# Composition of Monads

Jeremy E. Dawson

Logic & Computation Programme
National ICT Australia

Automated Reasoning Group
Computer Sciences Laboratory
Res. Sch. of Inf. Sci. and Eng.
Australian National University

`Jeremy.Dawson@nicta.com.au`
`http://rsise.anu.edu.au/~jeremy`

## Types of monad functions

$$
\begin{aligned}
\textit{unit} &: \alpha \longrightarrow \alpha M \\
\textit{map} &: (\alpha \longrightarrow \beta) \longrightarrow (\alpha M \longrightarrow \beta M) \\
\textit{join} &: \alpha M M \longrightarrow \alpha M \\
\textit{ext} &: (\alpha \longrightarrow \beta M) \longrightarrow (\alpha M \longrightarrow \beta M) \\
\textit{bind} &: \alpha M \longrightarrow (\alpha \longrightarrow \beta M) \longrightarrow \beta M \\
\odot &: (\beta \longrightarrow \gamma M) \longrightarrow (\alpha \longrightarrow \beta M) \longrightarrow (\alpha \longrightarrow \gamma M)
\end{aligned}
$$

## Examples for the list monad

let $g\ 1 = [1]$, $g\ 3 = [1, 2, 3]$, etc

$$
\begin{aligned}
\textit{unit}\ a &= [a] \\
\textit{map}\ f\ [x, y, z] &= [f\ x, f\ y, f\ z] \\
\textit{join}\ [[u, v], [w], [x, y]] &= [u, v, w, x, y] \\
\textit{ext}\ g\ [2, 0, 4] &= [1, 2, 1, 2, 3, 4]
\end{aligned}
$$

# Relationships between monad functions

$$
\begin{aligned}
m\ \textit{bind}\ f &= \textit{ext}\ f\ m \\
\textit{ext}\ f &= \textit{join} \circ \textit{map}\ f \\
\textit{join} &= \textit{ext}\ \textit{id} \\
\textit{map}\ f &= \textit{ext}\ (\textit{unit} \circ f) \\
g \odot f &= \textit{ext}\ g \circ f \\
\textit{ext}\ g &= g \odot \textit{id}
\end{aligned}
$$

# Monad rules for *unit*, *map* and *join*

$$\mathit{map}\ \mathit{id} = \mathit{id} \tag{1}$$

$$\mathit{map}\ f \circ \mathit{map}\ g = \mathit{map}\ (f \circ g) \tag{2}$$

$$\mathit{unit} \circ f = \mathit{map}\ f \circ \mathit{unit} \tag{3}$$

$$\mathit{join} \circ \mathit{map}\ (\mathit{map}\ f) = \mathit{map}\ f \circ \mathit{join} \tag{4}$$

$$\mathit{join} \circ \mathit{unit} = \mathit{id} \tag{5}$$

$$\mathit{join} \circ \mathit{map}\ \mathit{unit} = \mathit{id} \tag{6}$$

$$\mathit{join} \circ \mathit{map}\ \mathit{join} = \mathit{join} \circ \mathit{join} \tag{7}$$

$$\mathit{ext}\ f = \mathit{join} \circ \mathit{map}\ f \tag{8}$$

$$\mathit{ext}\ (g \circ f) = \mathit{ext}\ g \circ \mathit{map}\ f \tag{9}$$

# Examples of monad rules

$$\mathit{join} \circ \mathit{unit} = \mathit{id} \tag{5}$$

$$\mathit{join}\ [[3,5,7]] = [3,5,7]$$

$$\mathit{join} \circ \mathit{map}\ \mathit{unit} = \mathit{id} \tag{6}$$

$$\mathit{join}\ [[3],[5],[7]] = [3,5,7]$$

$$\mathit{join} \circ \mathit{map}\ (\mathit{map}\ f) = \mathit{map}\ f \circ \mathit{join} \tag{4}$$

$$[[u,v],[w],[x,y]] \qquad \begin{matrix} [[u',v'],[w'],[x',y']] \\ [u,v,w,x,y] \end{matrix} \qquad [u',v',w',x',y']$$

$$\mathit{join} \circ \mathit{map}\ \mathit{join} = \mathit{join} \circ \mathit{join} \tag{7}$$

$$[[[u,v],[w]],[[x],[y,z]]] \qquad \begin{matrix} [[[u,v,w]],[[x,y,z]] \\ [[u,v],[w],[x],[y,z]] \end{matrix} \qquad [u,v,w,x,y,z]$$

## Monad rules for *unit*, *ext* and $\odot$

$$ext\ f \circ unit = f \qquad \text{(E1)}$$

$$ext\ unit = id \qquad \text{(E2)}$$

$$ext\ (ext\ g \circ f) = ext\ g \circ ext\ f \qquad \text{(E3')}$$

$$join = ext\ id \qquad \text{(E4)}$$

$$map\ f = ext\ (unit \circ f) \qquad \text{(E5)}$$

$$ext\ f \circ unit \quad = \quad f \qquad \text{(E1)}$$

$$ext\ unit \quad = \quad id \qquad \text{(E2)}$$

$$ext\ (g \odot f) \quad = \quad ext\ g \circ ext\ f \qquad \text{(E3)}$$

$$g \odot f \quad = \quad ext\ g \circ f \qquad \text{(E6)}$$

# A useful theorem

**Theorem 1** *In a monad the following are equivalent*

*(i)* *ext g = g* $\circ$ *join*

*(ii)* *g = ext (g* $\circ$ *unit)*

*(iii)* *there exists* $f$ *such that g = ext f*

*(iv)* *for all* $h$*, ext (g* $\circ$ *h) = g* $\circ$ *ext h*

Refer monad rules (E4), (7), (E5) and (4)

# Identity and associativity in the Kleisli category $\mathcal{K}_M$

**Theorem 2** *Assuming rules (E1) to (E3)*

$$g \odot f = \mathit{ext}\ g \circ f \quad \text{(E6)}$$

$$(h \odot g) \circ f = h \odot (g \circ f) \quad \text{(A6)}$$

$$\mathit{ext}\ f = \mathit{ext}\ g \Rightarrow f = g \quad \text{(EI)}$$

$$f \odot \mathit{unit} = f \quad \text{(A1)}$$

$$\mathit{unit} \odot f = f \quad \text{(A2)}$$

$$h \odot (g \odot f) = (h \odot g) \odot f \quad \text{(A3)}$$

the Kleisli category $\mathcal{K}_M$: **objects** are types — $\alpha$, $\beta$, etc

**an arrow** from $\alpha$ to $\beta$ is a function of type $\alpha \to \beta M$

**identity arrow** on $\alpha$ is *unit* : $\alpha \to \alpha M$

**composition** is $\odot : (\beta \to \gamma M) \to (\alpha \to \beta M) \to (\alpha \to \gamma M)$, so
$g$ from $\beta$ to $\gamma$ and $f$ from $\alpha$ to $\beta$ compose to give $g \odot f$, from $\alpha$ to $\gamma$

Rules (A1) to (A3) give the properties required for a category.

# Monad Rules Based on the Kleisli Category $\mathcal{K}_M$

$$f \odot \textit{unit} = f \quad \text{(A1)}$$
$$\textit{unit} \odot f = f \quad \text{(A2)}$$
$$h \odot (g \odot f) = (h \odot g) \odot f \quad \text{(A3)}$$
$$(h \odot id) \circ f = h \odot f \quad \text{(A4)}$$

$$h \odot (\textit{unit} \circ f) = h \circ f \quad \text{(A4')}$$

$$\textit{ext}\ g = g \odot id \quad \text{(A5)}$$

$$(h \odot g) \circ f = h \odot (g \circ f) \quad \text{(A6)}$$

# The State Monad

Let State be a fixed type, eg, the program state.

$$\alpha S = \text{State} \to \alpha * \text{State}$$

$$\textit{unit}_S\ a\ s = (a, s)$$
$$(g \odot_S f)\ a\ s = \text{let}\ (b, s') = f\ a\ s\ \text{in}\ g\ b\ s'$$

Proof tedious, but

$$\textit{curry}\ g\ x\ y = g\ (x, y) \qquad \textit{unc}\ f\ (x, y) = f\ x\ y$$

(mutually inverse, and so 1-1)

for (A1) and (A2):

$$\textit{unc}\ (f \odot_S \textit{unit}_S) = \textit{unc}\ f \circ \textit{unc}\ \textit{unit}_S = \textit{unc}\ f \circ \textit{id} = \textit{unc}\ f$$
$$\textit{unc}\ (\textit{unit}_S \odot_S f) = \textit{unc}\ \textit{unit}_S \circ \textit{unc}\ f = \textit{id} \circ \textit{unc}\ f = \textit{unc}\ f$$

for (A3): $\textit{unc}\ (h \odot_S (g \odot_S f)) = \textit{unc}\ h \circ (\textit{unc}\ g \circ \textit{unc}\ f) =$
$(\textit{unc}\ h \circ \textit{unc}\ g) \circ \textit{unc}\ f = \textit{unc}\ ((h \odot_S g) \odot_S f)$

# The Compound State Monad

Let $M$ be any monad. Define $\alpha S_M = \text{State} \rightarrow (\alpha * \text{State})M$.

We can define $\odot_{SM}$ and $\mathit{unit}_{SM}$ by

$$\mathit{unc}\ (g \odot_{SM} f) = \mathit{unc}\ g \odot_M \mathit{unc}\ f$$
$$\mathit{unc}\ \mathit{unit}_{SM} = \mathit{unit}_M$$

Then the proofs are easy, using corresponding rules for monad $M$.

for (A1) and (A2):

$$\mathit{unc}\ (f \odot_{SM} \mathit{unit}_{SM}) = \mathit{unc}\ f \odot_M \mathit{unc}\ \mathit{unit}_{SM} = \mathit{unc}\ f \odot_M \mathit{unit}_M = \mathit{unc}\ f$$
$$\mathit{unc}\ (\mathit{unit}_{SM} \odot_{SM} f) = \mathit{unc}\ \mathit{unit}_{SM} \odot_M \mathit{unc}\ f = \mathit{unit}_M \odot_M \mathit{unc}\ f = \mathit{unc}\ f$$

for (A3):

$$\mathit{unc}\ (h \odot_{SM} (g \odot_{SM} f)) = \mathit{unc}\ h \odot_M (\mathit{unc}\ g \odot_M \mathit{unc}\ f) =$$
$$(\mathit{unc}\ h \odot_M \mathit{unc}\ g) \odot_M \mathit{unc}\ f = \mathit{unc}\ ((h \odot_{SM} g) \odot_{SM} f)$$

Other definitions more complicated, and proofs correspondingly so.

# A free theorem ??

Still need to prove (A4),

$$(h \odot id) \circ f = h \odot f \qquad f : \alpha \to \beta M \qquad h : \beta \to \gamma M$$

Can we use Wadler's "free theorems"?

Idea is that $h \odot \_ : (\alpha \to \beta M) \to \alpha \to \gamma M$ is polymorphic in $\alpha$, so in applying $h \odot f$ to $a : \alpha$, can only apply $f$ to $a$, and then do something else, call it $g$.

That is, $h \odot f = g \circ f$, so

$$(h \odot \mathit{id}) \circ f = (g \circ \mathit{id}) \circ f = g \circ f = h \odot f$$

Is this valid?

# Other Monads

List monad: $\alpha L = \alpha$ list

Reader monad: $\alpha R =$ param $\rightarrow \alpha$

Writer monad: $\alpha W = \alpha \times$ output

Error monad: $\alpha E = \alpha$ option (SOME $a$ for success, NONE for failure)

These can form compound monads with an arbitrary monad $M$ in different ways: $\alpha M R$, $\alpha W M$, $\alpha E M$

$\alpha E$ can represent termination of a program, in a final state, or non-termination

$\alpha E L$ represents corresponding non-deterministic program operation

$\alpha L E$ also represents such a program for considering *total correctness*:
if it *may* fail to terminate then never mind what else it might do, it is a failure.

$\alpha L E$ is also a compound monad

# Compound Monads *via* Partial Extension

compound monad type is $(\alpha N)M = \alpha NM$. To define a compound monad $NM$, need *ext*$_{NM}$, "extending" a function $f$ from a "smaller" domain, $\alpha$, to a "larger" one, $\alpha NM$.

Consider a "partial extension" function *pext* which does part of this job:

$$\mathit{ext}_{NM} : (\alpha \to \beta NM) \to (\alpha NM \to \beta NM)$$
$$\mathit{pext} : (\alpha \to \beta NM) \to (\alpha N \to \beta NM)$$

Then $\mathit{ext}_{NM}\ f = \mathit{ext}_M\ (\mathit{pext}\ f)$.

Rules (E1K) to (E3K) are enough to define $NM$.

About $\odot_{NM}$ or *unit*$_{NM}$, assume only they have the right types.

Need not assume that $N$ is a monad.

# Monad rules for a compound monad using *pext*

$$
\begin{aligned}
\textit{pext}\ f \odot_M \textit{unit}_{NM} &= f && \text{(E1K)}\\
\textit{pext}\ \textit{unit}_{NM} &= \textit{unit}_M && \text{(E2K)}\\
\textit{pext}\ (g \odot_{NM} f) &= \textit{pext}\ g \odot_M \textit{pext}\ f && \text{(E3K)}\\
\\
\textit{kjoin} &= \textit{pext}\ \textit{unit}_M && \text{(E4K)}\\
\textit{kmap}\ f &= \textit{pext}\ (\textit{unit}_{NM} \odot_M f) && \text{(E5K)}\\
\\
g \odot_{NM} f &= \textit{pext}\ g \odot_M f && \text{(E6K)}
\end{aligned}
$$

These are just the rules needed for a monad $N$ in $\mathcal{K}_M$

## Correspondence between monad $N$ and monad $N$ in $\mathcal{K}_M$

$$
\begin{aligned}
id &: \alpha \to \alpha & \textit{unit}_M &: \alpha \to \alpha M \\
\textit{unit}_N &: \alpha \to \alpha N & \textit{unit}_{NM} &: \alpha \to \alpha N M \\
\textit{map}_N &: (\alpha \to \beta) \to \alpha N \to \beta N & \textit{kmap} &: (\alpha \to \beta M) \to \alpha N \to \beta N M \\
\textit{join}_N &: \alpha N N \to \alpha N & \textit{kjoin} &: \alpha N N \to \alpha N M \\
\textit{ext}_N &: (\alpha \to \beta N) \to \alpha N \to \beta N & \textit{pext} &: (\alpha \to \beta N M) \to \alpha N \to \beta N M
\end{aligned}
$$

$$
\begin{aligned}
\textit{ext}_N\ g &= g \odot_N \textit{id} & \textit{pext}\ g &= g \odot_{NM} \textit{unit}_M \\
g \odot_N f &= \textit{ext}_N\ g \circ f & g \odot_{NM} f &= \textit{pext}\ g \odot_M f \\
\textit{join}_N &= \textit{ext}_N\ id & \textit{kjoin} &= \textit{pext}\ \textit{unit}_M \\
\textit{map}_N\ f &= \textit{ext}_N\ (\textit{unit}_N \circ f) & \textit{kmap}\ f &= \textit{pext}\ (\textit{unit}_{NM} \odot_M f) \\
\textit{ext}_N\ f &= \textit{join}_N \circ \textit{map}_N\ f & \textit{pext}\ f &= \textit{kjoin} \odot_M \textit{kmap}\ f \\
h \odot_N f &= (h \odot_N id) \circ f & h \odot_{NM} f &= (h \odot_{NM} \textit{unit}_M) \odot_M f
\end{aligned}
$$

## To show $NM$ is a monad, using the *pext* rules

$$f \odot_{NM} \textit{unit}_{NM} = f \quad \text{(A1K)}$$
$$\textit{unit}_{NM} \odot_{NM} f = f \quad \text{(A2K)}$$
$$h \odot_{NM} (g \odot_{NM} f) = (h \odot_{NM} g) \odot_{NM} f \quad \text{(A3K)}$$
$$(h \odot_{NM} \textit{unit}_M) \odot_M f = h \odot_{NM} f \quad \text{(A4K)}$$

To show $NM$ is a monad, want namely (A1NM) to (A4NM).

But (A1NM) to (A3NM) same as (A1K) to (A3K); only (A4NM) is different.

So need only (A4NM); to get it, have both (A4K) and (A4M).

$$(h \odot_{NM} \textit{id}) \circ f = h \odot_{NM} f \quad \text{(A4NM)}$$
$$(h \odot_M \textit{id}) \circ f = h \odot_M f \quad \text{(A4M)}$$

$$\begin{aligned} (h \odot_{NM} id) \circ f &= ((h \odot_{NM} \textit{unit}_M) \odot_M id) \circ f && \text{(A4K)} \\ &= (h \odot_{NM} \textit{unit}_M) \odot_M f = h \odot_{NM} f && \text{(A4M, A4K)} \end{aligned}$$

# What characterises such compound monads?

Such compound monads $NM$ satisfy

$$\mathit{ext}_{NM}\ f = \mathit{ext}_M\ (\mathit{pext}\ f) \qquad \text{(EC)}$$

$$\mathit{pext}\ f = \mathit{ext}_{NM}\ f \circ \mathit{unit}_M \qquad \text{(PE)}$$

$$\mathit{ext}_M\ (\mathit{ext}_{NM}\ f) = \mathit{ext}_{NM}\ f \circ \mathit{join}_M \qquad \text{(J1S)}$$

Note, (J1S) of the form of Theorem 1(i).

Conversely, if $M$ and $NM$ are monads, and (J1S) holds, then $\odot_{NM}$ also defines a monad in $\mathcal{K}_M$, and, using (PE) to define *pext*, (EC) holds.

Proof uses that (A1K) to (A3K) same as (A1NM) to (A3NM);
shows (A4K) and (EC) from (J1S) using Theorem 1.

# A more general set of rules

Three more functions of the following types:

$$\begin{aligned} \textit{dunit} &: \alpha M \longrightarrow \alpha N M \\ \textit{dmap} &: (\alpha \longrightarrow \beta M) \longrightarrow (\alpha N M \longrightarrow \beta N M) \\ \textit{djoin} &: \alpha N N M \longrightarrow \alpha N M \end{aligned}$$

$$\textit{dmap}\ \textit{unit}_M = \textit{id} \tag{G1}$$

$$\textit{dmap}\ (f \circ h) = \textit{dmap}\ f \circ \textit{map}_{NM}\ h \tag{G2}$$

$$\textit{dmap}\ f \circ \textit{unit}_{NM} = \textit{dunit} \circ f \tag{G3}$$

$$\textit{djoin} \circ \textit{dmap}\ (\textit{dmap}\ f) = \textit{dmap}\ f \circ \textit{join}_{NM} \tag{G4}$$

$$\textit{djoin} \circ \textit{dunit} = \textit{id} \tag{G5}$$

$$\textit{djoin} \circ \textit{dmap}\ \textit{unit}_{NM} = \textit{id} \tag{G6}$$

$$\textit{djoin} \circ \textit{dmap}\ \textit{djoin} = \textit{djoin} \circ \textit{join}_{NM} \tag{G7}$$

$$\textit{ext}_{NM}\ f = \textit{djoin} \circ \textit{dmap}\ f \tag{G8}$$

# These also give a monad $NM$

**Theorem 3** *Assume rules (G1) to (G8). Then* $\mathit{ext}_{NM}$, $\mathit{join}_{NM}$, $\mathit{map}_{NM}$ *and* $\mathit{unit}_{NM}$ *give a monad* $NM$*, where also*

$$\begin{aligned}
\mathit{djoin} &= \mathit{ext}_{NM}\ \mathit{unit}_M && \text{(G9)} \\
\mathit{dmap}\ f &= \mathit{ext}_{NM}\ (\mathit{dunit} \circ f) && \text{(G10)} \\
\mathit{unit}_{NM} &= \mathit{dunit} \circ \mathit{unit}_M && \text{(G11)} \\
\mathit{map}_{NM}\ f &= \mathit{dmap}\ (\mathit{unit}_M \circ f) && \text{(G12)}
\end{aligned}$$

Conversely, for a compound monad $NM$, when is this construction applicable?

**Theorem 4** *Assume that* $NM$ *is a monad. Also assume that rules (G5) and (G9) to (G11) hold. Then the remaining rules among (G1) to (G8) hold.*

# When is the construction applicable?

How to use Theorem 4? Assume (UC).

$$\mathit{unit}_{NM}\ f = \mathit{unit}_M\ (\mathit{unit}_N\ f) \tag{UC}$$

$$\mathit{dunit} = \mathit{map}_M\ \mathit{unit}_N \tag{DU}$$

$$\mathit{ext}_{NM}\ \mathit{unit}_M \circ \mathit{map}_M\ \mathit{unit}_N = \mathit{id} \tag{G5$'$}$$

If (J1S) and the *pext* construction hold, then *define* functions *dunit*, *dmap* and *djoin* by (DU), (G10) and (G9).

Then (G11) holds by (3M), and (G5) becomes (G5′) which holds: the proof uses $\mathit{ext}_{NM}\ f = \mathit{ext}_M(\mathit{pext}\ f)$.

So Theorem 4 applies.

# On the other hand . . .

$$\mathit{ext}_{NM}\ (\mathit{map}_M\ \mathit{join}_N) = \mathit{map}_M\ \mathit{join}_N \circ \mathit{join}_{NM} \qquad (\text{J2}')$$

Note (J2′) is also of the form of Theorem 1(i).

If $N$ is a monad and $M$ a premonad, and (J2′) holds, then
$\mathit{ext}_{NM}\ \mathit{unit}_M = \mathit{map}_M\ \mathit{join}_N$ (proved from Theorem 1).

In this case (G5)/(G5′) also hold, by a seemingly *different* proof.

$$\mathit{ext}_{NM}\ \mathit{unit}_M \circ \mathit{map}_M\ \mathit{unit}_N = \mathit{map}_M\ \mathit{join}_N \circ \mathit{map}_M\ \mathit{unit}_N$$
$$= \mathit{map}_M\ (\mathit{join}_N \circ \mathit{unit}_N) = \mathit{map}_M\ \mathit{id} = \mathit{id}$$

So again Theorem 4 applies.

Common feature: $\mathit{ext}_{NM}\ \mathit{unit}_M$ is of the form $\mathit{ext}_M\ f$, so Theorem 1 applies.

## When both (J1S) and (J2′) hold

If *both* (J1S) and (J2′) hold, and *both* $M$ and $N$ are monads, then we have a distributive law for the monads $M$, $N$ and $NM$.