Update monads: Cointerpreting directed containers

Danel Ahman, U. of Edinburgh
Tarmo Uustalu, Inst. of Cybernetics, Tallinn

Theory Days at Saka, 25–27 October 2013

Background: Three famous monads

Reader monad	State monad	Writer monad
S —a set	S —a set	$(P, \mathrm{o}, \oplus)$ —a monoid
$T\,X = S \to X$	$T\,X = S \to S \times X$	$T\,X = P \times X$

S —states, P —updates (alt. "programs")

This talk: A unification (+ a little more)

Update monad

S —a set

$(P, o, \oplus)$ —a monoid

$\downarrow$ —an action

$T\,X = S \to P \times X$

Reader monad

S —a set

$T\,X = S \to X$

State monad

S —a set

$T\,X = S \to S \times X$

Writer monad

$(P, o, \oplus)$ —a monoid

$T\,X = P \times X$

This talk: A unification (+ a little more)

Update monad

S —a set

$(P, o, \oplus)$ —a monoid

$\downarrow$ —an action

$$T\,X = S \to P \times X$$

cf. $T\,X = \Pi s : S.\,(s \downarrow P) \times X$ by Kammar and Plotkin

Reader monad	State monad	Writer monad
S —a set	S —a set	$(P, o, \oplus)$ —a monoid
$T\,X = S \to X$	$T\,X = S \to S \times X$	$T\,X = P \times X$

Monoids, monoid actions

- A monoid on a set P is given by

$$
\begin{aligned}
\mathrm{o} &: P, \\
\oplus &: P \to P \to P,
\end{aligned}
$$

$$
\begin{aligned}
p \oplus \mathrm{o} &= p \\
\mathrm{o} \oplus p &= p \\
(p \oplus p') \oplus p'' &= p \oplus (p' \oplus p'')
\end{aligned}
$$

- An action of a monoid $(P, \mathrm{o}, \oplus)$ on a set S is given by

$$\downarrow : S \to P \to S$$

$$
\begin{aligned}
s \downarrow \mathrm{o} &= s \\
s \downarrow (p \oplus p') &= (s \downarrow p) \downarrow p'
\end{aligned}
$$

Reader and writer monads

A set S and monoid $(P, \mathrm{o}, \oplus)$, $\downarrow$ give monads via

$$T_0\, X = S \to X \qquad\qquad T_1\, X = P \times X$$

$$\begin{aligned} \eta_0 &: \forall\{X\}.\, X \to S \to X \\ \eta_0\, x &= \lambda s.\, x \end{aligned} \qquad\qquad \begin{aligned} \eta_1 &: \forall\{X\}.\, X \to P \times X \\ \eta_1\, x &= (\mathrm{o}, x) \end{aligned}$$

$$\begin{aligned} \mu_0 &: \forall\{X\}.\, (S \to (S \to X)) \\ &\qquad\qquad\qquad \to S \to X \\ \mu_0\, f &= \lambda s.\, f\, s\, s \end{aligned} \qquad \begin{aligned} \mu_1 &: \forall\{X\}.\, P \times (P \times X) \\ &\qquad\qquad\qquad \to P \times X \\ \mu_1\, (p, (p', x)) &= (p \oplus p', x) \end{aligned}$$

State monads

- A set S gives a monad via

$$T\,X = S \to S \times X$$

$$\begin{gathered}\eta : \forall\{X\}.\, X \to S \to S \times X \\ \eta\, x = \lambda s.\,(s, x)\end{gathered}$$

$$\begin{aligned}\mu : \forall\{X\}.\,(S \to S \times (S \to S \times X)) \to S \to S \times X \\ \mu\, f = \lambda s.\,\mathrm{let}\quad (s', g) &= f\ s; \\ (s'', x) &= g\ s' \\ \mathrm{in}\ (s'', x)\end{aligned}$$

Update monads

- A set S, monoid $(P, \mathrm{o}, \oplus)$ and action $\downarrow$ give a monad via

$$T\,X = S \to P \times X$$

$$\begin{array}{c} \eta : \forall\{X\}.\, X \to S \to P \times X \\ \eta\, x = \lambda s.\, (\mathrm{o}, x) \end{array}$$

$$\begin{array}{c} \mu : \forall\{X\}.\, (S \to P \times (S \to P \times X)) \to S \to P \times X \\ \mu\, f = \lambda s.\, \mathrm{let} \quad \begin{array}{l} (p, g) = f\, s; \\ (p', x) = g\, (s \downarrow p) \end{array} \\ \mathrm{in}\ (p \oplus p', x) \end{array}$$

Reader and writer monads as instances

- Recall update monads:
  $T\,X = S \to P \times X$

- Reader monads:
  update monads with $(P, \mathrm{o}, \oplus)$ and $\downarrow$ trivial

- Writer monads:
  update monads with S and $\downarrow$ trivial

- State monads:
  embed into update monads
  for P the free monoid
  on the overwrite semi-group $(S, \bullet)$ with $s \bullet s' = s'$

Update monad example: writing into a buffer

- $S = E^* \times \mathrm{Nat}$ *(current buffer content and free space)*
- $P = E^*$ *(new values to write)*
- $\mathrm{o} = []$
- $p \oplus p' = p +\!\!+\, p'$
- $(s, n) \downarrow p = (s +\!\!+\, (p|n), n - \mathit{length}(p|n))$

($p|n$ is p truncated to length n)

Algebras of update monads

An algebra of such a monad is a set X with an operation

$$\mathsf{act} : (S \to P \times X) \to X$$

$$x = \mathsf{act}\,(\lambda s.\,(\mathsf{o}, x))$$

$$\begin{aligned}\mathsf{act}\,(\lambda s.\,(p, \mathsf{act}\,(\lambda s'.\,(p', x)))) \\ = \mathsf{act}\,(\lambda s.\,(p \oplus p'[s \downarrow p/s'], x[s \downarrow p/s']))\end{aligned}$$

or, equivalently a pair of operations (cf. algebraic effects)

$$\mathsf{lkp} : (S \to X) \to X$$
$$\mathsf{upd} : P \times X \to X$$

$$x = \mathsf{lkp}\,(\lambda s.\,\mathsf{upd}(\mathsf{o}, x))$$
$$\mathsf{upd}\,(p, \mathsf{upd}\,(p', x)) = \mathsf{upd}\,(p \oplus p', x)$$
$$\mathsf{lkp}\,(\lambda s.\,\mathsf{upd}\,(p, \mathsf{lkp}\,(\lambda s'.\,x))) = \mathsf{lkp}\,(\lambda s.\,\mathsf{upd}\,(p, x[s \downarrow p/s']))$$

Algebras of update monads cont'd

The operations

$$\mathsf{act} : (S \to P \times X) \to X$$

$$\begin{aligned} \mathsf{lkp} &: (S \to X) \to X \\ \mathsf{upd} &: P \times X \to X \end{aligned}$$

are interdefinable via

$$\begin{aligned} \mathsf{lkp}\,(\lambda s.\, x) &= \mathsf{act}\,(\lambda s.\, (\mathrm{o}, x)) \\ \mathsf{upd}\,(p, x) &= \mathsf{act}\,(\lambda s.\, (p, x)) \end{aligned}$$

$$\mathsf{act}\,(\lambda s.\, (p, x)) = \mathsf{lkp}\,(\lambda s.\, \mathsf{upd}\,(p, x))$$

Update monads as compatible compositions

The update monad for S, $(P, \mathsf{o}, \oplus)$, $\downarrow$ is the compatible composition the reader and writer monads

$$T_0\,X = S \to X \qquad\qquad T_1\,X = P \times X$$

$$\begin{array}{c}\eta_0 : \forall\{X\}.\, X \to S \to X \\ \eta_0\,x = \lambda s.\, x\end{array} \qquad \begin{array}{c}\eta_1 : \forall\{X\}.\, X \to P \times X \\ \eta_1\,x = (\mathsf{o}, x)\end{array}$$

$$\begin{array}{c}\mu_0 : \forall\{X\}.\, (S \to (S \to X)) \\ \qquad\qquad \to S \to X \\ \mu_0\,f = \lambda s.\, f\;s\;s\end{array} \qquad \begin{array}{c}\mu_1 : \forall\{X\}.\, P \times (P \times X) \\ \qquad\qquad \to P \times X \\ \mu_1\,(p, (p', x)) = (p \oplus p', x)\end{array}$$

for the distributive law

$$\begin{array}{c}\theta : \forall\{X\}.\, P \times (S \to X) \to (S \to P \times X) \\ \theta\,(p, f) = \lambda s.\, (p, f\,(s \downarrow p))\end{array}$$

Update algebras as compatible pairs of reader and writer algebras

An algebra of the update monad for S, $(P, \mathrm{o}, \oplus)$, $\downarrow$ is a set X carrying algebras of both the reader and writer monad

$$\mathsf{lkp} : (S \to X) \to X \qquad \mathsf{upd} : P \times X \to X$$

$$\begin{array}{ll} \mathsf{lkp}\,(\lambda s.\, x) = x & \mathsf{upd}\,(\mathrm{o}, x) = x \\ \mathsf{lkp}\,(\lambda s.\, \mathsf{lkp}\,(\lambda s'.\, x)) & \mathsf{upd}\,(p, \mathsf{upd}(p', x)) \\ \quad = \mathsf{lkp}\,(\lambda s.\, x[s/s']) & \quad = \mathsf{upd}\,(p \oplus p', x) \end{array}$$

satisfying an additional compatibility condition

$$\mathsf{upd}\,(p, \mathsf{lkp}\,(\lambda s'.\, x)) = \mathsf{lkp}\,(\lambda s.\, \mathsf{upd}\,(p, x[s \downarrow p/s']))$$

A finer version

- Rather than

$$\begin{array}{c} S \text{ —a set} \\ (P, \mathrm{o}, \oplus) \text{ —a monoid} \\ \downarrow \text{ —an action} \\ TX = S \to P \times X \end{array}$$

  consider

$$\begin{array}{c} (S, P, \downarrow, \mathrm{o}, \oplus) \text{ —a } \textit{directed container} \\ TX = \Pi s : S.\, P\, s \times X \end{array}$$

  S —states, $P\,s$ —updates *enabled* (or *safe*) in state s

Directed containers

- A directed container is

$$
\begin{gathered}
S \text{ a set,} \\
P \text{ a } S\text{-indexed family,} \\[1em]
\downarrow : \Pi s : S.\, P\, s \to S, \\
\mathsf{o} : \Pi\{s : S\}.\, P\, s \\
\oplus : \Pi\{s : S\}.\, \Pi p : P\, s.\, P\, (s \downarrow p) \to P\, s, \\[1em]
s \downarrow \mathsf{o} = s, \\
s \downarrow (p \oplus p') = (s \downarrow p) \downarrow p', \\[1em]
p \oplus \mathsf{o} = p, \\
\mathsf{o} \oplus p = p, \\
(p \oplus p') \oplus p'' = p \oplus (p' \oplus p'')
\end{gathered}
$$

Monads from directed containers

A directed container $(S, P, \downarrow, \mathrm{o}, \oplus)$ yields a monad via

$$T\,X = \Pi s : S.\, P\,s \times X$$

$$\begin{array}{c}\eta : \forall\{X\}.\, X \to \Pi s : S.\, P\,s \times X \\ \eta\,x = \lambda s.\, (\mathrm{o}, x)\end{array}$$

$$\begin{array}{c}\mu : \forall\{X\}.\, (\Pi s : S.\, P\,s \times (\Pi s' : S. P\,s' \times X)) \to \Pi s : S.\, P\,s \times X \\ \mu\,f = \lambda s.\, \mathrm{let}\quad \begin{array}{l}(p, g) = f\,s; \\ (p', x) = g\,(s \downarrow p)\end{array} \\ \mathrm{in}\ (p \oplus p', x)\end{array}$$

Example: writing into a buffer (a finer version)

- $S = E^* \times \mathrm{Nat}$ *(current buffer content and free space)*
- $P\ (s, n) = E^{\leq n}$ *(new values to write)*

- $\mathrm{o} = []$
- $p \oplus p' = p +\!\!+\ p'$

- $(s, n) \downarrow p = (s +\!\!+\ p, n - \mathit{length}(p))$

Monads from directed containers: Algebras

An algebra for the monad for the directed container $(S, P, \downarrow, \mathrm{o}, \oplus)$ is a set X with an operation

$$\mathsf{act} : (\Pi s : S.\, P\, s \times X) \to X$$

$$x = \mathsf{act}\,(\lambda s.\, \mathrm{o}, x)$$

$$\begin{array}{l} \mathsf{act}\,(\lambda s.\, p, \mathsf{act}\,(\lambda s'.\, p', x)) \\ \qquad = \mathsf{act}\,(\lambda s.\, p \oplus p'[s \downarrow p / s'], x[s \downarrow p / s']) \end{array}$$

Directed container morphisms, monad morphisms

- A morphism between two directed containers $(S', P', \downarrow', \mathsf{o}', \oplus')$ and $(S, P, \downarrow, \mathsf{o}, \oplus)$ is given by

$$t : S' \to S$$
$$q : \Pi\{s : S'\}.\, P\,(t\,s) \to P'\,s$$

$$t\,(s \downarrow' q\,p) = t\,s \downarrow p$$
$$\mathsf{o}' = q\,\mathsf{o}$$
$$q\,p \oplus' q\,p' = q\,(p \oplus p')$$

- It yields a morphism between the monads (T, η, μ) and (T', η', μ') via

$$\tau : \forall\{X\}.\,(\Pi s : S.\, P\,s \times X) \to \Pi s : S'.P'\,s \times X$$
$$\tau\,f = \lambda s.\,\mathsf{let}\;(p, x) = f\,(t\,s)\;\mathsf{in}\;(q\,p, x)$$

- Notice the reversal of arrow directions!

Directed containers and comonads

(A., C., U., FoSSaCS 2012)

$$\begin{tikzcd}
\begin{array}{c}\mathbf{DCont} \\ \cong \mathbf{Comonoids}(\mathbf{Cont})\end{array} \arrow[r, "U"] \arrow[d, "{[\![-]\!]^{\mathrm{dc}}\ \text{f.f.}}"'] \arrow[dr, phantom, "\text{pb}"] & \mathbf{Cont} \quad \text{mon.} \arrow[d, "{[\![-]\!]^{\mathrm{c}}\ \text{f.f., mon.}}"] \\
\begin{array}{c}\mathbf{Comonads}(\mathbf{Set}) \\ \cong \mathbf{Comonoids}([\mathbf{Set}, \mathbf{Set}])\end{array} \arrow[r, "U"'] & {[\mathbf{Set}, \mathbf{Set}]} \quad \text{mon.}
\end{tikzcd}$$

$$[\![S, P]\!]^{\mathrm{c}}\, X = \Sigma s : S.\, P\, s \to X$$

Directed containers and monads

(the new picture)

$$
\begin{array}{ccc}
\begin{array}{c}\mathbf{DCont}^{\mathrm{op}} \\ \cong (\mathbf{Comonoids}(\mathbf{Cont}))^{\mathrm{op}} \\ \cong \mathbf{Monoids}(\mathbf{Cont}^{\mathrm{op}})\end{array} & \xrightarrow{U} & \mathbf{Cont}^{\mathrm{op}} \quad \text{mon.} \\
\Big\downarrow {\scriptstyle \langle\!\langle - \rangle\!\rangle^{\mathrm{dc}}} & & \Big\downarrow {\scriptstyle \langle\!\langle - \rangle\!\rangle^{\mathrm{c}} \ \text{lax mon.}} \\
\begin{array}{c}\mathbf{Monads}(\mathbf{Set}) \\ \cong \mathbf{Monoids}([\mathbf{Set}, \mathbf{Set}])\end{array} & \xrightarrow[U]{} & [\mathbf{Set}, \mathbf{Set}] \quad \text{mon.}
\end{array}
$$

$$\langle\!\langle S, P \rangle\!\rangle^{\mathrm{c}}\, X = \Pi s : S.\, P\, s \times X$$