Distributive Laws of Monadic Containers

Stefania Damato
(University of Nottingham)

Chris Purdy
(Royal Holloway, University of London)

WG6 Meeting, Genoa
17 April 2025

arxiv.org/abs.2503.17191

Containers compose

Containers (aka polynomial functors) represent strictly positive types as shapes $S : \mathrm{Set}$ and positions $P : S \to \mathrm{Set}$, written $S \triangleleft P$.

Containers compose

Containers (aka polynomial functors) represent strictly positive types as shapes $S : \mathrm{Set}$ and positions $P : S \to \mathrm{Set}$, written $S \triangleleft P$.

The category Cont of containers carries a composition monoidal structure:

$$(S \triangleleft P) \circ (T \triangleleft Q) = \left(\sum_{s:S} (f : Ps \to T)\right) \triangleleft \left(\sum_{p:Ps} Q(f\,p)\right).$$

Functorial interpretation of containers

Containers have a fully-faithful interpretation into Set-endofunctors:

$$[\![\_]\!] : \mathrm{Cont} \longrightarrow [\mathrm{Set}, \mathrm{Set}]$$

$$[\![S \triangleleft P]\!] := X \longmapsto \sum_{s:S} (P_s \to X)$$

Functorial interpretation of containers

Containers have a fully-faithful interpretation into Set-endofunctors:

$$[\![\_]\!] : \mathrm{Cont} \to [\mathrm{Set}, \mathrm{Set}]$$

Containers whose functor interpretation carries a

- monad structure are called monadic.
- comonad structure are called directed.

Monadic containers

▶ **Definition 8** (⚙). *Let $S \triangleleft P$ be a container. A* <u>*monadic container*</u> *on $S \triangleleft P$ is a tuple* $(S \triangleleft P, \iota, \sigma, \mathsf{pr})$ *where*

$$\iota : S$$

$$\sigma : \prod_{s:S} (P\, s \to S) \to S$$

$$\mathsf{pr} : \prod_{\{s:S\}} \prod_{\{f:P\, s \to S\}} P\,(\sigma\, s\, f) \to \sum_{p:P\, s} P\,(f\, p)$$

+8 equations

[Uustalu, 2017]

Presented as lax Σ-universes

[Altenkirch & Pinyo, 2017]

Monadic containers

▶ **Definition 8** (⚙). *Let $S \triangleleft P$ be a container. A* <u>*monadic container*</u> *on $S \triangleleft P$ is a tuple* $(S \triangleleft P, \iota, \sigma, \mathsf{pr})$ *where*

$$\iota : S$$

$$\sigma : \prod_{s:S} (P\,s \to S) \to S$$

$$\mathsf{pr} : \prod_{\{s:S\}} \prod_{\{f:P\,s \to S\}} P\,(\sigma\,s\,f) \to \sum_{p:P\,s} P\,(f\,p)$$

+8 equations

[Uustalu, 2017]

Presented as lax Σ-universes

[Altenkirch & Pinyo, 2017]

Monadic containers form a category MCont.

Monadic containers (cont'd)

Monadic containers form a category **MCont**.

$$\mathbf{MCont} \cong$$

$$\begin{tikzcd}
\mathrm{Monoid}(\mathrm{Cont}, \mathrm{Id}^c, \circ^c) \arrow[r, "U"] \arrow[d, "[\![_]\!]^{mc}"'] & \mathrm{Cont} \arrow[d, "[\![_]\!]"] \\
\mathrm{Monoid}([\mathrm{Set}, \mathrm{Set}], \mathrm{Id}, \circ) \arrow[r, "U"'] & [\mathrm{Set}, \mathrm{Set}]
\end{tikzcd}$$

$$\cong \mathrm{Monad}(\mathrm{Set})$$

[Uustalu, 2017]

Monadic containers don't compose

Counterexample: writer monadic container on (Bool, false, xor) and terminal monadic container.

Monadic containers don't compose

Counterexample: writer monadic container on (Bool, false, xor) and terminal monadic container.

Monads do not compose either.
Beck's approach: distributive laws as a sufficient condition.

Distributive laws

1. Distributive laws, composite and lifted triples

A *distributive law of* **S** *over* **T** is a natural transformation $\ell\colon TS \longrightarrow ST$ such that

$$
\begin{tikzcd}
 & T \arrow[dl, "T\eta^S"'] \arrow[dr, "\eta^S T"] & \\
TS \arrow[rr, "\ell"'] & & ST
\end{tikzcd}
\qquad
\begin{tikzcd}
 & S \arrow[dl, "\eta^T S"'] \arrow[dr, "S\eta^T"] & \\
TS \arrow[rr, "\ell"'] & & ST
\end{tikzcd}
$$

$$
\begin{tikzcd}
TSS \arrow[r, "\ell S"] \arrow[d, "T\mu^S"'] & STS \arrow[r, "S\ell"] & SST \arrow[d, "\mu^S T"] \\
TS \arrow[rr, "\ell"'] & & ST
\end{tikzcd}
$$

$$
\begin{tikzcd}
TTS \arrow[r, "T\ell"] \arrow[d, "\mu^T S"'] & TST \arrow[r, "\ell T"] & STT \arrow[d, "S\mu^T"] \\
TS \arrow[rr, "\ell"'] & & ST
\end{tikzcd}
$$

commute.

[Beck, 1969]

However, when $\mathcal{T}$ is not free it can be rather difficult to prove the defining axioms of a distributive law. In this paper we describe how to obtain a distributive law for a monad with

[Bonsangue et. al, 2015]

Distributive laws (cont'd)

Theory and Applications of Categories, Vol. 18, No. 7, 2007, pp. 172–208.

MONAD COMPOSITIONS I: GENERAL CONSTRUCTIONS AND RECURSIVE DISTRIBUTIVE LAWS

ERNIE MANES AND PHILIP MULRY

Math. Struct. in Comp. Science (2008), *vol.* 18, *pp.* 613–643. © 2008 Cambridge University Press
doi:10.1017/S0960129508006695 First published online 26 February 2008 Printed in the United Kingdom

Monad compositions II: Kleisli strength

ERNIE MANES† and PHILIP MULRY‡

Logical Methods in Computer Science
Vol. 11(3:2)2015, pp. 1–23
www.lmcs-online.org

Submitted Jan. 13, 2013
Published Aug. 7, 2015

PRESENTING DISTRIBUTIVE LAWS

MARCELLO M. BONSANGUE[a], HELLE H. HANSEN[b], ALEXANDER KURZ[c], AND JURRIAAN ROT[d]

Layer by Layer – Combining Monads

Fredrik Dahlqvist, Louis Parlant, and Alexandra Silva*
University College London

Positive results

Distributing probability over nondeterminism

DANIELE VARACCA[1†] and GLYNN WINSKEL[2]
[1]*Department of Computing, Imperial College London.*
[2]*Computer Laboratory, Cambridge University.*

Received 24 October 2005

ELSEVIER
Available online at www.sciencedirect.com
ScienceDirect
Electronic Notes in Theoretical Computer Science 341 (2018) 261–276
Electronic Notes in Theoretical Computer Science
www.elsevier.com/locate/entcs

Iterated Covariant Powerset is not a Monad[1]

Bartek Klin[2]
*Faculty of Mathematics, Informatics, and Mechanics
University of Warsaw
Warsaw, Poland*

Julian Salamanca [3]
*Faculty of Mathematics, Informatics, and Mechanics
University of Warsaw
Warsaw, Poland*

Logical Methods in Computer Science
Volume 18, Issue 1, 2022, pp. 13:1–13:61
https://lmcs.episciences.org/

Submitted Mar. 30, 2020
Published Jan. 19, 2022

NO-GO THEOREMS FOR DISTRIBUTIVE LAWS

MAAIKE ZWART AND DAN MARSDEN

Negative results

Distributive laws & containers

Container Combinatorics: Monads and Lax Monoidal Functors

Tarmo Uustalu(✉)

Department of Software Science, Tallinn University of Technology, Akadeemia tee 21B, 12618 Tallinn, Estonia
tarmo@cs.ioc.ee

Monadic containers

Logical Methods in Computer Science
Vol. 10(3:14)2014, pp. 1–48
www.lmcs-online.org

Submitted Dec. 5, 2012
Published Sep. 03, 2014

WHEN IS A CONTAINER A COMONAD?*

DANEL AHMAN [a], JAMES CHAPMAN [b], AND TARMO UUSTALU [c]

[a] Laboratory for Foundations of Computer Science, School of Informatics, University of Edinburgh, 10 Crichton Street, Edinburgh EH8 9AB, United Kingdom
e-mail address: d.ahman@ed.ac.uk

[b,c] Institute of Cybernetics at Tallinn University of Technology, Akadeemia tee 21, 12618 Tallinn, Estonia
e-mail address: {james,tarmo}@cs.ioc.ee

Directed containers

This work:
Monadic distributive laws & mixed distributive laws.

Progress in Informatics, No. 10, pp.3–18, (2013)

Special issue: Advanced Programming Techniques for Construction of Robust, General and Evolutionary Programs

Research Paper

Distributive laws of directed containers

Danel AHMAN[1] and Tarmo UUSTALU[2]
[1] *Laboratory for Foundations of Computer Science, University of Edinburgh*
[2] *Institute of Cybernetics, Tallinn University of Technology*

Directed distributive laws

Our contribution

- Characterisation of
  - monadic,
  - monadic-directed, &
  - directed-monadic

  $\left.\begin{matrix}\\ \\ \end{matrix}\right\}$ 'mixed'

  container distributive laws.
- Positive, negative, & uniqueness results.
- Formalisation in Cubical Agda.

Our contribution in more detail

▶ **Definition 16** (⚙). *Let* $(S \triangleleft P, \iota^S, \sigma^S, \mathsf{pr}^S)$ *and* $(T \triangleleft Q, \iota^T, \sigma^T, \mathsf{pr}^T)$ *be monadic containers. A* *monadic container distributive law* *of* $S \triangleleft P$ *over* $T \triangleleft Q$ *is given by the data*

$$u_1 : \prod_{s:S} (P\,s \to T) \to T$$

$$u_2 : \prod_{s:S} \prod_{f:P\,s \to T} Q\,(u_1\,s\,f) \to S$$

$$v_1 : \prod_{\{s:S\}} \prod_{\{f:P\,s \to T\}} \prod_{q:Q\,(u_1\,s\,f)} P\,(u_2\,s\,f\,q) \to P\,s$$

$$v_2 : \prod_{\{s:S\}} \prod_{\{f:P\,s \to T\}} \prod_{q:Q\,(u_1\,s\,f)} \prod_{p:P\,(u_2\,s\,f\,q)} Q\,(f\,(v_1\,q\,p))$$

+18 equations

▶ **Definition 27** (⚙). *Let* $(T \triangleleft Q, \iota, \sigma, \mathsf{pr})$ *be a monadic container and* $(S \triangleleft P, o, \oplus, \downarrow)$ *be a directed container. We define a* *monadic-directed container distributive law* *of* $T \triangleleft Q$ *over* $S \triangleleft P$ *as the data*

$$u_1 : \prod_{s:S} (P\,s \to T) \to T$$

$$u_2 : \prod_{s:S} \prod_{f:P\,s \to T} Q\,(u_1\,s\,f) \to S$$

$$v_1 : \prod_{\{s:S\}} \prod_{\{f:P\,s \to T\}} \prod_{q:Q\,(u_1\,s\,f)} P\,(u_2\,s\,f\,q) \to P\,s$$

$$v_2 : \prod_{\{s:S\}} \prod_{\{f:P\,s \to T\}} \prod_{q:Q\,(u_1\,s\,f)} \prod_{p:P\,(u_2\,s\,f\,q)} Q\,(f\,(v_1\,q\,p))$$

+13 equations

& directed-monadic case.

Formalisation in Cubical Agda

```agda
ReaderDistr : ∀ {ℓs ℓp} (A : Set ℓp) (S : Set ℓs) (P : S → Set ℓp)
  → (MC : MndContainer ℓs ℓp (S ▷ P))
  → MndDistributiveLaw ℓs ℓp S P (⊤ {ℓs}) (const A) MC (ReaderM A)
u₁ (ReaderDistr A S P MC) s _ = tt
u₂ (ReaderDistr A S P MC) s _ a = s
v₁ (ReaderDistr A S P MC) a p = p
v₂ (ReaderDistr A S P MC) a p = a
unit-ıB-shape₂ (ReaderDistr A S P MC) s = refl
unit-ıB-shape₁ (ReaderDistr A S P MC) s = refl
unit-ıB-pos₁ (ReaderDistr A S P MC) s = refl
unit-ıB-pos₂ (ReaderDistr A S P MC) s i a p = a
unit-ıA-shape₂ (ReaderDistr A S P MC) tt = refl
unit-ıA-shape₁ (ReaderDistr A S P MC) tt = refl
unit-ıA-pos₁ (ReaderDistr A S P MC) tt = refl
unit-ıA-pos₂ (ReaderDistr A S P MC) tt = refl
mul-A-shape₁ (ReaderDistr A S P MC) s f g = refl
mul-A-shape₂ (ReaderDistr A S P MC) s f g = refl
mul-A-pos₁ (ReaderDistr A S P MC) s f g = refl
mul-A-pos₂₁ (ReaderDistr A S P MC) s f g = refl
mul-A-pos₂₂ (ReaderDistr A S P MC) s f g = refl
mul-B-shape₁ (ReaderDistr A S P MC) s f g = refl
mul-B-shape₂ (ReaderDistr A S P MC) s f g = refl
mul-B-pos₁ (ReaderDistr A S P MC) s f g = refl
mul-B-pos₂₁ (ReaderDistr A S P MC) s f g = refl
mul-B-pos₂₂ (ReaderDistr A S P MC) s f g = refl
```

```agda
MaybeDistr : ∀ {ℓs ℓp} (S : Set ℓs) (P : S → Set ℓp) (MC : MndContainer ℓs ℓp (S ▷ P)) →
             MndDistributiveLaw ℓs ℓp 2 JustOrNothing S P MaybeM MC
u₁ (MaybeDistr S P MC) true f = f tt
u₁ (MaybeDistr S P MC) false f = MC .ı
u₂ (MaybeDistr S P MC) true f _ = true
u₂ (MaybeDistr S P MC) false f _ = false
v₁ (MaybeDistr S P MC) {true} _ x = tt
v₂ (MaybeDistr S P MC) {true} {f} p x = p
unit-ıB-shape₁ (MaybeDistr S P MC) true = refl
unit-ıB-shape₁ (MaybeDistr S P MC) false = refl
unit-ıB-shape₂ (MaybeDistr S P MC) true = refl
unit-ıB-shape₂ (MaybeDistr S P MC) false = refl
unit-ıB-pos₁ (MaybeDistr S P MC) true i q tt = tt
unit-ıB-pos₂ (MaybeDistr S P MC) true i q tt = q
unit-ıA-shape₁ (MaybeDistr S P MC) _ = refl
unit-ıA-shape₂ (MaybeDistr S P MC) _ = refl
unit-ıA-pos₁ (MaybeDistr S P MC) s i q tt = tt
unit-ıA-pos₂ (MaybeDistr S P MC) s i q tt = q
mul-A-shape₁ (MaybeDistr S P MC) true f g = refl
mul-A-shape₁ (MaybeDistr S P MC) false f g = refl
mul-A-shape₂ (MaybeDistr S P MC) true f g = refl
mul-A-shape₂ (MaybeDistr S P MC) false f g = refl
mul-A-pos₁ (MaybeDistr S P MC) true f g = refl
mul-A-pos₁ (MaybeDistr {ℓs} {ℓp} S P MC) false f g i q ()
mul-A-pos₂₁ (MaybeDistr S P MC) true f g = refl
mul-A-pos₂₁ (MaybeDistr {ℓs} {ℓp} S P MC) false f g i q ()
mul-A-pos₂₂ (MaybeDistr S P MC) true f g = refl
mul-A-pos₂₂ (MaybeDistr S P MC) false f g i q ()
mul-B-shape₁ (MaybeDistr S P MC) true f g = refl
mul-B-shape₁ (MaybeDistr S P MC) false f g i = unit-r (isMndContainer MC) (MC .ı) (~ i)
mul-B-shape₂ (MaybeDistr S P MC) true f g = refl
mul-B-shape₂ (MaybeDistr S P MC) false f g i = λ _ → false
mul-B-pos₁ (MaybeDistr S P MC) true f g i q tt = tt
mul-B-pos₁ (MaybeDistr S P MC) false f g i q ()
mul-B-pos₂₁ (MaybeDistr S P MC) true f g i q tt = (MC .pr₁) (f tt) (g tt) q
mul-B-pos₂₁ (MaybeDistr S P MC) false f g i q ()
mul-B-pos₂₂ (MaybeDistr S P MC) true f g i q tt = (MC .pr₂) (f tt) (g tt) q
mul-B-pos₂₂ (MaybeDistr S P MC) false f g i q ()
```

Example 1

The reader monad is a monadic container:

$$\mathrm{Reader}_A\ X := A \to X$$

$$\cong \sum_{\mathbb{1}} (A \to X) = [\![\mathbb{1} \triangleleft \mathrm{const}\ A]\!]\ X .$$

Example 1

The reader monad is a monadic container:

$$\mathrm{Reader}_A\ X := A \to X$$
$$\cong \sum_{\mathbb{1}} (A \to X) = [\![\mathbb{1} \lhd \mathrm{const}\ A]\!]\ X.$$

There is a <u>unique</u> monadic container distributive law of the reader monadic container $\mathbb{1} \lhd A$ over any $S \lhd P$.

Example 2

The maybe monad is a monadic container:

$$\mathrm{Maybe}\ X := 1 + X$$
$$\cong [\![\mathrm{Bool} \lhd \lambda \begin{cases} \mathrm{true} \to \mathbf{1} \\ \mathrm{false} \to \mathbf{0} \end{cases}]\!]\ X$$

Example 2

The maybe monad is a monadic container:

$$\text{Maybe}\ X := 1 + X$$
$$\cong [\![\, \text{Bool} \lhd \lambda \begin{cases} \text{true} \to 1 \\ \text{false} \to 0 \end{cases}]\!]\ X$$

There is a <u>unique</u> monadic container distributive law of any $S \lhd P$ over the maybe monadic container.

Takeaway from examples

If you want to compose a monad M with the reader or maybe monad,

Takeaway from examples

If you want to compose a monad M with the reader or maybe monad, you should first check if the endofunctor of M can be expressed as a container, in which case we showed there will be a unique distributive law!

In summary

- The category of containers has a monoidal structure given by composition of objects.
- We characterise conditions under which monadic containers compose.
- We formalise our results in Cubical Agda, providing an interface for distributive laws.

In summary

arxiv.org/abs.2503.17191

- The category of containers has a monoidal structure given by composition of objects.
- We characterise conditions under which monadic containers compose.
- We formalise our results in Cubical Agda, providing an interface for distributive laws.

THANK YOU!

References

- Uustalu, Tarmo (2017). 'Container combinatorics: Monads and Lax Monoidal Functors'. In TTCS 2017.
- Altenkirch, Thorsten and Pinyo, Gun (2017). 'Monadic containers and Σ-universes'. TYPES 2017 abstract.
- Beck, Jon (1969). 'Distributive laws'. In Seminar on Triples and Categorical Homology Theory, p. 119-140, Berlin, Heidelberg.
- Bonsangue, Marcelo M. et. al (2013). 'Presenting Distributive Laws'. In CALCO 2013.