Monads and More: Part 1

Tarmo Uustalu, Tallinn

Nottingham, 14–18 May 2007

Outline

- Monads and why they matter for a working functional programmer
- Combining monads: monad transformers, distributive laws, the coproduct of monads
- Finer and coarser: Lawvere theories and arrows
- Comonadic notions of computation: dataflow notions of computation, notions of computation on trees

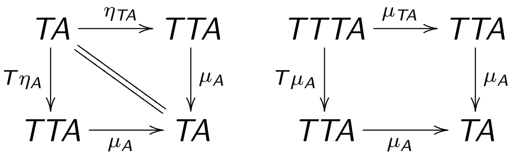
Prerequisites

- 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 $\eta : \mathsf{Id}_{\mathcal{C}} \stackrel{\cdot}{\to} T$ (the *unit*),
  - a natural transformation $\mu : TT \stackrel{\cdot}{\to} T$ (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
    $Tf =_{\mathrm{df}} (\, A \xrightarrow{f} B \xrightarrow{\eta_B} TB \,)^\star : TA \to TB$,
  - $\mu_A =_{\mathrm{df}} (\, TA \xrightarrow{\mathsf{id}_{TA}} TA \,)^\star : TTA \to TA$.

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.

Kleisli adjunction

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

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

- 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 adjunction.

Examples

- Exceptions monad:
  - $TA =_{\mathrm{df}} A + E$ where E is some object (of exceptions),
  - $\eta_A =_{\mathrm{df}} A \stackrel{\mathsf{inl}}{\longrightarrow} A + E$,
  - $\mu_A =_{\mathrm{df}} (A + E) + E \stackrel{[\mathsf{id},\mathsf{inr}]}{\longrightarrow} A + E$,
  - if $k : A \to B + E$, then $k^\star =_{\mathrm{df}} A + E \stackrel{[k,\mathsf{inr}]}{\longrightarrow} B + E$.
- 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,
  - $\eta_A =_{\mathrm{df}} A \stackrel{\mathsf{ur}}{\longrightarrow} A \times 1 \stackrel{\mathsf{id} \times e}{\longrightarrow} A \times E$,
  - $\mu_A =_{\mathrm{df}} (A \times E) \times E \stackrel{\mathsf{a}}{\longrightarrow} A \times (E \times E) \stackrel{\mathsf{id} \times m}{\longrightarrow} A \times E$,
  - if $k : A \to B \times E$, then
    $k^\star =_{\mathrm{df}} A \times E \stackrel{k \times \mathsf{id}}{\longrightarrow} (B \times E) \times E \stackrel{\mathsf{a}}{\longrightarrow} B \times (E \times E) \stackrel{\mathsf{id} \times m}{\longrightarrow} B \times E$.

- 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),
  - $\eta_A =_{\mathrm{df}} \Lambda(A \times S \xrightarrow{id} A \times S)$,
  - $\mu_A =_{\mathrm{df}} \Lambda(S \Rightarrow ((S \Rightarrow A \times S) \times S) \times S \xrightarrow{\mathsf{ev}} (S \Rightarrow A \times S) \times S \xrightarrow{\mathsf{ev}} A \times S)$,
  - if $k : A \to S \Rightarrow B \times S$, then $k^\star =_{\mathrm{df}} \Lambda((S \Rightarrow A \times S) \times S \xrightarrow{\mathsf{ev}} A \times S \xrightarrow{k} (S \Rightarrow B \times S) \times S \xrightarrow{\mathsf{ev}} B \times S)$.

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

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

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

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

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

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

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

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

(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

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

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 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:

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

- 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:

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

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

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

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

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

- 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 $k_0 =_{\mathrm{df}} \mathsf{raise} = \mathsf{inr} : E \to TA$,
  $k_1 =_{\mathrm{df}} \mathsf{id}^T = \mathsf{inl} : E \to TE$
  Then $\langle k_0, k_1 \rangle^T = \mathsf{inr} : E \to T(A \times E)$ 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

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

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

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

- 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 $J(- \times A) : \mathcal{C} \to \mathbf{Kl}(T)$:

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

  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 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:

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

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

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

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

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

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

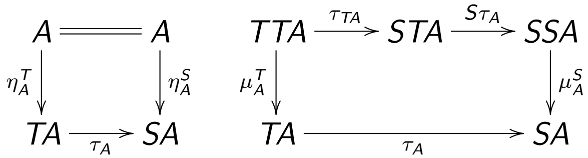
  so a closed term t of a type A denotes an element of $T[\![A]\!]^T$.

- 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

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

Monad maps

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



- Alternatively, a map between two monads (Kleisli triples) T, S is, for any object A, a map $\tau_A : TA \rightarrow SA$ satisfying
  - $\tau_A \circ \eta^T_A = \eta^S_A$,
  - if $k : A \rightarrow TB$, 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 **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
  - $\tau_A =_{\mathrm{df}} V(TA \xrightarrow{\mathrm{id}_{TA}} {}^T A) : TA \to^S A$.