mathsf{snd}} A_1 \xrightarrow{\eta} TA_1$$

$$\frac{k_0 : C \to TA_0 \quad k_1 : C \to TA_1}{\begin{array}{l}\langle k_0, k_1\rangle^T =_{\mathrm{df}} \\ \quad C \xrightarrow{\langle k_0, k_1\rangle} TA_0 \times TA_1 \xrightarrow{\mathsf{sr}_{A_0, TA_1}} T(A_0 \times TA_1) \xrightarrow{\mathsf{sl}^\star_{A_0, A_1}} T(A_0 \times A_1)\end{array}}$$

- The typing rules of products hold, but not all laws.
- In particular, we do not get the β-law of products. Effects cannot be undone!
- E.g., taking T to be the exception monad defined by $TA =_{\mathrm{df}} A + E$ for some fixed E we do not have $\mathsf{snd}^T \circ^T \langle k_0, k_1 \rangle^T = k_1$.
- Take $k_0 =_{\mathrm{df}} \mathsf{raise} = \mathsf{inr} : E \rightarrow TA$,
  $k_1 =_{\mathrm{df}} \mathsf{id}^T = \mathsf{inl} : E \rightarrow TE$
  Then $\langle k_0, k_1 \rangle^T = \mathsf{inr} : E \rightarrow T(A \times E)$ and hence $\mathsf{snd}^T \circ^T \langle k_0, k_1 \rangle^T = \mathsf{inr} \neq \mathsf{inl} = k_1$.
- In fact, $\times^T$ is not even a bifunctor unless T is commutative, although it is functorial in each argument separately. Effects do not commute in general!

- "Pre-exponents" are defined from the exponents of $\mathcal{C}$ by

$$
\begin{aligned}
A \Rightarrow^{T} B &=_{\mathrm{df}} A \Rightarrow TB \\
\mathsf{ev}^{T} &=_{\mathrm{df}} \mathsf{ev} \\
\Lambda^{T}(k) &=_{\mathrm{df}} \eta \circ \Lambda(k)
\end{aligned}
$$

$$
\mathsf{ev}^{T}_{A,B} =_{\mathrm{df}} (A \Rightarrow TB) \times A \xrightarrow{\mathsf{ev}_{A,TB}} TB
$$

$$
\frac{k : C \times A \to TB}{\Lambda^{T}(k) =_{\mathrm{df}} C \xrightarrow{\Lambda(k)} A \Rightarrow TB \xrightarrow{\eta} T(A \Rightarrow TB)}
$$

- It is not true that $A \Rightarrow^T - : \mathbf{Kl}(T) \to \mathbf{Kl}(T)$ is right adjoint to $- \times^T A : \mathbf{Kl}(T) \to \mathbf{Kl}(T)$.
  So $\Rightarrow^T$ is not a true exponent wrt. the preproduct $\times^T$.
- But $A \Rightarrow^T - : \mathbf{Kl}(T) \to \mathcal{C}$ is right adjoint to $J(- \times A) : \mathcal{C} \to \mathbf{Kl}(T)$:

$$\dfrac{\dfrac{\dfrac{J(C \times A) \to^T B}{C \times A \to TB}}{C \to A \Rightarrow TB}}{C \to A \Rightarrow^T B}$$

  We that say $A \Rightarrow^T B$ is the *Kleisli exponent* of A, B.
- More about the pre-[Cartesian closed] structure of Kleisli categories in the story about arrows.

CoCartesian structure of the Kleisli category of a monad

- If C is coCartesian (has coproducts), then $\mathbf{Kl}(T)$ is coCartesian too, since J as a left adjoint preserves colimits.
- Concretely, the coproduct on $\mathbf{Kl}(T)$ is defined by

$$
\begin{aligned}
A_0 +^T A_1 &=_{\mathrm{df}} A_0 + A_1 \\
\mathsf{inl}^T &=_{\mathrm{df}} \eta \circ \mathsf{inl} \\
\mathsf{inr}^T &=_{\mathrm{df}} \eta \circ \mathsf{inr} \\
[k_0, k_1]^T &=_{\mathrm{df}} [k_0, k_1]
\end{aligned}
$$

Semantics of an effectful language

- In the semantics of an effectful language, the semantic universe is the Kleisli category $\mathbf{Kl}(T)$ of the appropriate strong monad T on a Cartesian closed base category $\mathcal{C}$.
- The pure fragment is interpreted into $\mathbf{Kl}(T)$ as if the language was pure, using the pre-[Cartesian closed] structure:

$$
\begin{array}{rcl}
[\![K]\!]^T & =_{\mathrm{df}} & \text{an object of } \mathbf{Kl}(T) \\
 & & \qquad = \text{that object of } \mathcal{C} \\
[\![A \times B]\!]^T & =_{\mathrm{df}} & [\![A]\!]^T \times^T [\![B]\!]^T \\
 & & \qquad = [\![A]\!]^T \times [\![B]\!]^T \\
[\![A \Rightarrow B]\!]^T & =_{\mathrm{df}} & [\![A]\!]^T \Rightarrow^T [\![B]\!]^T \\
 & & \qquad = [\![A]\!]^T \Rightarrow T[\![B]\!]^T \\
[\![\underline{C}]\!]^T & =_{\mathrm{df}} & [\![C_0]\!]^T \times^T \ldots \times^T [\![C_{n-1}]\!]^T \\
 & & \qquad = [\![C_0]\!]^T \times \ldots \times [\![C_{n-1}]\!]^T
\end{array}
$$

$$
\begin{aligned}
[\![(\underline{x})\, x_i]\!]^T &=_{\mathrm{df}} \pi_i^T \\
&\qquad = \eta \circ \pi_i \\
[\![(\underline{x})\, let\ x \leftarrow t\ in\ u]\!]^T &=_{\mathrm{df}} [\![(\underline{x}, x)\, u]\!]^T \circ^T \langle \mathsf{id}^T, [\![(x)\ t]\!]^T \rangle^T \\
&\qquad = ([\![(\underline{x}, x)\, u]\!]^T)^\star \circ \mathsf{sl} \circ \langle \mathsf{id}, [\![(x)\ t]\!]^T \rangle \\
[\![(\underline{x})\, fst(t)]\!]^T &=_{\mathrm{df}} \mathsf{fst}^T \circ^T [\![(\underline{x})t]\!]^T \\
&\qquad = T\mathsf{fst} \circ [\![(\underline{x})t]\!]^T \\
[\![(\underline{x})\, snd(t)]\!]^T &=_{\mathrm{df}} \mathsf{snd}^T \circ^T [\![(\underline{x})t]\!]^T \\
&\qquad = T\mathsf{snd} \circ [\![(\underline{x})t]\!]^T \\
[\![(\underline{x})\, (t_0, t_1)]\!]^T &=_{\mathrm{df}} \langle [\![(\underline{x})t_0]\!]^T, [\![(\underline{x})t_1]\!]^T \rangle^T \\
&\qquad = \mathsf{sl}^\star \circ \mathsf{sr} \circ \langle [\![(\underline{x})t_0]\!]^T, [\![(\underline{x})t_1]\!]^T \rangle \\
[\![(\underline{x})\, \lambda x t]\!]^T &=_{\mathrm{df}} \Lambda^T([\![(\underline{x}, x)t]\!]^T) \\
&\qquad = \eta \circ \Lambda([\![(\underline{x}, x)t]\!]^T) \\
[\![(\underline{x})\, t\ u]\!]^T &=_{\mathrm{df}} \mathsf{ev}^T \circ^T \langle [\![(\underline{x})t]\!]^T, [\![(\underline{x})u]\!]^T \rangle^T \\
&\qquad = \mathsf{ev}^\star \circ \mathsf{sl}^\star \circ \mathsf{sr} \circ \langle [\![(\underline{x})t]\!]^T, [\![(\underline{x})u]\!]^T \rangle
\end{aligned}
$$

- As $\mathbf{Kl}(T)$ is only pre-Cartesian closed, for this pure fragment, soundness of typing holds, i.e.,

$$\underline{x} : \underline{C} \vdash t : A \text{ implies } [\![(\underline{x})\, t]\!]^T : [\![\underline{C}]\!]^T \rightarrow^T [\![A]\!]^T$$

  but not all equations of the pure typed lambda-calculus are validated.
- In particular,

$$\vdash t : A \text{ implies } [\![t]\!]^T : 1 \rightarrow^T [\![A]\!]^T$$

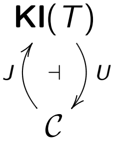
  so a closed term t of a type A denotes an element of $T[\![A]\!]^T$.

- Any effect-constructs must be interpreted specifically validating their desired typing rules and equations. E.g., for a language with exceptions we would use the exceptions monad and define

$$
\begin{aligned}
[\![(\underline{x})\, \mathit{raise}(e)]\!]^T \quad &=_{\mathrm{df}} \quad \mathrm{raise} \circ^T [\![(\underline{x})\, e]\!]^T \\
&= \mathrm{raise}^\star \circ [\![(\underline{x})\, e]\!]^T
\end{aligned}
$$

Kleisli adjunction

- Given a monad T on category $\mathcal{C}$, in the opposite direction to that of $J : \mathcal{C} \to \mathbf{Kl}(T)$ there is a functor $U : \mathbf{Kl}(T) \to \mathcal{C}$ defined by
  - $UA =_{\mathrm{df}} TA$,
  - if $k : A \to^T B$, then $Uk =_{\mathrm{df}} TA \xrightarrow{k^\star} TB$.
- U is right adjoint to J.



$$\frac{\dfrac{JA \to^T B}{A \to TB}}{A \to UB}$$

- Importantly, $UJ = T$. Indeed,
  - $UJA = TA$,
  - if $f : A \to B$, then $UJf = (\eta_B \circ f)^\star = Tf$.
- Moreover, the unit of the adjunction is η.
- $J \dashv U$ is the initial adjunction factorizing T in this way. There is also a final one, known as the Eilenberg-Moore

Kleisli categories

- In general one can define a *Kleisli category* on $\mathcal{C}$ to be
  - a category $\mathcal{D}$ with the same objects as $\mathcal{C}$
  - together with an identity-on-objects functor $J : \mathcal{C} \to \mathcal{D}$ with right adjoint U.
- To give a monad is the same as to give Kleisli category.
- We already know that a monad T induces a Kleisli category $\mathcal{D} =_{\mathrm{df}} \mathbf{Kl}(T)$.
- Given a Kleisli category $\mathcal{D}$, we obtain a monad by taking $T =_{\mathrm{df}} UJ$.

Monad maps

- A *monad map* between monads T, S on a category $\mathcal{C}$ is a natural transformation $\tau : T \stackrel{\cdot}{\to} S$ satisfying

$$
\begin{array}{ccc}
A & = & A \\
{\scriptstyle \eta^T_A}\downarrow & & \downarrow{\scriptstyle \eta^S_A} \\
TA & \xrightarrow[\tau_A]{} & SA
\end{array}
\qquad
\begin{array}{ccccc}
TTA & \xrightarrow{\tau_{TA}} & STA & \xrightarrow{S\tau_A} & SSA \\
{\scriptstyle \mu^T_A}\downarrow & & & & \downarrow{\scriptstyle \mu^S_A} \\
TA & & \xrightarrow[\tau_A]{} & & SA
\end{array}
$$

- Alternatively, a map between two monads (Kleisli triples) T, S is, for any object A, a map $\tau_A : TA \to SA$ satisfying
  - $\tau_A \circ \eta^T_A = \eta^S_A$,
  - if $k : A \to TB$, then $\tau_B \circ k^{\star T} = (\tau_B \circ k)^{\star S} \circ \tau_A$.

  (No explicit naturality condition on τ.)
- The two definitions are equivalent.
- Monads on $\mathcal{C}$ and maps between them form a category $\mathbf{Monad}(\mathcal{C})$.

Monad maps vs. functors between Kleisli categories

- There is a bijection between monad maps τ between T, S and functors $V : \mathbf{Kl}(T) \to \mathbf{Kl}(S)$ satisfying $VJ^T = J^S$.
- Given τ, one defines V by
  - $VA =_{\mathrm{df}} A$,
  - if $k : A \to TB$, then $Vk =_{\mathrm{df}} A \xrightarrow{k} TB \xrightarrow{\tau_B} SB$.
- Given V, one defines τ by
  - $\tau_A =_{\mathrm{df}} V(TA \xrightarrow{\mathrm{id}_{TA}} {}^T A) : TA \to^S A$.

Monads and More: Part 2

Tarmo Uustalu, Institute of Cybernetics, Tallinn

University of Nottingham, 14–18 May 2007
University of Udine, 2–6 July 2007

Monads from adjuctions (Huber)

- For any pair of adjoint functors $L : \mathcal{C} \to \mathcal{D}$, $R : \mathcal{D} \to \mathcal{C}$, $L \dashv R$ with unit $\eta : \mathsf{Id}_{\mathcal{C}} \stackrel{\cdot}{\to} RL$ and counit $\varepsilon : LR \stackrel{\cdot}{\to} \mathsf{Id}_{\mathcal{D}}$, the functor RL carries a monad structure defined by
  - $\eta^{RL} =_{\mathrm{df}} \mathsf{Id} \xrightarrow{\eta} RL$,
  - $\mu^{RL} =_{\mathrm{df}} RLRL \xrightarrow{R\varepsilon L} RL$.
- The Kleisli and Eilenberg-Moore adjunctions witness that any monad on $\mathcal{C}$ admits a factorization like this.

Examples

- State monad:
  - $L, R : \mathcal{C} \to \mathcal{C}$, $LA =_{\mathrm{df}} A \times S$, $RB =_{\mathrm{df}} S \Rightarrow B$,
  $$\frac{A \times S \to B}{A \to S \Rightarrow B}$$
  - $RLA = S \Rightarrow A \times S$,
- An exotic one:
  - $L, R : \mathcal{C} \to \mathcal{C}$, $LA =_{\mathrm{df}} \mu X. A + X \times S \cong A \times \mathsf{List} S$,
  $RB =_{\mathrm{df}} \nu Y. B \times (S \Rightarrow Y)$,
  $$\frac{\mu X. A + X \times S \to B}{A \to \nu Y. B \times (S \Rightarrow Y)}$$
  - $RLA = \nu Y. (\mu X. A + X \times S) \times (S \Rightarrow Y) \cong \nu Y. A \times \mathsf{List} S \times (S \Rightarrow Y)$.
  - What notion of computation does this correspond to?

- Continuations monad:
  - $L : \mathcal{C} \to \mathcal{C}^{\mathrm{op}}$, $LA =_{\mathrm{df}} A \Rightarrow E$,
    $R : \mathcal{C}^{\mathrm{op}} \to \mathcal{C}$, $RB =_{\mathrm{df}} B \Rightarrow E$,

$$
\begin{array}{c}
\hline\hline
A \Rightarrow E \leftarrow B \\
\hline\hline
E \leftarrow B \times A \\
\hline\hline
A \times B \to E \\
\hline\hline
A \to B \Rightarrow E
\end{array}
$$

  - $RLA = (A \Rightarrow E) \Rightarrow E$.

Monads from adjunctions ctd.

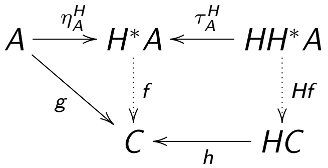
- Given two functors $L : \mathcal{C} \to \mathcal{D}$ and $R : \mathcal{D} \to C$, $L \dashv R$ and a monad T on $\mathcal{D}$, we obtain that RTL is a monad on $\mathcal{C}$.
- This is because T factorizes as UJ where $J \vdash U$ is the Kleisli adjunction.
  That means an adjoint situation $JL \vdash RU$ implying that $RUJL = RTL$ is a monad.
- The monad structure is
  - $\eta^{RTL} =_{\mathrm{df}} \mathsf{Id} \xrightarrow{\eta} RL \xrightarrow{R\eta^T L} RTL$,
  - $\mu^{RTL} =_{\mathrm{df}} RTLRTL \xrightarrow{RT\varepsilon TL} RTTL \xrightarrow{\mu^T} RTL$.

Examples

- State monad transformer:
  - $L, R : \mathcal{C} \to \mathcal{C}$, $LA =_{\mathrm{df}} A \times S$, $RB =_{\mathrm{df}} S \Rightarrow B$,
  - T – a monad on $\mathcal{C}$,
  - $RTLA = S \Rightarrow T(A \times S)$,
  - In particular, for T the exceptions monad we get $RTLA = S \Rightarrow (A \times S) + E$.
- Continuations monad transformer:
  - $L : \mathcal{C} \to \mathcal{C}^{\mathrm{op}}$, $LA =_{\mathrm{df}} A \Rightarrow E$,
    $R : \mathcal{C}^{\mathrm{op}} \to \mathcal{C}$, $RB =_{\mathrm{df}} B \Rightarrow E$,
  - T – a monad on $\mathcal{C}^{\mathrm{op}}$, i.e., a comonad on $\mathcal{C}$,
  - $RTLA =_{\mathrm{df}} T(A \Rightarrow E) \to E$.

Free algebras, free monads

- Given a endofunctor H on a category $\mathcal{C}$, let $(H^*A, [\eta^H_A, \tau^H_A])$ be the initial algebra of $A + H-$ (if it exists), so that, for any $A + H-$-algebra $(C, [g, h])$, there is a unique map $f : H^*A \to C$ satisfying



- H^*A is the type of wellfounded H-trees with mutable leaves from A, i.e., of H-terms over variables from A.

- $((H^* A, \tau_A^H), \eta_A^H)$ is the free H-algebra on A, i.e., $A \mapsto (H^* A, \tau^H A) : \mathcal{C} \to \mathbf{alg}(H)$ is left adjoint to the forgetful functor $U : \mathbf{alg}(H) \to \mathcal{C}$.

$$\frac{\dfrac{(H^* A, \tau_A) \to (C, h)}{A \to C}}{A \to U(C, h)}$$

and η^H is the unit of the adjunction.

- The pointed functor (H^*, η^H) carries a monad structure.
- The Kleisli extension $k^* : H^*A \to H^*B$ of any given map $k : A \to H^*B$ is defined as the unique map f satisfying

$$\begin{array}{ccccc} A & \xrightarrow{\eta_A} & H^*A & \xleftarrow{\tau_A} & HH^*A \\ & \searrow^{k} & \Big\downarrow{\scriptstyle f} & & \Big\downarrow{\scriptstyle Hf} \\ & & H^*B & \xleftarrow[\tau_B]{} & HH^*B \end{array}$$

Intuitively, this is grafting of trees into the mutable leaves of a tree or substitution of terms into the variables of a term.

- $((H^*, \eta^H, \mu^H), \tau^H)$ is the free monad on H, i.e., $H \mapsto (H^*, \eta^H, \mu^H) : [\mathcal{C}, \mathcal{C}] \to \mathbf{Monad}(\mathcal{C})$ is left adjoint to the forgetful functor $U : \mathbf{Monad}(\mathcal{C}) \to [\mathcal{C}, \mathcal{C}]$

$$\frac{\dfrac{(H^*, \eta^H, \mu^H) \to (S, \eta^S, \mu^S)}{H \to S}}{H \to U(S, \eta^S, \mu^S)}$$

and τ is the unit of the adjunction.

Free completely iterative algebras, free completely iterative monads (Adámek, Milius, Velebil)

- The final coalgebras $H^{\infty}A$ of $A + H-$ (the free completely iterative H-algebras over A) for each A also a give a monad (the free completely iterative monad on H).

Examples

- If $HX = 1 + X \times X$, then H^*A is the type of wellfounded binary trees with a termination option and with mutable leaves from A
(i.e., terms in the signature with one nullary, one binary operator over variables from A).
- If $HX =_{\mathrm{df}} \mathsf{List}X \cong \coprod_{i \in \mathbb{N}} X^i$, then H^*A is the type of wellfounded rose trees with mutable leaves from A
(i.e., terms in the signature with one operator of every finite arity over variables from A).

Monads from parameterized monads via initial algebras / final coalgebras (U.)

- A *parameterized monad* on $\mathcal{C}$ is a functor $F : \mathcal{C} \to \mathbf{Monad}(\mathcal{C})$.
- If F is a parameterized monad then the functors $F^*, F^\infty : \mathcal{C} \to \mathcal{C}$ defined by $F^* A =_{\mathrm{df}} \mu X.FXA$ and $F^\infty A =_{\mathrm{df}} \nu X.FXA$ carry a monad structure.
- In fact more can be said about them, but here we won't.

Examples

- Free monads:
  - $FXA =_{\mathrm{df}} A + HX$ where $H : \mathcal{C} \to \mathcal{C}$,
  - $F^* A =_{\mathrm{df}} \mu X.A + HX$, $F^\infty A =_{\mathrm{df}} \nu X.A + HX$.
  - These are the types of wellfounded/nonwellfounded H-trees with mutable leaves from A.
- Rose tree types:
  - $FXA =_{\mathrm{df}} A \times HX$ where $H : \mathcal{C} \to \mathbf{Monoid}(\mathcal{C})$,
  - $F^* A =_{\mathrm{df}} \mu X.A \times HX$, $F^\infty A =_{\mathrm{df}} \nu X.A \times HX$.
  - If $HX =_{\mathrm{df}} \mathrm{List} X$, these are the types of wellfounded/nonwellfounded A-labelled rose trees.

- Types of hyperfunctions with a fixed domain:
  - $FXA =_{\mathrm{df}} HX \Rightarrow A$ where $H : \mathcal{C} \to \mathcal{C}^{\mathrm{op}}$,
  - $F^* A =_{\mathrm{df}} \mu X.HX \Rightarrow A$, $F^\infty A =_{\mathrm{df}} \nu X.HX \Rightarrow A$.
  - If $FX =_{\mathrm{df}} X \Rightarrow E$, these are the types of wellfounded/nonwellfounded hyperfunctions from E to A. (Of course these types do no exist in **Set**.)

Distributive laws

- If T, S are monads on $\mathcal{C}$, it is not generally the case that ST is a monad. But sometimes it is.
- A *distributive law* of a monad T over a monad S is a natural transformation $\lambda : TS \xrightarrow{\cdot} ST$ satisfying

$$
\begin{array}{ccc}
T & = & T \\
{\scriptstyle T\eta^S}\downarrow & & \downarrow{\scriptstyle \eta^S T} \\
TS & \xrightarrow[\lambda]{} & ST
\end{array}
\qquad
\begin{array}{ccccc}
TSS & \xrightarrow{\lambda S} & STS & \xrightarrow{S\lambda} & SST \\
{\scriptstyle T\mu^S}\downarrow & & & & \downarrow{\scriptstyle \mu^S T} \\
TS & & \xrightarrow[\lambda]{} & & ST
\end{array}
$$

$$
\begin{array}{ccc}
S & = & S \\
{\scriptstyle \eta^T S}\downarrow & & \downarrow{\scriptstyle S\eta^T} \\
TS & \xrightarrow[\lambda]{} & ST
\end{array}
\qquad
\begin{array}{ccccc}
TTS & \xrightarrow{T\lambda} & TST & \xrightarrow{\lambda T} & STT \\
{\scriptstyle \mu^T S}\downarrow & & & & \downarrow{\scriptstyle S\mu^T} \\
TS & & \xrightarrow[\lambda]{} & & ST
\end{array}
$$

- A distributive law λ of T over S gives a monad structure on the endofunctor ST:
  - $\eta^{ST} =_{\mathrm{df}} \mathsf{Id} \stackrel{\eta^S \eta^T}{\longrightarrow} ST$,
  - $\mu^{ST} =_{\mathrm{df}} STST \stackrel{S\lambda T}{\longrightarrow} SSTT \stackrel{\mu^S \mu^T}{\longrightarrow} ST$.

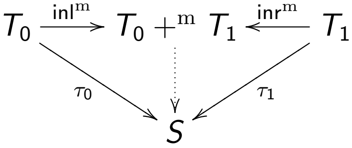
Examples

- The exceptions monad distributes over any monad.
  - S – a monad,
  - $TA =_{\mathrm{df}} A + E$ where E is an object,
  - $\lambda =_{\mathrm{df}} SA + E \xrightarrow{\mathsf{id}+\eta^S} SA + SE \xrightarrow{[S\mathsf{inl}, S\mathsf{inr}]} S(A + E)$,
  - $STA = S(A + E)$.
  - For T the state monad, this gives $ST = S \Rightarrow (A + E) \times S$, which is a different combination of exceptions and state than we saw before.
- The output monad distributes over any $(1, \times)$ strong monad.
  - (S, sl) – a strong monad,
  - $TA =_{\mathrm{df}} A \times E$ where E is a monoid,
  - $\lambda =_{\mathrm{df}} SA \times E \xrightarrow{\mathsf{sr}} S(A \times E)$,
  - $STA = S(A \times E)$.

- Any $(1, \times)$ strong monad distributes over the environment monad.
  - (T, sl) – a strong monad,
  - $SA =_{\mathrm{df}} E \Rightarrow A$ where E is an object,
  - $\lambda =_{\mathrm{df}} \Lambda(T(E \Rightarrow A) \times E \xrightarrow{\mathsf{sr}} T((E \Rightarrow A) \times E) \xrightarrow{T\mathsf{ev}} TA)$,
  - $STA = E \Rightarrow TA$.

Coproduct of monads

- An interesting canonical way to combine monads is the coproduct of monads.
- A coproduct of two monads T_0 and T_1 on $\mathcal{C}$ is their coproduct in $\mathbf{Monad}(\mathcal{C})$.
- I.e., it is a monad $T_0 +^{\mathrm{m}} T_1$ together with two monad maps $\mathsf{inl}^{\mathrm{m}} : T_0 \to^{\mathrm{m}} T_0 +^{\mathrm{m}} T_1$, $\mathsf{inr}^{\mathrm{m}} : T_0 \to^{\mathrm{m}} T_0 +^{\mathrm{m}} T_1$ such that for any monad S and monad maps $\tau_0 : T_0 \to^{\mathrm{m}} S$, $\tau_1 : T_1 \to^{\mathrm{m}} S$ there exists a unique monad map $T_0 +^{\mathrm{m}} T_1 \to^{\mathrm{m}} S$ satisfying



- The coproduct of two monads cannot be computed "pointwise", it is not the coproduct of the underlying functors.
- In fact, most of the time the coproduct of the underlying functors of two monads is not even a monad.

Coproduct of free monads

- The coproduct of the free monads on functors H_0, H_1 is the free monad on their coproduct:

$$H_0^\star +^{\mathrm{m}} H_1^\star = (H_0 + H_1)^*$$

(obvious, since the free monad delivering functor is a left adjoint and hence preserves colimits, in particular coproducts).

Coproduct of a free monad and an arbitrary monad (Power)

- More generally, the coproduct of a free monad H^* with an arbitary monad S is this (if $(HS)^*$ exists):

  $$H^* +^{\mathrm{m}} S = S(HS)^*$$

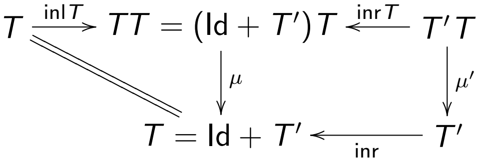
  i.e.,

  $$(H^* +^{\mathrm{m}} S)A = S(\mu X.A + HSX) = \mu X.S(A + HX)$$

- For $HX =_{\mathrm{df}} E$, $H^*A = \mu X.A + E \cong A + E$ (exceptions monad) and $(H^* +^{\mathrm{m}} S)A = \mu X.S(A + E) \cong S(A + E)$. This is the same combination of exceptions with any other monad as obtained from the canonical distributive law of the exceptions monad over another monad.

Ideal monads (Adámek, Milius, Velebil)

- Idea: to generalize the separation of variables from operator terms in term algebras.
- An *ideal monad* on $\mathcal{C}$ is a monad (T, η, μ) together with an endofunctor T' on $\mathcal{C}$ and a natural transformation $\mu' : T'T \dot{\to} T'$ such that
  - $T = \mathsf{Id} + T'$,
  - $\eta = \mathsf{inl}$,
  - $\mu = [id, \mathsf{inr} \circ \mu']$.



- An ideal monad map between $T = \mathsf{Id} + T'$ and $S = \mathsf{Id} + S'$ is monad map $\tau : T \dot{\to} S$ together with a nat. transf. $\tau' : T' \dot{\to} S'$ satisfying $\tau = \mathsf{id} + \tau'$.

Examples

- Free monads are ideal:
  - $TA =_{\mathrm{df}} \mu X.A + HX$ where $H : \mathcal{C} \to \mathcal{C}$
  - $TA \cong A + HTA$
- The finite powerset monad is not ideal:
  - $TA =_{\mathrm{df}} \mathcal{P}_{\mathrm{f}}$
  - $TA \cong A + 1 + \mathcal{P}_{\geq 2}A$, but $\mathcal{P}_{\geq 2}$ is not a functor: If for some $f : A \to B$ and $a_0, a_1 \in A$ we have $f(a_0) = f(a_1)$, then $\mathcal{P}_{\mathrm{f}} f$ sends a 2-element set $\{a_0, a_1\}$ to singleton.
- The finite multiset monad is not ideal:
  - $TA =_{\mathrm{df}} \mathcal{M}_{\mathrm{f}}$
  - $TA \cong A + 1 + \mathcal{M}_{\geq 2}A$, but μ does not restrict to a nat. transf. $\mathcal{M}_{\geq 2}\mathcal{M}_{\mathrm{f}} \stackrel{\cdot}{\to} \mathcal{M}_{\geq 2}$: If $a \in A$, then $\mu_A\{\{a\}, \emptyset\} = \{a\}$.

- The nonempty finite multiset monad is ideal:
  - $TA =_{\mathrm{df}} \mathcal{M}_{\geq 1}$
  - $TA \cong A + \mathcal{M}_{\geq 2}A$
- The nonempty list monad is ideal too.

Coproduct of ideal monads (Ghani, U.)

- Given two ideal monads $S_0 = \mathsf{Id} + S'_0$ and $S_1 = \mathsf{Id} + S'_1$, their coproduct is the ideal monad $T = \mathsf{Id} + T'_0 + T'_1$ defined by

$$(T'_0 A, T'_1 A) =_{\mathrm{df}} \mu(X_0, X_1).(S'_0(A + X_1)), S'_1(A + X_0))$$

Monads and More: Part 3

Tarmo Uustalu, Institute of Cybernetics, Tallinn

University of Nottingham, 14–18 May 2007
University of Udine, 2–6 July 2007

Arrows (Hughes)

- Arrows are a generalization of strong monads on symmetric monoidal categories (in their Kleisli triple form).
- An *arrow* on a symmetric monoidal category $(\mathcal{C}, I, \otimes)$ is given by
  - an object mapping $R : |\mathcal{C}| \times |\mathcal{C}| \to |\mathbf{Set}|$,
  - for any objects A, B of $\mathcal{C}$, a map $\mathrm{arr} : \mathrm{Hom}_{\mathcal{C}}(A, B) \to R(A, B)$ of **Set**,
  - for any objects A, B, C of $\mathcal{C}$, a map $\lll : R(A, B) \times R(B, C) \to R(A, C)$ of **Set**,
  - for any objects A, B, C of $\mathcal{C}$, a map $\mathrm{second}_C : R(A, B) \to R(C \otimes A, C \otimes B)$ of **Set**

  satisfying the conditions on the next slide.

- (ctd. from the previous slide)
  - if $k \in R(A,B)$, then $\mathsf{arr}\,\mathsf{id}_B \lll k = k$,
  - if $k \in R(A,B)$, then $k \lll \mathsf{arr}\,\mathsf{id}_A = k$,
  - if $k \in R(A,B)$, $\ell \in R(B,C)$, $m \in R(C,D)$, then $(m \lll \ell) \lll k = m \lll (\ell \lll k)$,
  - if $f : A \to B$, $g : B \to C$, then $\mathsf{arr}\,(g \circ f) = \mathsf{arr}\,g \lll \mathsf{arr}\ f$,
  - if $f : A \to B$, then $\mathsf{second}_C\,(\mathsf{arr}\,f) = \mathsf{arr}(\mathsf{id}_C \times f)$,
  - if $k \in R(A,B)$, $\ell \in R(B,C)$, $\mathsf{second}_D\,(\ell \lll k) = \mathsf{second}_D\,\ell \lll \mathsf{second}_D\,k$,
  - if $k \in R(A,B)$, $f : C \to D$, then $\mathsf{arr}\,(f \times \mathsf{id}_B) \lll \mathsf{second}_C\,k = \mathsf{second}_D\,k \lll \mathsf{arr}\,(f \times \mathsf{id}_A)$,
  - if $k \in R(A,B)$, $k \lll \mathsf{arr}\,\mathsf{ul}_A = \mathsf{ul}_B \lll \mathsf{second}_I\,k$,
  - if $k \in R(A,B)$, $\mathsf{second}_C\,(\mathsf{second}_D\,k) \lll \mathsf{a}_{C,D,A} = \mathsf{a}_{C,D,B} \lll \mathsf{second}_{C\otimes D}\,k$.

Examples

- Arrows from strong monoidal functors:
  - $R(A, B) =_{\mathrm{df}} \mathrm{Hom}_{\mathcal{C}}(FA, FB)$ where F is a monoidal endofunctor on $\mathcal{C}$ (i.e., there is a natural isomorphism $\mathsf{m}_{A,B} : FA \otimes FB \to F(A \otimes B)$),
  - if $f : A \to B$, then $\mathrm{arr}\, f = Ff : FA \to FB$,
  - if $k : FA \to FB$, $\ell : FB \to FC$, then $\ell \lll k =_{\mathrm{df}} FA \xrightarrow{k} FB \xrightarrow{\ell} FC$,
  - if $k : FA \to FB$, then $\mathrm{second}\, k =_{\mathrm{df}} F(C \otimes A) \xrightarrow{\mathsf{m}^{-1}} FC \otimes FA \xrightarrow{\mathsf{id} \otimes k} FC \otimes FB \xrightarrow{\mathsf{m}} F(C \otimes B)$.

- Kleisli maps of strong monads:
  - $R(A, B) =_{\mathrm{df}} \mathrm{Hom}_{\mathcal{C}}(A, TB)$ where T is a strong monad,
  - if $f : A \to B$, then $\mathrm{arr}\, f = Jf : A \to TB$ where J is the Kleisli inclusion of T,
  - if $k : A \to TB$, $\ell : B \to TC$, then $\ell \lll k =_{\mathrm{df}} A \xrightarrow{k} TB \xrightarrow{\ell^\star} TC$,
  - if $k : A \to TB$, then second $k =_{\mathrm{df}} C \otimes A \xrightarrow{\mathrm{id} \otimes k} C \otimes TB \xrightarrow{\mathrm{sr}} T(C \otimes B)$.
- CoKleisli maps of comonads on Cartesian categories:
  - $R(A, B) =_{\mathrm{df}} \mathrm{Hom}_{\mathcal{C}}(DA, B)$ where D is a comonad on $\mathcal{C}$,
  - if $f : A \to B$, then $\mathrm{arr}\, f = Jf : DA \to B$ where J is the coKleisli inclusion of D,
  - if $k : DA \to B$, $\ell : DB \to C$, then $\ell \lll k =_{\mathrm{df}} DA \xrightarrow{k^\dagger} DB \xrightarrow{\ell} C$,
  - if $k : DA \to B$, then second $k =_{\mathrm{df}} D(C \times A) \xrightarrow{\langle D\mathrm{fst}, D\mathrm{snd} \rangle} DC \times DA \xrightarrow{\varepsilon \times k} C \times B$.

- Output once more:
    - $R(A, B) =_{\rm df} E \times \mathrm{Hom}_{\mathcal{C}}(A, B)$ where (E, e, m) is a monoid in **Set**,
    - if $f : A \to B$, then $\mathrm{arr}\, f = (e, f) : E \times \mathrm{Hom}_{\mathcal{C}}(A, B)$,
    - if $(x, f) : E \times \mathrm{Hom}_{\mathcal{C}}(A, B)$, $(y, g) : E \times \mathrm{Hom}_{\mathcal{C}}(B, C)$, then $(y, g) \lll (x, f) =_{\rm df} (m(x, y), g \circ f) \in E \times \mathrm{Hom}_{\mathcal{C}}(A, C)$,
    - if $(x, f) : E \times \mathrm{Hom}_{\mathcal{C}}(A, B)$, then $\mathrm{second}\, (x, f) =_{\rm df} (x, C \otimes f) \in E \times \mathrm{Hom}_{\mathcal{C}}(C \otimes A, C \otimes B)$.

Arrows in the monoid form (Jacobs, Heunen, Hasuo)

- An alternative definition mimicks the definition of monads in the standard, i.e., monoid form.
- An *arrow* on a symmetric monoidal category $(\mathcal{C}, I, \otimes)$ is a strong monoid in the category of endoprofunctors on $(\mathcal{C}, I, \otimes)$.
- A *profunctor* from $\mathcal{C}$ to $\mathcal{D}$ is a functor $\mathcal{C}^{\mathrm{op}} \times \mathcal{D} \to \mathbf{Set}$.
  The identity profunctor on $\mathcal{C}$ is
  $\mathrm{Id} =_{\mathrm{df}} \mathrm{Hom}_{\mathcal{C}} : C^{\mathrm{op}} \times C \to \mathbf{Set}$.
  The composition of profunctors $R : \mathcal{C} \to \mathcal{D}$ and
  $S : \mathcal{D} \to \mathcal{E}$ is $SR(A, C) =_{\mathrm{df}} \int^{B} R(A, B) \times S(B, C)$.

- Accordingly, the data of an arrow are the following.
  - The carrier of an arrow is a profunctor R from $\mathcal{C}$ to $\mathcal{C}$, i.e., a functor $R : \mathcal{C}^{\mathrm{op}} \times \mathcal{C} \to \mathbf{Set}$.
  - The unit is a natural transformation from Id to R, i.e., a family of maps $\mathrm{arr}_{A,B} : \mathrm{Hom}_{\mathcal{C}}(A, B) \to R(A, B)$ natural in A, B.

    The multiplication is a nat. transf. from RR to R, i.e., a family of maps $\lll_{A,B,C}$: $R(A, B) \times R(B, C) \to R(A, C)$ natural in A, C and dinatural in B.

    The strength is a family of $\mathrm{second}_{A,B,C} :: R(A, B) \to R(C \otimes A, C \otimes B)$ natural in A, B and dinatural in C.

Symmetric premonoidal categories (Power, Robinson)

- Intuitively, a symmetric premonoidal category is the same as a symmetric monoidal category, except that the tensor is not necessarily a bifunctor, it must only be functorial in each argument separately.
- More officially: A *symmetric premonoidal category* is given by
  - a category $\mathcal{K}$,
  - an object I of $\mathcal{K}$,
  - for any object A, a functor $A \rtimes - : \mathcal{K} \to \mathcal{K}$,
  - natural isomorphisms a, ul, ur, c satisfying the laws of a symmetric monoidal category and have all their components central (see further).

- Symmetry yields symmetric functors $- \ltimes A : \mathcal{K} \to \mathcal{K}$ where $A_0 \ltimes A_1 = A_0 \rtimes A_1$ (which we also denote more symmetrically by $A_0 \otimes A_1$).
- A morphism $f : A \to B$ is called *central* if, for any $g : C \to D$, both

$$
\begin{array}{ccc}
A \otimes C & \xrightarrow{A \rtimes g} & A \otimes D \\
{\scriptstyle f \ltimes C}\downarrow & & \downarrow{\scriptstyle f \ltimes D} \\
B \otimes C & \xrightarrow[B \rtimes g]{} & C \otimes D
\end{array}
\qquad
\begin{array}{ccc}
C \otimes A & \xrightarrow{g \ltimes A} & D \otimes A \\
{\scriptstyle C \rtimes f}\downarrow & & \downarrow{\scriptstyle D \rtimes f} \\
C \otimes B & \xrightarrow[g \ltimes B]{} & D \otimes C
\end{array}
$$

Freyd categories

- A Freyd category on a symmetric monoidal category $\mathcal{C}$ is given by
  - a symmetric premonoidal category $(\mathcal{K}, I^{\mathcal{K}}, \otimes^{\mathcal{K}})$ with the same objects as $\mathcal{C}$
  - together with an identity-on-objects inclusion functor $J : \mathcal{C} \to \mathcal{K}$ that preserves centrality and strictly preserves its the $(I, \otimes)$ structure as premonoidal (meaning that $I^{\mathcal{K}} = I$, $A \otimes^{\mathcal{K}} B = A \otimes B$).

Freyd categories vs. arrows (Jacobs, Heunen, Hasuo)

- Freyd categories are in a bijection with arrows.
- For an arrow R on a symmetric monoidal category $(\mathcal{C}, I, \otimes)$, the Freyd category $((\mathcal{K}, I^{\mathcal{K}}, \otimes^{\mathcal{K}}), J)$ is defined by
  - an object is an object of $\mathcal{C}$,
  - a map from A to B is an element of $R(A, B)$,
  - $\mathrm{id}^{\mathcal{K}}_A =_{\mathrm{df}} \mathrm{arr}\,\mathrm{id}_A$,
  - if $k : A \to^{\mathcal{K}} B$, $\ell : B \to^{\mathcal{K}} C$, then $\ell \circ^{\mathcal{K}} k =_{\mathrm{df}} \ell \lll k$,
  - $I^{\mathcal{K}} = I$, $A \otimes^{\mathcal{K}} B =_{\mathrm{df}} A \otimes B$,
  - if $k : A \to^{\mathcal{K}} B$, then $C \rtimes^{\mathcal{K}} k =_{\mathrm{df}} \mathrm{second}\, k$,
  - if $f : A \to B$, then $Jf =_{\mathrm{df}} \mathrm{arr}\, f$.

- Given a Freyd category $((\mathcal{K}, I^{\mathcal{K}}, \otimes^{\mathcal{K}}), J)$ on $\mathcal{C}$, the corresponding arrow R is defined by
  - $R(A, B) =_{\mathrm{df}} \mathrm{Hom}_{\mathcal{K}}(A, B)$,
  - if $f : A' \to A$, $g : B \to B'$, $k \in \mathrm{Hom}_{\mathcal{K}}(A, B)$, then $R(f, g)\, k =_{\mathrm{df}} Jg \lll k \lll Jf$,
  - if $f : A \to B$, then $\mathrm{arr}\, f =_{\mathrm{df}} Jf \in \mathrm{Hom}_{\mathcal{K}}(A, B)$,
  - if $k \in \mathrm{Hom}_{\mathcal{K}}(A, B)$, $\ell \in \mathrm{Hom}_{\mathcal{K}}(B, C)$, then $\ell \lll k =_{\mathrm{df}} \ell \circ^{\mathcal{K}} k \in \mathrm{Hom}_{\mathcal{K}}(A, C)$,
  - if $k \in \mathrm{Hom}_{\mathcal{K}}(A, B)$, then $\mathrm{second}\, k =_{\mathrm{df}} C \rtimes^{\mathcal{K}} k \in \mathrm{Hom}_{K}(C \otimes A, C \otimes B)$.

When is Freyd Kleisli? (Power)

- Given a Freyd category $((\mathcal{K}, I^{\mathcal{K}}, \otimes^{\mathcal{K}}), J)$ on a symmetric monoidal category $(\mathcal{C}, I, \otimes)$, when is it the Kleisli category of a strong monad?
- A simple condition is in terms of Kleisli exponents.
- Suppose $J(- \ltimes A) : \mathcal{C} \to \mathcal{K}$ has a right adjoint $A \Rightarrow^{\mathcal{K}} -$. In this case we say the Freyd category is *closed*. Then also $TB =_{\mathrm{df}} I \Rightarrow^{\mathcal{K}} B$ is a strong monad with Kleisli exponents and $((\mathcal{K}, I^{\mathcal{K}}, \otimes^{\mathcal{K}}), J)$ is its Kleisli category.

Monads and More: Part 4

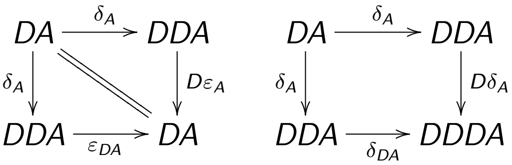
Tarmo Uustalu, Institute of Cybernetics, Tallinn

University of Nottingham, 14–18 May 2007
University of Udine, 2–6 July 2007

Comonads

- Comonads are the dual of monads.
- A *comonad* is a
  - a functor $D : \mathcal{C} \to \mathcal{C}$ (the *underlying functor*),
  - a natural transformation $\eta : D \stackrel{.}{\to} \mathsf{Id}_{\mathcal{C}}$ (the *counit*),
  - a natural transformation $\delta : D \stackrel{.}{\to} DD$ (the *comultiplication*)

  satisfying these conditions:



- In other words, a comonad is comonoid in $[\mathcal{C}, \mathcal{C}]$ (a monoid in $[\mathcal{C}, \mathcal{C}]^{\mathrm{op}}$).

CoKleisli triples

- A *coKleisli triple* is given by
  - an object mapping $D : |\mathcal{C}| \to |\mathcal{C}|$,
  - for any object A, a map $\varepsilon_A : DA \to A$,
  - for any map $k : DA \to B$, a map $k^\dagger : DA \to DB$ (the *coKleisli extension* operation)

  satisfying
  - if $k : DA \to B$, then $\varepsilon_B \circ k^\dagger = k$,
  - $\varepsilon_A^\dagger = \mathrm{id}_{DA}$,
  - if $k : DA \to B$, $\ell : DB \to C$, then $(\ell \circ k^\dagger)^\dagger = \ell^\dagger \circ k^\dagger$.
- There is a bijection between comonads and coKleisli triples.

CoKleisli category of a comonad

- A comonad D on a category $\mathcal{C}$ induces a category $\mathbf{CoKl}(D)$ called the *coKleisli category* of D defined by
  - an object is an object of $\mathcal{C}$,
  - a map of from A to B is a map of $\mathcal{C}$ from DA to B,
  - $\mathsf{id}^D_A =_{\mathrm{df}} DA \xrightarrow{\varepsilon_A} A$,
  - if $k : A \to^D B$, $\ell : B \to^D C$, then $\ell \circ^D k =_{\mathrm{df}} DA \xrightarrow{\mu_A} DDA \xrightarrow{Dk} DB \xrightarrow{\ell} C$.
- From $\mathcal{C}$ there is an identity-on-objects *inclusion functor* J to $\mathbf{CoKl}(D)$, defined on maps by
  - if $f : A \to B$, then $Jf =_{\mathrm{df}} DA \xrightarrow{\varepsilon_A} A \xrightarrow{f} B = DA \xrightarrow{Df} DB \xrightarrow{\varepsilon_B} B$.
- The functor J has a left adjoint $U : \mathbf{CoKl}(D) \to \mathcal{C}$ given by $UA =_{\mathrm{df}} DA$, if $k : A \to^D B$, then $Uk =_{\mathrm{df}} DA \xrightarrow{k^\dagger} DB$.

Comonadic notions of computation

- We think of $\mathcal{C}$ as the category of pure functions and of DA as the type of coeffectful computations of values of type A (values in context).
- **CoKl**(D) is the category of coeffectful or context-dependent functions.
- $\varepsilon_A : DA \to A$ is the identity on A seen as trivially context-dependent (discarding the context).
- $Jf : DA \to B$ is a general pure function $f : A \to B$ regarded as trivially context-dependen.
- $\delta_A : DA \to DDA$ blows the context of a value up (duplicates the context).
- $k^\dagger : DA \to DB$ is a context-dependent function $k : DA \to B$ extended into one that can output a value of in a context (e.g., for a postcomposed context-dependent function).

Examples

- Product comonad, for dependency on an environment:
  - $DA =_{\mathrm{df}} A \times E$ where E is an object of $\mathcal{C}$,
  - $\varepsilon_A =_{\mathrm{df}} A \times E \xrightarrow{\mathsf{fst}} A$,
  - $\delta_A =_{\mathrm{df}} A \times E \xrightarrow{\langle \mathsf{id}, \mathsf{snd} \rangle} (A \times E) \times E$,
  - if $k : A \times E \to B$, then $k^\dagger =_{\mathrm{df}} A \times E \xrightarrow{\langle k, \mathsf{snd} \rangle} B \times E$.
- This is the dual of the exceptions monad.
- It is not very interesting, as $\mathbf{CoKl}(D) \cong \mathbf{Kl}(T)$ for $TA =_{\mathrm{df}} E \Rightarrow A$ (the reader monad).

- Exponent comonad:
  - $DA =_{\mathrm{df}} E \Rightarrow A$ where (E, e, m) is a monoid in $\mathcal{C}$,
  - $\varepsilon_A =_{\mathrm{df}} (E \Rightarrow A) \xrightarrow{\mathsf{ur}^{-1}} (E \Rightarrow A) \times 1 \xrightarrow{\mathsf{id} \times e} (E \Rightarrow A) \times E \xrightarrow{\mathsf{ev}} A$,
  - $\delta_A =_{\mathrm{df}} \Lambda(\Lambda(((E \Rightarrow A) \times E) \times E \xrightarrow{\mathsf{a}} (E \Rightarrow A) \times (E \times E) \xrightarrow{\mathsf{id} \times m} (E \Rightarrow A) \times E \xrightarrow{\mathsf{ev}} A))$,
- Interesting special cases are $(E, e, m) =_{\mathrm{df}} (\mathsf{Nat}, 0, +)$ and $(E, e, m) =_{\mathrm{df}} (\mathsf{Nat}, 0, \mathsf{max})$.

- "Costate" comonad:
  - $DA =_{\mathrm{df}} (S \Rightarrow A) \times S$ where S is an object of $\mathcal{C}$,
  - $\varepsilon_A =_{\mathrm{df}} (S \Rightarrow A) \times S \xrightarrow{\mathsf{ev}} A$,
  - if $k : (S \Rightarrow A) \times S \to B$, then
    $k^\dagger =_{\mathrm{df}} (S \Rightarrow A) \times S \xrightarrow{\Lambda(k) \times \mathrm{id}} (S \Rightarrow B) \times S$.
- This comonad arises from the adjunction
  $S \times - \dashv S \Rightarrow -$.

Symmetric monoidal functors

- A *strong/lax symmetric monoidal functor* between symmetric monoidal categories $(\mathcal{C}, I, \otimes)$ and $(^{D}, I', \otimes')$ is
  - a functor on $F : \mathcal{C} \to^{D}$
  - together with an isomorphism/map $\mathrm{e} : I' \to FI$
  - and a natural isomorphism/transformation with components $\mathrm{m}_{A,B} : FA \otimes' FB \to F(A \otimes B)$

satisfying

$$\begin{array}{ccccc} FA \otimes' I' & \xrightarrow{\mathrm{id}\otimes'\mathrm{e}'} & FA \otimes' FI & \xrightarrow{\mathrm{m}_{A,I}} & F(A \otimes I) \\ \downarrow{\scriptstyle \mathrm{ur}'_{FA}} & & & & \downarrow{\scriptstyle F\mathrm{ur}_{A}} \\ FA & & = & & FA \end{array}$$

$$\begin{array}{ccc} FA \otimes' FB & \xrightarrow{\mathrm{m}_{A,B}} & F(A \otimes B) \\ \downarrow{\scriptstyle \mathrm{c}'_{FA,FB}} & & \downarrow{\scriptstyle F\mathrm{c}_{A,B}} \\ FB \otimes' FA & \xrightarrow[\mathrm{m}_{B,A}]{} & F(B \otimes A) \end{array}$$

$$\begin{array}{ccccc} (FA \otimes' FB) \otimes' FC & \xrightarrow{\mathrm{m}_{A,B}\otimes\mathrm{id}} & F(A \otimes B) \otimes' FC & \xrightarrow{\mathrm{m}_{A\otimes B,C}} & F((A \otimes B) \otimes C) \\ \downarrow{\scriptstyle \mathrm{a}'_{FA,FB,FC}} & & & & \downarrow{\scriptstyle F\mathrm{a}_{A,B,C}} \\ FA \otimes' (FB \otimes' FC) & \xrightarrow[\mathrm{id}\otimes\mathrm{m}_{B,C}]{} & FA \otimes' F(B \otimes C) & \xrightarrow[\mathrm{m}_{A,B\otimes C}]{} & F(A \otimes (B \otimes C)) \end{array}$$

- A *symmetric monoidal natural transformation* between two (strong or lax) symmetric monoidal functors $(F, \mathrm{e}, \mathrm{m})$, $(G, \mathrm{e}', \mathrm{m}')$ is a natural transformation $\tau : F \stackrel{.}{\to} G$ satisfying

$$
\begin{array}{ccc}
I' & \xrightarrow{\mathrm{e}} & FI \\
\Vert & & \downarrow{\scriptstyle \tau_I} \\
I' & \xrightarrow[\mathrm{e}']{} & GI
\end{array}
\qquad
\begin{array}{ccc}
FA \otimes' FB & \xrightarrow{\mathrm{m}_{A,B}} & F(A \otimes B) \\
{\scriptstyle \tau_A \otimes' \tau_B}\downarrow & & \downarrow{\scriptstyle \tau_{A \otimes B}} \\
GA \otimes' GB & \xrightarrow[\mathrm{m}'_{A,B}]{} & G(A \otimes B)
\end{array}
$$

Symmetric monoidal comonads

- A *strong/lax symmetric monoidal comonad* on a symmmetric monoidal category $(\mathcal{C}, I, \otimes)$ is a comonad (D, ε, δ) where D is a strong/lax symmetric monoidal functor (with I, $\otimes$ preserved by e, m) and ε, δ are symmetric monoidal natural transformations, i.e., satisfy

$$
\begin{array}{ccc}
I & \xrightarrow{\mathrm{e}} & DI \\
\Vert & & \downarrow{\scriptstyle \varepsilon_I} \\
I & = & I
\end{array}
\qquad
\begin{array}{ccccc}
I & & \xrightarrow{\mathrm{e}} & & DI \\
\Vert & & & & \downarrow{\scriptstyle \delta_I} \\
I & \xrightarrow[\mathrm{e}]{} & DI & \xrightarrow[D\mathrm{e}]{} & DDI
\end{array}
$$

$$
\begin{array}{ccc}
DA \otimes DB & \xrightarrow{\mathrm{m}_{A,B}} & D(A \otimes B) \\
{\scriptstyle \varepsilon_A \otimes \varepsilon_B}\downarrow & & \downarrow{\scriptstyle \varepsilon_{A \otimes B}} \\
A \otimes B & = & A \otimes B
\end{array}
$$

$$
\begin{array}{ccccc}
DA \otimes DB & & \xrightarrow{\mathsf{m}_{A,B}} & & D(A \otimes B) \\
\downarrow{\scriptstyle \delta_A \otimes \delta_B} & & & & \downarrow{\scriptstyle \delta_{A\otimes B}} \\
DDA \otimes DDB & \xrightarrow[\mathsf{m}_{DA,DB}]{} & D(DA \otimes DB) & \xrightarrow[D\mathsf{m}_{A,B}]{} & DD(A \otimes B)
\end{array}
$$

- (Note that Id is always symmetric monoidal and F, G being symmetric monoidal imply that GF is symmetric monoidal too.)
- A strong/lax symmetric *semi*monoidal comonad is as a strong/lax symmetric monoidal comonad, but without e (on a category which may be without I).

Dataflow computations

Dataflow computation = discrete-time signal transformations = stream functions.

The output value at a time instant (stream position) is determined by the input value at the same instant (position) plus further input values.

Example dataflow programs

$$
\begin{array}{rcl}
pos & = & 0 \text{ fby } (pos + 1) \\
sum\ x & = & x + (0 \text{ fby } (sum\ x)) \\
fact & = & 1 \text{ fby } (fact * (pos + 1)) \\
fibo & = & 0 \text{ fby } (fibo + (1 \text{ fby } fibo))
\end{array}
$$

`pos`	0	1	2	3	4	5	6	...
`sum pos`	0	1	3	6	10	15	21	...
`fact`	1	1	2	6	24	120	720	...
`fibo`	0	1	1	2	3	5	8	...

We want to consider functions $\mathsf{Str}\ A \to \mathsf{Str}\ B$ as impure functions from A to B.

Streams are naturally isomorphic to functions from natural numbers: $\mathsf{Str}A =_{\mathrm{df}} \nu X.A \times X \cong \mathsf{Nat} \Rightarrow A$.

General stream functions $\mathsf{Str}A \to \mathsf{Str}B$ are thus in natural bijection with maps $\mathsf{Str}A \times \mathsf{Nat} \to B$.

Comonad for general stream functions

- Functor:

$$DA =_{\mathrm{df}} (\mathsf{Nat} \Rightarrow A) \times \mathsf{Nat} \cong \mathsf{List}A \times \mathsf{Str}A$$

- Input streams with past/present/future:

$$a_0, a_1, \ldots, a_{n-1}, \boxed{a_n}, a_{n+1}, a_{n+2}, \ldots$$

- Counit:

$$\begin{array}{rcrcl} \varepsilon_A & : & (\mathsf{Nat} \Rightarrow A) \times \mathsf{Nat} & \to & A \\ & & (a, n) & \mapsto & a(n) \end{array}$$

- CoKleisli extension:

$$\frac{k \ : \ (\mathsf{Nat} \Rightarrow A) \times \mathsf{Nat} \ \to \ B}{\begin{array}{rcrcl} k^\star & : & (\mathsf{Nat} \Rightarrow A) \times \mathsf{Nat} & \to & (\mathsf{Nat} \Rightarrow B) \times \mathsf{Nat} \\ & & (a, n) & \mapsto & (\lambda m\, k(a, m), n) \end{array}}$$

Comonad for causal stream functions

- Functor: $DA =_{\mathrm{df}} \mathsf{NEList} \cong \mathsf{List}A \times A$
- Input streams with past and present but no future
- Counit:

$$\begin{array}{rcrcl} \varepsilon_A & : & \mathsf{NEList}A & \longrightarrow & A \\ & & [a_0, \ldots, a_n] & \mapsto & a_n \end{array}$$

- CoKleisli extension:

$$\frac{k \;:\; \mathsf{NEList}A \longrightarrow B}{\begin{array}{rcrcl} k^\star & : & \mathsf{NEList}A & \longrightarrow & \mathsf{NEList}B \\ & & [a_0, \ldots, a_n] & \mapsto & [k[a_0], k[a_0, a_1], \ldots, k[a_0, \ldots, a_n]] \end{array}}$$

Comonad for anticausal stream functions

- Input streams with present and future but no past
- Functor: $DA =_{\mathrm{df}} \mathsf{Str}A \cong A \times \mathsf{Str}A$

Relabelling tree transformations

- Let $H : \mathcal{C} \to \mathcal{C}$. Define $\mathsf{Tree}A =_{\mathrm{df}} \mu X.A \times HX$. We are interested in relabelling functions $\mathsf{Tree}A \to \mathsf{Tree}B$. (Alt. we can define $\mathsf{Tree}^\infty A =_{\mathrm{df}} \nu X.A \times HX$ and interest ourselves in relabelling functions $\mathsf{Tree}^\infty A \to \mathsf{Tree}^\infty B$.)
- Comonad for general relabelling functions:

$$DA =_{\mathrm{df}} \mathsf{Tree}'A \times A \cong \mathsf{Path}A \times \mathsf{Tree}A$$

  where $\mathsf{Path}A =_{\mathrm{df}} \mu X.1 + A \times H'(\mathsf{Tree}A) \times X$ (Huet's zipper).
- E.g., for $HX =_{\mathrm{df}} 1 + X \times X$, $H'X \cong 2 \times X$ and $\mathsf{Path}A \cong \mu X.1 + A \times 2 \times \mathsf{Tree}A \times X$.
- Comonad for bottom-up relabelling functions:

$$DA =_{\mathrm{df}} \mathsf{Tree}A$$

Cartesian preclosed structure of the coKleisli category of a strong/lax (semi)monoidal comonad

- Let D be a comonad on a Cartesian closed category $\mathcal{C}$.
- Since $J : \mathcal{C} \to \mathbf{CoKl}(D)$ is a right adjoint and preserves limits, $\mathbf{CoKl}(D)$ inherits products from $\mathcal{C}$. Explicitly, we can define

$$
\begin{array}{rcl}
A \times^D B & =_{\mathrm{df}} & A \times B \\
\pi_0^D & =_{\mathrm{df}} & \mathsf{fst} \circ \varepsilon \\
\pi_1^D & =_{\mathrm{df}} & \mathsf{snd} \circ \varepsilon \\
\langle k_0, k_1 \rangle^D & =_{\mathrm{df}} & \langle k_0, k_1 \rangle
\end{array}
$$

- If D is $(1, \times)$ strong/lax symmetric semimonoidal, then we can also define

$$\begin{array}{rcl} A \Rightarrow^D B & =_{\mathrm{df}} & DA \Rightarrow B \\ \mathsf{ev}^D & =_{\mathrm{df}} & \mathsf{ev} \circ \langle \varepsilon \circ D\mathsf{fst}, D\mathsf{snd} \rangle \\ \Lambda^D(k) & =_{\mathrm{df}} & \Lambda(k \circ \mathsf{m}) \end{array}$$

$$D((DA \Rightarrow B) \times A) \xrightarrow{\langle \varepsilon \circ D\mathsf{fst}, D\mathsf{snd} \rangle} (DA \Rightarrow B) \times DA \xrightarrow{\mathsf{ev}} B$$

$$\frac{DC \times DA \xrightarrow{\mathsf{m}} D(C \times A) \xrightarrow{k} B}{DC \xrightarrow{\Lambda(k \circ \mathsf{m})} DA \Rightarrow B}$$

- Using a strength (if available) is not a good idea: We have no multiplication

$$DC \times DA \xrightarrow{\mathsf{sl}} D(C \times DA) \xrightarrow{D\mathsf{sr}} DD(C \times A) \xrightarrow{?} D(C \times A)$$

and applying ε or $D\varepsilon$ gives a solution where the order of arguments of a function is important and coeffects do not combine:

$$DC \times DA \xrightarrow{\mathsf{id}\times\varepsilon} DC \times A \xrightarrow{\mathsf{sl}} D(C \times A)$$

or

$$DC \times DA \xrightarrow{\varepsilon\times\mathsf{id}} C \times DA \xrightarrow{\mathsf{sr}} D(C \times A)$$

- If D is strong semimonoidal (in which case it is automatically strong symmetric semimonoidal as well), then $A \Rightarrow^D -$ is right adjoint to $- \times^D A$ and hence $\Rightarrow^D$ is an exponent functor:

$$\frac{\dfrac{D(C \times A) \to B}{DC \times DA \to B}}{DC \to DA \Rightarrow B}$$

- This is the case, e.g., if $DA \cong \nu X.A \times (E \Rightarrow X)$ for some E (e.g., $DA \cong \mathsf{Str} A \cong \nu X.A \times (1 \Rightarrow X)$).

- More typically, D is only lax symmetric semimonoidal.
- Then it suffices to have m satisfying

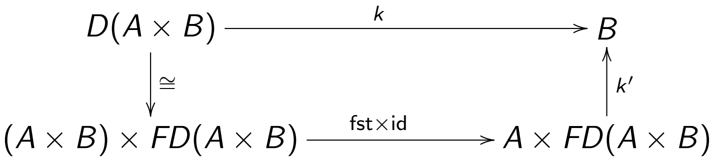
$$\begin{array}{ccc} DA & = & DA \\ {\scriptstyle \Delta_{DA}}\downarrow & & \downarrow{\scriptstyle D\Delta_A} \\ DA \times DA & \xrightarrow[\mathrm{m}_{A,A}]{} & D(A, A) \end{array}$$

where $\Delta = \langle \mathrm{id}, \mathrm{id} \rangle : A \to A \times A$ is part of the comonoid structure on the objects of $\mathcal{C}$, to get that $m \circ \langle D\mathrm{fst}, D\mathrm{snd} \rangle = \mathrm{id}$ and that $\Rightarrow^D$ is a weak exponent operation on objects. It is not functorial (not even in each argument separately).

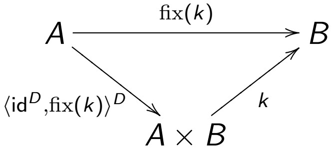
Partial uniform parameterized fixpoint operator

Let $F : \mathcal{C} \to \mathcal{C}$. Define $DA =_{\mathrm{df}} \nu Z.A \times FZ$.

Call a coKleisli map $k : A \times B \to^D B$ *guarded* if for some k' we have



For any guarded $k : A \times B \to^D B$, there is a unique map $\mathrm{fix}(k) : A \to^D B$ satisfying



fix is a partial *Conway operator* defined on guarded maps, i.e., besides the *fixpoint property*, for any guarded $k : A \times^D B \to^D B$,

$$\mathrm{fix}(k) = k \circ^D \langle \mathsf{id}^D, \mathrm{fix}(k) \rangle^D$$

it satisfies *naturality* in A, *dinaturality* in B, and the *diagonal property*: for any guarded $k : A \times^D B \times^D B \to^D B$,

$$\mathrm{fix}(k \circ^D (\mathsf{id}^D \times^D \Delta^D)) = \mathrm{fix}(\mathrm{fix}(k))$$

Wrt. pure maps, fix is also *uniform* (i.e., strongly dinatural in B instead of dinatural), i.e., for any guarded $k : A \times^D B \to^D B$, $\ell : A \times^D B' \to^D B'$ and $h : B \to B'$

$$Jh \circ^D k = \ell \circ^D (\mathsf{id}^D \times^D Jh) \quad \Longrightarrow \quad Jh \circ^D \mathrm{fix}(k) = \mathrm{fix}(\ell)$$

Comonadic semantics

- As in the case of monadic semantics, we interpret the lambda-calculus into **CoKl**(D) in the standard way (using its Cartesian preclosed structure), getting

$$
\begin{array}{rcl}
[\![K]\!]^D & =_{\mathrm{df}} & \text{an object of } \mathbf{CoKl}(D) \\
& & \quad = \text{that object of } \mathcal{C} \\
[\![A \times B]\!]^D & =_{\mathrm{df}} & [\![A]\!]^D \times^D [\![B]\!]^D \\
& & \quad = [\![A]\!]^D \times [\![B]\!]^D \\
[\![A \Rightarrow B]\!]^D & =_{\mathrm{df}} & [\![A]\!]^D \Rightarrow^D [\![B]\!]^D \\
& & \quad = D[\![A]\!]^D \Rightarrow [\![B]\!]^D \\
[\![\underline{C}]\!]^D & =_{\mathrm{df}} & [\![C_0]\!]^D \times \ldots \times [\![C_{n-1}]\!]^D \\
& & \quad = [\![C_0]\!] \times \ldots \times [\![C_{n-1}]\!]
\end{array}
$$

$$
\begin{aligned}
[\![(\underline{x})\, x_i]\!]^D &=_{\mathrm{df}} \pi_i^D \\
&\qquad\qquad = \pi_i \circ \varepsilon \\
[\![(\underline{x})\, \mathit{fst}(t)]\!]^D &=_{\mathrm{df}} \pi_0^D \circ^D [\![(\underline{x})\, t]\!]^D \\
&\qquad\qquad = \mathsf{fst} \circ [\![(\underline{x})\, t]\!]^D \\
[\![(\underline{x})\, \mathit{snd}(t)]\!]^D &=_{\mathrm{df}} \pi_1^D \circ^D [\![(\underline{x})\, t]\!]^D \\
&\qquad\qquad = \mathsf{snd} \circ [\![(\underline{x})\, t]\!]^D \\
[\![(\underline{x})\, (t_0, t_1)]\!]^D &=_{\mathrm{df}} \langle [\![(\underline{x})\, t_0]\!]^D, [\![(\underline{x})\, t_1]\!]^D \rangle^D \\
&\qquad\qquad = \langle [\![(\underline{x})\, t_0]\!]^D, [\![(\underline{x})\, t_1]\!]^D \rangle \\
[\![(\underline{x})\, \lambda x t]\!]^D &=_{\mathrm{df}} \Lambda^D([\![(\underline{x}, x)\, t]\!]^D) \\
&\qquad\qquad = \Lambda([\![(\underline{x}, x)\, t]\!]^D \circ \mathsf{m}) \\
[\![(\underline{x})\, t\; u]\!]^D &=_{\mathrm{df}} \mathsf{ev}^D \circ^D \langle [\![(\underline{x})\, t]\!]^D, [\![(\underline{x})\, u]\!]^D \rangle^D \\
&\qquad\qquad = \mathsf{ev} \circ \langle [\![(\underline{x})\, t]\!]^D, ([\![(\underline{x})\, u]\!]^D)^\dagger \rangle
\end{aligned}
$$

- Coeffect-specific constructs are interpreted specifically.
- E.g., for the constructs of a general/causal/anticausal dataflow language we can use the appropriate comonad and define:

$$
\begin{array}{rcl}
[\![(\underline{x})\, t \;\; \mathit{fby} \;\; u]\!]^D & =_{\mathrm{df}} & \mathsf{fby} \circ \langle [\![(\underline{x})\, t]\!]^D, ([\![(\underline{x})\, u)]\!]^D)^\dagger \rangle^D \\
[\![(\underline{x})\, t \;\; \mathit{next} \;\; u]\!]^D & =_{\mathrm{df}} & \mathsf{next} \circ ([\![(\underline{x})\, t]\!]^D)^\dagger
\end{array}
$$

- Again, we have soundness of typing, in the form $\underline{x} : \underline{C} \vdash t : A$ implies $[\![(\underline{x})t]\!]^D : [\![\underline{C}]\!]^D \to^D [\![A]\!]^D$, but not all equations of the lambda-calculus are validated.
- For a closed term $\vdash t : A$, soundness of typing says that $[\![t]\!]^D : 1 \to^D [\![A]\!]^D$, i.e., $D1 \to [\![A]\!]^D$, so closed terms are evaluated relative to a coeffect over 1.
- In case of general or causal stream functions, an element of $D1$ is a list over 1, i.e., a natural number, the time elapsed.
- If D is strong or lax symmetric monoidal (not just semimonoidal), we have a canonical choice $\mathrm{e} : 1 \to D1$.

- Comonadic dataflow language semantics: The first-order language agrees perfectly with Lucid and Lustre by its semantics.
  The meaning of higher-order dataflow computation has been unclear. We get a neat semantics from mathematical considerations (cf. Colaço, Pouzet's design with two flavors of function spaces).

Symmetric monoidal comonads (and strong monads) in linear / modal logic

- Strong symmetric monoidal comonads are central in the semantics of intuitionistic linear logic and modal logic to interpret the ! and $\Box$ ($\Diamond$) operators.
- Linear logic: Benton, Bierman, de Paiva, Hyland; Bierman; Benton; Mellies; Maneggia; etc.
- Modal logic: Bierman, de Paiva.
- Applications to staged computation and semantics of names: Pfenning, Davies, Nanevski.