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 $\eta : \mathsf{Id}_{\mathcal{C}} \stackrel{.}{\to} RL$ and counit $\varepsilon : LR \stackrel{.}{\to} \mathsf{Id}_{\mathcal{D}}$, 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:
  - $L : \mathcal{C} \to \mathcal{C}^{\mathrm{op}}$, $LA =_{\mathrm{df}} A \Rightarrow E$,
    $R : \mathcal{C}^{\mathrm{op}} \to \mathcal{C}$, $RB =_{\mathrm{df}} B \Rightarrow E$,

$$
\dfrac{\dfrac{\dfrac{\dfrac{}{A \Rightarrow E \leftarrow B}}{E \leftarrow B \times A}}{A \times B \to E}}{A \to B \Rightarrow E}
$$

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

Monads from adjunctions ctd.

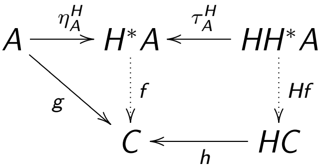
- Given two functors $L : \mathcal{C} \to \mathcal{D}$ and $R : \mathcal{D} \to C$, $L \dashv R$ and a monad T on $\mathcal{D}$, we obtain that RTL is a monad on $\mathcal{C}$.
- This is because T factorizes as UJ where $J \vdash U$ is the Kleisli adjunction.
  That means an adjoint situation $JL \vdash RU$ 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 $U : \mathbf{alg}(H) \to \mathcal{C}$.

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

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

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

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 $HX =_{\mathrm{df}} \mathsf{List}X$, 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 $\lambda : TS \dot{\to} ST$ satisfying

$$\begin{CD}
T @= T \\
@V{T\eta^S}VV @VV{\eta^S T}V \\
TS @>>{\lambda}> ST
\end{CD}
\qquad
\begin{CD}
TSS @>{\lambda S}>> STS @>{S\lambda}>> SST \\
@V{T\mu^S}VV @. @VV{\mu^S T}V \\
TS @>>{}> @>>{\lambda}> ST
\end{CD}$$

$$\begin{CD}
S @= S \\
@V{\eta^T S}VV @VV{S\eta^T}V \\
TS @>>{\lambda}> ST
\end{CD}
\qquad
\begin{CD}
TTS @>{T\lambda}>> TST @>{\lambda T}>> STT \\
@V{\mu^T S}VV @. @VV{S\mu^T}V \\
TS @>>{}> @>>{\lambda}> ST
\end{CD}$$

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

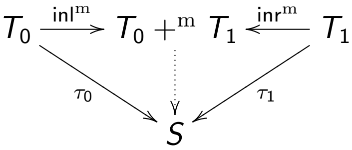
Examples

- 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):

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

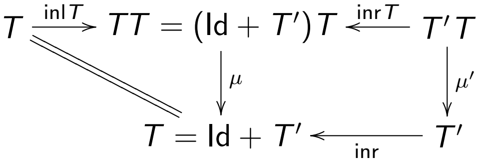
i.e.,

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

- 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:
  - $TA =_{\rm df} \mu X.A + HX$ where $H : \mathcal{C} \to \mathcal{C}$
  - $TA \cong A + HTA$
- The finite powerset monad is not ideal:
  - $TA =_{\rm df} \mathcal{P}_{\rm f}$
  - $TA \cong A + 1 + \mathcal{P}_{\geq 2}A$, 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 $\mathcal{P}_{\rm f} f$ sends a 2-element set $\{a_0, a_1\}$ to singleton.
- The finite multiset monad is not ideal:
  - $TA =_{\rm df} \mathcal{M}_{\rm f}$
  - $TA \cong A + 1 + \mathcal{M}_{\geq 2}A$, but μ does not restrict to a nat. transf. $\mathcal{M}_{\geq 2}\mathcal{M}_{\rm f} \stackrel{.}{\to} \mathcal{M}_{\geq 2}$: If $a \in A$, then $\mu_A\{\{a\}, \emptyset\} = \{a\}$.

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

$$(T_0'A, T_1'A) =_{\mathrm{df}} \mu(X_0, X_1).(S_0'(A + X_1)), S_1'(A + X_0))$$